

# A manufacturable surface code architecture for spin qubits with fast transversal logic

Jason D. Chadwick, Willers Yang, Joshua Vízslai, and Frederic T. Chong  
 University of Chicago, Chicago, IL, USA  
 {jchadwick, willers, jvizslai, ftchong}@uchicago.edu

**Abstract**—Spin qubits in silicon quantum dot arrays are a promising quantum computation platform for long-term scalability due to their small qubit footprint and compatibility with advanced semiconductor manufacturing. However, spin qubit devices face a key architectural bottleneck: the large physical footprint of readout components relative to qubits prevents a dense layout where all qubits can be measured simultaneously, complicating the implementation of quantum error correction. This challenge is offset by the platform’s unique rapid shuttling capability, which can be used to transport qubits to distant readout ports. In this work, we explore the design constraints and capabilities of spin qubits in silicon and propose the SNAQ (Shuttling-capable Narrow Array of spin Qubits) surface code architecture, which relaxes the 1:1 readout-to-qubit assumption by leveraging spin shuttling to time-multiplex ancilla qubit initialization and readout. Our analysis shows that, given sufficiently high (experimentally demonstrated) qubit coherence times, SNAQ delivers an orders-of-magnitude reduction in chip area per logical qubit. Additionally, by using a denser grid of physical qubits, SNAQ enables fast transversal logic for short-distance logical operations, achieving over  $10\times$  improvement in local logical clock speed while still supporting global operations via lattice surgery. This translates to a  $3.2\text{--}5.7\times$  improvement in key Adder and Lookup fault-tolerant subroutines. Our work pinpoints critical hardware metrics and provides a compelling path toward high-performance fault-tolerant computation on near-term-manufacturable spin qubit arrays.

## I. INTRODUCTION

Scaling quantum computers up to the million-qubit processors needed to enable powerful applications [1]–[5] is an immense architectural challenge. Spin qubits in silicon quantum dot arrays are a promising candidate for large-scale quantum computing, primarily due to their small footprint (on the order of  $100 \times 100$  nm per qubit) and their compatibility with existing advanced semiconductor fabrication techniques [6]–[11]. However, the small footprint that makes spin qubits attractive creates a fundamental architectural challenge: physical readout components, such as a single-electron transistor (SET) or single-electron box (SEB), require large reservoirs and ohmic contacts that are many times the size of a quantum dot [12]–[17]. This makes it difficult to build a dense array of dots while dedicating a unique readout component to each qubit, instead incentivizing asymmetric fixed-width arrays as shown in Figure 1(a). On the other hand, spin qubits also offer the unique and compelling capability of extremely fast spin shuttling between quantum dots [18], [19], which can enable novel shuttling-based hardware layouts that can overcome the readout bottleneck [20]–[25]. Shuttling comes with a cost,

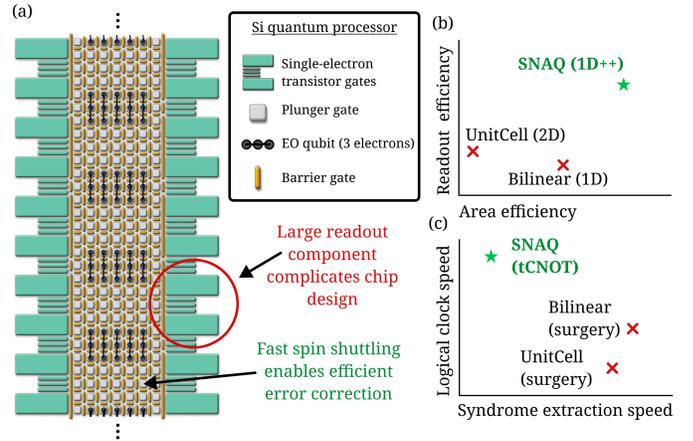


Fig. 1. (a) Depiction of a possible implementation of a 7-dot-wide array of quantum dots, similar to devices fabricated by industrial and academic groups [26]–[30]. Electrons (black) are held in place at the plunger gates (light gray) and can interact with their neighbors or shuttle to adjacent plungers by manipulating the voltages on the barrier gates (orange). Single-electron transistors (green) allow for qubit readout on the two sides of the array. (b) SNAQ achieves similar performance to baselines while improving chip area efficiency and readout component efficiency per logical qubit. (c) SNAQ avoids the downside of a longer syndrome extraction cycle by enabling transversal CNOTs (tCNOTs), achieving significantly faster logical clock speeds compared to lattice-surgery-based approaches.

though. Qubits accumulate errors proportional to the distance shuttled, making it crucial to minimize shuttling distance when possible.

Previous spin qubit architecture proposals have addressed these challenges by introducing sparser arrays, creating large interior spaces to fit the needed components, but at the cost of a large shuttling overhead required to enable the dense 2D qubit connectivity needed for error correction. In this work, instead of designing an entirely new hardware layout, we consider the proven design of a fixed-width, limited-readout array [26]–[29], posing the central question: can we efficiently run the surface code in this constrained geometry by sacrificing measurement parallelism in exchange for a denser qubit array?

We propose a hardware-aware surface code architecture named SNAQ (Shuttling-capable Narrow Array of spin Qubits) that leverages fast spin shuttling to overcome the limited readout capabilities in silicon quantum dot hardware. We show that for sufficiently low idle error rates demonstrated by existing spin qubit hardware, the initialization and mea-

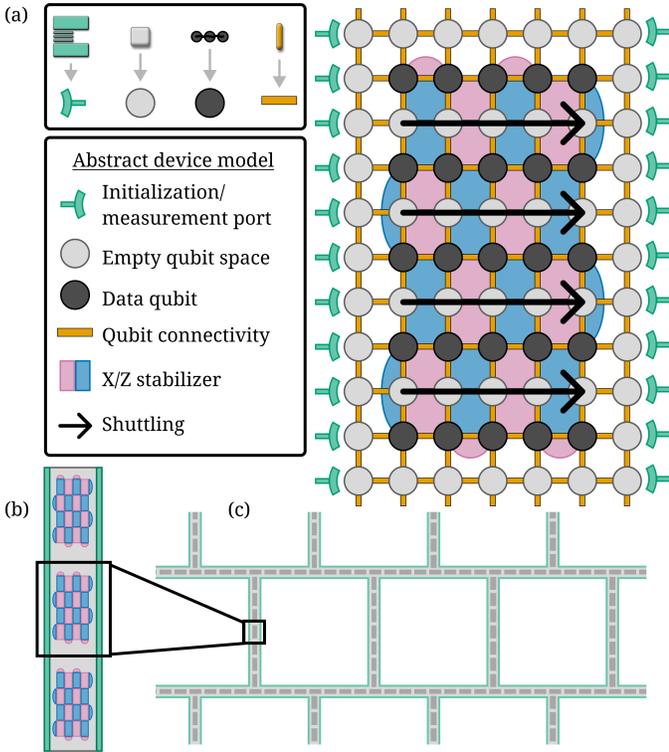


Fig. 2. Proposed SNAQ (Shuttling-capable Narrow Array of spin Qubits) architecture. (a) For ease of visualization, we translate from the specific chip components in Figure 1 to the abstract components we will use in the rest of the paper. Depicted is a layout of a distance-5 surface code patch on a 7-dot-wide array. Initialization and readout components (green) are only available on the edges of the array. Data qubits (dark gray) are interleaved with channels of empty dots (light gray) used to shuttle surface code ancilla qubits from edge to edge in the direction of the black arrows. Surface code X and Z stabilizers (blue and pink) are shown behind the array, supported on the data qubits. (b) A logical  $1 \times N$  layout of surface codes in a narrow array. (c) Possible design of a fully scalable architecture consisting of loops of SNAQ channels, with large spaces in between to allow for integration of control wires and readout.

surement of physical qubits can be serialized with manageable impact on the logical performance of the surface code. This creates a significantly denser qubit array, which reduces the required amount of spin shuttling for error correction and enables the transversal CNOT for short-range logical multi-qubit gates, delivering logical clock speed improvements and greater potential parallelism. This work makes the following main contributions:

- We identify and characterize the qubit-to-readout area mismatch as a key design challenge for scalable spin qubit architectures.
- We propose SNAQ, a novel surface code architecture that leverages fast spin shuttling to time-multiplex syndrome extraction and overcome this bottleneck in near-term-manufacturable arrays.
- We analyze the physical parameters for this new design with detailed circuit-level simulations, pinpointing the readout density  $\rho$  and the qubit idle error rate  $p_{id}$  as the most critical hardware metrics for its viability. Although

the idle error restricts the lowest achievable logical error rates, our method shows competitive performance against baselines for algorithmically relevant error rates.

- We demonstrate through simulation that SNAQ enables low-latency transversal CNOTs (tCNOTs) for local multi-qubit gates, creating a two-tiered latency hierarchy between fast-local tCNOTs and slow-global lattice surgery.
- We perform a thorough performance evaluation, showing that SNAQ can achieve over  $10\times$  faster logical clock speed. This translates to 57-84% reduced cost for key fault-tolerant primitives, enabling an estimated  $2.4\text{-}4.2\times$  improvement in resource estimates for large-scale factoring algorithms.

## II. BACKGROUND

In this section, we explain the components and terminology of silicon quantum dot devices and provide some high-level background on quantum error correction and the surface code. We encourage the reader to refer to Ref. [31] for a more thorough background on spin qubits and to Refs. [32], [33] for more information on the surface code.

### A. Spin qubits in silicon

Semiconductor spin qubits generally refer to nanoscale qubits encoded in the spin states of electrons or holes in a semiconductor substrate. In this work, we will specifically consider electrons held in quantum dots in silicon, which is a platform that has seen considerable recent progress [30], [31], [34]–[37]. Here we will describe the general terminology and architectural implications of the silicon spin qubit platform, leaving a specific discussion of state-of-the-art performance metrics to Section III. From an architectural perspective, these devices present a compelling substrate for large-scale fault-tolerant quantum computing: the qubit footprint is comparably small (dot pitches on the order of tens of nanometers), fabrication is compatible with existing CMOS processes, and spin shuttling is a novel and powerful tool. We refer the interested reader to Ref. [31] for a more thorough review of semiconductor spin qubits.

A silicon quantum dot array is defined by electrostatic gates as depicted in Figure 1, in which we identify three important categories of these gates: “plunger” gates, each of which is tuned to create a quantum dot that can hold one electron; “barrier” gates, which control inter-dot tunnel coupling, used for shuttling and for electron-electron entanglement via the exchange interaction; and the gates that define the large electron reservoirs and quantum dot that form a single-electron transistor, which is used to measure the spin states of electrons. The simplest way to compute with electrons in quantum dots is to treat individual electrons as qubits, where the two logical states are the spin states  $|\uparrow\rangle$  and  $|\downarrow\rangle$ ; alternative encodings can exploit up to four electrons per qubit to trade control complexity, magnetic gradient requirements, and coupling speed [31]. Generally, increasing the number of electrons that encode each qubit simplifies the control requirements, but may reduce best achievable qubit coherence.

The exchange-only (EO) qubit encoding uses three electrons to define each qubit, and is a promising choice for scalability because it relaxes the requirement for localized magnetic field control and instead can be operated by only tuning DC voltages on barriers and plungers [38], [39]. In all qubit encodings, logical operations between qubits are mediated by the exchange interaction: by tuning the inter-dot tunnel barrier and dot detuning, one can dynamically adjust the exchange coupling between two adjacent electrons and thereby implement controlled-phase, CNOT-equivalent, or SWAP-like operations. It may even be beneficial to use multiple different encodings in one architecture [40].

Silicon quantum processors introduce several unique architectural considerations that we account for in this work. First, readout of spin qubits typically uses spin-to-charge conversion, which is implemented using components that are significantly larger than those that hold the qubits [12]–[17]. Designing the layout of a silicon spin processor is thus similar to the “pitch matching” exercise studied extensively in classical architecture, where physical components of different sizes must be integrated into a cohesive layout [41]–[45]. Because the qubit-qubit interactions are fundamentally constrained to be short-distance, the large readout size has motivated the current approach of placing readout sensors on the edges of a quantum dot array to allow for a dense qubit grid, which has enabled impressive experimental demonstrations [26]–[29] but has not previously been expected to scale to the fault-tolerant era. Second, the capability of “shuttling” (moving an electron spin coherently between quantum dots) has a different cost model than qubit movement in neutral atoms or trapped ions. In spin qubits, the shuttling is very fast (a few nanoseconds per dot-to-dot transfer), but shuttling errors accumulate more rapidly with distance than in atomic systems.

Silicon spin qubit technology is a highly promising candidate for truly scalable quantum processors. At the same time, its distinct challenges and opportunities call for novel architectural ideas to make the best use of this hardware and enable efficient fault-tolerant computation.

### B. Quantum error correction and the surface code

Quantum error correction (QEC) is the process by which many physical qubits are used to encode a smaller number of logical qubits with much higher lifetimes and operating fidelities. A typical QEC code is encoded into a large number of *data qubits* and operates by repeatedly measuring a set of stabilizers of the code, which are joint observables on multiple data qubits. Typically, each stabilizer is measured using an *ancilla qubit* that interacts with each of the relevant data qubits before being measured. Measurement of all stabilizers of the code produces an error syndrome that indicates the difference between the expected stabilizer parities and the observed values. This syndrome must then be decoded by a classical algorithm to determine which physical qubits experienced errors. We will refer to this stabilizer measurement process as syndrome extraction (SE). Repeatedly performing SE rounds and decoding allows individual physical qubit errors

to be corrected, stabilizing the logical qubit and extending its lifetime.

Important metrics of a QEC code in the context of this work are the code distance  $d$ , which determines the error robustness and size of a code, and the threshold  $p^*$ , which is the physical error rate below which the code can be scaled up to increase logical lifetimes. When the physical error rate  $p$  is below  $p^*$ , the logical error rate  $p_L$  scales approximately as

$$p_L \approx A \left( \frac{p}{p^*} \right)^{(d+1)/2}. \quad (1)$$

Among the many QEC codes proposed, the rotated surface code stands out for its easy-to-build planar connectivity requirements, relatively high threshold, ease of decoding [32], [33], and well-understood logical operations [46]–[48]. One logical qubit in a distance- $d$  surface code is encoded on a square  $d \times d$  grid of physical qubits, which we will refer to as a “surface code patch”. Each stabilizer within this grid is either a joint X or Z operator supported on four neighboring qubits (or two on a boundary). Including edge stabilizers, there are  $d^2 - 1$  stabilizers that must be measured in each SE round. Conventionally, each of these stabilizers is measured using a unique physical ancilla qubit which is initialized, entangled with the involved data qubits, and measured. Serialization of syndrome extraction has been studied before to reduce physical qubit overheads [49], [50] or to adapt to control constraints [51]; in this work, we study the forced serialization of the measurement of ancilla qubits due to the limited readout capacity of the chip.

Two main methods have been proposed to entangle logical surface code qubits, both of which are supported in SNAQ. Hardware that is restricted to nearest-neighbor connectivity can perform lattice surgery [46]–[48], which involves the “merging” and “splitting” of surface code patches to perform multiqubit Pauli measurements, each of which takes  $d$  SE rounds. Alternatively, hardware with long-range connections can perform a transversal CNOT (tCNOT), where each data qubit of one patch interacts with its corresponding data qubit in the second patch. When combined with specialized decoding techniques [52], [53], tCNOTs only require  $\Theta(1)$  SE rounds after each operation. Prior work has found that less than one SE round is needed per tCNOT [52], saving significant temporal costs compared to lattice surgery. Additionally, while parallel shuttling of qubits is possible in a transversal computation, parallel lattice surgery through the same shared routing space is not, so tCNOTs may have compilation benefits for certain workloads in restricted logical layouts. Decoding a tCNOT-based logical circuit ostensibly requires use of a more complex decoder than in a lattice-surgery-based system, but methods have recently been developed to reduce the transversal decoding problem to one closely resembling an idling logical qubit, allowing for fast decoding [54].

### III. HARDWARE REQUIREMENTS

SNAQ is designed to be compatible with current or near-future silicon spin qubit technology. This section reviews state-

of-the-art spin qubit capabilities and lists the concrete device-level capabilities we assume for SNAQ and the experimental evidence that these capabilities are achievable in the near term.

#### A. Array geometry and readout placement

A SNAQ device consists of a narrow and dense two-dimensional grid of quantum dots with readout components on two edges of the array, such as the 7-dot-wide example shown in Figure 1(a). This design maximizes compatibility with existing semiconductor fabrication techniques [30], [55], and is a directly scaled-up version of existing devices such as Intel’s 4x27-dot array [28] or HRL’s 3x3-dot array [29]. The choice of a fixed width means that the array can be constructed using a fixed number of interconnect layers, which are required to control the interior dots in the array, regardless of the size along the second axis of the array. A key parameter that affects the performance of SNAQ is the *readout density*,  $\rho$ , which is the density of readout devices along the sides of the array relative to the rows of qubits. Readout density can be increased by building more complex routing on the array edges or by using a qubit encoding that occupies more dots, such as the three-electron exchange-only qubits depicted in Figure 1.

#### B. Spin shuttling

SNAQ relies on coherent electron transport to move ancilla qubits between readout/initialization regions and interior interaction sites. There are two distinct methods for shuttling an electron between dots. “Bucket-brigade” shuttling is performed by a series of discrete single-dot jumps, which has the advantage of only requiring discrete control pulses on plungers and barriers, and has achieved error rates of less than 0.3% per hop [56], [57]. “Conveyor-mode” shuttling involves operating the metal gates along the path with smoothly oscillating voltages to engineer a continuously moving potential wave that carries the electron along. This second approach is more complex to engineer, but can reach faster shuttling rates and has been achieved with per-hop fidelities of 99.99% [19]. At a nominal dot pitch of 100 nm, these results correspond to per-dot shuttling latencies around 2 ns. We find that these experimentally achieved speeds and fidelities are already sufficient to enable effective quantum error correction, and we expect further improvement in the future.

#### C. Coherence

Data qubits in SNAQ are idle for longer intervals than a standard surface code due to serialized ancilla initialization and measurement. The idling coherence time of the data qubits is therefore a crucial parameter that will significantly affect the performance of a SNAQ device. Achieved spin qubit coherence times vary significantly between device types and choice of qubit encoding. State-of-the-art coherence times range from around 1 ms to 0.56 s for one and two-electron qubits [58]–[61]. Three-electron EO qubits, whose appeal has grown recently, have been stabilized along one axis for up to 720  $\mu$ s [62]. Coherence is expected to improve further as device uniformity and material purity improve [31]. We

find that a coherence time of at least 200  $\mu$ s is required to enable competitive error correction on SNAQ, with the specific number depending on the code distance, readout element density, and the strength of other error sources.

#### D. Physical qubit gate fidelities

SNAQ does not impose any unique constraints on the fidelity of single- and two-qubit gates compared to surface code proposals on other hardware modalities, simply requiring that the gate fidelity is below the threshold of the surface code (typically around  $10^{-2}$ ). We set a realistic gate error target of  $10^{-3}$ , which has been nearly reached or surpassed in many experimental spin qubit demonstrations [34]–[37], [63], [64].

#### E. Initialization and readout

Spin qubit readout via spin-dependent tunneling [12], [65] offers a promising path towards fast high-fidelity readout. Early experiments have demonstrated readout fidelity above 99% in 6  $\mu$ s with a single-electron box (SEB) [16], 1.6  $\mu$ s using a single-electron transistor (SET) [14], and 990 ns using a dot-charge sensor [66]. Theoretical analyses predict that a readout fidelity of 99.97% and duration as short as 100 ns could be achieved with the SEB [16], and readout above 99% fidelity could be performed in only 36 ns with fully optimized SET device parameters [67]. In this work, we assume that initialization and readout each take 500 ns and have fidelities comparable to physical quantum gates.

#### F. Practical summarized requirements for near-term SNAQ

The following target parameters capture the regime in which serialized readout with shuttling yields competitive logical performance, as we will show in simulation in the following sections:

- **Array geometry:** Dense fixed-width rectangular array of nearest-neighbor-connected dots with readout elements confined to edges [28]–[30]. Readout density  $\rho$  at least 1, ideally closer to 2.
- **Shuttling:** Per-hop error  $p_{\text{sh}} \leq 10^{-4}$  at 2 ns latency [19].
- **Idling:** Coherence time exceeding 200  $\mu$ s (effective idle-error-per- $\mu$ s  $p_{\text{id}} \lesssim 5 \times 10^{-3}$ ) under active stabilization such as dynamical decoupling [58]–[62].
- **Physical logic gates:** CNOT error  $p_{\text{g}} \gtrsim 99.9\%$  [34]–[37], [63], [64].
- **Initialization and measurement:** Initialization and readout fidelity equal to  $p_{\text{g}}$  with 500 ns latency each [67].

While the state-of-the-art metrics cited above have been demonstrated across different devices, they rely on compatible semiconductor fabrication techniques. We perform sensitivity studies to variations in each of these variables, finding SNAQ to be competitive across wide parameter ranges.

## IV. THE SNAQ ARCHITECTURE

The layout of our proposed architecture is shown in Figure 2(a), where a 7-qubit wide array is used to support a  $d = 5$  surface code.

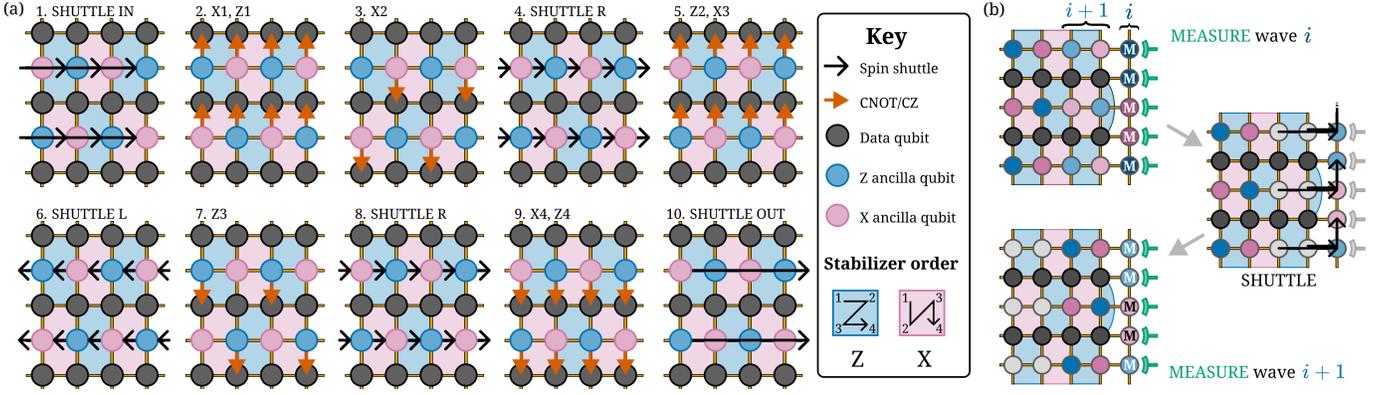


Fig. 3. (a) Schedule of shuttle and CNOT/CZ operations that implement the surface code X and Z checks in a distance-preserving order. Each ancilla qubit is responsible for interacting with four nearby data qubits, and it must interact in a specific order to avoid hook errors [32], as shown in the lower right. (b) Serialized measurement of a group of ancilla qubits for readout density  $\rho = 1$ . Ancilla qubits in group  $i$  are measured first, followed by qubits in group  $i + 1$ .

### A. Hardware layout

SNAQ consists of a dense fixed-width array of quantum dots with readout components on both sides, as shown in Figure 1(a). Although previous work has proposed building long shuttling channels to route around these large readout zones [23], [68], in this work we explore the more near-term friendly approach of placing readout elements along two outer edges of a fixed-width array [28], keeping the interior free for uniform dot placement and nearest-neighbor exchange coupling. A fixed array width  $w$  limits the largest surface code that we can embed in the array using the mapping of Figure 2(a) to  $d_{\max} = w - 2$ . A two-column layout to enable multiqubit interactions via lattice surgery would support a maximum distance of  $d_{\max} = \lceil \frac{w-3}{2} \rceil$ .

The denser qubit grid of SNAQ compared to alternative architectures introduces additional control concerns, most importantly crosstalk between nearby control lines when addressing qubits in parallel. However, this is not a fundamental obstacle, as recent work has shown that, with proper calibration of crosstalk compensation terms, voltage pulses can be applied in parallel in very close range [69].

The readout density  $\rho$ , the ratio of readout sensors to array rows, is a key hardware parameter. A density of 1 means that each row has a readout sensor, as shown in Figure 2(a). The readout density directly determines the amount of serialization required to initialize or measure some fixed number of spin qubits, so it has a significant effect on the speed and error rate of the surface code. Achieving  $\rho \gg 1$  in a manufacturable device is difficult, but a small constant-factor increase (such as  $\rho = 2$ ) may be attainable with careful routing.

### B. Scheduling a surface code on SNAQ

Figure 3(a) shows the specific schedule of shuttles and CNOTs that perform a stabilizer measurement cycle of the surface code. The ancilla qubits are first initialized on the left side of the array and are shuttled into place as they become ready. The ancilla qubits and data qubits then perform several

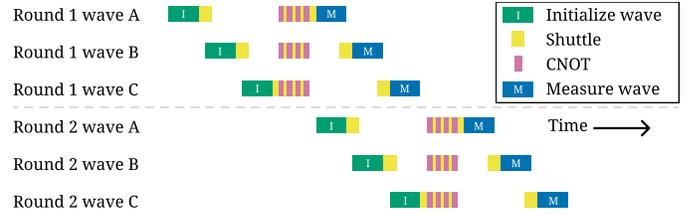


Fig. 4. Example pipeline schedule showing initialization, data entanglement, and measurement for two successive rounds of ancilla qubits, split into three waves each. The measurement of the first round of ancillae can be done simultaneously with the initialization of the second round. Any blank space in the schedule indicates idling, during which dynamical decoupling techniques could be used to improve coherence.

layers of CNOTs. Finally, the ancilla qubits are shuttled to the right side of the array, where they can be measured. Note that the depiction of shuttle operations here is slightly simplified; in reality, buffer space would be needed in between adjacent ancilla qubits for them to both be shuttled at the same time. This means that the group shuttle operation cannot be fully parallelized.

To avoid harmful “hook errors”, where errors propagate in detrimental patterns [32], we implement the commonly chosen CNOT ordering shown in the lower right of Figure 3(a), where the number in each corner of a stabilizer indicates the order of the gates from the perspective of the ancilla qubit. Ancilla qubits on the edges of the patch perform the same shuttling pattern, simply skipping some CNOT gates.

The primary bottleneck of the SNAQ surface code is the initialization and measurement of  $d^2 - 1$  ancilla qubits in each round of syndrome extraction. In the proposed syndrome extraction schedule, these ancilla qubits are all initialized on the left side of the array, shuttled into place in the interior, and then eventually measured and removed on the right side of the array. Because a given surface code patch will only have  $O(d)$  available initialization/measurement devices on one of its edges for any constant density  $\rho$ , a sufficiently large

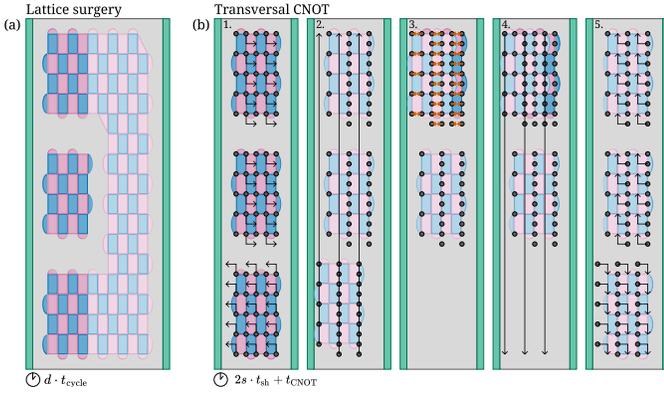


Fig. 5. Two possible implementations of multiqubit operations in the SNAQ architecture. (a) Lattice surgery can connect distant logical qubits via a channel of “routing patches” (lighter shaded stabilizers), which requires a wider dot array to support a second column of surface codes. (b) Spin shuttling enables transversal CNOT operations for sufficiently close tiles. Between stabilizer measurement rounds, physical qubits can shuttle past intermediate tiles to reach the target tile, then perform the transversal operation and shuttle back.

code distance will require multiple serial waves of ancilla qubit initialization/measurement, as depicted in Figure 3(b). The number of ancilla waves is described by

$$n_w = \left\lceil \frac{d^2 - 1}{2\rho(d + 1)} \right\rceil, \quad (2)$$

where  $d^2 - 1$  is the total number of ancilla qubits in a distance- $d$  surface code and  $2\rho(d + 1)$  is the number of readout ports on one side of a SNAQ surface code. This serialization is the primary novel source of error that the SNAQ surface code must mitigate. Logical initialization and measurement of a surface code require serialized readout and measurement of the  $d^2$  data qubits as well, which can be done in a similar method to the ancilla qubits (and can be done in roughly half the number of waves by using both edges of the array at the same time).

Importantly, we note that steps 1 and 10 of Figure 3(a), which are by far the longest due to serialization, can be pipelined: while layers of ancillae are being moved to the readout locations on the right side of the array, fresh waves of ancillae can be initialized at the same time to fill in the empty spots in the array, as shown in Figure 4. Our circuit-level simulations account for this pipelining.

### C. Multiqubit logical gates

Both lattice surgery and transversal operations are supported in the SNAQ architecture, and their implementations are depicted in Figure 5. Importantly, a two-column logical layout is required to support lattice surgery, while a processor that only uses tCNOTs can be operated in a single-column layout, which allows twice as large a code distance on a chip of the same width. The tradeoff to using tCNOTs is that they natively only work well for relatively small separation distances before shuttling errors accumulate.

	UnitCell	Bilinear	SNAQ (this work)
Near-term	×	✓	✓
Low- $d$ perf.	×	✓	✓✓
Ultrahigh- $d$ perf.	✓	×	×
Logical operations	LS	LS	LS + tCNOT
Logical clock cycle time ( $d = 11$ )	28.6 $\mu$ s	20.8 $\mu$ s	2.5 $\mu$ s (local) - 55.6 $\mu$ s (global)
Logical parallelism	2D	1D LS	1D transversal

TABLE I  
COMPARISON TO PRIOR SILICON ARCHITECTURE PROPOSALS

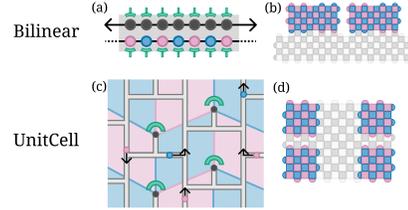


Fig. 6. Two prior spin qubit architecture proposals. (a) The Bilinear architecture, based on Refs. [24], [25], consists of two rails of qubits, one of which can shuttle back and forth. Data and ancilla qubits of a surface code are placed column-by-column into the linear array. (b) The Bilinear architecture supports a linear logical-level arrangement, where each (wide) surface code patch is connected to others by a shared lattice surgery routing bus (gray). (c) The UnitCell architecture, similar to Refs. [20], [23], [68], consists of large unit cells, each of which contains a readout port. A data qubit can be placed in each unit cell and the ancilla qubits must shuttle long distances between unit cells. (d) The two-dimensional physical qubit array of the UnitCell architecture naturally produces a two-dimensional logical array, with lattice surgery routing space between surface code patches.

### D. Prior proposals

We compare SNAQ to two baseline spin qubit architectures based on prior proposals, which we call UnitCell and Bilinear, depicted in Figure 6. These architectures both support a 1:1 ratio of readout ports to qubits, but achieve this by different means. The Bilinear architecture restricts the physical qubits to a  $2 \times N$  physical layout, as proposed in Ref. [24]. This design makes fabrication relatively straightforward and cost-effective, but performance depends more heavily on shuttling (as not all data-ancilla pairs can be placed near each other) and yields a logical layout with only one shared lattice surgery routing bus, limiting achievable logical parallelism.

The approach of the UnitCell architecture is to create large unit cells that each contain a readout component and have enough space for all readout and qubit wiring, as proposed in Refs. [20], [23], [68]. Each unit cell is surrounded by shuttling channels, allowing qubits to be moved between nearby unit cells. We assume a relatively optimistic unit cell size of  $5 \times 5 \mu\text{m}$ , corresponding to 50 dot-to-dot shuttling distances per side. The surface code can be implemented by assigning each data qubit to its own unit cell and shuttling ancilla qubits between data cells. The benefit of this approach is that the amount of required shuttling does not change with the code distance, so this architecture can in principle support arbitrarily high code distance. The downside is a very large constant amount of shuttling, which can be thought of as effectively adding to the baseline physical error rate, thereby shifting the code threshold. This reduces UnitCell’s relative performance

for smaller code distances. The resulting logical layout is two-dimensional, as shown in Figure 6d, which allows for improved logical parallelism compared to Bilinear.

SNAQ’s layout lies somewhat in between these two baselines with its fixed-width design, though fundamentally differs from both in purposefully deviating from the 1:1 readout-to-qubit ratio of the baselines. This allows SNAQ to have a much higher qubit density than either baseline, which is critical to enable its transversal logic capabilities.

We provide a high-level, qualitative comparison of these architectural tradeoffs in Table I. This summary highlights our focus on near-term manufacturability, strong low-distance performance, and efficient logical operations. Unlike the baselines, which are limited to lattice surgery (LS), SNAQ’s dense design supports fast transversal CNOTs (tCNOTs), enabling a more flexible, high-parallelism logical computation.

We note that the Bilinear architecture proposal has been extended to a two-dimensional architecture supporting transversal operations [25], but this comes at the cost of increased shuttling, significant added scheduling complexity, and extra qubit costs required to avoid potentially defective shuttling paths. The notion of “separation distance” between two logical qubits, which we use to evaluate SNAQ’s tCNOT, is less well-defined for this architecture, as the logical qubits do not live in a sea of qubits but rather continually move on sparse shuttling tracks. We therefore cannot perform direct comparisons to SNAQ like we did for Bilinear and UnitCell. The logical memory performance will be similar to Bilinear, and we expect the tCNOT performance to vary significantly with specific hardware parameters like the sparsity of the layout and the overlap between adjacent loops.

Several proposals have been made for spin qubit architectures based on “crossbar” wiring layouts, which offer compelling reductions in wiring complexity [7], [8]. However, in the absence of near-perfect fabrication uniformity across the chip, parallel operation of the physical qubits becomes impossible, leading to severe serialization overhead [8], [51] that Ref. [70] estimated would limit the best achievable logical error rates to  $2 \times 10^{-5}$  (at code distance  $d = 63$ ) even with a relatively high coherence time of  $10 \mu\text{s}$ . Integrating readout components also remains challenging, with current physical crossbar implementations relegating readout to the edges of the array similar to the SNAQ arrays considered here [27].

## V. PERFORMANCE EVALUATIONS

We develop a custom simulation framework that can accurately model the pipelined schedule of shuttled waves of ancilla qubits, allowing us to precisely quantify both the clock speed of the code and the logical performance. We simulate the performance of our proposed architecture using Stim [71] and decode with PyMatching [72]. For a given code distance  $d$ , we assume a fixed array width of  $d+2$ . For all experiments, we perform both X and Z-basis experiments and combine the two error rates to obtain an overall error rate.

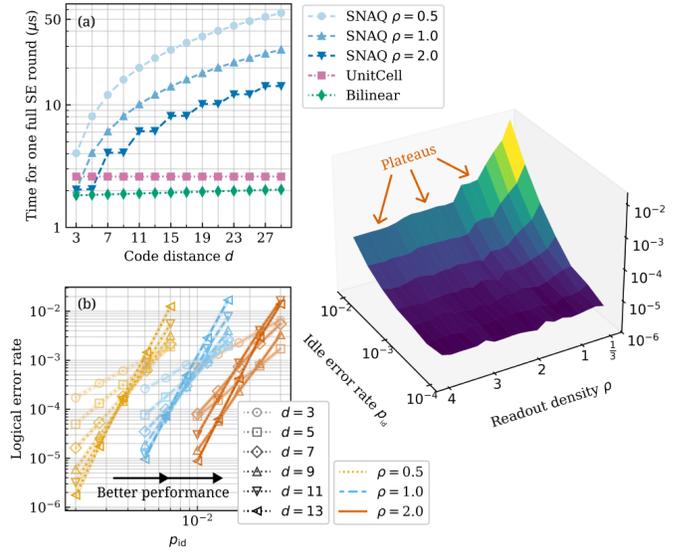


Fig. 7. Exploring the effect of readout serialization in SNAQ. (a) Time to complete one (non-pipelined) SE round in SNAQ compared to the baselines, for various values of  $\rho$ . (b) Logical error rate as a function of  $p_{id}$  for various code distances and densities. Doubling  $\rho$  can give an order of magnitude improvement in logical error rate. (c) Logical error rate of a  $d = 7$  surface code on SNAQ, showing “plateaus” where ranges of density settings lead to the same number of ancilla waves, yielding the same performance.

### A. Noise model

Our noise model is inspired by exchange-only spin qubits [34]; while this limits the direct application of our results for other spin encodings, our general takeaways should apply for a wide range of possible implementations. We use a noise model with three error parameters  $p_g$ ,  $p_{sh}$ , and  $p_{idle}$ . The parameter  $p_g$  sets the strength of gate and readout errors. Each CNOT gate causes a two-qubit depolarizing error with probability  $p_g$ , each readout encounters a bit flip with probability  $p_g$ , and each Hadamard gate causes a one-qubit depolarizing error with probability  $p_g/10$ . The parameter  $p_{sh}$  is the error-per-shuttle such that shuttling a qubit over a distance of  $m$  dots incurs a depolarizing error with probability  $m \cdot p_{sh}$ . Finally, the idle error  $p_{idle}$  is defined as the chance that a qubit experiences a depolarizing idle error in a  $1 \mu\text{s}$  idle interval (in our noise model, this is equivalent to the time for the five layers of CNOTs in one SE round), so a coherence timescale of  $T$  would correspond to  $p_{idle} = e^{-(1 \mu\text{s})/T}$ .

We set the durations of physical operations assuming an exchange pulse duration of  $10 \text{ ns}$ , which means that a CNOT takes approximately  $200 \text{ ns}$  (the exact CNOT latency depends on dot-level connectivity [73], [74], but here we leave it fixed), a single-qubit Hadamard gate takes  $30 \text{ ns}$ , and the shuttling latency is  $2 \text{ ns}$  per dot. We assume that initialization and readout can each be done in  $500 \text{ ns}$ . Although our numerical simulation results depend on these specific choices, the important underlying ratio is the idle coherence timescale relative to the rate of syndrome extraction rounds. Our results can therefore be interpreted for different hardware latency

assumptions by rescaling  $p_{id}$  accordingly.

In some qubit encodings, the shuttling and idling errors can be strongly biased towards dephasing noise [19], which would incentivize asymmetric surface codes with  $d_z \neq d_x$ . A benefit of SNAQ is that the readout serialization would be determined by the smaller of  $d_x$  or  $d_z$ . Other options to consider include bias-tailored variants of the surface code [75]–[78].

### B. Impact of readout serialization

The readout density  $\rho$  is a critical architectural parameter that directly affects logical performance by changing the duration of each syndrome extraction (SE) round, as shown in Figure 7(a). For  $\rho = 2$ , we see that the SE duration remains below 10  $\mu\text{s}$  up to  $d = 21$ . Compared to the two baseline architectures, SNAQ’s SE duration at the same code distance is still significantly longer even with a relatively high  $\rho$ ; however, we will show later that SNAQ can still achieve a faster logical clock cycle due to its ability to execute transversal gates. Note that pipelining (Figure 4) will reduce SNAQ’s effective SE latency by half.

To quantify the impact of readout serialization on SNAQ’s logical performance, we simulate logical memory performance at different readout densities  $\rho$  and different idle error rates  $p_{id}$ , with  $p_g = p_{sh} = 0$ . Due to the increase in serialization for larger distances, we do not expect to observe a clear threshold (a point where the lines of the same group would all cross). The results are shown in Figure 7(b), where we see that doubling the density can improve the logical performance at the same distance by more than an order of magnitude, underscoring the importance of  $\rho$  for the SNAQ architecture. Figure 7(c) visualizes this relationship for fixed  $d = 7$ , showing performance “plateaus” where ranges of  $\rho$  yield the same number of ancilla waves.

### C. Comparison to other architectures

Although we cannot directly simulate code performance at larger distances with current Monte Carlo methods, as the number of required shots becomes impractical, we can instead fit simulated lower-distance logical error rates to the expected scaling behavior of SNAQ. With  $p_g$ ,  $p_{sh}$ , and  $p_{id}$  fixed, we can model the idling logical error rate of SNAQ as a modification of Eq. 1:

$$p_L(d) = A \left( \frac{p_g}{p_g^*} + \frac{p_{sh}^{\text{eff}}}{p_{sh}^*} + \frac{p_{id}^{\text{eff}}}{p_{id}^*} \right)^{(d+1)/2} \quad (3)$$

where  $p_{sh}^{\text{eff}}$  is the cumulative shuttling error experienced by the physical qubits and  $p_{id}^{\text{eff}}$  is the cumulative idling error experienced by the qubits. Here we have made the first-order assumption of a linear threshold surface [79] characterized by three thresholds  $p_g^*$ ,  $p_{sh}^*$ , and  $p_{id}^*$ . To fit circuit-level simulation data to this model, we use the form

$$p_L(d) = A(\alpha + \beta d + \gamma n_w(\rho, d))^{(d+1)/2}, \quad (4)$$

where  $A$ ,  $\alpha$ ,  $\beta$ , and  $\gamma$  are fitting parameters, and  $n_w(\rho, d)$  is given by Eq. 2. This model accounts for the three ways in which physical errors in SNAQ scale with increasing code

distance: gate errors remain constant, shuttling errors increase linearly in  $d$ , and idle errors increase with the amount of ancilla serialization  $n_w$ . We have derived this model from the structure of the SE circuit, and we find that it fits well to the available simulation data, but we caution that it is an approximation and circuit-level simulation still remains the most reliable performance predictor. To fit the Bilinear baseline, we fix  $\gamma = 0$ , and for the UnitCell baseline, we fix  $\beta = \gamma = 0$ .

In Figure 8(a), we simulate the performance of SNAQ and baselines for distances up to  $d = 11$  under  $(\rho, p_g, p_{sh}) = (2, 10^{-3}, 10^{-5})$  and various values of the critical parameter  $p_{id}$ , fitting each set of results to Eq. 4 and extrapolating to large distances. At a fixed code distance, SNAQ’s logical error rate is far more sensitive to  $p_{id}$  than the baselines. As expected, we can clearly see that  $p_{id}$  defines an error floor for SNAQ, limiting the achievable logical error rate. However, we see that it can still reach utility-scale error rates below  $10^{-6}$  with  $p_{id} = 5 \times 10^{-3}$ , roughly corresponding to a coherence time of 200  $\mu\text{s}$ , which is well within achievable ranges for spin qubits. To reach extremely low error rates below  $10^{-12}$  for long-term resource-intensive applications,  $p_{id} = 10^{-3}$  or  $10^{-4}$  will be required. This translates to a spin qubit coherence time near the single-ms range. Although this has already been demonstrated in small devices [58]–[61], it remains to be seen whether the same performance can be achieved at scale. Reductions in gate and measurement duration compared to the assumptions in this work would relax these coherence time targets.

However, comparing these architectures at a fixed code distance is misleading because it obscures the starkly different physical resource costs of the architectures. In SNAQ, where I/O costs and qubit count are decoupled, the number of qubits may no longer be the primary cost driver. We therefore turn to more relevant architectural metrics by studying two fabrication-limited resources: the total number of readout components, which is a direct driver of I/O complexity, and the chip area per logical qubit. Figure 8(b) shows the same simulation data but with code distance converted to the number of readout components required per logical qubit, revealing SNAQ’s much more efficient use of readout components. Finally, in Figure 8(c), we convert code distance to chip area by assuming that each physical qubit (plunger and adjacent barrier gates) takes up a  $100 \times 100 \text{ nm} = 0.01 \mu\text{m}^2$  space and using an order-of-magnitude estimate of the readout footprint as  $1 \mu\text{m}^2$  based on Ref. [17] and chip images from e.g. Refs. [12]–[16]. These estimates account for the footprint of the components themselves, but do not include interconnect routing, which will be implementation-specific. For UnitCell, we therefore only consider the area taken up by the readout component and shuttling channels (not the large interior spaces reserved for wiring). This analysis reveals that, for achievable  $p_{id}$ , SNAQ produces logical qubits that are orders of magnitude more area-efficient than the baselines, already providing a clear advantage at  $p_{id} = 5 \times 10^{-3}$  that becomes much stronger with lower  $p_{id}$ .

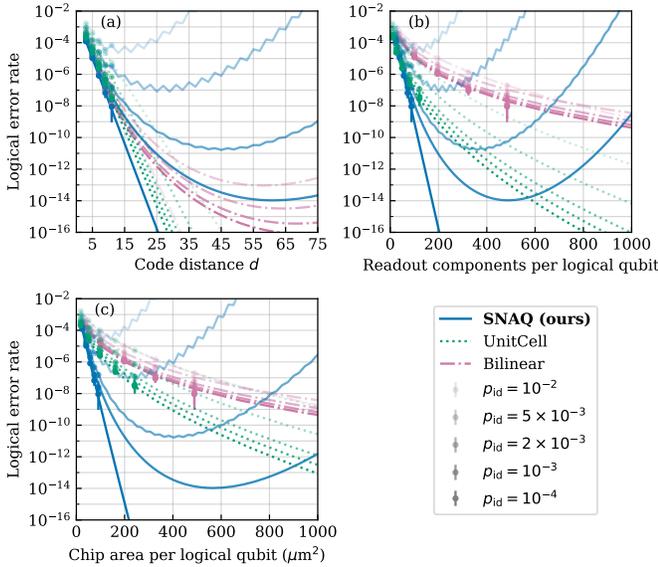


Fig. 8. Projecting logical performance to higher code distances under fixed  $(p_g, p_{sh}) = (10^{-3}, 10^{-5})$  for various  $p_{id}$  with SNAQ  $\rho = 2$ . (a) SNAQ and Bilinear give diminishing returns as code distance increases due to an increase in shuttling and idling errors with larger code block size, while UnitCell exhibits consistent error suppression as distance is increased. (b) Logical error rate as a function of readout count, which may be an important driver of packaging cost, showing that SNAQ outperforms Bilinear and is competitive with UnitCell. (c) Logical error rate as a function of total component area per logical qubit. For sufficiently low  $p_{id}$ , SNAQ achieves better logical performance at significantly reduced total chip area compared to both baselines.

#### D. Logical operations in SNAQ

An important difference between lattice surgery and transversal CNOTs in SNAQ is the scaling of their error rates as the separation between the two surface code patches increases: in SNAQ, the *physical* shuttling and idling errors that occur during a transversal CNOT will increase linearly with the distance between the surface code patches, meaning that the *logical* error rate will increase polynomially with degree  $(d+1)/2$  once the shuttling errors become dominant, which implies that code distance cannot be increased indefinitely to extend the tCNOT’s range. In contrast, lattice surgery errors increase linearly with the total merge volume due to a linearly increasing number of possible error chains, so it is always possible to increase the code distance to enable further-distance communication. We thus expect transversal operations to perform better for sufficiently close patches, while lattice surgery will be preferred for sufficiently long-distance communication.

To investigate this tradeoff, we perform circuit-level simulations of tCNOT and LS operations in the SNAQ architecture using parameters  $(p_g, p_{id}, p_{sh}, \rho) = (10^{-3}, 10^{-4}, 10^{-5}, 1.0)$ . We prepare two surface code patches in the logical  $|0\rangle$  state and either perform a tCNOT or an LS merge-split XX measurement between them, calculating the resulting error rate, which is the chance that either logical qubit experiences a bit

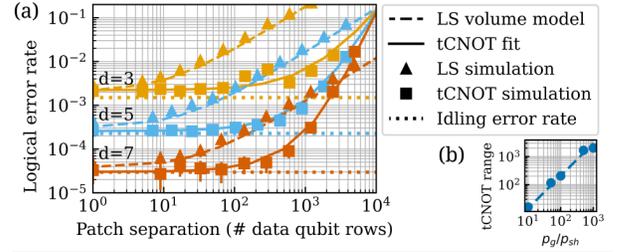


Fig. 9. (a) Error rates of transversal CNOT (tCNOT) and lattice surgery merge-split (LS) as a function of patch separation, which is defined as the number of data qubit rows between the two patches. (b) The range in which tCNOT incurs comparable error to idling is limited by the accumulation of shuttling errors, and therefore scales linearly with the relative strength of  $p_{sh}$ .

flip. We include  $d$  rounds of idling on each patch before and after the operation. We compare the error rate of this experiment to that of two patches simply idling for  $2d$  rounds. We decode the lattice surgery experiment with PyMatching and the transversal CNOT with BP+OSD [80], [81]. The results are shown in Figure 9(a): the error rate of distance- $d$  lattice surgery scales linearly with increasing separation distance, while the tCNOT’s error rate remains low (comparable to the idling surface code) until accumulated shuttling errors become stronger than the gate error  $p_g$ , at which point the error rate increases polynomially. The tCNOT’s limit is specified by the relative strength of  $p_{sh}$  to other error mechanisms, which we demonstrate in Figure 9(b) by calculating the  $d = 5$  tCNOT’s effective range for different ratios of  $p_g/p_{sh}$ . We define the range as the largest separation distance such that the tCNOT adds a negligible (within sampling uncertainty) additional error compared to the idling experiment.

We find that the LS error rate in Figure 9(a) is well-modeled by a linearly scaling increase in logical error rate  $p_L(s) = \frac{(s+d)d^2}{4d^3} \cdot p_L(1) + p_L^{\text{idle}}$  for a separation distance of  $s$  data qubit rows, where the prefactor gives the relative spacetime volume of the merge operation,  $p_L(1)$  is the lattice surgery error rate at minimal separation distance, and  $p_L^{\text{idle}}$  is the idling logical error rate. We find that the tCNOT’s error rate is well-modeled by  $p_L = A(B(s+d) + C)^{(d+1)/2}$  after fitting  $A$ ,  $B$ , and  $C$ , revealing the expected polynomial scaling behavior once  $s$  is sufficiently large.

We envision several potential ways to extend the tCNOT to longer effective ranges, all of which involve a space tradeoff in the SNAQ array. First, the shuttled surface code could stop along the way to perform multiple SE rounds during a long-distance tCNOT, which would avoid accumulating shuttle errors [82]. Second, extra spaces could be used to prepare Bell pairs to use for gate teleportation or entanglement swapping [83]. Finally, the most aggressive method is to designate every other surface code as a dedicated logical ancilla for GHZ state preparation, allowing for constant-depth CNOTs [84]–[86]. These methods may enable long-distance communication at significantly reduced cost compared to lattice surgery; however, a thorough evaluation of the tradeoffs involved is beyond the scope of this work.

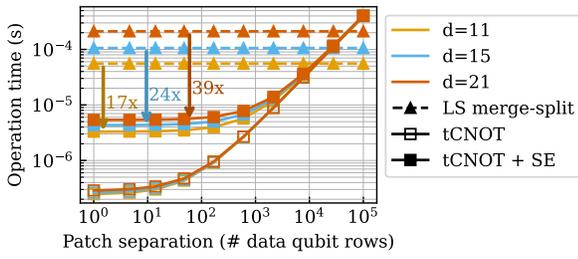


Fig. 10. Duration of lattice surgery merge-split and tCNOT operations in SNAQ as a function of patch separation, assuming a per-shuttle latency of 2 ns and a density of  $\rho = 1$ . tCNOT+SE is the latency for one tCNOT and half an SE round, which is the effective operation latency in a compiled program. For separation distances under 200 dots, tCNOT takes less than a microsecond (without the SE round), and tCNOT+SE is competitive with lattice surgery out to separation distances over 10,000.

### E. Logical clock speed

To further compare the two modes of logical operation in SNAQ, we calculate the latencies of the operations, which directly determine the logical clock speed of the fault-tolerant processor. Figure 10 shows the durations of the tCNOT and LS merge-split operations in SNAQ over varying separation distance for three surface code distances  $d \in (11, 15, 21)$ . The LS latency is the time to complete  $d$  pipelined SE rounds. For the tCNOT, we show both the time to complete the tCNOT itself as well as the overall average operation latency assuming that two tCNOTs are performed for every SE round [5], [52]. We can speed up these individual SE rounds by using both edges of the array for initialization and measurement of the ancilla qubits, roughly halving the number of serialization waves. For separation distances within the tCNOT’s fidelity-limited range of 100, the tCNOT+SE latency is effectively constant and is consistently over  $10\times$  faster than LS.

In Figure 11, we compare logical clock speed to the baseline architectures, varying each hardware parameter across a wide range, for three target logical error rates  $p_L \in (10^{-4}, 10^{-5}, 10^{-6})$ . For each instance, we simulate code distances up to 15, find the smallest distance that reaches the target  $p_L$ , and calculate the logical cycle time of the code. SNAQ tCNOT consistently maintains an advantage across the parameter space, with the exception of high  $p_{id}$  (lower right), where no instances of SNAQ could reach the target logical error rate. SNAQ can operate even with very high shuttling error, where the baselines fail (lower left). This demonstrates the key tradeoff of the SNAQ architecture compared to the baselines: physical qubits are packed much more closely, reducing required shuttling costs but increasing idling time due to readout serialization. SNAQ’s improvement factor is most sensitive to the readout density  $\rho$  (upper left).

To further investigate SNAQ’s dependence on  $\rho$ , we study this tradeoff in more depth in Figure 12. For fixed target  $p_L = 10^{-6}$  and varying  $p_{id}$  and  $\rho$ , we again calculate the required code distance and then convert to logical clock speeds. Across the studied range of  $\rho$ , we see that SNAQ’s tCNOT speedup varies from  $2\times$  to  $15\times$ , and its LS slowdown varies from  $10\times$

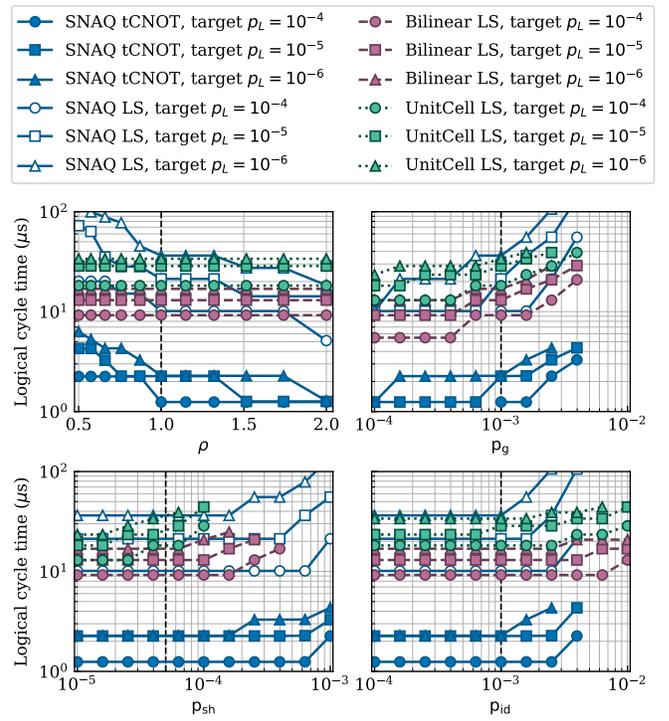


Fig. 11. Logical cycle times of proposed spin qubit architectures under varying noise parameters at fixed target logical error rates. Each plot shows sensitivity to one of the four hardware parameters ( $\rho, p_g, p_{sh}, p_{id}$ ). Vertical dashed lines indicates default parameter values when held constant in other plots. Points are omitted for parameter settings where no simulated code distance could reach the target logical error rate. SNAQ’s tCNOT maintains a consistent latency advantage across a wide range of hardware parameters, while SNAQ’s LS operation is generally slower than that of the baselines. SNAQ is more resilient to high  $p_{sh}$  than the baselines (lower left), but fails for sufficiently-high  $p_{id}$  (lower right).

to negligible. We also see that SNAQ fails for  $p_{id}$  significantly above  $10^{-3}$ , with the exact failure point depending on  $\rho$ .

Finally, we consider performance at lower  $p_L$  by extrapolating the simulated data using the models discussed in Section V-C. For this analysis, we use fixed physical error rates of  $(p_g, p_{sh}, p_{id}) = (10^{-3}, 10^{-5}, 10^{-4})$ , simulate distances up to 11, and fit the model in Eq. 4 to the data for a range of values of  $\rho$ . These fits allow us to determine the code distance needed for each architecture to achieve a certain logical error rate, which we can then convert to logical clock speeds. Figure 13 shows the comparison of the logical operation speeds in SNAQ to those of UnitCell and Bilinear. We find that SNAQ’s tCNOT is significantly faster than either of the baselines across the entire studied range. For a near-term target error rate of  $10^{-6}$ , we find consistent  $7.5\times$  and  $10.6\times$  speedups across the density range. For lower  $p_L$ , the improvement is smaller but still remains above  $4\times$ . Because the interaction distance of the tCNOT is limited, we also compare the lattice surgery time between SNAQ and the baselines, finding that SNAQ’s lattice surgery still outperforms the baselines for a target  $p_L$  of  $10^{-6}$  but is less competitive for lower  $p_L$  and lower  $\rho$ .

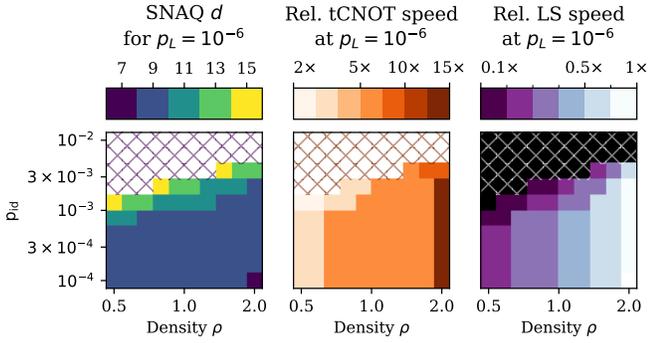


Fig. 12. Quantifying SNAQ’s logical performance as  $\rho$  and  $p_{id}$  vary. Parameters  $p_{sh}$  and  $p_g$  are held constant at  $5 \times 10^{-5}$  and  $10^{-3}$ , respectively. Hatched area shows parameter range where no SNAQ instance reached the target logical error rate of  $p_L \leq 10^{-6}$ . Center plot shows relative speedup of SNAQ’s tCNOT compared to a lattice surgery operation on the baselines (choosing the faster baseline at each point), demonstrating logical clock speedups between  $2\times$  and  $15\times$ . Rightmost plot shows relative speed of SNAQ’s LS operation compared to the best baseline, showing a significant slowdown up to  $10\times$  for low  $\rho$  and high  $p_{id}$ .

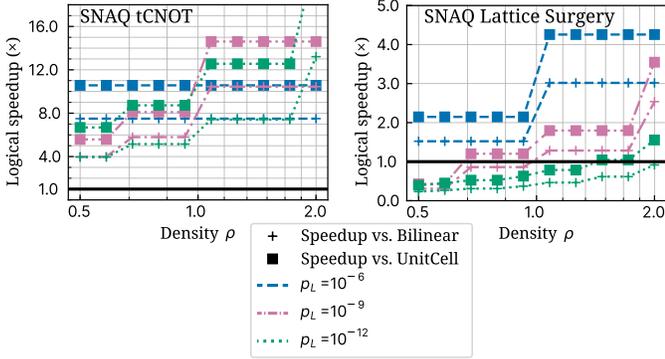


Fig. 13. Comparing projected performance at lower logical error rates under  $(p_g, p_{sh}, p_{id}) = (10^{-3}, 10^{-5}, 10^{-4})$  for various  $\rho$ , assuming patches are adjacent. SNAQ’s tCNOT outperforms baseline lattice surgery methods by over  $4\times$  even for low  $\rho$  and can provide up to  $22\times$  improvement in the range studied here. Lattice surgery on SNAQ is fast for higher  $p_L$  and  $\rho$ , but becomes less efficient for lower  $p_L$  due to the significant increase in required code distance (making each SE round take much longer).

### F. Architectural implications

The fidelity and latency studies discussed above suggest that both tCNOTs and lattice surgery are valid options for logical operations on the SNAQ architecture, with tCNOTs providing a significant speed advantage for sufficiently short separation distance and LS operations enabling longer-distance communication without degrading fidelity, but at the cost of over  $10\times$  slower operation speed.

A SNAQ device of width  $w$  can support a surface code distance of up to  $d = w - 2$  in a logical  $1 \times N$  configuration if logical operations are restricted to only tCNOTs. This processor would have a maximum allowed interaction distance within which the tCNOT’s performance is acceptable, which would depend on the relative strengths of shuttling

errors to other sources of error. This limited connectivity is reminiscent of nearest-neighbor-connected noisy intermediate-scale quantum (NISQ) processors, where long-range interactions require multi-hop, higher-cost operations, except in this case the extra cost is in latency instead of added error. In these NISQ processors, locality-aware mapping and routing techniques were developed to minimize the amount of long-distance communication needed [87]; SNAQ may benefit from the adaptation of these techniques to make use of fast tCNOTs whenever possible. Compiling to a transversal-equipped architecture presents several interesting opportunities to further improve processor speed. First, logical shuttles in the same direction can be performed in parallel. Second, only one of the two logical qubits is actively in-use during the shuttling periods of a tCNOT, allowing the other qubit to take part in other operations in the meantime. Third, the end-to-end latency of a longer-distance tCNOT can be nearly halved if the mobile qubit is not shuttled back to its original location, but instead deposited close to the stationary qubit after the operation.

On the other hand, the same width- $w$  SNAQ device could also support a surface code distance of up to  $d = \lfloor \frac{w-3}{2} \rfloor$  in a logical  $2 \times N$  layout, enabling both tCNOTs and lattice surgery if one logical channel is left free to use as routing space. A processor with two available modes of communication is reminiscent of the latency hierarchies in classical computer architecture, where care must be taken to minimize the use of main memory accesses or network connections. This presents a compelling challenge for future compilation research.

## VI. FAULT-TOLERANT RESOURCE ESTIMATION

The numerics in the prior section showed that, using transversal CNOTs, the SNAQ architecture can perform individual logical operations  $2\text{--}15\times$  as fast as competing architectures for sufficiently close logical qubits. Such close-range tCNOTs support key application subroutines including quantum arithmetics and state preparation, which in turn can implement important quantum applications such as factoring and chemistry simulation [5]. In this section, we explicitly compile several key primitives—T and CCZ distillation, the Toffoli gate, GHZ state generation, and long-range transport—and use them to calculate the costs of larger subroutines (Adder and Lookup), with fixed  $\rho = 1$  and  $d = 27$ . With these subroutines as building blocks, we heuristically estimate the improvements SNAQ can achieve compared against baselines. Although large scale algorithms generally be nonlocal, we find that the qubit transport costs are sufficiently low that the tCNOTs still provide a speed advantage. In particular, we estimate up to  $2.4\times$  and  $2.9\times$  reductions over the Bilinear and UnitCell baselines for factoring, and expect significant reductions for various chemistry simulation algorithms as well.

### A. Primitives: state preparation, distillation, and logic

We simulate the runtime of several key fault-tolerant circuit primitives by compiling them to either transversal or lattice surgery instruction sets and calculating the execution time for each architecture. For SNAQ, we insert an SE round after

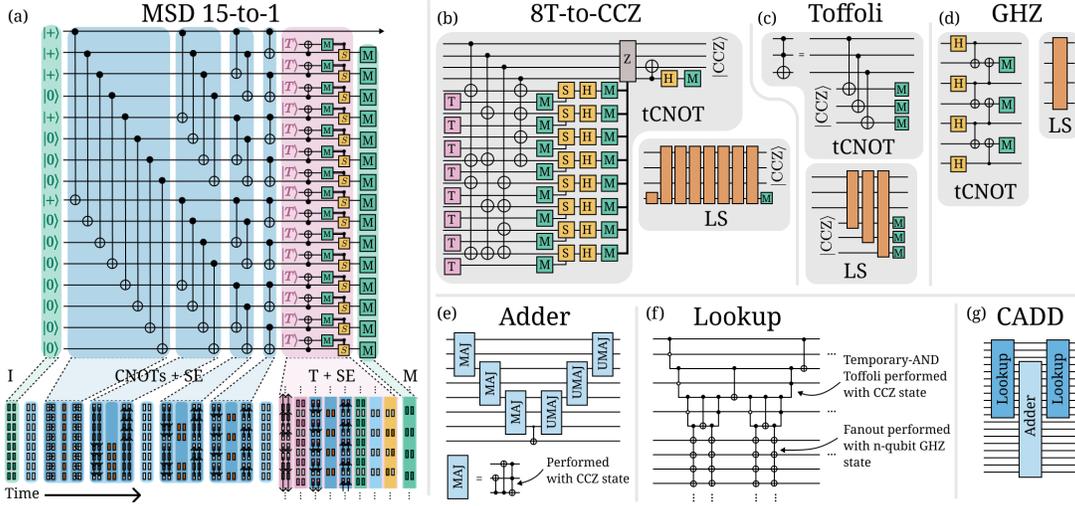


Fig. 14. (a) Logical circuit implementing 15-to-1 T distillation. CNOTs are shaded in groups to show those that can be performed in parallel using transversal CNOTs. *Bottom*: A  $2 \times 8$  SNAQ layout executing the circuit. The schedule involves parallel tCNOTs and T gate injection, interleaved with four SE rounds. (b-d) Logical primitives described in Section VI-A, expressed in CNOT and lattice surgery (LS) circuits: 8T-to-CCZ distillation, a Toffoli gate (note: auto-corrections not shown), and GHZ state preparation. (e-f) Larger logical subroutines described in Section VI-B: Adder [88] and Lookup [5]. (g) The lookup-controlled adder (CADD) is the key component of the quantum factoring algorithm.

every two layers of tCNOTs. In the following section, we describe the primitives we consider.

1) *15-to-1 T state distillation*: The 15-to-1 distillation circuit involves preparing 15 noisy T states and performing an encoding circuit to produce one higher-fidelity T state. For SNAQ, we use a tCNOT-compatible circuit shown in Figure 14a. We assume T state preparation time comparable to surface code initialization time and no decoding delay for conditional S corrections. Our construction assumes that four SE rounds are performed during the distillation, as shown in Figure 14. We assume the implementation of a fold-transversal S gate [89], [90], which we set to have a latency equal to  $2d \cdot t_{\text{shuttle}} + t_{\text{CNOT}}$ . Neither of the two prior spin qubit architecture proposals is compatible with parallelizable transversal CNOTs, so lattice surgery is needed instead. For the UnitCell architecture, we can use the implementation from Figure 11 of Ref. [91], which requires  $6d$  SE rounds on a 15-patch two-dimensional layout of surface codes. For the Bilinear architecture, using a virtual  $2 \times N$  layout of logical surface code qubits as depicted in Figure 3 of Ref. [24], we can implement the same circuit restricted to a  $2 \times 5$  layout, which extends the temporal cost to  $12d$  SE rounds.

2) *8T-to-CCZ factory*: The 8T-to-CCZ factory [92], [93] (Figure 14b) is an alternative magic state preparation method that outputs a  $|\text{CCZ}\rangle$  state, which can be used to perform a Toffoli gate. We follow the transversal implementation from Ref. [5] and compare to the lattice surgery implementation from Ref. [91], which involves 8 lattice surgery operations. For both transversal and lattice surgery implementations, we assume the input T states are prepared via cultivation [94], which is approximately  $15 \times$  lower-cost on transversal-equipped architectures [95] (accounting for the added shuttling overhead, which is small on SNAQ). This results in a  $4.3\text{-}5.6 \times$

more efficient  $|\text{CCZ}\rangle$  factory at  $d = 27$ , as shown in Table II.

3) *GHZ states*: An  $n$ -qubit GHZ state (Figure 14d) can either be prepared using linear-nearest neighbour parallel tCNOTs [84] or implicitly created when performing a Pauli product measurement via lattice surgery, which we use as baseline. GHZ states are used in CNOT fan-outs, which, together with Toffoli gates (Figure 14c), form the Lookup subroutine, which is important for factoring [4] and chemistry simulation tasks that require PREP and SELECT oracles [96]. GHZ states are also used in Pauli exponentials in Trotterized Hamiltonian simulation [97]. SNAQ’s GHZ state generation is  $2.9\text{-}5.6 \times$  more efficient than the baselines.

4) *Logical qubit transport*: The final primitive we study is the transport of logical information across large distances. A long-range transport operation can either be done by shuttling with intermediate SE rounds to avoid the accumulation of errors, or through lattice surgery [48]. Therefore, the latency of a transport operation increases with distance, but is upper-bounded by the cost of an LS operation. Assuming we perform one SE round for every 100 qubit distances traveled (Fig. 9),  $k$ -patch transport via shuttling in SNAQ takes  $\lfloor kd/100 \rfloor \cdot t_{\text{SE}} + 2kd \cdot t_{\text{sh}}$  time. The SE round cost has been amortized for the result shown in Table II.

## B. Subroutines

The two subroutines we consider are built of basic logical operations and the primitives discussed previously. We simulate the execution of their circuits, accounting for the previously calculated costs of primitives.

1) *Adder*: The quantum adder (Figure 14e) performs a reversible addition. It can be implemented as repeated applications of the MAJ and UMA blocks, which require  $|\text{CCZ}\rangle$  states and local CNOT interactions. We use a transversal adder for SNAQ [5], and a volume-optimized adder using lattice surgery

	MSD 15-to-1		8T-to-CCZ		GHZ, $n = 100$		$k$ -patch transport	
	Time ( $\mu$ s)	Vol. (qubit-s)	Time ( $\mu$ s)	Vol. (q-s)	Time ( $\mu$ s)	Vol. (q-s)	Time ( $\mu$ s)	Vol. (q-s)
Bilinear	654.5	9.536	2182.1	31.793	54.5	15.893	54.5	$0.0795(k+2)$
UnitCell	421.2	9.205	2808.5	40.920	70.2	20.456	39.0	$0.102(k+2)$
<b>SNAQ</b>	<b>302.2</b>	<b>7.045</b>	<b>250.0</b>	<b>7.284</b>	<b>18.7</b>	<b>5.426</b>	$4.0k$	<b><math>0.00586k</math></b>

TABLE II  
COST COMPARISON OF FAULT-TOLERANT PRIMITIVES AT  $d = 27$ .

	Adder, $n = 139$		Lookup(7, 139)	
	Time (ms)	Vol. (q-s)	Time (ms)	Vol. (q-s)
Bilinear	<b>3.767</b>	21350.9	3.463	1535.0
UnitCell	4.002	23055.4	4.458	2014.7
<b>SNAQ</b>	7.584	<b>13446.9</b>	<b>0.711</b>	<b>354.4</b>

	CADD(10, 1067) Ref. [4]		CADD(7, 139) Ref. [5]	
	Time (ms)	Vol. (q-s)	Time (ms)	Vol. (q-s)
Bilinear	79.1	1838772	10.7	60918
UnitCell	95.4	2225879	13.0	74821
<b>SNAQ</b>	<b>76.5</b>	<b>756051</b>	<b>10.0</b>	<b>17718</b>

TABLE III  
FULL COSTS OF LOGICAL SUBROUTINES AND CONTROLLED ADDERS (CADD) AT  $d = 27$ . INCLUDES CCZ FACTORIES AND QUBIT TRANSPORT COSTS. CADD DOMINATES THE COST OF QUANTUM FACTORING.

as the baseline [4]. The duration of both adders depends on the *reaction time*, or how quickly the architecture can resolve a series of measurements where each measurement outcome determines the following measurement basis. Due to the need for a reactive CZ gate and measurement serialization, the SNAQ adder incurs a  $7\times$  longer reaction time than the baseline, leading to a  $2\times$  slower overall duration. However, due to cheaper  $|CCZ\rangle$  factories and lower  $|CCZ\rangle$  consumption rates, SNAQ achieves a lower volume with a much smaller  $|CCZ\rangle$  factory, leading to a  $3.2\times$  reduction in cost.

2) *Lookup*: The quantum Lookup (Figure 14f) appears in many algorithms due to its use in both quantum arithmetic [98] and chemistry [96]. It consists of a Toffoli ladder, implemented using  $|CCZ\rangle$  states, and a CNOT-fanout that can be implemented using either a  $|GHZ\rangle$  state or lattice surgery. The state-of-the-art transversal implementation is given in [5], and we use an established lattice surgery construction for the baselines [4], [99]. While the Toffoli part of the Lookup is reaction-time limited, this cost is hidden by the much costlier CNOT-fanout operation, which can be done in parallel; as such, the look-up is said to be Clifford-limited [99]. Due to faster transversal implementation of the CNOT-fanout, SNAQ achieves a  $4.87\times$  to  $6.26\times$  reduction in duration compared to baseline, leading to a  $4.37\times$  to  $5.74\times$  volume reduction.

### C. Algorithm impact

We can heuristically estimate the impact on large-scale algorithms by considering the fraction of total algorithm volume spent on the above subroutines and the speedup achieved for that subroutine. The key subroutine for factoring is the lookup-controlled adder (CADD), which performs a quantum addition on states in specific superpositions prepared using a Lookup (Figure 14g). With one additional round of qubit shuttling, the data qubits can be laid out as illustrated in Figure 14g. The Lookup and Adder sizes are chosen to jointly

optimize compilation cost and logical error-rate performance under hardware constraints such as reaction time and gate error rates. Although sweeping this full parameter space to determine the optimum for various factoring problems is an important direction, it is beyond the scope of this work. Instead, we use the optimized hyperparameters reported in [4], [5] as reasonable reference points to estimate the range of improvements that SNAQ may achieve relative to the baselines, as shown in Table III. Aside from two QFTs that contribute negligible volume, the entire factoring algorithm can be seen as repeated applications of CADD, which therefore account for the overwhelming majority of the total volume [4], [5]. We therefore estimate that the  $2.43$ - $4.22\times$  improvements in CADD (Table III) propagate to very similar volume reductions for the entire factoring algorithm relative to baselines. This estimate accounts for all required qubit transport costs, which are included in the CADD cost.

Algorithms for chemistry simulation usually employ either a qubitization [100], [101] or Trotterization [97] approach. The state-of-the-art qubitization compilation technique uses repeated applications of the PREPARE and SELECT oracle [96], where Lookup operations and single qubit rotations that require  $|T\rangle$  states account for  $\geq 90\%$  the cost [5], [102]. On the other hand, Trotterization employs repeated applications of Pauli exponentials, which are implementable using GHZ states and magic states. We could expect time and volume reductions for both approaches, since SNAQ provides speedups to  $|T\rangle$  distillation, GHZ state preparation, and Lookups. However, the costs we estimate in Table III are specific to the register sizes needed for factoring; the exact algorithmic speedup will depend on the specific chemistry problem being studied.

## VII. CONCLUSION

In this work, we proposed the SNAQ surface code, a novel surface code implementation tailored for spin qubits. To overcome the readout component size problem, we introduced the idea of serialized ancilla qubit initialization and readout, showing that this is an effective way to enable error correction on a near-term-manufacturable silicon quantum dot array. SNAQ is more space-efficient than prior proposals while providing a substantial logical clock speedup through transversal logic by leveraging rapid spin shuttling and a dense dot array.

SNAQ is sensitive to coherence time but tolerates higher shuttling errors than the baselines, demonstrating the tradeoff between readout efficiency and qubit density. Concatenating SNAQ with a higher-level QEC code could provide a path towards achieving ultra-low logical error rates beyond the coherence-limited error floor. The fixed-width spin qubit array

may also be amenable to other specific classes QEC codes, in particular Floquet codes [103] or adaptive concatenated schemes [104]. These codes may be harmed less by the serialization of ancilla measurement due to their already-partitioned syndrome extraction schedules.

Overall, SNAQ demonstrates that the often-assumed 1:1 readout-to-qubit ratio is neither necessary nor optimal on a shuttling-equipped architecture such as silicon spin qubits. Enabled by the unique capability of fast spin shuttling, SNAQ is both orders-of-magnitude more area-efficient and significantly faster for common subroutines than proposals that mimic 2D grid layouts. Our work highlights qubit coherence and readout density as the most critical device metrics to enable this new architecture, providing a compelling, practical path toward fault-tolerant spin qubit processors.

#### ACKNOWLEDGEMENTS

The authors thank Prithvi Prabhu, Fahd Mohiyaddin, and Nathaniel Bishop at Intel for helpful discussions on spin qubit manufacturing constraints and performance metrics. The authors are also grateful to Mariesa Teo for her assistance in understanding the significance of the adder and QROM benchmarks.

This work is funded in part by the STAQ project under award NSF Phy-232580; 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), in part by the NSF National Virtual Quantum Laboratory program, in part based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, and in part by the Army Research Office under Grant Number W911NF-23-1-0077. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

FTC is the Chief Scientist for Quantum Software at Infleqtion and an advisor to Quantum Circuits, Inc.

#### REFERENCES

[1] Isaac H. Kim, Ye-Hua Liu, Sam Pallister, William Pol, Sam Roberts, and Eunseok Lee. Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules. *Physical Review Research*, 4(2):023019, April 2022.

[2] Michael E. Beverland, Prakash Murali, Matthias Troyer, Krysta M. Svore, Torsten Hoefler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. Assessing requirements to scale to practical quantum advantage, November 2022. arXiv:2211.07629 [quant-ph].

[3] Tyler Leblond, Christopher Dean, George Watkins, and Ryan Bennink. Realistic Cost to Execute Practical Quantum Circuits using Direct Clifford+T Lattice Surgery Compilation. *ACM Transactions on Quantum Computing*, 5(4):1–28, December 2024.

[4] Craig Gidney. How to factor 2048 bit RSA integers with less than a million noisy qubits, May 2025. arXiv:2505.15917 [quant-ph].

[5] Hengyu Zhou, Casey Duckering, Chen Zhao, Dolev Bluvstein, Madelyn Cain, Aleksander Kubica, Sheng-Tao Wang, and Mikhail D. Lukin. Resource Analysis of Low-Overhead Transversal Architectures for Reconfigurable Atom Arrays. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, pages 1432–1448, Tokyo Japan, June 2025. ACM.

[6] R. Maurand, X. Jehl, D. Kotekar-Patil, A. Corna, H. Bohuslavskiy, R. Laviéville, L. Hutin, S. Barraud, M. Vinet, M. Sanquer, and S. De Franceschi. A CMOS silicon spin qubit. *Nature Communications*, 7(1):13575, November 2016.

[7] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak. Silicon CMOS architecture for a spin-based quantum computer. *Nature Communications*, 8(1):1766, December 2017.

[8] Ruoyu Li, Luca Petit, David P. Franke, Juan Pablo Dehollain, Jonas Helsen, Mark Steudtner, Nicole K. Thomas, Zachary R. Yoscovits, Kanwal J. Singh, Stephanie Wehner, Lieven M. K. Vandersypen, James S. Clarke, and Menno Veldhorst. A crossbar network for silicon quantum dot qubits. *Science Advances*, 4(7):eaar3960, July 2018.

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

[10] Samuel Neyens, Otto K. Zietz, Thomas F. Watson, Florian Luthi, Aditi Nethewewala, Hubert C. George, Eric Henry, Mohammad Islam, Andrew J. Wagner, Felix Borjans, Elliot J. Connors, J. Corrigan, Matthew J. Curry, Daniel Keith, Roza Kotlyar, Lester F. Lampert, Mateusz T. Mądzik, Kent Millard, Fahd A. Mohiyaddin, Stefano Pellerano, Ravi Pillarisetty, Mick Ramsey, Rostyslav Savvitsky, Simon Schaal, Guoji Zheng, Joshua Ziegler, Nathaniel C. Bishop, Stephanie Bojarski, Jeanette Roberts, and James S. Clarke. Probing single electrons across 300-mm spin qubit wafers. *Nature*, 629(8010):80–85, May 2024.

[11] Paul Steinacker, Nard Dumoulin Stuyck, Wee Han Lim, Tuomo Tanttu, MengKe Feng, Andreas Nick, Santiago Serrano, Marco Candido, Jesus D. Cifuentes, Fay E. Hudson, Kok Wai Chan, Stefan Kubicek, Julien Jussot, Yann Canvel, Sofie Beyne, Yosuke Shimura, Roger Loo, Clement Godfrin, Bart Raes, Sylvain Baudot, Danny Wan, Arne Laucht, Chih Hwan Yang, Andre Saraiva, Christopher C. Escott, Kristiaan De Greve, and Andrew S. Dzurak. A 300 mm foundry silicon spin qubit unit cell exceeding 99% fidelity in all operations, October 2024. arXiv:2410.15590.

[12] Andrea Morello, Jarryd J. Pla, Floris A. Zwanenburg, Kok W. Chan, Kuan Y. Tan, Hans Huebl, Mikko Möttönen, Christopher D. Nugroho, Changyi Yang, Jessica A. van Donkelaar, Andrew D. C. Alves, David N. Jamieson, Christopher C. Escott, Lloyd C. L. Hollenberg, Robert G. Clark, and Andrew S. Dzurak. Single-shot readout of an electron spin in silicon. *Nature*, 467(7316):687–691, October 2010.

[13] M. J. Curry, M. Rudolph, T. D. England, A. M. Mounce, R. M. Jock, C. Bureau-Oxton, P. Harvey-Collard, P. A. Sharma, J. M. Anderson, D. M. Campbell, J. R. Wendt, D. R. Ward, S. M. Carr, M. P. Lilly, and M. S. Carroll. Single-Shot Readout Performance of Two Heterojunction-Bipolar-Transistor Amplification Circuits at Millikelvin Temperatures. *Scientific Reports*, 9(1):16976, November 2019.

[14] Elliot J. Connors, JJ Nelson, and John M. Nichol. Rapid High-Fidelity Spin-State Readout in  $\text{Si}/\text{SiGe}$  Quantum Dots via rf Reflectometry. *Physical Review Applied*, 13(2):024019, February 2020.

[15] N. W. Hendrickx, W. I. L. Lawrie, L. Petit, A. Sammak, G. Scappucci, and M. Veldhorst. A single-hole spin qubit. *Nature Communications*, 11(1):3478, July 2020.

[16] G. A. Oakes, V.N. Ciriano-Tejel, D.F. Wise, M. A. Fogarty, T. Lundberg, C. Lainé, S. Schaal, F. Martins, D.J. Ibberson, L. Hutin, B. Bertrand, N. Stelmashenko, J.W.A. Robinson, L. Ibberson, A. Hashim, I. Siddiqi, A. Lee, M. Vinet, C. G. Smith, J.J.L. Morton, and M.F. Gonzalez-Zalba. Fast High-Fidelity Single-Shot Readout of Spins in Silicon Using a Single-Electron Box. *Physical Review X*, 13(1):011023, February 2023.

[17] Quentin Schmidt, Baptiste Jadot, Brian Martinez, Thomas Houriez, Adrien Morel, Tristan Meunier, Gaël Pillonnet, Gérard Billiot, Aloysius

- Jansen, Xavier Jehl, Yvain Thonnart, and Franck Badets. Compact frequency multiplexed readout of silicon quantum dots in monolithic FD-SOI 28nm technology. In *2024 IEEE European Solid-State Electronics Research Conference (ESSERC)*, pages 157–160, September 2024.
- [18] Inga Seidler, Tom Struck, Ran Xue, Niels Focke, Stefan Trelenkamp, Hendrik Bluhm, and Lars R. Schreiber. Conveyor-mode single-electron shuttling in Si/SiGe for a scalable quantum computing architecture. *npj Quantum Information*, 8(1):100, August 2022.
- [19] Maxim De Smet, Yuta Matsumoto, Anne-Marije J. Zwerver, Larysa Tryputen, Sander L. de Snoo, Sergey V. Amitonov, Amir Sammak, Nodar Samkharadze, Önder Gül, Rick N. M. Wasserman, Maximilian Rimbach-Russ, Giordano Scappucci, and Lieven M. K. Vandersypen. High-fidelity single-spin shuttling in silicon, June 2024. arXiv:2406.07267 [cond-mat].
- [20] Jelmer M. Boter, Juan P. Dehollain, Jeroen P.G. Van Dijk, Yuanxing Xu, Toivo Hensgens, Richard Versluis, Henricus W.L. Naus, James S. Clarke, Menno Veldhorst, Fabio Sebastiano, and Lieven M.K. Vandersypen. Spiderweb Array: A Sparse Spin-Qubit Array. *Physical Review Applied*, 18(2):024053, August 2022.
- [21] Zhenyu Cai, Adam Siegel, and Simon Benjamin. Looped Pipelines Enabling Effective 3D Qubit Lattices in a Strictly 2D Device. *PRX Quantum*, 4(2):020345, June 2023.
- [22] Armands Strikis and Lucas Berent. Quantum Low-Density Parity-Check Codes for Modular Architectures. *PRX Quantum*, 4(2):020321, May 2023.
- [23] Matthias Künne, Alexander Willmes, Max Oberländer, Christian Goraew, Julian D. Teske, Harsh Bhardwaj, Max Beer, Eugen Kammerloher, René Otten, Inga Seidler, Ran Xue, Lars R. Schreiber, and Hendrik Bluhm. The SpinBus architecture for scaling spin qubits with electron shuttling. *Nature Communications*, 15(1):4977, June 2024.
- [24] Adam Siegel, Armands Strikis, and Michael Fogarty. Towards Early Fault Tolerance on a  $\text{Si}$  Array of Qubits Equipped with Shuttling. *PRX Quantum*, 5(4):040328, November 2024.
- [25] Adam Siegel, Zhenyu Cai, Hamza Jnane, Balint Koczor, Shaun Pexton, Armands Strikis, and Simon Benjamin. Snakes on a Plane: mobile, low dimensional logical qubits on a 2D surface, January 2025. arXiv:2501.02120 [quant-ph] version: 1.
- [26] Stephan G. J. Phillips, Mateusz T. Mądzik, Sergey V. Amitonov, Sander L. de Snoo, Maximilian Russ, Nima Kalhor, Christian Volk, William I. L. Lawrie, Delphine Brousse, Larysa Tryputen, Brian Paquelet Wuetz, Amir Sammak, Menno Veldhorst, Giordano Scappucci, and Lieven M. K. Vandersypen. Universal control of a six-qubit quantum processor in silicon. *Nature*, 609(7929):919–924, September 2022.
- [27] Francesco Borsoi, Nico W. Hendrickx, Valentin John, Marcel Meyer, Sayr Motz, Floor van Riggelen, Amir Sammak, Sander L. de Snoo, Giordano Scappucci, and Menno Veldhorst. Shared control of a 16 semiconductor quantum dot crossbar array. *Nature Nanotechnology*, 19(1):21–27, January 2024.
- [28] Eric Henry. 2D Silicon Spin Qubits. In *APS March Meeting 2025, MAR-C19:8*. American Physical Society, March 2025.
- [29] Sieu D. Ha, Edwin Acuna, Kate Raach, Zachery T. Bloom, Teresa L. Brecht, James M. Chappell, Maxwell D. Choi, Justin E. Christensen, Ian T. Counts, Dominic Daprano, J. P. Dodson, Kevin Eng, David J. Fialkow, Christina A. C. Garcia, Wonill Ha, Thomas R. B. Harris, Nathan Holman, Isaac Khalaf, Justine W. Matten, Christi A. Peterson, Clifford E. Plesha, Matthew J. Ruiz, Aaron Smith, Bryan J. Thomas, Samuel J. Whiteley, Thaddeus D. Ladd, Michael P. Jura, Matthew T. Rakher, and Matthew G. Borselli. Two-dimensional Si spin qubit arrays with multilevel interconnects, February 2025. arXiv:2502.08861 [quant-ph].
- [30] Hubert C. George, Mateusz T. Mądzik, Eric M. Henry, Andrew J. Wagner, Mohammad M. Islam, Felix Borjans, Elliot J. Connors, J. Corrigan, Matthew Curry, Michael K. Harper, Daniel Keith, Lester Lampert, Florian Luthi, Fahd A. Mohiyaddin, Sandra Murcia, Rohit Nair, Rambert Nahm, Aditi Nethewala, Samuel Neyens, Bishnu Patra, Roy D. Raharjo, Carly Rogan, Rostyslav Savvitsky, Thomas F. Watson, Josh Ziegler, Otto K. Zietz, Stefano Pellerano, Ravi Pillarisetty, Nathaniel C. Bishop, Stephanie A. Bojarski, Jeanette Roberts, and James S. Clarke. 12-Spin-Qubit Arrays Fabricated on a 300 mm Semiconductor Manufacturing Line. *Nano Letters*, 25(2):793–799, January 2025.
- [31] Guido Burkard, Thaddeus D. Ladd, Andrew Pan, John M. Nichol, and Jason R. Petta. Semiconductor spin qubits. *Reviews of Modern Physics*, 95(2):025003, June 2023.
- [32] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, September 2002.
- [33] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, September 2012.
- [34] Aaron J. Weinstein, Matthew D. Reed, Aaron M. Jones, Reed W. Andrews, David Barnes, Jacob Z. Blumoff, Larken E. Euliss, Kevin Eng, Bryan H. Fong, Sieu D. Ha, Daniel R. Hulbert, Clayton A. C. Jackson, Michael Jura, Tyler E. Keating, Joseph Kerckhoff, Andrey A. Kiselev, Justine Matten, Golam Sabbir, Aaron Smith, Jeffrey Wright, Matthew T. Rakher, Thaddeus D. Ladd, and Matthew G. Borselli. Universal logic with encoded spin qubits in silicon. *Nature*, 615(7954):817–822, March 2023.
- [35] Yi-Hsien Wu, Leon C. Camenzind, Patrick Büttler, Ik Kyeong Jin, Akito Noiri, Kenta Takeda, Takashi Nakajima, Takashi Kobayashi, Giordano Scappucci, Hsi-Sheng Goan, and Seigo Tarucha. Simultaneous High-Fidelity Single-Qubit Gates in a Spin Qubit Array, July 2025. arXiv:2507.11918 [quant-ph].
- [36] Joseph D. Broz, Jesse C. Hoke, Edwin Acuna, and Jason R. Petta. Demonstration of an always-on exchange-only spin qubit, August 2025. arXiv:2508.01033 [quant-ph].
- [37] Paul Steinacker, Nard Dumoulin Stuyck, Wee Han Lim, Tuomo Tanttu, MengKe Feng, Santiago Serrano, Andreas Nickl, Marco Candido, Jesus D. Cifuentes, Ensar Vahapoglu, Samuel K. Bartee, Fay E. Hudson, Kok Wai Chan, Stefan Kubicek, Julien Jussot, Yann Carvel, Sofie Beyne, Yosuke Shimura, Roger Loo, Clement Godfrin, Bart Raes, Sylvain Baudot, Danny Wan, Arne Laucht, Chih Hwan Yang, Andre Saraiva, Christopher C. Escott, Kristiaan De Greve, and Andrew S. Dzurak. Industry-compatible silicon spin-qubit unit cells exceeding 99% fidelity. *Nature*, 646(8083):81–87, October 2025.
- [38] D. Bacon, J. Kempe, D. A. Lidar, and K. B. Whaley. Universal Fault-Tolerant Quantum Computation on Decoherence-Free Subspaces. *Physical Review Letters*, 85(8):1758–1761, August 2000.
- [39] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley. Universal quantum computation with the exchange interaction. *Nature*, 408(6810):339–342, November 2000.
- [40] Mauricio Gutiérrez, Juan S. Rojas-Arias, David Obando, and Chien-Yuan Chang. Comparison of spin-qubit architectures for quantum error-correcting codes, June 2025. arXiv:2506.17190 [quant-ph].
- [41] Daniel P. Siewiorek, C. Gordon. Bell, and Allen. Newell. Computer structures : principles and examples / [edited by] daniel P. Siewiorek, C. Gordon bell, and allen newell. In *Computer structures : principles and examples*, McGraw-Hill computer science series. McGraw-Hill Book Company, Principles and examples, 1982. tex.lccn: 80027926.
- [42] Mark Oskin, Frederic T. Chong, Isaac L. Chuang, and John Kubiatowicz. Building quantum wires: the long and the short of it. *SIGARCH Comput. Archit. News*, 31(2):374–387, May 2003.
- [43] Neil H. E. Weste and David Money Harris. *CMOS VLSI design: a circuits and systems perspective*. Addison Wesley, Boston, 4th ed edition, 2011.
- [44] Hossein SeyyedAghaei, Mahmood Naderan-Tahan, Magnus Jahre, and Lieven Eeckhout. Memory-Centric MCM-GPU Architecture. *IEEE Computer Architecture Letters*, 24(1):101–104, January 2025.
- [45] Aditya Sridharan Iyer, Daehyun Kim, Saibal Mukhopadhyay, and Sung Kyu Lim. Multi-Tier 3D SRAM Module Design: Targeting Bit-Line and Word-Line Folding. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9. Association for Computing Machinery, New York, NY, USA, April 2025.
- [46] Dominic Horsman, Austin G. Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, December 2012.
- [47] Austin G. Fowler and Craig Gidney. Low overhead quantum computation using lattice surgery, August 2019. arXiv:1808.06709 [quant-ph].
- [48] Daniel Litinski. A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery. *Quantum*, 3:128, March 2019. arXiv:1808.02892 [cond-mat, physics:quant-ph].
- [49] Shintaro Sato and Yasunari Suzuki. Scheduling of syndrome measurements with a few ancillary qubits, August 2025. arXiv:2508.07913 [quant-ph].
- [50] Min Ye and Nicolas Delfosse. Quantum error correction for long chains of trapped ions. *Quantum*, 9:1920, November 2025.

- [51] Jonas Helsen, Mark Steudtner, Menno Veldhorst, and Stephanie Wehner. Quantum error correction in crossbar architectures. *Quantum Science and Technology*, 3(3):035005, July 2018.
- [52] Madelyn Cain, Chen Zhao, Hengyun Zhou, Nadine Meister, J. Pablo Bonilla Ataides, Arthur Jaffe, Dolev Bluvstein, and Mikhail D. Lukin. Correlated decoding of logical algorithms with transversal gates, March 2024. arXiv:2403.03272 [cond-mat, physics:quant-ph].
- [53] Hengyun Zhou, Chen Zhao, Madelyn Cain, Dolev Bluvstein, Casey Duckering, Hong-Ye Hu, Sheng-Tao Wang, Aleksander Kubica, and Mikhail D. Lukin. Algorithmic Fault Tolerance for Fast Quantum Computing, June 2024. arXiv:2406.17653 [quant-ph].
- [54] Madelyn Cain, Dolev Bluvstein, Chen Zhao, Shouzhen Gu, Nishad Maskara, Marcin Kalinowski, Alexandra A. Geim, Aleksander Kubica, Mikhail D. Lukin, and Hengyun Zhou. Fast correlated decoding of transversal logical algorithms, June 2025. arXiv:2505.13587 [quant-ph].
- [55] A. Elsayed, M. M. K. Shehata, C. Godfrin, S. Kubicek, S. Massar, Y. Canvel, J. Jussot, G. Simion, M. Mongillo, D. Wan, B. Govoreanu, I. P. Radu, R. Li, P. Van Dorpe, and K. De Greve. Low charge noise quantum dots with industrial CMOS manufacturing. *npj Quantum Information*, 10(1):70, July 2024.
- [56] A.M.J. Zwerver, S.V. Amitonov, S.L. De Snoo, M.T. Mądzik, M. Rimbach-Russ, A. Sammak, G. Scappucci, and L.M.K. Vandersypen. Shuttling an Electron Spin through a Silicon Quantum Dot Array. *PRX Quantum*, 4(3):030303, July 2023.
- [57] Akito Noiri, Kenta Takeda, Takashi Nakajima, Takashi Kobayashi, Amir Sammak, Giordano Scappucci, and Seigo Tarucha. A shuttling-based two-qubit logic gate for linking distant silicon quantum processors. *Nature Communications*, 13(1):5740, September 2022.
- [58] Juha T. Muhonen, Juan P. Dehollain, Arne Laucht, Fay E. Hudson, Rachpon Kalra, Takeharu Sekiguchi, Kohei M. Itoh, David N. Jamieson, Jeffrey C. McCallum, Andrew S. Dzurak, and Andrea Morello. Storing quantum information for 30 seconds in a nano-electronic device. *Nature Nanotechnology*, 9(12):986–991, December 2014.
- [59] M. Veldhorst, J. C. C. Hwang, C. H. Yang, A. W. Leenstra, B. De Ronde, J. P. Dehollain, J. T. Muhonen, F. E. Hudson, K. M. Itoh, A. Morello, and A. S. Dzurak. An addressable quantum dot qubit with fault-tolerant control-fidelity. *Nature Nanotechnology*, 9(12):981–985, December 2014.
- [60] Filip K. Malinowski, Frederico Martins, Peter D. Nissen, Edwin Barnes, Łukasz Cywiński, Mark S. Rudner, Saeed Fallahi, Geoffrey C. Gardner, Michael J. Manfra, Charles M. Marcus, and Ferdinand Kuemmeth. Notch filtering the nuclear environment of a spin qubit. *Nature Nanotechnology*, 12(1):16–20, January 2017.
- [61] Arne Laucht, Rachpon Kalra, Stephanie Simmons, Juan P. Dehollain, Juha T. Muhonen, Fahd A. Mohiyaddin, Solomon Freer, Fay E. Hudson, Kohei M. Itoh, David N. Jamieson, Jeffrey C. McCallum, Andrew S. Dzurak, and A. Morello. A dressed spin qubit in silicon. *Nature Nanotechnology*, 12(1):61–66, January 2017.
- [62] Bo Sun, Teresa Brecht, Bryan H. Fong, Moonmoon Akmal, Jacob Z. Blumoff, Tyler A. Cain, Faustin W. Carter, Dylan H. Finestone, Micha N. Fireman, Wonil Ha, Anthony T. Hatke, Ryan M. Hickey, Clayton A. C. Jackson, Ian Jenkins, Aaron M. Jones, Andrew Pan, Daniel R. Ward, Aaron J. Weinstein, Samuel J. Whiteley, Parker Williams, Matthew G. Borselli, Matthew T. Rakher, and Thaddeus D. Ladd. Full-Permutation Dynamical Decoupling in Triple-Quantum-Dot Spin Qubits. *PRX Quantum*, 5(2):020356, June 2024.
- [63] K. Takeda, A. Noiri, J. Yoneda, T. Nakajima, and S. Tarucha. Resonantly Driven Singlet-Triplet Spin Qubit in Silicon. *Physical Review Letters*, 124(11):117701, March 2020.
- [64] Adam R. Mills, Charles R. Guinn, Michael J. Gullans, Anthony J. Sigillito, Mayer M. Feldman, Erik Nielsen, and Jason R. Petta. Two-qubit silicon quantum processor with operation fidelity exceeding 99%. *Science Advances*, 8(14):eabn5130, April 2022.
- [65] J. M. Elzerman, R. Hanson, L. H. Willems van Beveren, B. Witkamp, L. M. K. Vandersypen, and L. P. Kouwenhoven. Single-shot read-out of an individual electron spin in a quantum dot. *Nature*, 430(6998):431–435, July 2004.
- [66] Jacob Z. Blumoff, Andrew S. Pan, Tyler E. Keating, Reed W. Andrews, David W. Barnes, Teresa L. Brecht, Edward T. Croke, Larken E. Euliss, Jacob A. Fast, Clayton A.C. Jackson, Aaron M. Jones, Joseph Kerckhoff, Robert K. Lanza, Kate Raach, Bryan J. Thomas, Roland Velunta, Aaron J. Weinstein, Thaddeus D. Ladd, Kevin Eng, Matthew G. Borselli, Andrew T. Hunter, and Matthew T. Rakher. Fast and High-Fidelity State Preparation and Measurement in Triple-Quantum-Dot Spin Qubits. *PRX Quantum*, 3(1):010352, March 2022.
- [67] D. Keith, M. G. House, M. B. Donnelly, T. F. Watson, B. Weber, and M. Y. Simmons. Single-Shot Spin Readout in Semiconductors Near the Shot-Noise Sensitivity Limit. *Physical Review X*, 9(4):041003, October 2019.
- [68] Rubén M. Otxoa, Josu Etxezarreta Martinez, Paul Schnabl, Normann Mertig, Charles Smith, and Frederico Martins. SpinHex: A low-crosstalk, spin-qubit architecture based on multi-electron couplers, April 2025. arXiv:2504.03149 [quant-ph].
- [69] Mateusz T. Mądzik, Florian Luthi, Gian Giacomo Guerreschi, Fahd A. Mohiyaddin, Felix Borjans, Jason D. Chadwick, Matthew J. Curry, Joshua Ziegler, Sarah Atanasov, Peter L. Bavdaz, Elliot J. Connors, J. Corrigan, H. Ekmel Ercan, Robert Flory, Hubert C. George, Benjamin Harpt, Eric Henry, Mohammad M. Islam, Nader Khammassi, Daniel Keith, Lester F. Lampert, Todor M. Mladenov, Randy W. Morris, Aditi Nethewala, Samuel Neyens, René Otten, Linda P. Osuna Ibarra, Bishnu Patra, Ravi Pillarisetty, Shavindra Premaratne, Mick Ramsey, Andrew Risinger, John D. Rooney, Rostyslav Savvitsky, Thomas F. Watson, Otto K. Zietz, Anne Y. Matsuura, Stefano Pellerano, Nathaniel C. Bishop, Jeanette Roberts, and James S. Clarke. Operating two exchange-only qubits in parallel. *Nature*, 647(8091):870–875, November 2025.
- [70] Dávid Pataki and András Pályi. Compiling the surface code to crossbar spin qubit architectures. *Physical Review B*, 111(11):115307, March 2025.
- [71] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021. arXiv:2103.02202 [quant-ph].
- [72] Oscar Higgott. PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching, July 2021. arXiv:2105.13082 [quant-ph].
- [73] F. Setiawan, Hoi-Yin Hui, J. P. Kestner, Xin Wang, and S. Das Sarma. Robust two-qubit gates for exchange-coupled qubits. *Physical Review B*, 89(8):085314, February 2014.
- [74] Jason D. Chadwick, Gian Giacomo Guerreschi, Florian Luthi, Mateusz T. Mądzik, Fahd A. Mohiyaddin, Prithviraj Prabhu, Albert T. Schmitz, Andrew Litteken, Shavindra Premaratne, Nathaniel C. Bishop, Anne Y. Matsuura, and James S. Clarke. Short two-qubit pulse sequences for exchange-only spin qubits in two-dimensional layouts. *Physical Review A*, 111(5):052616, May 2025.
- [75] Xiao-Gang Wen. Quantum Orders in an Exact Soluble Model. *Physical Review Letters*, 90(1):016803, January 2003.
- [76] Alexey A. Kovalev, Ilya Dumer, and Leonid P. Pryadko. Design of additive quantum codes via the code-word-stabilized framework. *Physical Review A*, 84(6):062319, December 2011.
- [77] Barbara M. Terhal, Fabian Hassler, and David P. DiVincenzo. From Majorana fermions to topological order. *Physical Review Letters*, 108(26):260504, June 2012.
- [78] J. Pablo Bonilla Ataides, David K. Tuckett, Stephen D. Bartlett, Steven T. Flammia, and Benjamin J. Brown. The XZZX surface code. *Nature Communications*, 12(1):2172, April 2021.
- [79] Bence Hetényi and James R. Wootton. Tailoring quantum error correction to spin qubits. *Physical Review A*, 109(3):032433, March 2024.
- [80] Joschka Roffe, David R. White, Simon Burton, and Earl Campbell. Decoding across the quantum low-density parity-check code landscape. *Physical Review Research*, 2(4):043423, December 2020.
- [81] Joschka Roffe. LDPC: Python tools for low density parity check codes, 2022.
- [82] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627(8005):778–782, March 2024.
- [83] Austin G. Fowler, David S. Wang, Charles D. Hill, Thaddeus D. Ladd, Rodney Van Meter, and Lloyd C. L. Hollenberg. Surface Code Quantum Communication. *Physical Review Letters*, 104(18):180503, May 2010.
- [84] Willers Yang and Patrick Rall. Harnessing the Power of Long-Range Entanglement for Clifford Circuit Synthesis. *IEEE Transactions on Quantum Engineering*, 5:1–10, 2024.
- [85] Elisa Bäumer and Stefan Woerner. Measurement-based long-range entangling gates in constant depth. *Physical Review Research*, 7(2):023120, May 2025.

- [86] Rich Rines, Benjamin Hall, Mariesa H. Teo, Joshua Vizslai, Daniel C. Cole, David Mason, Cameron Barker, Matt J. Bedalov, Matt Blakely, Tobias Bothwell, Caitlin Carnahan, Frederic T. Chong, Samuel Y. Eubanks, Brian Fields, Matthew Gillette, Palash Goiporia, Pranav Gokhale, Garrett T. Hickman, Marin Iliev, Eric B. Jones, Ryan A. Jones, Kevin W. Kuper, Stephanie Lee, Martin T. Lichtman, Kevin Loeffler, Nate Mackintosh, Farhad Majdeteimouri, Peter T. Mitchell, Thomas W. Noel, Ely Novakoski, Victory Omole, David Owusu-Antwi, Alexander G. Radnaev, Anthony Reiter, Mark Saffman, Bharath Thotakura, Teague Tomesh, and Ilya Vinogradov. Demonstration of a Logical Architecture Uniting Motion and In-Place Entanglement: Shor’s Algorithm, Constant-Depth CNOT Ladder, and Many-Hypercube Code, September 2025. arXiv:2509.13247 [quant-ph].
- [87] Gushu Li, Yufei Ding, and Yuan Xie. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’19, pages 1001–1014, New York, NY, USA, April 2019. Association for Computing Machinery.
- [88] Steven A. Cuccaro, Thomas G. Draper, Samuel A. Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit, October 2004. arXiv:quant-ph/0410184.
- [89] Jonathan E. Moussa. Transversal Clifford gates on folded surface codes. Physical Review A, 94(4):042316, October 2016.
- [90] Zi-Han Chen, Ming-Cheng Chen, Chao-Yang Lu, and Jian-Wei Pan. Transversal Logical Clifford gates on rotated surface codes with reconfigurable neutral atom arrays, December 2024. arXiv:2412.01391 [quant-ph] version: 1.
- [91] Daniel Litinski. Magic State Distillation: Not as Costly as You Think. Quantum, 3:205, December 2019. arXiv:1905.06903 [quant-ph].
- [92] Cody Jones. Low-overhead constructions for the fault-tolerant Toffoli gate. Physical Review A, 87(2):022328, February 2013.
- [93] Craig Gidney and Austin G. Fowler. Efficient magic state factories with a catalyzed  $|CCZ\rangle$  to  $|T\rangle$  transformation. Quantum, 3:135, April 2019. arXiv:1812.01238 [quant-ph].
- [94] Craig Gidney, Noah Shutty, and Cody Jones. Magic state cultivation: growing T states as cheap as CNOT gates, September 2024. arXiv:2409.17595 [quant-ph].
- [95] Kaavya Sahay, Pei-Kai Tsai, Kathleen Chang, Qile Su, Thomas B. Smith, Shraddha Singh, and Shruti Puri. Fold-transversal surface code cultivation, September 2025. arXiv:2509.05212 [quant-ph].
- [96] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity. Physical Review X, 8(4):041015, October 2018.
- [97] Seth Lloyd. Universal Quantum Simulators. Science, 273(5278):1073–1078, August 1996.
- [98] Craig Gidney. Windowed quantum arithmetic, May 2019. arXiv:1905.07682 [quant-ph].
- [99] Craig Gidney and Austin G. Fowler. Flexible layout of surface code computations using AutoCCZ states, May 2019. arXiv:1905.08916 [quant-ph].
- [100] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. Quantum, 3:163, July 2019. arXiv:1610.06546 [quant-ph].
- [101] Matthew P. Harrigan, Tanuj Khattar, Charles Yuan, Anurudh Peduri, Noureldin Yosri, Fionn D. Malone, Ryan Babbush, and Nicholas C. Rubin. Expressing and Analyzing Quantum Algorithms with Qualtran, September 2024. arXiv:2409.04643 [quant-ph].
- [102] Joonho Lee, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction. PRX Quantum, 2(3):030305, July 2021.
- [103] Matthew B. Hastings and Jeongwan Haah. Dynamically Generated Logical Qubits. Quantum, 5:564, October 2021. arXiv:2107.02194 [quant-ph].
- [104] Noah Berthussen, Shi Jie Samuel Tan, Eric Huang, and Daniel Gottesman. Adaptive Syndrome Extraction. PRX Quantum, 6(3):030307, July 2025.