2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA) | 978-1-6654-7652-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/HPCA56546.2023.10071036

Co-Designed Architectures for Modular Superconducting Quantum Computers

Evan McKinney[†], Mingkang Xia[‡], Chao Zhou[‡], Pinlei Lu[‡], Michael Hatridge[‡], Alex K. Jones[†]
[†]Department of Electrical and Computer Engineering, [‡]Department of Physics and Astronomy
University of Pittsburgh

Abstract—**Noisy, Intermediate Scale Quantum (NISQ) computers have reached the point where they can show the potential for quantum advantage over classical computing. Unfortunately, NISQ machines introduce sufficient noise that even for moderate size quantum circuits the results can be unreliable. We propose a collaboratively designed superconducting quantum computer using a Superconducting Nonlinear Asymmetric Inductive eLement (SNAIL) modulator. The SNAIL modulator is designed by considering both the ideal fundamental qubit gate operation while maximizing the qubit coupling capabilities. First, the SNAIL natively implements $\sqrt[n]{\texttt{iSWAP}}$ gates realized through proportionally scaled pulse lengths. This naturally includes $\sqrt{\texttt{iSWAP}}$, which provides an advantage over CNOT as a basis gate. Second, the SNAIL enables high-degree couplings that allow rich and highly parallel qubit connection topologies without suffering from frequency crowding. Building on our previously demonstrated SNAIL-based quantum state router we propose a quantum 4-ary tree and a hypercube inspired corral built from interconnected quantum modules. We compare their advantage in data movement based on necessary SWAP gates to the traditional lattice and heavy-hex lattice used in latest commercial quantum computers. We demonstrate the co-design advantage of our SNAIL-based machine with $\sqrt{\texttt{iSWAP}}$ basis gates and rich topologies against CNOT/heavy-hex and FSIM/lattice for 16-20 qubit and extrapolated designs circa 80 qubit architectures. We compare total circuit time and total gate count to understand fidelity for systems dominated by decoherence and control imperfections, respectively. Finally, we provide a gate duration sensitivity study on further decreasing the SNAIL pulse length to realize $\sqrt[n]{\texttt{iSWAP}}$ qubit systems to reduce decoherence times.**

I. Introduction

Quantum Computers (QCs) leverage the physics of quantum information with the promise to change the computing landscape by solving problems that are intractable for classical computers. The field is currently in the Noisy Intermediate-Scale Quantum (NISQ) era where quantum machines with more than a hundred qubits exist and the fidelities of quantum operations among one or more qubits, referred to as *gates*, above 99.9% are possible. Yet these QCs remain too small and sensitive to error to perform error-correction [39]. Therefore, NISQ machines are crucially constrained by the duration of the circuits to limit gate noise and qubit decoherence.

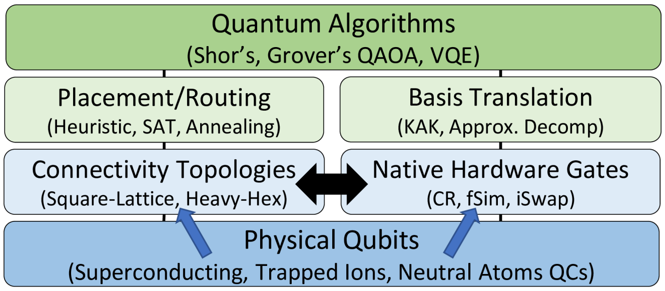


Fig. 1: Traditional quantum computing stack. The SNAIL allows co-design of rich connectivity topologies and native hardware *basis* gates for improved basis translation, placement, and routing of important quantum algorithms.

NISQ QCs operate by qubit coupling mechanisms that produce different gate operations and neighborhoods of qubit connectivity. Practically, this qubit-qubit coupling arises from a physical connection between them, such as a simple capacitive coupling to a more elaborate nonlinear circuit [10], [11], [21]. We refer to them generally as *modulators*, which may target pairs of qubits via layout geometry, unique frequencies or frequency differences, and which, together with applied control signals, govern the fundamental gate operations implemented in the QC. Due to the strict constraints of duration and decoherence, it is necessary to advance the design of modulators to produce high-fidelity qubits and couplings.

The realized potential of quantum computing is furthered when elements of the quantum computing stack, shown in Fig. 1, are designed in synergy. This requires development of the physical mechanisms and modulators, which realize the *physical qubits*, and the *native hardware gates* and *connectivity topologies*, respectively are collaboratively designed with the *quantum algorithms* we want to implement. These QCs more successfully execute a quantum program with higher reliability to advance the applicability of NISQ computers.

In this paper we propose ***a novel co-designed superconducting QC architecture*** based on a "SNAIL" (Superconducting Nonlinear Asymmetric Inductive eLement) quantum modulator [15]. We leverage our recent experimental demonstration of a SNAIL-based quantum state router [50] and prototype SNAIL-based four qubit modules to characterize and construct modular quantum topologies such as various designs based on a 4-ary tree with native iSWAP-family gates.

We explore the benefit of $\sqrt{\texttt{iSWAP}}$ as a basis operation as an improved decomposition alternative to CNOT, or SYC gates [19]. We then explore how decomposition to $\sqrt[n]{\texttt{iSWAP}}$ where $n \geq 3$ can further improve fidelity through reduced pulse time. Furthermore, we consider alternate topologies in a search for designs which are both physically reasonable with demonstrated technology and maximize the computational power of SNAIL-based modular NISQ machines. All results are benchmarked against existing large-scale machines from Google [1] based on a Square-Lattice of qubit couplings and the FSIM gate family and IBM's recent "Heavy-Hex" based machines with native CR gates [9].

In particular, our contributions are as follows:

- Demonstrate novel and scalable high connectivity 4-ary tree and Corral topology modules enabled through the co-designed QC using SNAIL modulators.
- Explore $\sqrt[n]{\texttt{iSWAP}}$ as the basis gate in our co-designed QC to increase fidelity through improving overall time to algorithm completion.
- Demonstrate our SNAIL-based modular QCs potential improvement for moderate 16-20 qubit QCs versus normalized versions of IBM and Google architectures as representative systems on a set of NISQ-algorithms.
- Explore the scalability of collaboratively designed modular superconducting QCs for large numbers of qubits.

We find that on an average of Quantum Volume (QV) circuits ranging from 16 to 80 qubits, a hypercube topology with a $\sqrt{\texttt{iSWAP}}$ basis gate requires $3.16\times$ less total 2Q gates and $6.11\times$ less total duration-dependent 2Q gates than a Heavy-Hex topology with a CNOT basis. Additionally, we find that for randomly sampled 2Q unitaries and for a 99% fidelity iSWAP basis, a $\sqrt[4]{\texttt{iSWAP}}$ basis decreases infidelity on average by 58% compared to iSWAP.

II. Background and Preliminaries

Here we provide a relevant background on QC principles and preliminary statements about co-design methodologies.

A. Transmon Qubits

Qubits have been realized from a variety of quantum systems, including atoms, electrons, nuclear spins, etc. In this work we focus on superconducting qubits which are nanofabricated, nonlinear microwave-frequency circuits [13]. The nonlinear element of choice in these circuits is the Josephson junction (JJ), which can be thought of as a non-linear inductor. The most commonly used superconducting qubit is the transmon, which consists simply of a JJ shunted with a large capacitance [22]. The transmon is ubiquitous because it is insensitive to charge and flux noise and can achieve very high coherence [38], [48]. It is also easy to fabricate, control, and couple to other circuit elements [6].

Superconducting qubits are almost always controlled and read out by embedding them in a linear oscillator; this platform of a qubit + cavity unit goes by the name circuit Quantum Electrodynamics or cQED [6]. The cavity is coupled to the qubit dispersively via the "cross-Kerr term"which shifts the mode of the cavity when the qubit gains a photon. This allows for a pulse transiting the cavity to encode the state of the qubit, and is the basis for qubit readout. More, dispersive interactions and cross-Kerr interactions can also be used to couple qubits to each other, to a resonator, or to another nonlinear object, and so are often the physical basis for two qubit gates as well.

B. Quantum Gates

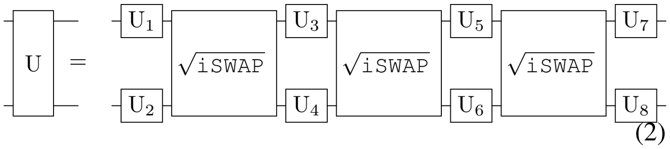
Quantum gates are unitary matrix operations that act on quantum state vectors. In general, for NISQ machines single-qubit gates (1Q) and two-qubit gates (2Q) form the building blocks of quantum circuits [34]. A native QC gate set, analogous to a classical computer's instruction set architecture, defines which unitary operations are available to use on a particular machine. A basic universal gate set consists of arbitrary single qubit gates plus a single two qubit gate, most often CNOT [2]. A different, potentially more convenient, physical coupling of a QC may result in a different *basis* gate. For example, the family of fractional-iSWAP gates, as in Eq. 1, that include iSWAP and $\sqrt{\texttt{iSWAP}}$, are also universal gates.

$$\sqrt[n]{\texttt{iSWAP}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\pi/2n) & i\sin(\pi/2n) & 0 \\ 0 & i\sin(\pi/2n) & \cos(\pi/2n) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

C. Gate Translation and Decomposition

Translating a quantum program between basis gates such as CNOT or $\sqrt{\texttt{iSWAP}}$ uses gate decomposition. This process converts arbitrary unitaries into an equivalent sequence of unitaries only compromised of gates from the basis set of the target system. Note, in superconducting QCs, including the SNAIL module, 1Q gates are generally much faster and higher fidelity (e.g., ~10–100× for the SNAIL) than 2Q gates and are often treated as negligible or perfect [30].

Prior work has shown at most only 3 $\sqrt{\texttt{iSWAP}}$ gates plus four interleaved rounds of 1Q gates (given as U_1 to U_8) are required to decompose any 2Q unitary [19], depicted in Eq. 2.



(2)

When direct decomposition for a basis gate has not been solved, an alternative is to use approximate decomposition using numerical optimization techniques. An optimizer is run to minimize the distance, or variance between the requested 2Q gate and the approximation, on the candidate circuit structure. This is done by either converging on a set of 1Q gate parameters or expanding the circuit template to include more target 2Q gate instances [43].

Next, we describe the coupling neighborhoods, formed from the combination of modulator and physical connections.

D. Qubit Coupling and Topologies

To perform both 1Q and 2Q gates, the control signals applied to all qubits and their associated modulators must be unique. For example, in the case of two qubits coupled via a central modulator, a 1Q gate on qubit 1 must operate without creating spurious 2Q gates or driving qubit 2. This is typically accomplished via a combination of (1) spatial selectivity, in which drive lines couple only to a single qubit or modulator and/or (2) frequency selectivity, in which nearest-neighbor qubits have deliberately spaced frequencies to suppress cross-talk among 1Q gates. Accordingly, a graph $G = \{V, E\}$ is used to represent the organization of the QC where physical qubits form the vertices in V and a coupling capability to perform 2Q gates between qubits are edges in E. In practice, connectivity is limited by these requirements and is typically relatively small, with qubits having 2–6 couplings. A complicating factor in discussing QC topology is that *the choice of gate type and coupling topology are not independent*, as they are both determined by the choice of modulator. Thus, in the remainder of this section we will introduce common qubit modulators as comparison points.

1) Cross-resonance gate: The first modulator/gate, originally proposed by IBM [11], is the 'cross-resonance' (CR) or ZX gate. It utilizes the dispersive cross-Kerr interaction [6] between two qubits to realize a 2Q gate. CR drives the second qubit at the first qubit's frequency. In the driven frame, the interaction drive couples the Z-component of the first qubit to an X rotation on the second qubit ($H = Z_1X_2$). This action resembles a CNOT gate but is most often translated into a true CNOT by adding 1Q gates.

$$\texttt{ZX}(\theta) = \begin{bmatrix} \cos\theta/2 & 0 & -i\sin\theta/2 & 0 \\ 0 & \cos\theta/2 & 0 & i\sin\theta/2 \\ -i\sin\theta/2 & 0 & \cos\theta/2 & 0 \\ 0 & i\sin\theta/2 & 0 & \cos\theta/2 \end{bmatrix} \quad (3)$$

This gate has successfully realized high fidelities in large systems, and is used throughout IBM's fleet of QCs. The challenges this gate faces are that: (a) the un-driven cross-Kerr interaction creates Z_1Z_2 errors continuously while off, (b) the qubits should be close in frequency, which does not allow for many-to-many interactions, and (c) given the former, CR gates require very precise fabrication to avoid cross-talk, which has motivated IBM's shift to more sparsely connected Heavy-Hex architectures [10].

2) Direct photon exchange (iSWAP and FSIM gates): Another category of gate is direct photon exchange. In this process, two qubits are coupled resonantly for a period of time to exchange light via the photon-exchange interaction. To form a gate requires turning this interaction on and off. Typically either the qubit frequencies must be tuned together in frequency to exchange, and then far apart to stop the interaction [3], [14], or by using a 'tunable coupler' in between [5]. Direct exchange naturally yields iSWAP-family gates. The coupler approach has been adopted by Google Quantum AI [1] in their Sycamore (SYC) architecture among other groups [7], [25], [35], [46]. SYC gates accrue a phase on the $|11\rangle$ state in addition to iSWAP, termed FSIM given in Eq. 4. θ and ϕ are determined by the pulses applied to the coupler.

$$\texttt{FSIM}(\theta, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{bmatrix} \quad (4)$$

SYC sets $\theta = \pi/2, \phi = \pi/6$. The FSIM gate set yields respectable on/off ratios, but suffers from challenges due to: (a) the difficulty of implementing rapid, extremely precise base-band flux control [20] and strong sensitivity to flux noise in these controls [29], (b) the requirement for equal-frequency qubit and concomitant qubit-qubit crosstalk issues, and (c) the recent demonstration of strong flux-noise based noise and qubit dephasing in the couplers [49].

3) Data Movement on the Topology: Movement between physical qubits is accomplished using non-entangling SWAP gates. A quantum algorithm, represented as a graph $G' = \{V', E'\}$, is mapped to hardware topology by embedding G' in G and inducing SWAP gates when edges cannot be directly realized. As SWAP gates consist themselves of noisy 2Q hardware gates, it is important to minimize the introduced SWAP gate cost to maximize the overall fidelity of the circuit. Increasing the connectivity in the QC topology will reduce the SWAP gate cost overhead compared to a sparsely connected graph, similarly impacting fidelity.

4) Common Qubit Topologies: A simple topology that couples qubits to each of their four nearest-neighbors is the Square-Lattice, shown in Fig. 2a, which is regular and straightforward to expand. IBM's early Penguin machines attempted higher connectivities with diagonals on alternating tiles of the Square-Lattice, shown in Fig. 2c, with limited success due to issues of frequency crowding at the cost of fidelity. For this reason, IBM has over time reduced the connectivity, moving to a Hex-Lattice, shown in Fig. 2d, and now currently to Heavy-Hex topologies, shown in Fig. 2b [33]. All of these topologies have been demonstrated experimentally with varying degrees of success using FSIM and/or CR drive protocols.

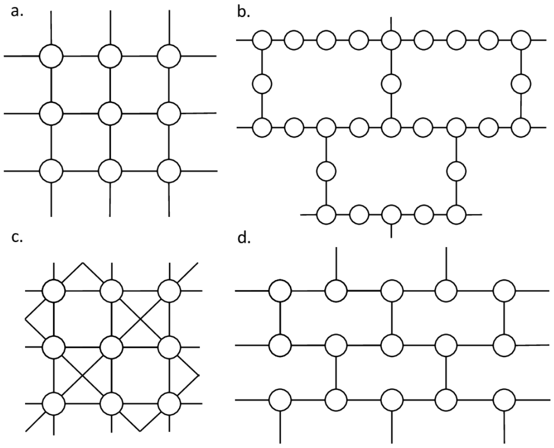


Fig. 2: Standard Qubit Coupling Topologies including (a). Square-Lattice, (b). Heavy-Hex, (c). Lattice with Alternating Diagonals, (d). Hex-Lattice

These topologies are guided by the constraints of 2D circuits whose modulators do not, in general, cross each other or span long distances across the chip. Instead, a richly connected topology of interest in classical networking and parallel computing is the Hypercube. Hypercubes are of interest to qubit coupling topologies because for 2^n nodes, both the

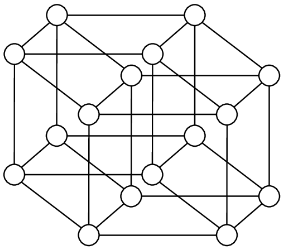


Fig. 3: Hypercube Topology

number of edges incident on every node and the distance between any pair of nodes are exactly n, hence efficiently scaling the neighborhoods of qubit couplings and induced SWAP operations. Implementing such a topology requires us to be able to link a given qubit to n neighbors, which in turn requires a modulator with this connectivity.

Besides just comparing structural properties of each topology, we demonstrate this experimentally in Section III-B to seed our study of the effects of connectivity on computational efficiency. Lattice, Hex-Lattice, and Heavy-Hex topologies and various gates have been examined *independently* to study the efficiency of routing and decomposition algorithms, respectively [31]. In contrast, we demonstrate the value in considering both gate and topology *together* to benefit quantum workloads given the state-of-the-art transpilation—*i.e.,* decomposition, placement, routing—algorithms.

III. Motivation

To develop a NISQ computer with improved fidelity and scalability there are several factors which lead to the need for collaborative design methodology. First, it is desirable to design a machine that has a target gate type in which it is efficient to map relevant quantum algorithms. Second, it is important to provide a flexible topology to minimize the need for SWAP gates. In this section we direct the design of our proposed quantum system described in Section IV.

A. Normalizing Native Gate Sets

To understand the design choice of physically realizable gates we normalize to avoid decision making based on the impact of fabrication engineering effort that impacts underlying gate speed and qubit decoherence and instead establish whether a basis gate is more computationally useful than another. As previously noted, we treat 1Q gates as negligible.

Unlike classical computing, where the primary concerns are performance and energy consumption, the principal concern for quantum calculation is fidelity of the gates which perform computation. Moreover, infidelity in QCs can come from different sources. Some are only present during the gate operation, *e.g.*, driving the qubits to unwanted/error states and the imprecision/instability of the control electronics. Other sources of error are always present, *e.g.*, the loss of information from bits due to decoherence and energy loss. Common measures of gate fidelity, such as those experimentally determined by randomized benchmarking [28], combine the two together, further confusing the issue. However, if one source of error dominates over the other then the strategy for circuit design must change. Qubits which are idle retain their coherence, and a good figure of merit is just the *total number of gates* in the circuit [29]. In contrast, if time is the dominant source of error for all qubits in the system, then *circuit duration* is the best figure of merit, irrespective of the number of gates involved [40], [41], [45]. To address these two scenarios we produce throughout the remainder of the paper two parallel datasets: first, the total gate count, and second, the critical path gate count *i.e.* total circuit duration, for a given circuit size, topology, algorithm, and basis. These metrics of total and critical path gate count provide insight into and normalized comparisons among the expected achievable system fidelities as realizable quantum gate fidelities improve.

Observation 1. We consider decomposition efficiency of the basis gates realized by different modulators to predict their relative success. The choice of basis gate is SYC, CNOT, and $\sqrt{\texttt{iSWAP}}$ for FSIM, CR, and SNAIL modulators, respectively. Both CNOT and $\sqrt{\texttt{iSWAP}}$ require at most three instances to implement an arbitrary 2Q gate, whereas the best known analytical decomposition for SYC requires exactly four [12]. In NISQ machines, data movement via SWAP gates can dominate many algorithms which requires three uses of either CNOT and $\sqrt{\texttt{iSWAP}}$. However, for a random distribution of 2Q gates, the $\sqrt{\texttt{iSWAP}}$ requires only two uses far more often than the CNOT [19], providing a slight information theoretic advantage.

B. Impact of Topology on Data Movement

We have transpiled several quantum benchmarks onto the set of 84-qubit topologies and observed the required SWAP gates induced for data movement, independent of choice of basis gate, shown in Fig. 4. Hex-Lattice and Heavy-Hex perform poorly in important benchmarks such as QV and QAOA. Unlike in the smaller problem sizes where the Square-Lattice is harder to distinguish from the richer topologies, it tends to follow trends of the hex configurations as the size scales. Only hypercube performs the best as size scales. Interestingly, while it does not dramatically reduce total SWAPs for QFT, it scales comparatively better for critical path SWAPs. On average for an 80-qubit QAOA circuit, Heavy-Hex required 1.92×, 1.53×, and 2.83× critical path SWAPs more than Square-Lattice, Lattice+AltDiagonals, and Hypercube, respectively.

Observation 2. Unsurprisingly, topologies with higher connectivity generally scale better than sparse meshes. However, topologies that prioritize reducing distance everywhere rather than dense neighborhoods of connectivity, *i.e.,* avoiding bottlenecks of data movement, are more tolerant to scaling.

IV. Quantum Co-Design with SNAILs

Based on the observations in the prior section, there are several important factors to consider in the co-design a quantum architecture. From observation 1, we should select a basis gate that minimizes the expected duration for decomposed 2Q gates. From observation 2, we should construct a topology that efficiently scales in diameter while providing rich local connectivity. Collectively, our choice of basis gate should be

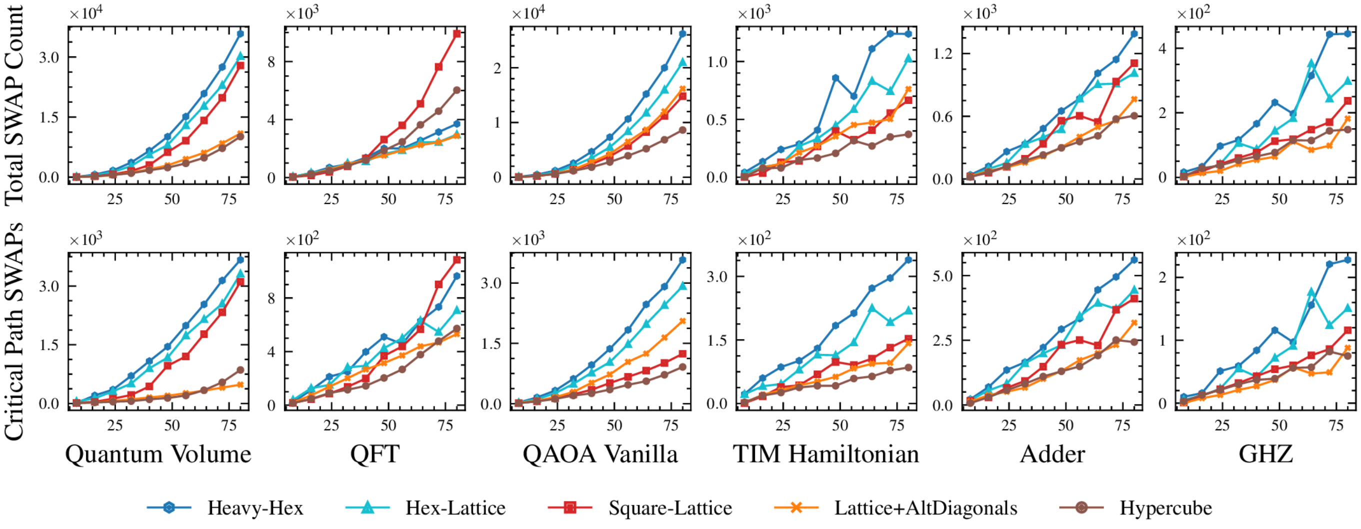


Fig. 4: Total **(top)** and critical path **(bottom)** `SWAP` gates required for 80-qubit implementations on basic topologies. The count of induced `SWAP` gates is independent of the gate set and measures the efficiency of a topology subject to placement and routing transpilation passes.

designed collaboratively with a modulator that allows for increased qubit-qubit coupling neighborhoods. Next, we propose building a novel quantum architecture using the SNAIL quantum modulator. Using the SNAIL allows natively implementing the $\sqrt[n]{\texttt{iSWAP}}$ family and for a rich qubit couplings without frequency crowding. Additionally, the connectivity of SNAILs introduces the exploration of new topology configurations.

A. SNAIL Parametric Modulator

Quantum mechanics describes how a state evolves in time by *unitary* transformations (gates). The interactions of a system, specified by a Hamiltonian, yield a set of allowed energy eigenstates which determine the time-evolution unitary operator. SNAILs offer a way to control the Hamiltonian of superconducting circuit elements, such that the unitary transformations, *i.e.,* quantum gates, are controlled.

The SNAIL is a flux-tunable device which can, for a certain applied flux, create a *strong third-order* Hamiltonian term while canceling *all* 4^{th} *and higher-order even terms*. The third-order term allows many discrete coupling frequencies while the fourth-order and higher even terms *eliminated in the SNAIL* (but found in the CR and other modulators) create crosstalk interactions which lead to *frequency crowding*. The Hamiltonian of a SNAIL can be represented as:

$$H_{SNAIL}/\hbar = \omega_s \hat{s}^\dagger \hat{s} + g_3(\hat{s}^\dagger + \hat{s})^3 \tag{5}$$

When coupled with other linear objects (*e.g.,* harmonic oscillators) and non-linear objects (*e.g.,* qubits), the total system inherits all possible three-body interaction terms from the SNAIL. For instance, driving at the difference of two qubit resonant frequencies $\omega_{\text{drive}} = \omega_1 - \omega_2$ creates the effective interaction:

$$H_{int}^{eff} = g_{12}^{eff}(\hat{a_1}^\dagger \hat{a_2} + \hat{a_1}\hat{a_2}^\dagger) \tag{6}$$

where a, b, s are ladder operators on qubit 1, qubit 2 and the SNAIL, respectively. The $\hat{a_1}^\dagger \hat{a_2} + \hat{a_1}\hat{a_2}^\dagger$ term creates an `iSWAP` relationship between qubits 1 and 2 with a rotation intensity governed by g. If $g = \frac{\pi}{2n}$ radians, the unitary U = $\sqrt[n]{\texttt{iSWAP}}$ follows the transformation matrix:

$$U(t) = e^{iHt/\hbar} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(gt) & i\sin(gt) & 0 \\ 0 & i\sin(gt) & \cos(gt) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Additionally, a strong third-order term results in a higher coupling strength g, which is inversely proportional to time. In other words, a strong pump power, with a faster rate of gate interaction, reduces errors due to decoherence loss.

SNAILs are based on the concept of parametric coupling. IBM's `CR` modulator uses fixed capacitive coupling and Google's `FSIM` modulator uses tunable coupling. The parametric coupling idea of SNAIL modulators has long been used in parametric amplifiers [4], [24], [42], and has recently been used to demonstrate qubit-qubit [36], cavity-cavity [16], and qubit-cavity [8], [32] gates. However, the SNAIL [15] has been designed to *increase the frequency difference to several GHz* for distinguishing qubit-qubit coupling pairs, which increases their *resilience against frequency crowding*. Compared to the $\sim$ hundred MHz differences in cross-resonance systems [17].

In this system, the gate produced is strictly controlled from the driving frequency of the SNAIL; to create an addressable series of gates among many modes each qubit-pair must have a unique frequency difference not shared by another term in the Hamiltonian. Compared to the `CR` modulator, which operates via a cross-Kerr term of order ($\sim$100kHz) [18], SNAIL third-order parametric couplers allow us to create much smaller cross-Kerrs ($\leq$ 1kHz). This results in a rate of 100 times less unwanted accumulating errors due to always on cross-Kerr (or ZZ) interactions. Thus, SNAIL modulators *neatly*

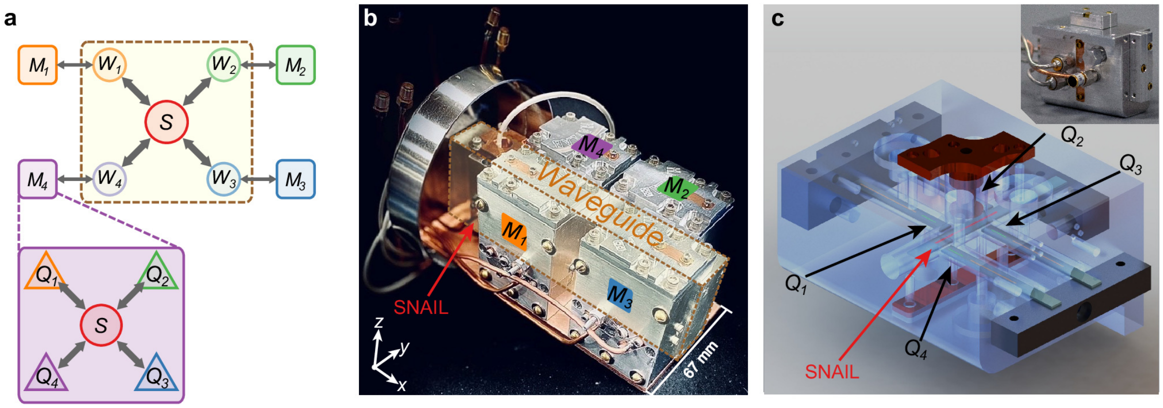


Fig. 5: **Proposed modular QC with quantum router and module** (a) Schematic of one quantum router coupled to four quantum modules. Each module has the same structure as module 4, forming a tree-like architecture. (b) Photograph of the SNAIL based quantum router and four simple modules (adapted from Ref. [27]). (c) Rendered representation and picture of the four qubit SNAIL-based quantum module.

allow operation of multiple gates in parallel in the same neighborhood, or even can create three- or more-mode (≥3Q) gates by applying multiple, simultaneous drivers to the SNAIL.

Combined with low frequency crowding, these features of the SNAIL modulator allow flexible, parallel topologies with many to many superconducting qubit interactions, even across modules each with their own SNAILs. Thus, some qubits can participate in multiple modules as topologies scale.

To characterize the capabilities of a superconducting QC using SNAIL modulators, in previous work [50], we constructed a quantum state router and a number of quantum modules with an overall architecture shown in Fig. 5a. In each module, four qubits are directly coupled to the same SNAIL using unique frequency modes allowing $\sqrt[n]{\texttt{iSWAP}}$ gates between each qubit pair. The router is made with a SNAIL chip placed inside a 3D superconducting waveguide. The SNAIL and the TE_{10k} modes of the waveguide all couple to the SNAIL. Thus, all qubits in module M_k can couple with waveguide mode W_k. W_k is coupled both to the SNAIL in M_k as well as the SNAIL in the quantum state router and can form a gate with any element in either the module or quantum state router.

Fig. 5b shows our preliminary physical implementation of the "Tree," only with each module replaced by a simpler design to evaluate the performance of the router independently. In future experiments, an advanced module prototype depicted in Fig. 5c will be coupled to the router, which, together form the Tree architecture as depicted in Fig. 5a. In each module, a SNAIL couples to all qubits for intra-module communication. Then, the modules are connected to the central SNAIL in the router through the piece marked as the waveguide.

B. Example SNAIL Quantum Computer

The 4-port state router and 4-qubit module sub-systems have been physically realized in two separate experiments. Preliminary router results [27] demonstrate all-to-all exchange interactions among four modules with an average inter-module gate fidelity of $\sim$ 97%. For the 4-qubit module experiment, data from a qubit-qubit exchange is shown in Fig. 6, which

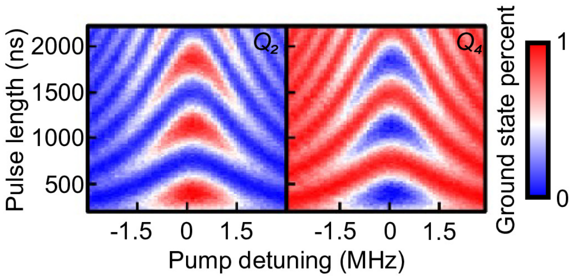


Fig. 6: Parametrically driven exchange between two qubits, Q_2 and Q_4 of the quantum module. Q_2 is prepared in the (red) excited state, and the `iSWAP` process drives exchange between the two devices such that red→blue on left while blue→red on right as time moves up the y-axis (six `iSWAP`s are shown).

shows an excitation swapping between qubits when the SNAIL is pumped at varying durations and detunings. This figure depicts how gate operations are continuous in time, as we see each qubit oppositely alternating states along the y-axis duration of the gate. Note the swapping of red→blue and *vice versa* demonstrate that the router is capable of performing `iSWAP` gates, while the white bands represent the $\sqrt{\texttt{iSWAP}}$. We have also shown that by keeping the W_k device empty, it is possible to build `SWAP` gates using a single `iSWAP`.

The primary fidelity limit in this device is the ratio of gate time to qubit coherence time [50]. Qubit lifetimes depend on their internal loss which comes from many sources such as loss from the metal package or drive ports as well as through coupling to the SNAIL. One source of decoherence is due to flux-noise, which can detune the SNAIL and cause qubits to dephase. Addressing these challenges along with scaling 3D qubit links and increasing gate speeds requires engineering effort to move from devices created in the lab to industry research products on their way to commercialization.

With a similar amount of effort SNAILs can reach pulse speeds and decoherence of other modulators that have received this engineering investment. Evidence from other QCs exploiting similar physics of nonlinear coupling and parametric driving provide confidence of this rapid progress potential [26], [44].In particular, there is headroom in the SNAIL system to increase coherence times by 2–3× [38] and SNAIL-qubit coupling strengths by a similar factor to yield a total fidelity ∼0.99–0.999, competitive with the best existing technologies.

To evaluate core *potential* of these approaches without overemphasizing the engineering effort an implementation has received, we normalize the evaluation to duration in terms of pulses, gate count, etc. as discussed in Section III-A.

C. SNAIL-based Topologies

As we explored the potential of the hypercube topology in Section III, we now explore topologies enabled by the SNAIL with similar properties. For our modular designs, a module will contain one SNAIL with all-to-all connections between its qubits, as described by our current prototype (Fig. 5). Then to realize larger topologies, qubits link between modules and thus are coupled with a connected module's qubits.

The first basic design is the 20-qubit Tree topology, seen in Fig. 7a. The major differences between this topology and the demonstrated prototype in Fig. 5a is that central cyan nodes, which are analogs to the W_k elements, are considered to be qubits rather than cavities; as well as the light-green modules coupling between a feasible 5 qubits instead of 4. Further, the standard Tree design contains bottlenecks in the router qubits, so we explore a theoretical alternative where every qubit in a neighborhood is connected to a different $W_0 \dots W_3$ in an interleaved fashion using colors to denote the qubit connections from the center to the leaf nodes. The goal is to eliminate the bottleneck of the W_k qubits. This design, called a "Interleaved Tree" (Tree-I), Fig. 7b, decreases the maximum distance between all pairs but requires an additional 4 SNAILs and creates some implementation challenges.

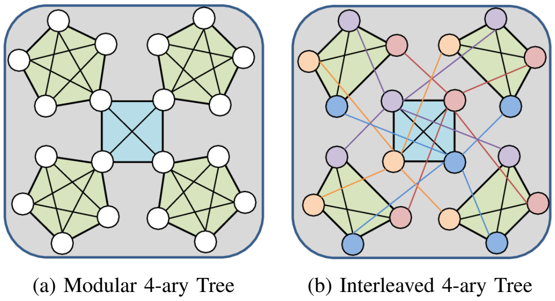

(a) Modular 4-ary Tree (b) Interleaved 4-ary Tree

Fig. 7: Two-level Tree (20-qubit) Topologies

We append additional levels, where each module is connected to the following level's router, to create 84-qubit versions of both the Tree (Fig. 8) and Tree-I topologies. Note, the top (level 3) module of the tree can also be extended with a fifth qubit as with the levels 1 and 2, to connect with a higher level. For the 84-qubit Tree-I, each module couples to a different second-level router qubit, and each second-level router qubit is coupled to a different first-level router qubit in the same interleaved fashion as before.

For our second modular "Corral" design, we aimed to build a topology that maintained the low-diameter property of hypercubes without the required connectivity dimension scaling. In Fig. 9, red vertical cylinders represent SNAILs and green or yellow horizontal bars are the qubits coupled between them. By building a octagonal ring of modules, each with 4 qubits, is defined by a pattern of fence-post connections. Whereas Fig. 9a is the easiest to physically realize, with each qubit coupled to the nearest adjacent SNAIL, denoted Corral$_{1,1}$, and topology shown in Fig. 9b, we might also realize differing strides, reminiscent of the hypercube. The Corral$_{1,2}$ shown in Fig. 9c, connects its second fence to the second-nearest neighbor, decreasing the average distance between all pairs of qubits. The resulting topology is shown in Fig. 9d.

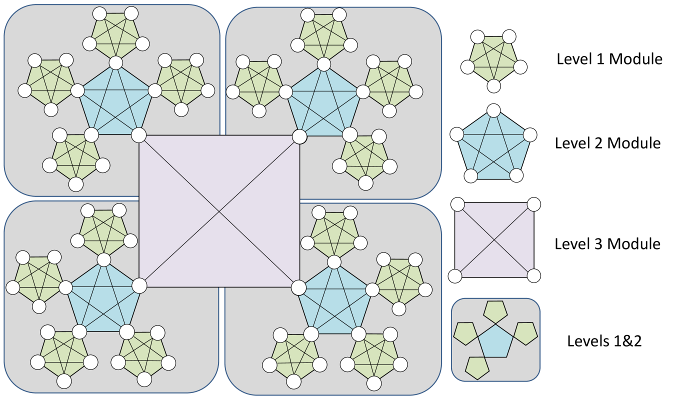
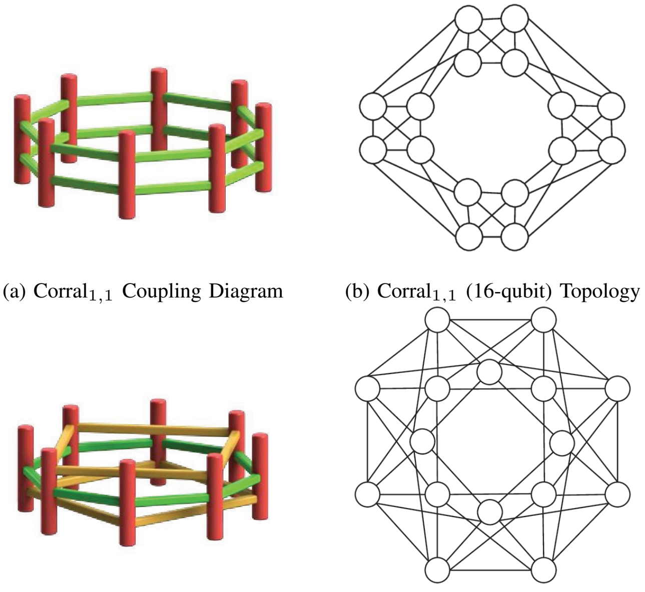


Fig. 8: Three-level Tree (84-qubit) Topology


(a) Corral$_{1,1}$ Coupling Diagram (b) Corral$_{1,1}$ (16-qubit) Topology

(c) Corral$_{1,2}$ Coupling Diagram (d) Corral$_{1,2}$ (16-qubit) Topology

Fig. 9: Corral SNAIL-qubit architectures

Despite initially appearing similar to a ring, we note that the Corral topologies actually exhibit a coupling structure similar to a 4-D hypercube. Corrals could be scaled by adding more posts (modules) in the ring; or with more complex designs that connect Corrals with Tree-like modules or layouts with Corrals in a lattice pattern. In the next section we describe our methodology to compare SNAIL-based NISQ QCs using our proposed modular topologies against QCs using `CR` (IBM) and `FSIM` (Google) modulators.

V. Experimental Setup

We evaluate the SNAIL-based superconducting QC using the metrics for normalization as described in Section III-A. For simplicity, we presume all gates have uniform fidelity. We explore two sizes of machine: one of the scale of our hardware prototype with 16-20 qubits (QBs) and a scaled size of 84 qubits. These designs are summarized in Table I. The table includes the number of nodes (qubits), the diameter of the topology (Dia.), the average distance between any two qubits (Avg_D), and the average connectivity of a qubit (Avg_C). Tree, Tree-I, $\text{Corral}_{1,1}$, and $\text{Corral}_{1,2}$, are all topologies realizable by the SNAIL. Square-Lattice is included as a baseline and hypercube is included for comparison against the corrals. For 84 qubits, Tree and Tree-I use a third level router (Section IV-C). We retain the hypercube as a Corral-like design to see the potential of building larger scaled Corrals in addition to scaled Heavy-Hex, Hex-Lattice, Square-Lattice, and Lattice+AltDiag as comparison points.

TABLE I: Topologies and Connectivities

	QBs	**Dia.**	**Avg_D**	**Avg_C**	**QBs**	**Dia.**	**Avg_D**	**Avg_C**
Heavy-Hex	20	8.0	3.77	2.1	84	21.0	8.47	2.26
Hex-Lattice	20	7.0	3.37	2.45	84	17.0	6.95	2.71
Square-Lattice	16	6.0	2.5	3.0	84	17.0	6.26	3.55
Lattice+AltDiag	-	-	-	-	84	11.0	4.62	5.12
Tree	20	3.0	2.15	4.6	84	5.0	3.91	4.71
Tree-I	20	3.0	2.03	4.6	84	5.0	3.65	4.71
$\text{Corral}_{1,1}$	16	4.0	2.06	5.0	-	-	-	-
$\text{Corral}_{1,2}$	16	2.0	1.5	6.0	-	-	-	-
Hypercube	16	4.0	2.0	4.0	84	7.0	3.32	6.0

To generate circuits for these QCs, we extended the Qiskit Terra 0.20.0 transpiler. We provided new backends that support analytical $\sqrt{\texttt{iSWAP}}$ and SYC decompositions using Cartan's KAK method. We also extended the transpiler to include our proposed Tree, Tree-I, Corrals, Hypercube, and Lattice+AltDiagonals topologies. For design-space exploration we ensure that each basis gate can be assigned to each topology. Note Qiskit already includes an Cartan's KAK CX decomposition backend, as well as Square-, Hex-Lattice, and Heavy-Hex topologies. We use Qiskit's DenseLayout for initial qubit mapping and StochasticSwap for routing SWAPs.

We tested the machine configurations with workloads that include widely used quantum circuits that can be scaled to different problem sizes. Our parameterized circuits are QuantumVolume, QFT, and CDKMRippleCarryAdder from Qiskit and QAOAVanillaProxy, HamiltonianSimulation, and GHZ from Supermarq [47]. We select these circuit benchmarks over other popular algorithms, such as VQE, because they can be parameterized as a function of qubit size and be generated automatically, while VQE or other similar benchmarks would require hand-coded designs for all problem sizes.

During transpilation we collect 4 sets of data over each backend, for each circuit of incremental size. We use Qiskit's functionality to count total gates and critical path gates. After the routing pass, we count the total induced SWAP and critical path SWAP gates. After the final basis translation pass, we count the total 2Q gates and critical path 2Q gates. Note, we use 2Q basis gate count and its associated pulse duration as a surrogate for determining overall reliability as described in Section III-A. These experimental settings were used to generate the charts shown in Section III and next in Section VI.

VI. Results

To evaluate the potential of the SNAIL QCs we explore the impact of the newly possible topologies similarly to Sec. III, in a gate agnostic fashion, and then combine the topology and gate impacts to represent the full co-design advantage. Finally, we explore the potential of other basis gates in the $\sqrt[n]{\texttt{iSWAP}}$ family by investigating fidelity of $n > 2$ decompositions.

A. Evaluation of SNAIL-enabled topologies

We evaluate the proposed SNAIL-coupling corral topologies against the previously discussed topologies, shown in Fig. 10. The scaling of total and critical SWAP gates moderately obeys the expected ordering, as when average connectivity goes up and average distance goes down, less SWAP gates are required. Despite unsteady trends on small circuit sizes, the corral topologies are unambiguously the best performers. Noticeably, the transpiler finds an initial mapping that often requires zero SWAP gates for $\text{Corral}_{1,1}$, indicating its rich connectivity.

To extrapolate to larger topologes, we revisit the topologies from Section III-B, now including the SNAIL topologies Tree and Tree-I, shown in Fig. 11. Hex-Lattice and Lattice+Diag are not shown for readability and redundancy. This gives us a comparison relating to the previous set of benchmarks. Once again, the constant properties of the topologies appear to generally coincide with performance. In fact, for an 80-qubit QV circuit, we compute from Heavy-Hex to Tree a 54.3% decrease in total SWAP gates or a 79.8% decrease in critical path SWAP gates. However, the Tree designs do not quite match the performance of the hypercube, as from Tree to hypercube experiences an additional 42.5% decrease in total SWAP or 54.3% decrease of critical path SWAP gates.

B. Evaluation of Collaborative Design

Next, we continue the decomposition into each topology's basis gate, to count the total pulse duration of the circuit, shown in Fig. 12 and Fig. 13. As noted before, there is a small advantage to$\sqrt{\texttt{iSWAP}}$ over CNOT, and CNOT over SYC, effectively adding a scaling factor enough to bring the SYC gate on Square-Lattice above the CR gate on Heavy-Hex. This data exhibits critical parallelism on the topologies, *i.e.* when a curve flattens from total gate count to duration, more gates are not contributing to the duration and therefore are in parallel time steps. As an example, the Tree on the QV benchmark flattens its total 2Q count onto the duration plot, suggesting a comparatively high degree of gate parallelism. Finally, our co-designed Corral topology combined with the SNAIL-enabled $\sqrt{\texttt{iSWAP}}$ consistently outperforms others on all benchmarks.

We conducted a fidelity simulation on the transpiled circuits based on the product of fidelities over each qubit [26] to achieve a total circuit fidelity of 0.9 and 0.5, respectively and reported in Table II along with the fidelity improvement ratios of our co-designed systems normalized to Heavy-Hex-CR. The gates require $< 10^{-4}$ infidelity to achieve these circuit

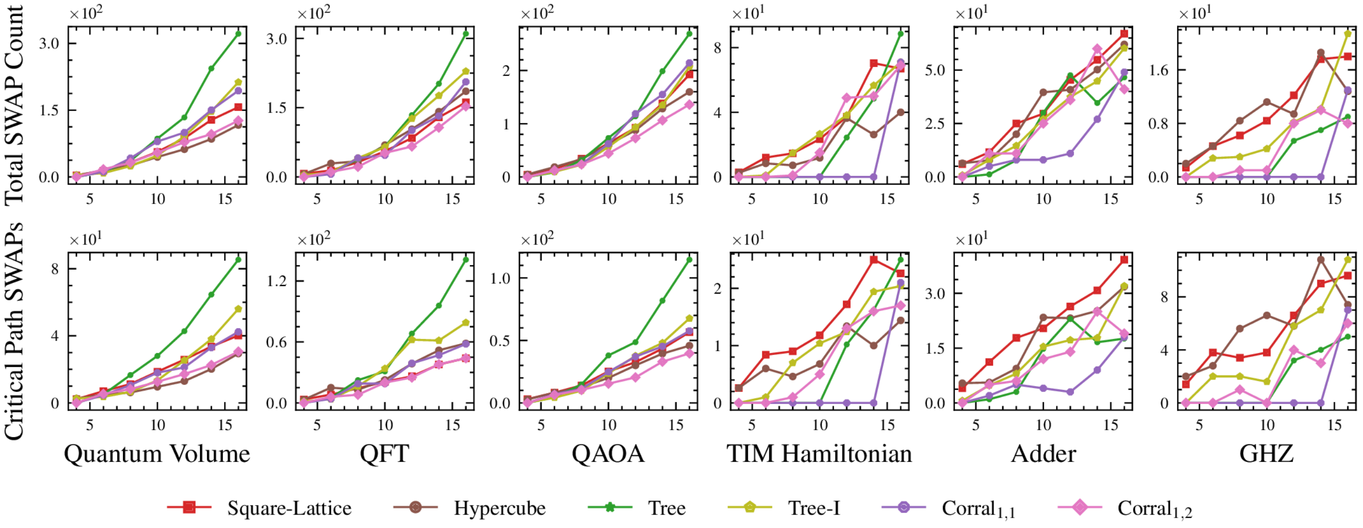


Fig. 10: Total **(top)** and critical path **(bottom)** SWAP gates required for 16-qubit implementations of proposed SNAIL topologies.

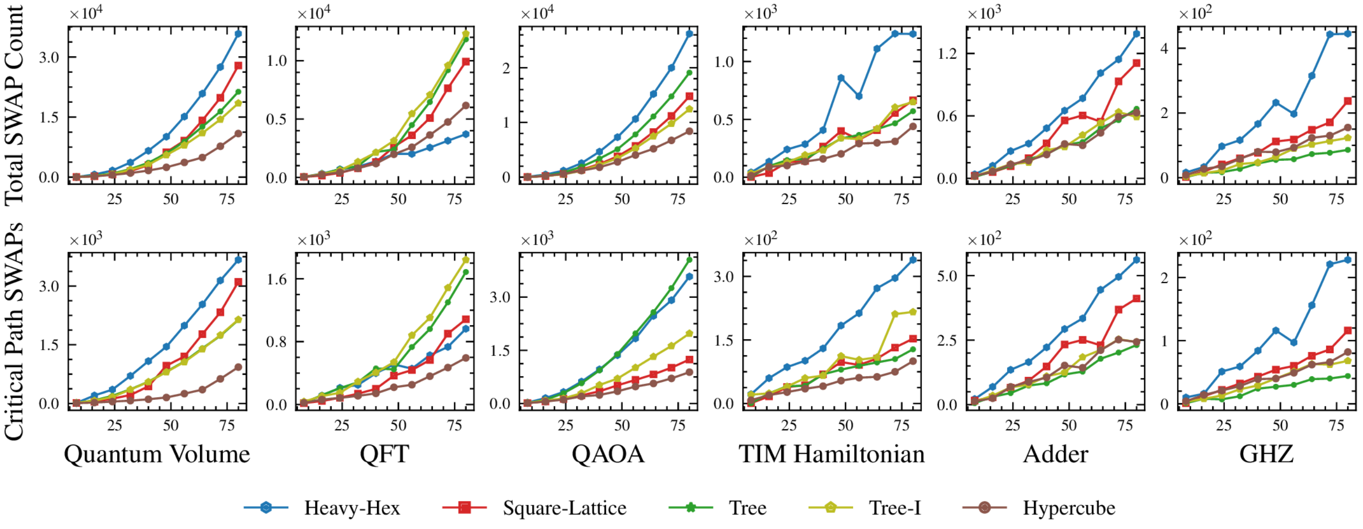


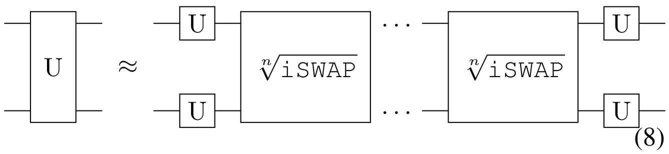
Fig. 11: Total **(top)** and critical path **(bottom)** SWAP gates required for 84-qubit implementations comparing proposed SNAIL topologies against common topology baseline.

fidelities, highlighting why current production NISQ machines at $\geq 10^{-3}$ are limited to shallow depth. Corrals can provide 1.27–2.99× improvement, often beating a Hypercube, while Trees also show benefits reaching $> 2\times$; each gain allows circuit depth to increase by similar factors. Lattice-FSIM's lower fidelity is from inefficient FSIM decomposition.

When scaling to larger topologies, Fig. 13 we see less consistent trends. The transpilation heuristics appear more noisy with problem size, *i.e.* gate counts are not always monotonic on the same topologies, indicating inefficient routing. The variability on a topology and might temper excitement about a singular benchmark succeeding without evidence that more workloads also have positive trends. Nonetheless, the hypercube is generally among the best for all benchmarks. Finally, modular QCs could eventually have reconfigurable architectures changing on an application basis to adapt to the application.

C. Pulse Duration Sensitivity Study

Shown in Eq. 7, using the SNAIL to realize gate couplings yields a $\sqrt[n]{\texttt{iSWAP}}$ family of gates. $\sqrt{\texttt{iSWAP}}$ has been studied as a naturally good candidate for forming a basis, which decreases the basis gate pulse duration by half. However, exploring even smaller fractions of iSWAP for decomposition to decrease decoherence time has not been studied. While $\sqrt{\texttt{iSWAP}}$ is the smallest fraction that is a "perfect entangler" [19], decompositions to shorter $\sqrt[n]{\texttt{iSWAP}}$ gates may still result in high-fidelity decompositions. No analytical decomposition to these has been discovered, thus we use an approximate decomposition engine to explore this possibility.



(8)

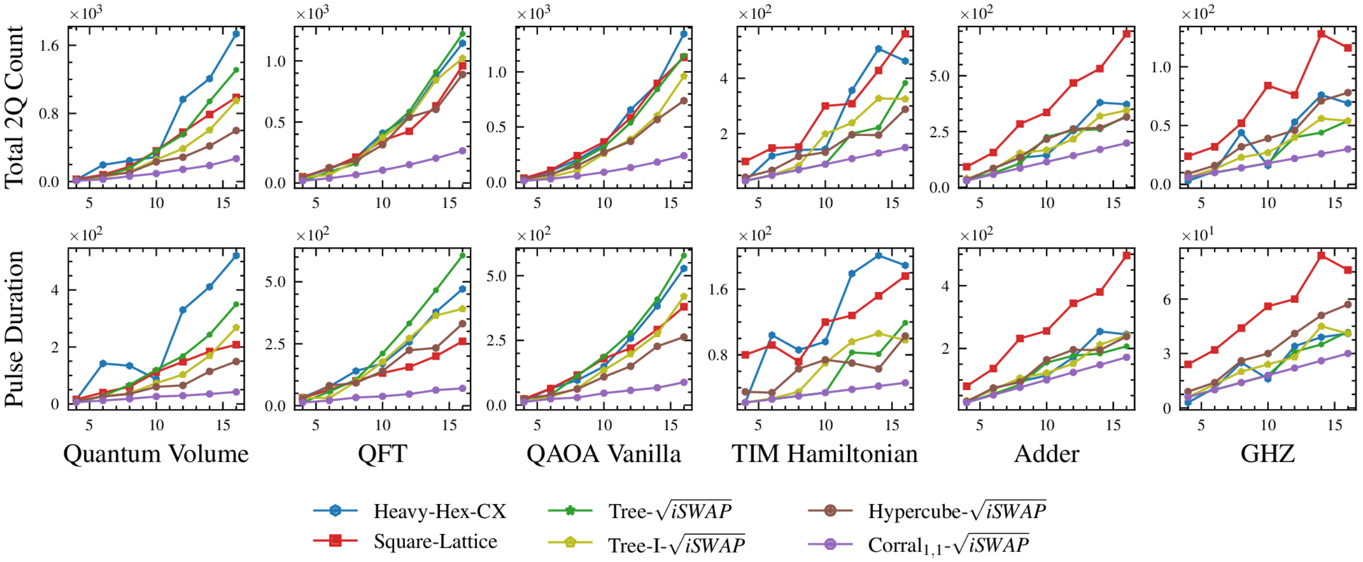


Fig. 12: Total **(top)** and critical path **(bottom)** 2Q gate counts, decomposed into the respective native basis sets, required for 16-qubit implementations comparing proposed SNAIL topologies against common topology baseline.

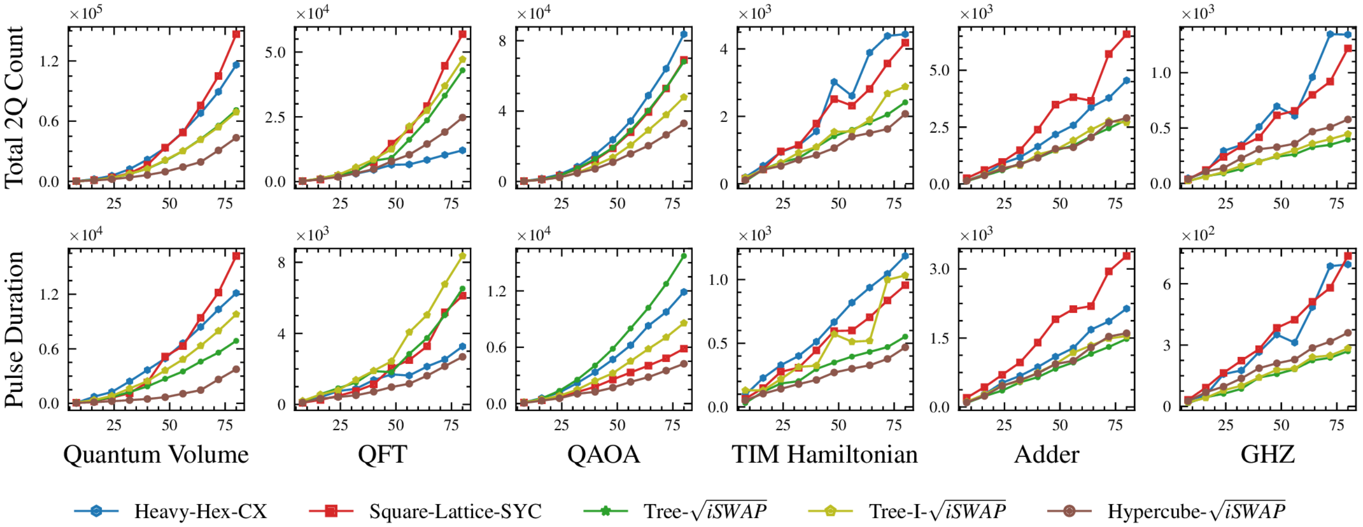


Fig. 13: Total **(top)** and critical path **(bottom)** 2Q gate counts, decomposed into the respective native basis sets, required for 84-qubit implementations comparing proposed SNAIL topologies against common topology baseline.

We reproduce a version of NuOp [23], [37] to build template circuits, which interleave the desired $\sqrt[n]{\texttt{iSWAP}}$ gate with 1Q gates (Eq. 8) similar to the exact decomposition method in Section II-C. However, because the decomposition is approximate, this introduces another form of error beyond decoherence, the error from the decomposition approximation. Thus, the similarity between unitaries, used as the decomposition fidelity is defined using the Hilbert-Schmidt inner product between the template and target from Eq. 9.

$$F_d(U_d, U_t) = \frac{\mathrm{Tr}(U_d^\dagger U_t)}{\dim(U_d)} \tag{9}$$

However, given our goal is to improve fidelity by reducing decoherence, we include an approximation that decoherence scales linearly over time, as shown in Eq. 10. To illustrate, consider an `iSWAP` duration that reduces fidelity to 90%. A gate with half the duration has approximately half the decoherence, hence infidelity is reduced from 10%, to 5%, yielding a 95% fidelity $\sqrt{\texttt{iSWAP}}$ gate.

$$F_b(\sqrt[n]{\texttt{iSWAP}}) = 1 - \frac{1 - F_b(\texttt{iSWAP})}{n} \tag{10}$$

As a result, the best total fidelity of the unitary decomposition is the product of the fidelity relating to the total duration of k applications of the basis gate times the fidelity relating to the approximate decomposition error of each of the k basis gates, described in Eq. 11. As before, we ignore the delay of the 1Q gates. To study this, we generated circuits for the "Haar distribution" of 2Q unitaries, which is a weighted average distribution over the quantum computational space. We use this find the best total fidelity, then iterate the template size k to find the best approximate decomposition.

$$F_t = \max_k F_d^{(k)} (F_b)^k \tag{11}$$

TABLE II: Required gate fidelity on different co-designed, circa 20Q architectures computed using an adapted technique from the literature [26] to achieve 0.9 and 0.5 total circuit fidelity. "Ratio" is ratio of infidelity versus Heavy-Hey-CR.

	QV			QFT			QAOA			TIM			Adder			GHZ		
	0.9	0.5	Ratio	0.9	0.5	Ratio	0.9	0.5	Ratio	0.9	0.5	Ratio	0.9	0.5	Ratio	0.9	0.5	Ratio
Corral$_{1,1}$ $-\sqrt{\texttt{iSWAP}}$	0.99992	0.9995	2.30	0.99992	0.9995	1.63	0.99993	0.9996	1.54	0.99984	0.9990	1.31	0.99984	0.9989	1.32	0.99924	0.9950	1.74
Corral$_{1,2}$ $-\sqrt{\texttt{iSWAP}}$	0.99990	0.9993	2.99	0.99991	0.9994	1.73	0.99991	0.9994	2.01	0.99980	0.9987	1.65	0.99984	0.9990	1.28	0.99908	0.9939	2.10
Hypercube-$\sqrt{\texttt{iSWAP}}$	0.99990	0.9994	2.88	0.99991	0.9994	1.71	0.99991	0.9994	2.01	0.99982	0.9988	1.50	0.99987	0.9992	1.01	0.99937	0.9959	1.43
Tree-$\sqrt{\texttt{iSWAP}}$	0.99995	0.9997	1.45	0.99994	0.9996	1.20	0.99995	0.9997	1.18	0.99984	0.9989	1.35	0.99986	0.9991	1.12	0.99912	0.9942	2.00
Tree-I-$\sqrt{\texttt{iSWAP}}$	0.99992	0.9995	2.31	0.99993	0.9995	1.45	0.99992	0.9995	1.88	0.99983	0.9989	1.40	0.99987	0.9991	1.08	0.99930	0.9954	1.60
Lattice-FSIM	0.99995	0.9997	1.35	0.99995	0.9997	0.96	0.99996	0.9997	0.96	0.99990	0.9993	0.85	0.99993	0.9995	0.60	0.99966	0.9978	0.77
Heavy-Hex-CR	0.99997	0.9998	1.00	0.99995	0.9997	1.00	0.99996	0.9997	1.00	0.99988	0.9992	1.00	0.99988	0.9992	1.00	0.99956	0.9971	1.00

As evidenced by Fig. 14(top left), smaller fractional $\sqrt[n]{\texttt{iSWAP}}$ need more repetitions (larger k) to reach near-exact decompositions, visible by reaching $< 10^{-6}$ fidelity with higher values of k. However, the need for more gates is generally outweighed by the shorter basis gate durations, reducing the overall total pulse duration. For example, given high-fidelity decomposition for $\sqrt{\texttt{iSWAP}}$ in $k = 3$ and $\sqrt[3]{\texttt{iSWAP}}$ in $k = 4$, the total duration is reduced from 1.5 to 1.33. This is verified again in Fig. 14(top right), where as n grows, the total pulse duration decreases. Fig. 14(bottom) shows the total fidelity of decoherence and approximate decomposition with decoherence due to iSWAP pulse length on the x-axis and total fidelity on the y-axis. We find that for Haar sampled 2Q unitaries and for a 99% fidelity iSWAP basis, while a $\sqrt{\texttt{iSWAP}}$ basis reduces infidelity by 51%, $\sqrt[4]{\texttt{iSWAP}}$ provides a 58% reduction, each compared to an iSWAP basis gate. This evidence continues to support the SNAIL modulator for its realization of a powerful basis set, with the ability to modify duration of the continuous operator to maximize gate fidelities.

VII. Conclusion

In this work, we demonstrate the data movement overheads and penalties from lattice-based NISQ machines on a range of algorithms. The SNAIL-based modulator and modular architectures provide significant improvements over 2D lattices, particularly for smaller node sizes, whereas a directly scalable Tree structure shows mixed performance over the benchmark workloads for larger node sizes. We also observe that a hypercube structure, which has rich local connections and low diameter, is superior to both lattices and the Tree and its variants, with exception of the highly ordered GHZ-state creation. We found that on an average of Quantum Volume circuits ranging from 16 to 80 qubits, a hypercube topology induces 2.57× less total SWAP gates and 5.63× less critical path SWAP gates compared to Heavy-Hex. We have explored hypercube inspired 'Corral' structures which are both feasible given the current 4-way coupling capabilities of SNAIL modulators for 16 nodes and provide superior computational performance particularly when coupled with $\sqrt{\texttt{iSWAP}}$ (see Fig. 12).

All of our connectivity designs are physically realizable with our SNAIL modulators, and represent excellent targets for next-generation QCs. Our results point to the need for both dense connectivity and a mix of short- and long-range links for future NISQ QCs. Finally, the strong performance of $\sqrt{\texttt{iSWAP}}$, which is native to the SNAIL modulator, inspired us to explore whether smaller fractions of $\sqrt[n]{\texttt{iSWAP}}$ can yield superior approximate implementations. We found that for 2Q unitaries and for a 99% fidelity iSWAP basis, $\sqrt[4]{\texttt{iSWAP}}$ decreases infidelity on average by 58% compared to iSWAP, leading to further co-design advancements.

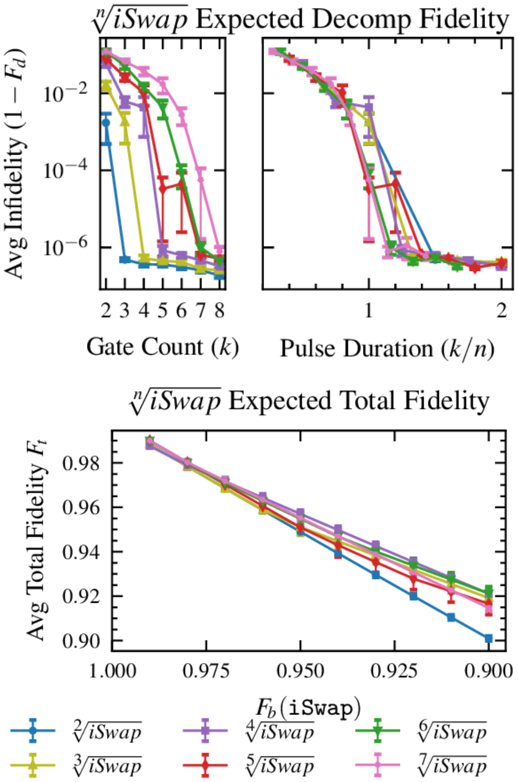


Fig. 14: Fidelity comparisons for Haar-random 2Q unitaries on a set of $\sqrt[n]{\texttt{iSWAP}}$ basis gates (N=50).

In future work, exploring methods to scale Corrals or develop new SNAIL-realizable topologies to compete with aspirational hypercube topologies for larger numbers of qubits and compatibility with error-correcting codes are important next steps.

Acknowledgements

This work is partially supported by the Laboratory of Physical Sciences and NSF Award CNS-1822085. MX, MJH, and PL are partially supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers Co-Design Center for Quantum Advantage under contract DE-SC0012704.

References

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019. [Online]. Available: https://doi.org/10.1038/s41586-019-1666-5

[2] A. Barenco, "A universal two-bit gate for quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 449, no. 1937, pp. 679–683, 1995.

[3] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, O. P., P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, "Superconducting quantum circuits at the surface code threshold for fault tolerance," *Nature*, vol. 508, no. 7497, pp. 500–503, 2014.

[4] N. Bergeal, R. Vijay, V. E. Manucharyan, I. Siddiqi, R. J. Schoelkopf, S. M. Girvin, and M. H. Devoret, "Analog information processing at the quantum limit with a Josephson ring modulator," *Nature Phys*, vol. 6, no. 4, pp. 296–302, Apr. 2010. [Online]. Available: https://www.nature.com/articles/nphys1516

[5] R. Bialczak, M. Ansmann, M. Hofheinz, M. Lenander, E. Lucero, M. Neeley, A. O'Connell, D. Sank, H. Wang, M. Weides, J. Wenner, T. Yamamoto, A. Cleland, and J. Martinis, "Fast tunable coupler for superconducting qubits," *Physical review letters*, vol. 106, no. 6, p. 060501, 2011.

[6] A. Blais, A. L. Grimsmo, S. Girvin, and A. Wallraff, "Circuit quantum electrodynamics," *Reviews of Modern Physics*, vol. 93, no. 2, p. 025005, 2021.

[7] A. Blais, A. M. van den Brink, and A. M. Zagoskin, "Tunable coupling of superconducting qubits," *Physical review letters*, vol. 90, no. 12, p. 127901, 2003.

[8] L. D. Burkhart, J. D. Teoh, Y. Zhang, C. J. Axline, L. Frunzio, M. Devoret, L. Jiang, S. Girvin, and R. Schoelkopf, "Error-detected state transfer and entanglement in a superconducting quantum network," *PRX Quantum*, vol. 2, no. 3, p. 030321, 2021.

[9] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, "Topological and subsystem codes on low-degree graphs with flag qubits," *Physical Review X*, vol. 10, no. 1, p. 011022, 2020.

[10] J. Chow, O. Dial, and J. Gambetta, "Ibm quantum breaks the 100-qubit processor barrier," *IBM Research Blog, available in https://research.ibm.com/blog/127-qubit-quantum-process-or-eagle*, 2021.

[11] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, "Simple all-microwave entangling gate for fixed-frequency superconducting qubits," *Physical review letters*, vol. 107, no. 8, p. 080502, 2011.

[12] G. E. Crooks, "Gates, states, and circuits," 2020.

[13] M. H. Devoret and R. J. Schoelkopf, "Superconducting circuits for quantum information: an outlook," *Science*, vol. 339, no. 6124, pp. 1169–1174, 2013.

[14] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin, , and R. Schoelkopf, "Demonstration of two-qubit algorithms with a superconducting quantum processor," *Nature*, vol. 460, no. 7252, pp. 240–244, 2009.

[15] N. Frattini, U. Vool, S. Shankar, A. Narla, K. Sliwa, and M. Devoret, "3-wave mixing josephson dipole element," *Applied Physics Letters*, vol. 110, no. 22, p. 222603, 2017.

[16] Y. Y. Gao, B. J. Lester, K. S. Chou, L. Frunzio, M. H. Devoret, L. Jiang, S. Girvin, and R. J. Schoelkopf, "Entanglement of bosonic modes through an engineered exchange interaction," *Nature*, vol. 566, no. 7745, pp. 509–512, 2019.

[17] J. B. Hertzberg, E. J. Zhang, S. Rosenblatt, E. Magesan, J. A. Smolin, J.-B. Yau, V. P. Adiga, M. Sandberg, M. Brink, J. M. Chow *et al.*, "Laser-annealing josephson junctions for yielding scaled-up superconducting quantum processors," *npj Quantum Information*, vol. 7, no. 1, pp. 1–8, 2021.

[18] K. Heya and N. Kanazawa, "Cross-cross resonance gate," *PRX Quantum*, vol. 2, no. 4, p. 040336, 2021.

[19] C. Huang, D. Ding, F. Wu, L. Kong, F. Zhang, X. Ni, Y. Shi, H.-h. Zhao, and J. Chen, "Towards ultra-high fidelity quantum operations: Sqisw gate as a native two-qubit gate," *arXiv preprint arXiv:2105.06074*, 2021.

[20] D. Kafri, C. Quintana, Y. Chen, A. Shabani, J. M. Martinis, and H. Neven, "Tunable inductive coupling of superconducting qubits in the strongly nonlinear regime," *Phys. Rev. A*, vol. 95, p. 052333, May 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.95.052333

[21] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," *Nature*, vol. 567, no. 7749, pp. 491–495, 2019.

[22] J. Koch, M. Y. Terri, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Charge-insensitive qubit design derived from the cooper pair box," *Physical Review A*, vol. 76, no. 4, p. 042319, 2007.

[23] L. Lao, P. Murali, M. Martonosi, and D. Browne, "Designing calibration and expressivity-efficient instruction sets for quantum computing," *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, Jun 2021. [Online]. Available: http://dx.doi.org/10.1109/ISCA52012.2021.00071

[24] F. Lecocq, L. Ranzani, G. Peterson, K. Cicak, R. Simmonds, J. Teufel, and J. Aumentado, "Nonreciprocal microwave signal processing with a field-programmable Josephson amplifier," *Physical Review Applied*, vol. 7, no. 2, p. 024028, 2017.

[25] X. Li, T. Cai, H. Yan, Z. Wang, X. Pan, Y. Ma, W. Cai, J. Han, Z. Hua, X. Han, Y. Wu, H. Zhang, H. Wang, Y. Song, L. Duan, and L. Sun, "Tunable coupler for realizing a controlled-phase gate with dynamically decoupled regime in a superconducting circuit," *Physical Review Applied*, vol. 14, no. 2, p. 024070, 2020.

[26] S. F. Lin, S. Sussman, C. Duckering, P. S. Mundada, J. M. Baker, R. S. Kumar, A. A. Houck, and F. T. Chong, "Let each quantum bit choose its basis gate," *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1042–1058, Oct 2022.

[27] P. Lu, C. Zhou, M. Xia, R. Kaufman, I. Yusuf, P. Patel, B. Mesits, M. Mucci, and M. Hatridge, "A quantum module with all-to-all gates via parametric control," *Bulletin of the American Physical Society*, 2022.

[28] E. Magesan, J. M. Gambetta, and J. Emerson, "Characterizing quantum gates via randomized benchmarking," *Physical Review A*, vol. 85, no. 4, p. 042311, 2012.

[29] T. McCourt, C. Neill, K. Lee, C. Quintana, Y. Chen, J. Kelly, V. Smelyanskiy, M. Dykman, A. Korotkov, I. L. Chuang, and A. Petukhov, "Learning noise via dynamical decoupling of entangled qubits," *arXiv preprint arXiv:2201.11173*, 2022.

[30] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient z gates for quantum computing," *Physical Review A*, vol. 96, no. 2, Aug 2017. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.96.022330

[31] P. Murali, J. M. Baker, A. J. Abhari, F. T. Chong, and M. Martonosi, "Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers," 2019.

[32] A. Narla, S. Shankar, M. Hatridge, Z. Leghtas, K. M. Sliwa, E. Zalys-Geller, S. O. Mundhada, W. Pfaff, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret, "Robust concurrent remote entanglement between two superconducting qubits," *Physical Review X*, vol. 6, no. 3, p. 031036, 2016.

[33] P. Nation, H. Paik, A. Cross, and Z. Nazario, "The ibm quantum heavy hex lattice," Jul 2021. [Online]. Available: https://research.ibm.com/blog/heavy-hex-lattice

[34] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[35] A. Niskanen, K. Harrabi, F. Yoshihara, Y. Nakamura, S. Lloyd, and J. S. Tsai, "Quantum coherent tunable coupling of superconducting qubits," *Science*, vol. 316, no. 5825, pp. 723–726, 2007.

[36] A. Noguchi, A. Osada, S. Masuda, S. Kono, K. Heya, S. P. Wolski, H. Takahashi, T. Sugiyama, D. Lachance-Quirion, and Y. Nakamura, "Fast parametric two-qubit gates with suppressed residual interaction using the second-order nonlinearity of a cubic transmon," *Physical Review A*, vol. 102, no. 6, p. 062408, 2020.

[37] E. C. Peterson, G. E. Crooks, and R. S. Smith, "Fixed-depth two-qubit circuits and the monodromy polytope," *Quantum*, vol. 4, p. 247, 2020.

[38] A. P. Place, L. V. Rodgers, P. Mundada, B. M. Smitham, M. Fitzpatrick, Z. Leng, A. Premkumar, J. Bryon, A. Vrajitoarea, S. Sussman, G. Cheng, T. Madhavan, H. K. Babla, X. H. Le, Y. Gang, B. Jäck, A. Gyenis, N. Yao, R. Cava, N. P. de Leon, and A. A. Houck, "New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds," *Nature communications*, vol. 12, no. 1, pp. 1–6, 2021.

[39] J. Preskill, "Quantum computing 40 years later," *arXiv preprint arXiv:2106.10522*, 2021.

[40] M. Rol, C. C. Bultink, T. E. O'Brien, S. De Jong, L. S. Theis, X. Fu, F. Luthi, R. F. Vermeulen, J. De Sterke, A. Bruno, D. Deurloo, R. N. Schouten, R. N. Wilhelm, and L. DiCarlo, "Restless tuneup of high-fidelity qubit gates," *Physical Review Applied*, vol. 7, no. 4, p. 041001, 2017.

[41] S. Sheldon, L. S. Bishop, E. Magesan, S. Filipp, J. M. Chow, and J. M. Gambetta, "Characterizing errors on qubit operations via iterative randomized benchmarking," *Physical Review A*, vol. 93, no. 1, p. 012301, 2016.

[42] K. Sliwa, M. Hatridge, A. Narla, S. Shankar, L. Frunzio, R. Schoelkopf, and M. Devoret, "Reconfigurable Josephson circulator/directional amplifier," *Physical Review X*, vol. 5, no. 4, p. 041020, 2015.

[43] E. Smith, M. G. Davis, J. M. Larson, E. Younis, C. Iancu, and W. Lavrijsen, "Leap: Scaling numerical optimization based synthesis using an incremental approach," *arXiv preprint arXiv:2106.11246*, 2021.

[44] K. N. Smith, G. S. Ravi, J. M. Baker, and F. T. Chong, "Scaling superconducting quantum computers with chiplet architectures," *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct 2022.

[45] A. Somoroff, Q. Ficheux, R. A. Mencia, H. Xiong, R. V. Kuzmin, and V. E. Manucharyan, "Millisecond coherence in a superconducting qubit," *arXiv preprint arXiv:2103.08578*, 2021.

[46] Y. Sung, L. Ding, J. Braumüller, A. Vepsäläinen, B. Kannan, M. Kjaergaard, A. Greene, G. O. Samach, C. McNally, D. Kim *et al.*, "Realization of high-fidelity cz and z z-free iswap gates with a tunable coupler," *Physical Review X*, vol. 11, no. 2, p. 021058, 2021.

[47] T. Tomesh, P. Gokhale, V. Omole, G. S. Ravi, K. N. Smith, J. Viszlai, X.-C. Wu, N. Hardavellas, M. R. Martonosi, and F. T. Chong, "Supermarq: A scalable quantum benchmark suite," *arXiv preprint arXiv:2202.11045*, 2022.

[48] C. Wang, X. Li, H. Xu, Z. Li, J. Wang, Z. Yang, Z. Mi, X. Liang, T. Su, C. Yang, G. Wang, W. Wang, Y. Li, M. Chen, C. Li, K. Linghu, J. Han, Y. Zhang, Y. Feng, Y. Song, T. Ma, J. Zhang, R. Wang, P. Zhao, W. Liu, G. Zue, Y. Jin, and H. Yu, "Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds," *npj Quantum Information*, vol. 8, no. 1, pp. 1–6, 2022.

[49] F. Yan, P. Krantz, Y. Sung, M. Kjaergaard, D. L. Campbell, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "Tunable coupling scheme for implementing high-fidelity two-qubit gates," *Physical Review Applied*, vol. 10, no. 5, p. 054062, 2018.

[50] C. Zhou, P. Lu, M. Praquin, T.-C. Chien, R. Kaufman, X. Cao, M. Xia, R. Mong, W. Pfaff, D. Pekker, and M. Hatridge, "A modular quantum computer based on a quantum state router," 2021.

VIII. Artifact Appendix

A. Abstract

The artifact contains the source code used to generate, and evaluate the benchmarks presented in this paper. Since our benchmarks concern topology and gate based hardware designs, we use existing quantum circuit software to build circuits and model our transpilation passes after. The artifact provides a Jupyter notebook, python files, and data sets to recreate the plots shown in Figures 4, 11-15.

B. Artifact check-list (meta-information)

- **Program: Qiskit**
- **Run-time environment: Jupyter kernel**
- **Hardware: Quad-Core Intel Core i7**
- **Execution: Quantum circuit transpilation**
- **Output: Replication of paper figures**
- **Experiments: Transpilation gate count and best approximate compilation fidelity calculation**
- **How much disk space required (approximately)?: 1 GB to store the artifact directory and python virtual environment**
- **How much time is needed to prepare workflow (approximately)?: 10 minutes**
- **How much time is needed to complete experiments (approximately)?: 5 hours**
- **Publicly available?: Yes**
- **Workflow framework used?: Jupyter notebook**
- **Archived?: https://doi.org/10.5281/zenodo.7269557**

C. Description

1) How to access: The artifact is available on Zenodo (https://doi.org/10.5281/zenodo.7269557). The source code and artifact notebook are zipped within transpilation_EM-main.zip. Moreover, a maintained repository is availabe here: https://github.com/Pitt-JonesLab/clonk_transpilation.

2) Hardware dependencies: Any system which can run python programs should be able to evaluate the artifact, but has only been tested on Ubuntu 20.04.

3) Software dependencies: The artifact requires the installation of the source code as a python package. The dependencies are listed within requirements.txt. Additionally, there is a dependency on a public github repository (https://github.com/evmckinney9/NuOp) with more instructions provided in the main Jupyter notebook.

4) Data sets: We use external packages with quantum circuit constructors to build the circuits which we transpile and benchmark. QuantumVolume, QFT, and CDKMRippleCarryAdder are from Qiskit while QAOAVanillaProxy, HamiltonianSimulation, and GHZ are from Supermarq. The creation of QAOA circuits requires a modification, detailed in the Jupyter notebook, which prevents lengthy optimization over 1Q gate parameters, but will not effect final transpiled gate counts reported.

D. Installation

After downloading the artifact zipfile, and extracting the contents, the source code package can be installed via:

```
# pip install -r requirements.txt
# pip install -e .
```

The user can then open the jupyter lab with the command:

```
# jupyter lab
```

The file HPCA_artifact.ipynb contains an overview of the benchmarks and figures used in this paper.

E. Evaluation and expected results

The notebook HPCA_artifact.ipynb contains examples showing how our benchmarks are generated. The notebook is divided into two parts. The first section, describes how the target topologies are created, with a visualization tool to print the qubit interconnections. Then, we show the validity of our custom transpilation pass by showing the expected Haar score matches the correct value. Next, we demonstrate how to assemble benchmark objects consisting of circuits and target backends including the set of benchmarks which created each figure in the paper. The second section is the collection of data for approximate compiling and creation of the figure. We use a fork of 'NuOp' which iterates through basis gates and number of repeated applications while tracking decomposition fidelity. Finally, we also point out which parameters to modify which will either regenerate the figures from the original data or run from scratch to reproduce the results.

F. Methodology

Submission, reviewing and badging methodology:

- https://www.acm.org/publications/policies/artifact-review-badging
- http://cTuning.org/ae/submission-20201122.html
- http://cTuning.org/ae/reviewing-20201122.html