

Short two-qubit pulse sequences for exchange-only spin qubits in 2D layouts

Jason D. Chadwick,^{1,2,*} Gian Giacomo Guerreschi,¹ Florian Luthi,¹ Mateusz T. Mađzik,¹ Fahd A. Mohiyaddin,¹ Prithviraj Prabhu,¹ Albert T. Schmitz,¹ Andrew Litteken,¹ Shavindra Premaratne,¹ Nathaniel C. Bishop,^{1,†} and James S. Clarke¹

¹*Technology Research, Intel Corporation, Hillsboro, OR 97124*

²*Department of Computer Science, University of Chicago, Chicago, IL 60637*

(Dated: December 20, 2024)

Exchange-only (EO) spin qubits in quantum dots offer an expansive design landscape for architecting scalable device layouts. The study of two-EO-qubit operations, which involve six electrons in six quantum dots, has so far been limited to a small number of the possible configurations, and previous works lack analyses of design considerations and implications for quantum error correction. Using a simple and fast optimization method, we generate complete pulse sequences for CX, CZ, iSWAP, leakage-controlled CX, and leakage-controlled CZ two-qubit gates on 450 unique planar six-dot topologies and analyze differences in sequence length (up to 43% reduction) across topology classes. In addition, we show that relaxing constraints on post-operation spin locations can yield further reductions in sequence length; conversely, constraining these locations in a particular way generates a CXSWAP operation with minimal additional cost over a standard CX. We integrate this pulse library into the Intel quantum stack and experimentally verify pulse sequences on a Tunnel Falls chip for different operations in a linear-connectivity device to confirm that they work as expected. Finally, we explore architectural implications of these results for quantum error correction. Our work guides hardware and software design choices for future implementations of scalable quantum dot architectures.

I. INTRODUCTION

Fault-tolerant quantum computation is expected to require a large-scale quantum computer consisting of up to tens of millions of physical qubits [1–6]. Spin qubits in semiconductor quantum dots have emerged as a promising candidate for a scalable quantum architecture due to the small characteristic size of individual qubits [7], long coherence times, nanosecond operation timescales, operating temperature above one kelvin [8–12], and compatibility with existing fabrication techniques in the well-established semiconductor industry [7, 13–18]. The fidelity of quantum operations on spin qubit platforms has progressed rapidly [12, 19–30], and device sizes of up to 12 have recently been demonstrated in Si/SiGe dots [18, 31, 32].

Exchange-only (EO) spin qubits, in particular, present a compelling platform for quantum computation, as they can be controlled solely through the exchange interaction between electron spins, eliminating the need for precise local magnetic field or microwave control and consequently reducing hardware complexity [33, 34]. Each EO qubit occupies three adjacent quantum dots, and one- and two-qubit operations are performed via a series of exchange interactions between pairs of dots. Recent work has experimentally demonstrated high-fidelity one- and two-EO-qubit operations [32, 35].

The emergence of two-dimensional quantum dot arrays [36, 37] underscores the need for thoughtful layout

choices in large arrays. Key considerations include integration of internal readout components [38], minimizing interconnects and control lines [15, 39–42], and enabling high-speed, high-fidelity qubit operations.

In this work, we focus on the latter, investigating how the connectivity of the relevant dots within a larger array influences the performance of qubit operations. The architecture of quantum dot arrays for EO qubits presents a rich design space with numerous possibilities for arranging and connecting EO qubits. However, exploration of two-EO-qubit operations has been limited to a small subset of potential configurations [43, 44]. In this work, we investigate the interplay between architecture layout and two-qubit gate quality by optimizing pulse sequences directly and using the results to inform the design of layouts that admit better error correction.

We introduce an efficient method for mapping fixed all-to-all pulse sequences to arbitrary restricted dot connectivities, enabling the exploration of a vast array of potential device layouts. This enables us to generate pulse sequences for several two-qubit gates across 450 unique planar six-dot topologies. We find significant differences in pulse sequence length across different dot connectivities, showing that EO qubit connectivity should be considered when designing large dot arrays. We also investigate the impact of relaxing constraints on post-operation spin locations, yielding free additional reductions in sequence length with the help of the compiler. A specific choice of post-operation spin locations leads to a rapid CXSWAP operation that is almost the same cost as a standard CX. We explore the architectural implications of our findings for quantum error correction by comparing surface code performance on different dot layouts.

Our research aims to guide both hardware and soft-

* jchadwick@uchicago.edu

† nathaniel.bishop@intel.com

ware design choices for future implementations of scalable quantum dot architectures. By providing a thorough analysis of the effects of EO qubit connectivity on two-qubit pulse sequences, we contribute to the ongoing effort to realize practical, scalable, and fault-tolerant quantum computers based on semiconductor quantum dots.

II. BACKGROUND

A. Exchange-only spin qubits

Exchange-only spin qubits are a promising platform to scalably encode qubits in silicon quantum dots. Each EO qubit is encoded in a decoherence-free subspace (DFS) of a three-electron-spin system, distributed over three quantum dots [33, 34, 45]. The $|0\rangle$ and $|1\rangle$ states of the EO qubit correspond to the singlet and triplet states of the first two spins in the DFS, while the third spin is known as the *gauge*. We refer to Ref. [45] for a more comprehensive explanation of the structure of the DFS.

Single-EO-qubit operations are performed via exchange interactions between component spins. Each pulse is a partial swap operation $U_{\text{SWAP}}(j, k, \theta) = \cos(\theta/2)I_{jk} + i \sin(\theta/2)\text{SWAP}_{jk}$ between two spins with some angle θ [32]. Note that an exchange pulse with $\theta = \pi$ corresponds to a complete swap (up to global phase) of the two spin states. In hardware, the exchange interaction is performed by lowering a potential barrier between two spins in adjacent quantum dots. Either the barrier voltage or the duration of the operation can be tuned to generate exchange interactions of different angles. The allowed exchange interactions between spins are thus defined by the presence of barriers between adjacent dots. In this work, we assume that the barrier voltage is the tunable knob, so that each exchange pulse is of the same duration regardless of the rotation angle.

In the encoded EO qubit picture, an exchange interaction $U_{12}(\theta)$ between spins 1 and 2 generates a Pauli Z rotation $Z(\theta)$, and interactions $U_{23}(\theta)$ and $U_{13}(\theta)$ generate rotations about two other axes in the X-Z plane of the Bloch sphere [46]. Any two of these three exchange axes yields universal single-qubit control. A two-EO-qubit operation consists of a sequence of pairwise exchange pulses in a six-spin (two-qubit) system [45, 47]; several pulse schedules have previously been found that implement CX and CZ operations on a small number of dot connectivities [43–45]. However, because the set of possible exchange interactions is restricted by the set of inter-dot connections (barriers) present in the hardware, not every pulse sequence can be applied as-is to a given configuration of EO qubits on a device. Existing pulse sequences can be applied to any connected configuration by inserting spin swaps (exchange interactions with angle π) on demand to bring spins near each other; however, this method can lead to extremely long pulse sequences (see Section III B), which is undesirable as exchange errors accumulate linearly with pulse count [46] and magnetic

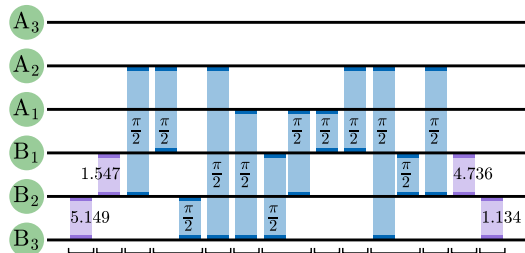


FIG. 1. A 16-pulse sequence implementing a CX gate on two exchange-only qubits encoded in six spins, obtained by adding local corrections to the 12-pulse sequence from [44]. Each dot is labeled according to its qubit (A or B) and spin index, with gauge spins labeled A3 and B3. Each pulse is a partial swap operation between two spins [32], which can be implemented with the exchange interaction [33]. Brackets underneath show groups of sequential pulses which act on disjoint dots, and thus can in principle be applied in parallel. A different choice of local corrections can instead yield CZ; see Appendix .

noise error scales quadratically with sequence duration [32]. In this work, we focus on generating pulse sequences with minimal pulse count with the aim of reducing both sources of error.

B. Previous work on two-qubit gate sequences for EO qubits

Ref. [33] provided the first exchange-only two-qubit pulse sequence construction, consisting of 19 exchange pulses in 13 layers that generates a unitary locally equivalent to CX on a linear dot array (six dots connected in a line). Using a genetic evolution algorithm, Ref. [45] discovered a new pulse sequence (now known as the FW sequence) for a linear dot array that is valid regardless of the state of the gauge qubit. This sequence implements CX exactly and uses 22 pulses in 13 layers. Ref. [43] used a similar method to find CX pulse sequences for two nonlinear connectivities. Ref. [44] used a more efficient constrained search to generate *locally equivalent* (correct up to single-qubit gates) pulse sequences for all-to-all connectivity and several constrained dot connectivities, recovering the FW-CX for the linear connectivity case. Figure 1 shows how the 12-pulse locally equivalent CX from [44] can be extended to exactly implement CX on an all-to-all dot connectivity. Ref. [48] formalized the FW construction and Ref. [49] generalized the construction to give sequences for CPHASE gates of arbitrary angle.

More recently, Ref. [32] demonstrated the first experimental two-EO-qubit gates, performing FW-CX and SWAP operations on six dots in a linear array with fidelities 96.3% and 99.3% respectively. This work also introduced the leakage-controlled CX and CZ (LCCX and LCCZ), which prevent the spread of leakage from

qubit to qubit at the cost of a 65-70% increase in pulse count (and comparable increase in interleaved randomized benchmarking error) relative to the FW-CX and CZ.

In this work, we describe and evaluate an alternative method to obtain pulse sequences for arbitrary dot connectivities, and we present pulse sequences on a much larger set of unique connectivities that are relevant to consider when designing upcoming two-dimensional dot arrays.

C. Quantum error correction on spin qubits

It is generally assumed that physical quantum hardware will never reach the algorithmically relevant error rates of 10^{-6} to 10^{-12} needed for useful applications [2, 5] due to unavoidable noise. To bridge the gap, quantum error correction (QEC) will be used to build a logical abstraction layer upon which high-fidelity programs can be implemented. With sufficiently-low physical error rates, QEC can exponentially suppress errors by operating with *logical qubits* composed of many physical qubits that undergo repeated error detection and correction cycles. [50–52].

The surface code [53, 54] is a leading approach to quantum error correction and is attractive due to its high physical error threshold, planar connectivity requirements, and relative ease of decoding, as well as added benefits of several specialized flavors that have recently been developed to adapt to specific noise channels, basis gates, or connectivities [55, 56]. The fundamental action in quantum error correction is *syndrome extraction*, which primarily consists of layers of parallel two-qubit gates followed by measurement of specific *ancilla* qubits. Logical operations in the surface code consist of different patterns of syndrome extraction on different sets of qubits. It is important to note that, at the physical hardware level, QEC (and, more specifically, repeated syndrome extraction) is the *only* algorithm we expect to run directly on the physical hardware itself. The exponential suppression of errors means that every gain (or loss) in performance at the physical level is exponentially amplified at the logical level, so it is critical to design the physical architecture to provide the best performance when running quantum error correction.

Several works have proposed novel encodings of QEC on spin qubit arrays [57–60], but none have yet considered exchange-only qubits. Here, we explore simple mappings of the standard surface code and the 3-CX surface code [56] to several abstract hardware layouts, focusing on how differences in pulse schedules translate to differences in QEC resource estimates. A more thorough study of QEC tradeoffs on exchange-only qubits is left as future work.

III. METHODOLOGY

A. Enumerating dot connectivities

We consider a broad selection of configurations of EO qubits, which we call *dot topologies*. A dot topology consists of six dots (holding two EO qubits) and a set of allowed exchange interactions, which we represent as edges connecting pairs of dots. In other words, a dot topology is a unique six-node undirected graph. We consider a total of 450 unique topologies in this work (all 6-node connected subgraphs of the square lattice, as well as several selected subgraphs of the triangular lattice), so we condense them into *permutation equivalence classes* for easier visualization. An equivalence class consists of all topologies that are equivalent up to intraqubit spin permutations (changing positions of spins within each EO qubit, but maintaining the same overall qubit-level footprints). Each class thus consists of up to $3!^2 = 36$ topologies. Figure 2a shows all permutation equivalence classes that we consider in this work.

B. Generating pulse sequences

Many methods could be used to generate pulse sequences for arbitrary connectivities. Refs. [45] and [43] used a genetic algorithm to find the FW sequence for a CX on a linear connectivity. In principle, this approach could be applied to find pulse sequences for any connectivity of interest. However, the approach is expensive (Ref. [45] reports a runtime of several weeks), and there is no guarantee that each optimization will converge to a desirable-length solution, which is a concern when considering a large number of different hardware connectivities. Ref. [44] uses a constrained exhaustive search method instead, and successfully finds locally-equivalent pulse sequences for several different dot connectivities. However, any exhaustive method will scale poorly for operations that require longer sequence lengths, such as the leakage-controlled operations from [32]; additionally, the constraints placed on the optimization (time-symmetric sequences with pulse angles only allowed to be multiples of $\pi/2$) restrict the possibilities and require that the optimization can only be done up to local corrections, not allowing the pre- and post-sequence single-qubit corrections to be incorporated into the optimization.

Instead, inspired by the qubit routing problem in quantum compiler research [61, 62], we begin with a pulse sequence designed for all-to-all dot connectivity and aim to find an efficient set of spin-swaps (exchange pulses of angle π) to map this sequence to any restricted connectivity of interest. For example, consider a linear connectivity with spins assigned $A_3 - A_2 - A_1 - B_1 - B_2 - B_3$. Suppose we wish to apply the pulse pair (A_2, B_2) , but these two spins are not directly connected. We can insert the spin swaps (A_2, A_1) and (A_1, B_1) to bring the spin originally in A_2 to the location of B_1 , where it can now interact

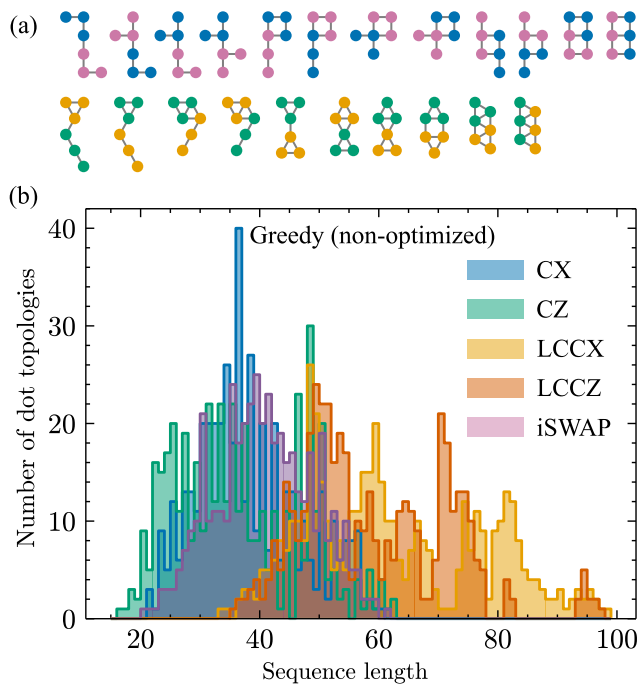


FIG. 2. (a) All permutation classes of dot topologies considered in this work. Dots of the same color indicate that they hold spins of the same EO qubit. Top row contains all topologies that can arise in a 2D rectangular grid of dots. Bottom row consists of additional topologies that require triangular EO qubits. (b) Starting with an all-to-all pulse sequence and greedily inserting spin swaps to adapt it to a restricted dot topologies yields poor results. Each colored histogram shows the distribution of resulting sequence lengths across all 450 dot topologies for a specific two-qubit operation. See Appendix for initial all-to-all pulse sequences.

with B_2 .

Swap-adding is motivated by the observation that every known exchange-only CX sequence fundamentally aligns with the theory of the FW sequence [48], which involves three “quasi-Fredkin” [32] operations on sets of the component spins. We therefore attempt to efficiently map these fundamental operations onto each connectivity rather than solving the problem anew each time.

There are several practical advantages to the swap-adding approach. First, compared to the constrained search method used by Ref. [44], we no longer need to work under a symmetry constraint. Second, our swap-adding optimization runs very quickly; on average, it takes seconds or a few minutes to optimize a pulse sequence for a single topology, and parallelization across different topologies and different gate operations is trivial. This improvement in runtime compared to earlier works makes it far easier to iterate with new optimization goals and could even enable more fine-grained calibration such as generating unique hardware-aware pulse sequences for different quantum dots on a device. Finally, it is easy to find a valid solution for any connectivity, so

we will never be left without any solution; for example, the simplest swap-adding algorithm is to step through the all-to-all sequence and greedily add spin swaps to enable each new pulse. However, the greedy swap-adding approach mentioned previously is highly suboptimal, as we show in Figure 2b; this approach yields sequence lengths of up to 60 for a CX, depending on the topology, while we expect sequence lengths to be 28 at worst (existing linear connectivity pulse sequence). This motivates a more intelligent approach.

1. Formulating swap routing as a shortest path problem

There are two important considerations when determining which spin swaps to insert to map a sequence to a dot topology. First, two exchange pulses that act in succession on the same dot pair can be *merged* into one pulse with rotation angle equal to the sum of the original two; in effect, on a hardware where exchange pulses are fixed duration, the second pulse has become “free”. When incrementally constructing a sequence, we say that a previous exchange pulse is *unblocked* if neither dot has participated in a different exchange pulse since then (meaning that if we were to apply a new pulse on the same pair, we could absorb it into this previous pulse). Second, when two spins need to interact but are not currently in adjacent dots, there are often several choices of swap sequences that will bring them into contact. The choice of swap sequence can either help or hinder later operations in the pulse sequence, depending on where all six spins end after the application of the swaps. A no-lookahead greedy approach will rarely choose the best intermediate swap sequence at every step.

We encode the swap-adding optimization problem as a shortest-path problem on a layered directed acyclic graph, where each path from the source node to the destination node encodes a valid sequence of exchange pulses that realizes the desired gate. We begin by specifying the target dot connectivity (Figure 3a-b) and the *reference sequence*, the all-to-all pulse sequence that implements the desired operation (Figure 3c). Each layer of the optimization graph corresponds to one *reference pulse* from this sequence. Within each layer are many nodes, each representing the application of the reference pulse for a different configuration of spins within the dots (Figure 3d-f). A configuration is only valid for a given layer if the two spins involved in the corresponding reference pulse are in adjacent dots, guaranteeing that the reference pulse can be applied. Each node has outgoing directed edges that lead to some nodes in the next layer, with edge weights related to the number of spin swaps (exchange pulses of angle π) needed to transition from the first node to the second by permuting the spins to the new configuration. Figure 3 shows several nodes of an example optimization graph.

Each node also has information on the current “unblocked exchange pulses” that are available to absorb a

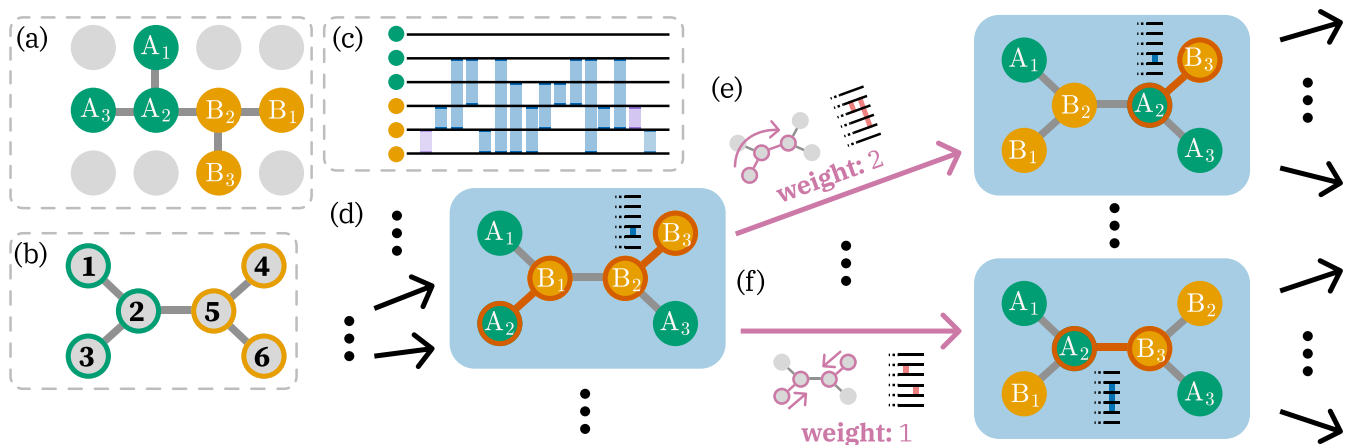


FIG. 3. The optimization graph used to find a pulse sequence for a given dot configuration, described in Section III B 1. (a) Configuration of the two EO qubits of interest within a larger 2D grid of dots, with labeled spins comprising qubits A and B. Allowed exchange interactions are indicated by lines between dots. (b) Connectivity of the selected (numbered) dots, encoding the allowed exchange interactions. (c) Ideal *all-to-all* CX pulse sequence (assuming any exchange interaction is allowed). Not all of these edges are present in the connectivity of interest, so spin swaps must be interleaved to shuffle the spins between dots and bring the desired spins into contact. (d) A node of the optimization graph. Each node corresponds to the application of one of the reference pulses from the all-to-all sequence. At this layer, the pair (B_2, B_3) is applied, so B_2 and B_3 must be in neighboring dots. Note the highlighted dot pairs (A_2, B_1) and (B_2, B_3) in red, corresponding to the most-recently-applied pulses. (e) An edge between nodes represents the insertion of spin-swap operations (exchange interactions). In this case, we perform spin swaps to move spin A_2 from dot 3 to dot 5, where it can interact with B_3 as desired. The weight of this edge is the *total number of new pulses* that will be added to the sequence: the first spin swap on dots (2,3) can be absorbed the previous pulse (A_2, B_1) ; however, the second spin swap and the (A_2, B_3) reference pulse each add one to the sequence length, so the overall edge weight is 2. (f) An edge to a different node in the next layer. In this case, we apply two spin swaps which can both be absorbed into previous pulses. The weight is 1 due to the (A_2, B_3) pulse that is applied in the next node.

potential future pulse. The weight of an edge is the total number of additional pulses required to perform the specified spin swaps *and* the destination reference pulse, after absorbing any possible pulses into previous unblocked pulses.

Finally, there is a single source node that specifies the initial locations of all six spins in the dot topology, and there is a single destination node which does not correspond to a reference pulse but instead enforces that the final locations of all six spins are identical to the initial locations. In this optimization graph, any path from source to destination must contain one and only one node from each layer, and so the sum of edge weights along any path from source to destination is equal to the length of the pulse sequence formed by the pulses on the nodes and edges of the path. We can therefore use a simple shortest-path algorithm to find the shortest sequence. The exact pulses of the sequence can then be constructed by traversing this path and inserting the specified swaps and reference pulses from each edge and node along the way.

Our code is written in Python using the `networkx` package. In the problem instances that we consider, we find that we can exactly solve any single swap-adding optimization in only a few minutes. However, for larger problems, such as a single-step weight-four parity check [63], the search graph could be constructed on demand

instead of ahead of time, and approximate or heuristic methods could be used for pathfinding.

2. Note on optimality

This graph-based search will find the optimal solution to map a fixed all-to-all sequence to a specified dot topology. However, it does not account for the flexibility of changing the order of certain pulses in the all-to-all sequence. For example, the fourth and fifth pulses in the all-to-all CX shown in Figure 1 can be applied in either order, but our current optimizer does not directly account for this. We experimented with changing the ordering of pulses in the all-to-all sequence and did not see significant variation in results. We intend to extend the optimizer to account for this flexibility in the future.

C. Modifying final spin locations

We now discuss two examples of modifications to the search graph that can yield interesting alternative operations. In the implementation described previously, there is a single destination node in the search graph that corresponds to the state where all six spins are located in the dots where they began. This ensures that any valid

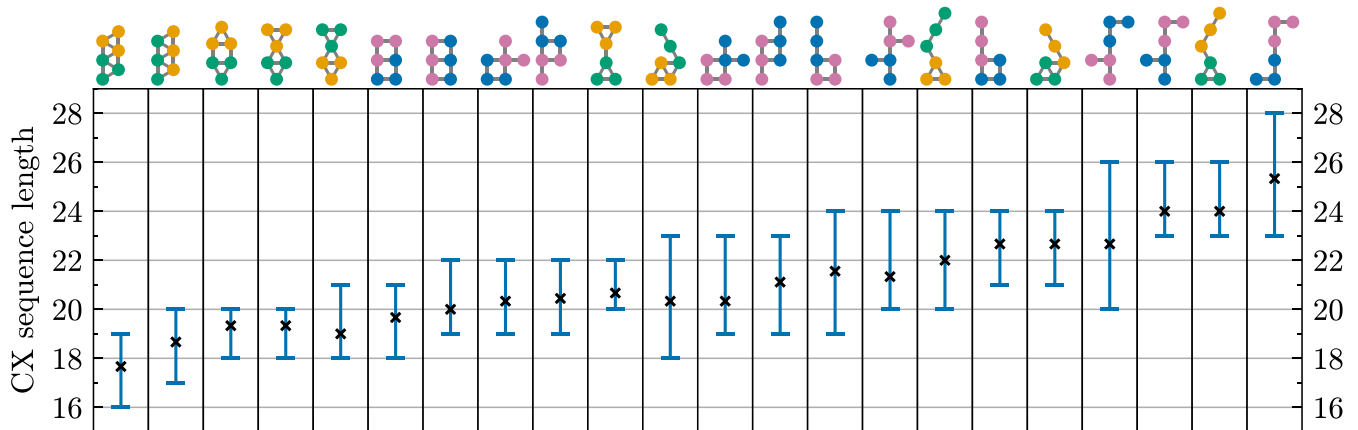


FIG. 4. Optimized CX sequence lengths, grouped by permutation equivalence class. Within each class, blue line spreads from minimum to maximum sequence length and black x denotes mean. For rectangular-grid topologies, dark blue represents the “control” qubit of the CX and magenta the “target”. For non-rectangular-grid topologies, green represents the “control” and gold the “target”.

pulse sequence must return the spins to their initial locations. However, we can change this final destination node or add other possible destination nodes to specify other optimization targets.

1. CXSWAP

Alternatively, we can set the mapping so that spins A_1 and B_1 are exchanged, A_2 and B_2 are exchanged, and A_3 and B_3 are exchanged. This corresponds to a qubit-level SWAP operation in addition to the original reference operation. Using this optimization graph, the resulting pulse sequences will yield CXSWAP operations instead of CX, potentially with shorter pulse sequences than those required to do the distinct operations sequentially, by permuting the qubits *during* the application of the operation. This yields the following two-qubit gate matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

SWAP CX CXSWAP

CXSWAP-like gates are the fourth and final class (up to single-qubit Cliffords) of Clifford two-qubit operations along with Identity-like, SWAP-like, and CX-like gates. Consequently, a fast CXSWAP operation provides interesting opportunities for QEC design, where syndrome extraction circuits are typically made up entirely of Clifford gates. Ref. [56] showed that the surface code can be expressed with CXSWAP (or iSWAP, which is locally equivalent) as the two-qubit basis gate rather than CX, suggesting that these gates may be useful for other QEC circuits. For example, by replacing some CXs with

CXSWAPs, a 4+1 flag fault-tolerant code [64] could be adapted to degree-4 connectivity, or the stepping surface code [56] could be performed in place (maintaining its advantageous leakage reduction properties while eliminating the extra qubit overhead typically needed for this circuit). A fast CXSWAP in place of a CX may also prove useful when adapting QEC to sparsely connected qubit arrays that may be necessary in spin qubits due to the relatively large size of current state-of-the-art readout components [38].

2. Relaxing spin location constraints

Alternatively, instead of modifying the default spin location constraint in the final layer, we can relax the constraint by allowing *several* possible spin permutations in the final layer. In the search graph, we add several additional destination nodes, each of which corresponds to a different final mapping of spins. We can then search for the shortest path from the source to *any* destination node (which is not significantly harder than the original shortest path problem). We find that this relaxation can lead to sequences that are several pulses shorter.

We specifically focus on the case where we allow any *intraqubit* spin permutations such that the qubit-level footprints remain the same but spins 1, 2, and 3 can be shuffled within each qubit. Each optimized pulse sequence must choose a specific spin permutation to apply. When compiling a quantum program to the device, the compiler can simply track the known repositioning of spins as these operations are successively applied, and pick the correct next pulse sequence given the prior spin locations.

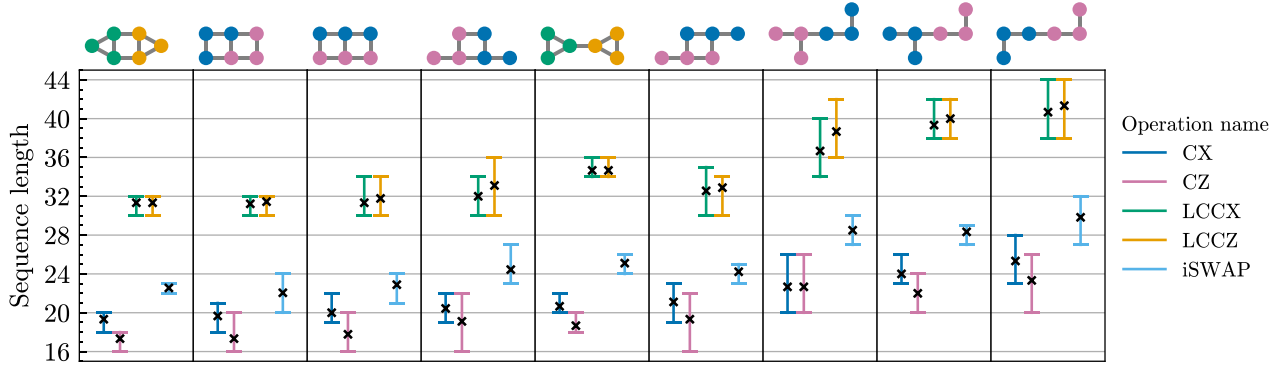


FIG. 5. Optimized sequence lengths for five different two-qubit operations, for a selected set of permutation equivalence classes.

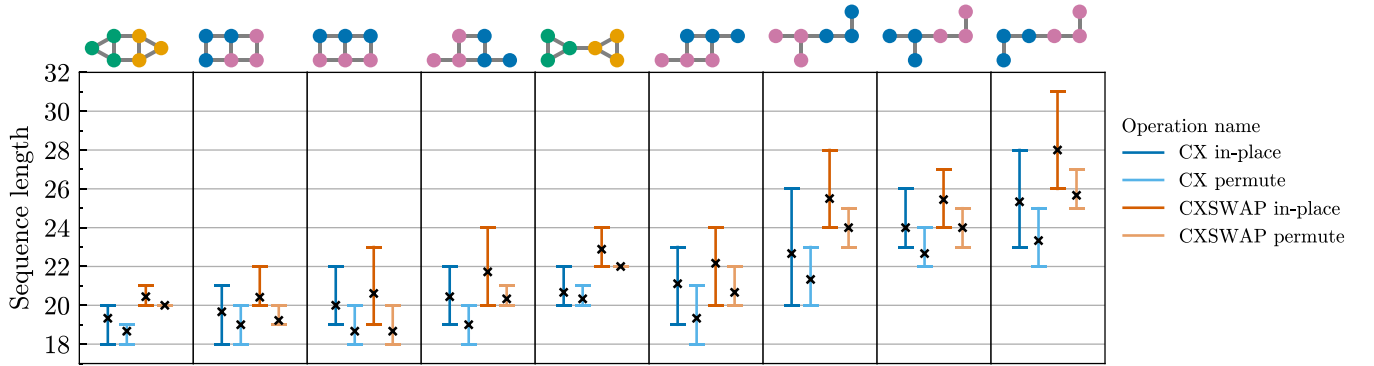


FIG. 6. CX and CXSWAP sequence lengths for selected permutation equivalence classes, with and without intraqubit spin permutations allowed. Across all topologies, CXSWAP is 22.2% shorter than applying CX and SWAP operations sequentially and is only 7.34% longer on average than CX. For both CX and CXSWAP, allowing permutations reduces the average and maximum sequence length within each equivalence class.

IV. OPTIMIZED SEQUENCE RESULTS

We run our optimizations over the full collection of dot topologies and present the results aggregated within each permutation equivalence class.

A. Standard operation sequence lengths for different topology classes

Figure 4 shows the optimized sequence lengths for a CX operation on all 450 dot topologies in consideration. The results are organized by permutation equivalence class (see Section III A). Observe that the linear dot topology class (rightmost) yields the longest pulse sequences. Generally, as the dots become more connected, shorter CX sequences become possible. For example, the “linear-parallel” permutation class (seventh from left) has a maximum-in-class length of 22 compared to 28 for the fully linear class, corresponding to a reduction of 21.4%. For the furthest left topology class consisting of two densely connected triangular qubits, maximum-in-class sequence length is reduced to 19, a 32% reduction.

Figure 5 shows optimized sequence lengths for an extended set of gates (CX, CZ, iSWAP, LCCX, and LCCZ) over a selected subset of relevant topology classes (the connectivities that we believe to be the most promising when considering integration into a larger array). Across all five two-qubit operations, a similar trend emerges: denser dot connectivities yield faster two-qubit operations. LCCX and LCCZ, the “leakage-controlled” versions of CX and CZ that avoid the spread of leakage from one qubit to another [32], are constructed by applying two CX-like sequences with a single-qubit operation in between; thus, they are significantly longer. On average, our optimized leakage-controlled variants of CX and CZ are 60-70% longer than the standard CX and CZ.

These results show the importance of considering dot connectivity when designing a large quantum dot array for exchange-only qubits. Significant improvements in two-qubit operation fidelity can be achieved simply by changing the way two EO qubits are connected. In addition, if SWAP operations are expected to be necessary during computation, consider that a “linear-parallel” pair of qubits can be fully swapped with 3 exchange pulses in a single maximally-parallel step, while a fully linear pair

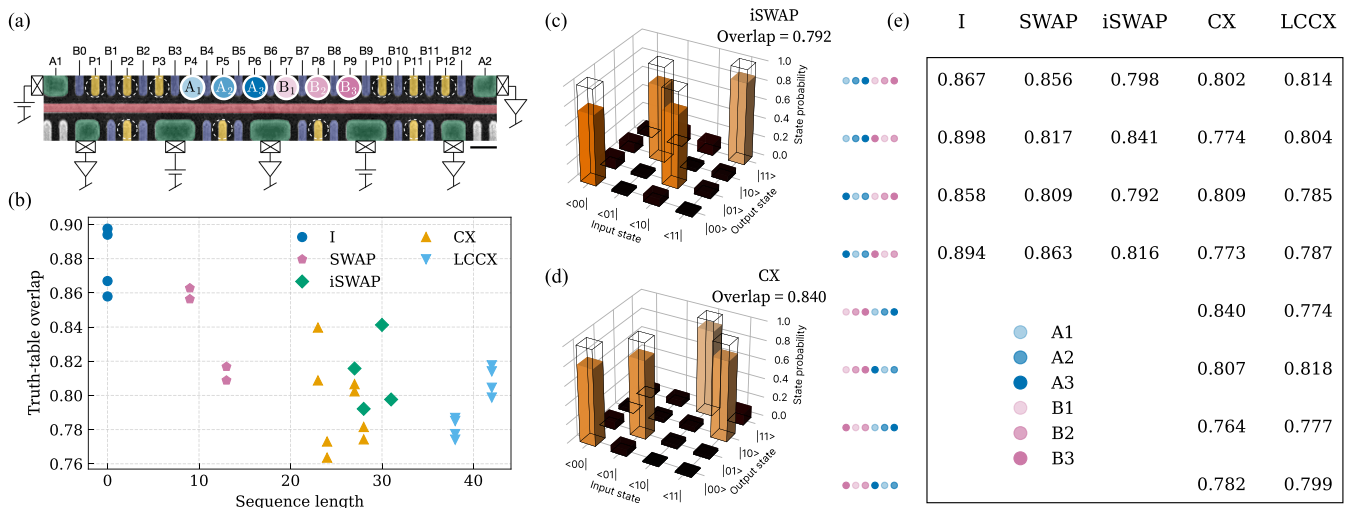


FIG. 7. Experimental truth-table validation of pulse sequences. (a) 12 quantum dot Tunnel Falls device. Dots P4,P5,P6 and P7,P8,P9 are used for the two EO qubits. (b) Summarized results comparing sequence length and truth-table overlap. (c-d) Example truth tables for iSWAP and CX two-qubit gates for specific spin configurations. (e) Truth-table overlap results for the five gate types and eight spin configurations, without SPAM corrections. CX-type gates are performed with qubit A as control and qubit B as target.

of qubits will require 9 exchange pulses in 6 steps.

Finally, we note that this particular optimization did not find a 22-pulse CX for linearly connected EO qubits, although such a sequence is known to exist [45]. This is due to the choice of the initial all-to-all sequence (i.e. the choice of the local corrections at the start and end which are needed to generate the full CX); choosing a sequence equivalent to that from [45] would indeed result in our optimizer finding a 22-pulse sequence for that connectivity. For simplicity, the results we present here are all obtained from a single all-to-all CX sequence (Figure 1), but for a more heavily optimized pulse library, our optimizer could be run with many variations of initial sequences and the best result for each topology could be kept, which is still feasible due to the fast runtime of our optimizer.

B. Effects of spin permutation constraints

Figure 6 shows optimized sequence lengths for variants of CX with different final spin location constraints (explained in Section III C). Across all 450 dot topologies, allowing intraqubit spin permutations (“CX-permute”) yields a 0-13% (mean 5.7%) reduction in sequence length. Notably, the worst linear topology CX can be performed in 25 pulses instead of 28. We emphasize that there is no penalty to replacing CX with CX-permute; the compiler simply needs to track the applied permutations and account for them in future operations.

Additionally, observe that the CXSWAP operation (which is a CX sequence with final spin location constraints that enforce a qubit SWAP) is similar in length

to the CX itself. Across all topologies, CXSWAP is between 8.7% *shorter* and 27% longer (mean 7.34% longer) compared to a standard CX. Compared to a standard CX and a standard SWAP applied sequentially, the optimized CXSWAP is 4.8-35.7% shorter (mean 22.2%). When intraqubit permutations are allowed, the CXSWAP-permute is an average of 7.2% longer than the CX-permute and 23.5% shorter than CX-permute and SWAP in series. With these optimized sequences, a SWAP operation can be applied for very low cost whenever a CX (or any other two-qubit operation) is being applied to a pair of qubits, which may be beneficial for certain QEC applications, as discussed in Section III C 1.

V. EXPERIMENTAL VALIDATION

The pulse sequences presented in the previous section have been compiled into a library available at [65]. This library has been integrated into the Intel Quantum Compiler and the Intel quantum hardware stack, allowing the pulse sequences to be queried on demand based on the target dot connectivity.

This allows us to experimentally demonstrate the described pulse sequences for the CX, LCCX, iSWAP and SWAP gates. We use an Intel-fabricated Tunnel Falls device [17, 18] to bring a linear array of six quantum dots into the (1,3,1,1,3,1) electron regime, and tune the exchange couplings between the quantum dots so that we can encode two exchange-only qubits [32, 46]. The Pauli spin blockade (PSB) technique [7] is used to perform the spin-to-charge conversion, allowing both qubits to be read out using a sensing quantum dot. With PSB

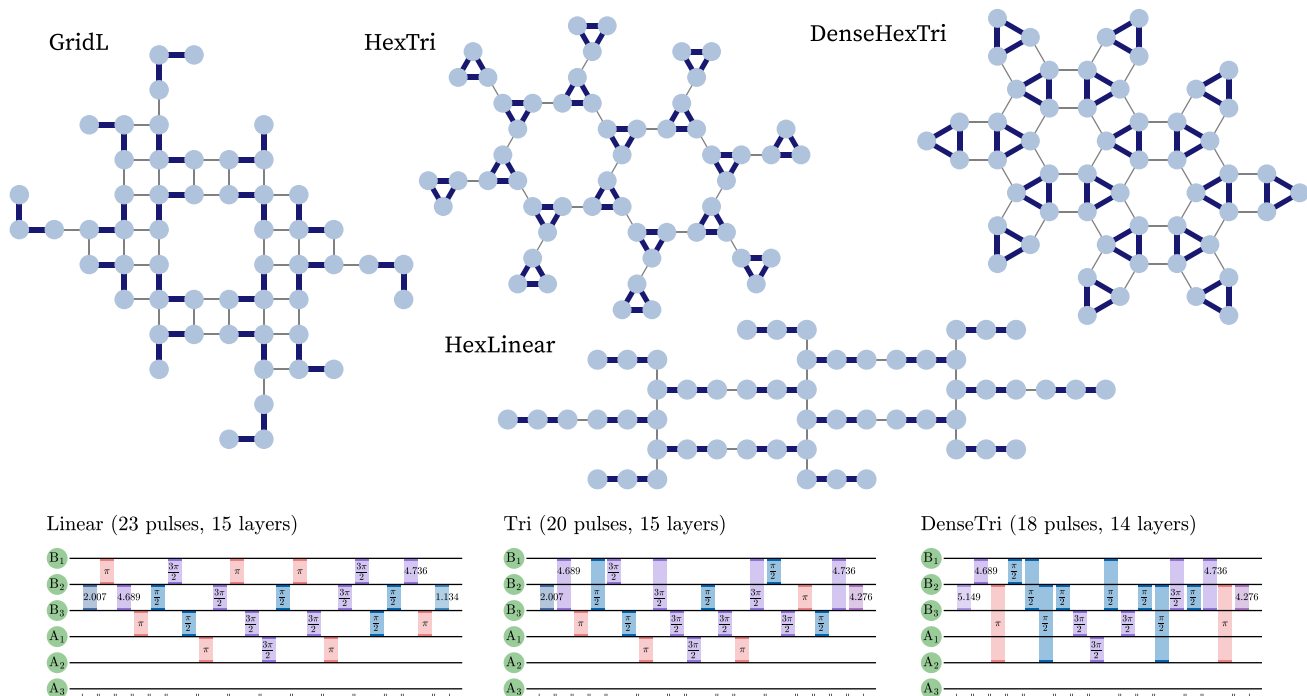


FIG. 8. *Top*: The four dot layouts considered in this study. Names denote the qubit-level connectivity (Grid or Hex) and the dot-level connectivity (Linear, L, Tri). Thick dark blue lines indicate intraqubit connections and thin gray lines indicate inter-qubit connections. *Bottom*: Example CX pulse sequences for three different two-EO-qubit connectivities. More densely-connected qubits yield shorter pulse counts (23 for Linear vs. 18 for DenseTri); however, note that the number of fully-parallel layers (14-15) remains similar.

and post-selection, we avoid initialization into leakage states and can reliably initialize the system into its $|00\rangle$ state.

Figure 7 summarizes the experiments used to verify the gate action of the two-qubit gates of interest. Depending on the positions of the gauge dots in the two qubits, different pulse sequences are required to perform the desired gate action. In the linear dot connectivity, we query pulse sequences for eight different gauge spin configurations for the CX and LCCX gates and four different configurations for the SWAP and iSWAP gates (as they act symmetrically on the two qubits). The EO qubits are initialized in a fixed configuration; to efficiently access the different configurations, the spin states of the electrons on the respective qubits are selectively swapped by applying a series of three π -pulses after initialization. The gate action characterization is performed using truth-table measurements in which the computational basis states are prepared, the gate under question is applied, and the resulting two-qubit state is measured in the computational basis [28, 66, 67]. We calculate the *truth-table overlap* of the measured classical probabilities P_{meas} with the probabilities of the ideal gate action P_{ideal} over the basis states as $\text{Tr}(P_{\text{meas}}P_{\text{ideal}}^{-1})/2^{n_{\text{qubits}}}$. State preparation and measurement (SPAM) error contributions can be characterized using the above scheme with an identity gate on both qubits, which yields an average

truth-table overlap of 87.9% for the four different configurations. Without SPAM corrections, we characterize the classical fidelities averaged over the respective configurations for the SWAP, iSWAP, CX, and LCCX to be 83.6%, 81.2%, 79.4%, and 79.5% respectively. These results confirm that the pulse sequences work as expected; a full characterization of the gate action using quantum process tomography or similar techniques is beyond the scope of this work.

Figure 7b shows that truth-table overlap tends to decrease as pulse sequence length increases; however, the LCCX appears to exhibit some degree of noise resilience compared to the standard CX, achieving similar error rates despite much longer sequence length, as observed in Ref. [32].

VI. CASE STUDY: ERROR CORRECTION ON DIFFERENT HARDWARE LAYOUTS

While we have shown that changing EO qubit connectivity can significantly reduce pulse count for two-qubit gates, it remains unclear how this will translate to system-level, error-corrected device performance. In this section, we present several different abstract hardware layouts with different EO qubit topologies and evaluate the number of qubits needed to reach the teraquop regime

(10^{-12} logical error rate) in different noise regimes.

A. Evaluated dot layouts

Figure 8 shows the four layouts that we consider in this study. Layout name denotes the qubit-level connectivity (Grid or Hex) and the dot-level connectivity (L, Tri, Linear). Depending on the qubit connectivity, we map a section of the bulk of either the 4CX (grid) or 3CX (hex) surface code [56] to the layout. We then specify the schedule of two-qubit gates that corresponds to one round of syndrome extraction and compile the corresponding pulse-level schedule consisting of pairwise exchange pulses. Each schedule consists of four layers of CX gates. We allow post-operation intraqubit spin permutations to reduce pulse count, tracking the positions of spins during the schedule to apply the correct pulse sequence at each step. Figure 9 shows the average number of pulses per CX for a QEC cycle on each layout. As expected, the linear-connectivity layout has the highest average length, and the dense triangular connectivity yields the lowest average length.

B. Parallelism restrictions

Current EO experiments [32] have used fully-sequential pulse sequences, meaning that the duration of a sequence is directly determined by its pulse count. However, some degree of parallelism across a device will be necessary for large-scale quantum computation. In a two-dimensional quantum dot array, considerations such as classical control signal crosstalk [32] or gate virtualization [68–71] may limit the parallelism of exchange pulses. Pulse parallelism is important to reduce the gate duration t_{gate} ; since T_2^* errors scale with $(t_{\text{gate}}/T_2^*)^2$, reducing the gate duration can significantly improve performance. We consider three conditions: *Full parallelism*, where each pulse in the schedule is performed as early as possible; *N-restricted*, where two neighboring quantum dots cannot both perform exchange interactions with other dots at the same time; and *NN-restricted*, where this restriction extends to next-nearest neighbors. We compile the pulse schedule for all CX gates in a full QEC round, scheduling each individual pulse as early as possible within the parallelism restrictions.

Figure 10 shows the total durations of the compiled QEC cycle schedules for each layout under the three parallelism conditions. Interestingly, with full parallelism, all layouts have almost identical durations, so we would expect them to all have similar levels T_2^* -induced errors. However, with parallelism restrictions, the HexTri layout becomes significantly shorter than the others. The DenseHexTri layout, which has the shortest pulses-per-CX count, performs relatively poorly in restricted parallelism settings, implying that denser connectivities may suffer more from neighbor-based parallelism restrictions.

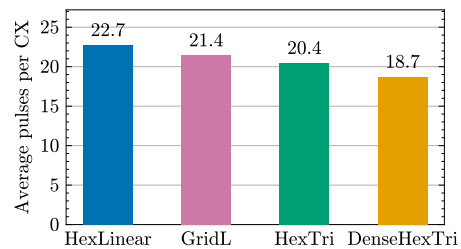


FIG. 9. Average pulse count per CX for QEC round on each layout. As expected, more densely-connected layouts yield shorter pulse sequences.

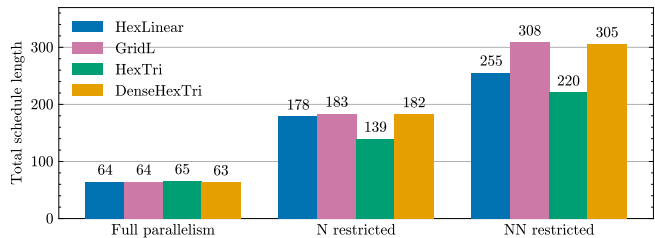


FIG. 10. Schedule length for one full QEC cycle of CX gates, under various parallelism restrictions. Here, “length” refers to the duration of the schedule in terms of the number of steps, where many pulses can be applied in parallel during each step.

From these full-QEC-cycle pulse schedules, we then recover the schedule for each individual two-qubit gate, preserving the parallelism or serialization of pulses and including any idle time induced by parallelism restrictions, to simulate individually under noise.

C. Noisy simulation

We consider noise characterized by T_2^* , the ergodic dephasing time of an individual spin, and by Δ^J , the unitless exchange error. For fixed values of these parameters, we simulate evolution under the time-varying Hamiltonian

$$\begin{aligned}
 H(t) = & \left. \epsilon^z \sum_i S_i^z + \sum_{ij} J_{ij}(t) \mathbf{S}_i \cdot \mathbf{S}_j \right\} H_{\text{ideal}} \\
 & + \left. \sum_i \delta_i^{\epsilon^z} S_i^z + \sum_{ij} \delta_{ij}^J J_{ij}(t) \mathbf{S}_i \cdot \mathbf{S}_j \right\} H_{\text{noise}},
 \end{aligned} \tag{2}$$

where ϵ^z is the constant Zeeman term (assuming no magnetic field gradients or g-factor variations between the spins), \mathbf{S}_i (S_i^z) is the spin operator (z component) for spin i , and $J_{ij}(t)$ is the ideal schedule of exchange pulses between spins i and j . Zeeman noise $\delta_i^{\epsilon^z}$ is sampled from the normal distribution $\mathcal{N}(0, \sigma = \sqrt{2}\hbar/T_2^*)$, and unitless exchange error δ_{ij}^J is sampled from $\mathcal{N}(0, \sigma = \Delta^J)$ for each quasi-static simulation shot. We consider the high mag-

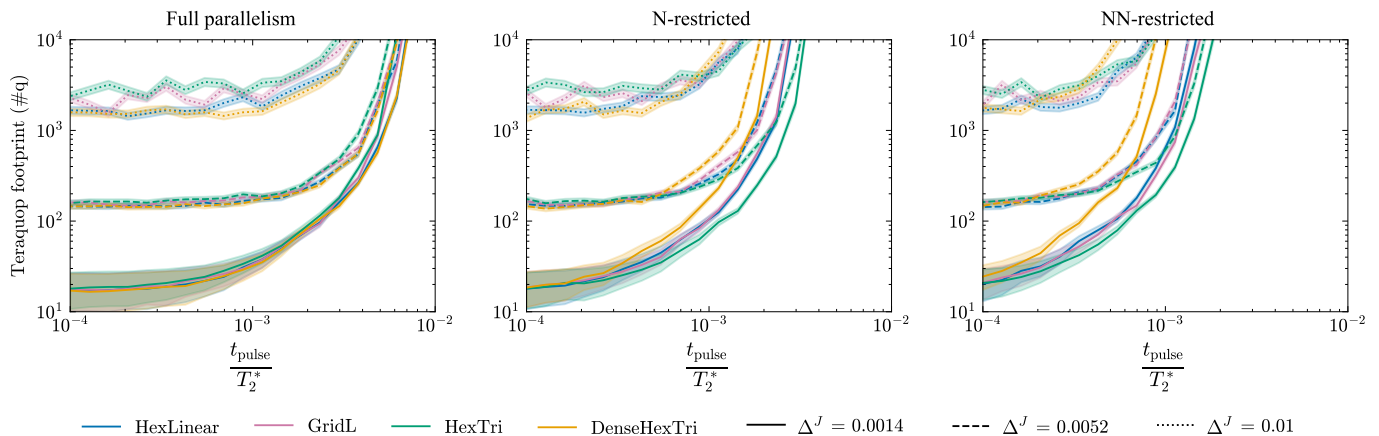


FIG. 11. Teraquop footprints for all four layouts, under different parallelism conditions and in different noise regimes. The ratio $\frac{t_{\text{pulse}}}{T_2^*}$ varies along the x axis and each plot contains three groups of lines corresponding to different Δ^J strengths.

netic field (\sim mT) limit such that the impact of $\delta_i^{\varepsilon^{x,y}}$ is negligible compared to $\delta_i^{\varepsilon^z}$.

For each gate two-qubit gate, we simulate the pulse schedule as a unitary evolution of the 6 electron spins, and then compute its effect on the qubit subspace. The two-EO-qubit gate infidelity is calculated as the sum of leakage and unitary error within the qubit subspace. We calculate the average fidelity over 20 quasistatic instances, which we then use in the Stim simulations of the surface code described in the next section.

D. Teraquop footprint calculation

We adapt the Stim simulation code from [72] to calculate the teraquop footprints. After each two-qubit gate in the Stim circuit, we add two-qubit depolarizing noise with probability corresponding to the simulated fidelity of that particular gate. For several choices of code distance, we perform X- and Z-basis logical memory experiments and combine the X and Z error rates to obtain the overall logical failure rate per QEC round. We then project the code distance that would be needed to reach a logical failure rate of 10^{-12} and calculate the corresponding number of physical qubits needed to construct one logical qubit (the *teraquop footprint*). This footprint is a function of the simulated two-qubit gate error rates from the previous section.

The results are shown in Figure 11. The three plots show data for the three different parallelism restrictions in different noise regimes. We sweep t_{pulse}/T_2^* on the x axis for three fixed values for Δ^J . Due to the many simplifying assumptions used in this analysis, we do not intend these results to be exact resource estimates, but rather we wish to study the extent to which the different pulse schedules in the same noise regime can affect the quantum error correction overhead.

For the full parallelism case, we see little difference between hardware layouts, except in the regime of very

strong Δ^J noise. However, with parallelism restrictions, differences begin to emerge. For NN-restricted parallelism, with $\Delta^J \approx 0.5\%$ and $t_{\text{pulse}}/T_2^* \approx 0.1\%$, HexTri (green) has a teraquop footprint of around 300 qubits, while DenseHexTri's (yellow) footprint exceeds 10,000 qubits to create a logical qubit of equivalent error rate. HexLinear and GridL would require about 1,000 qubits each.

Overall, these results show that program-level performance cannot be predicted simply by looking at individual pulse sequences for linear or triangular connectivities. It is critical to evaluate pulse sequences in the context in which they will be performed in real applications to draw meaningful conclusions.

VII. DISCUSSION

In this work, we provided a complete set of pulse sequences to implement two-qubit gates on exchange-only spin qubits in hundreds of unique connectivities using an efficient optimization method. Compared to linear connectivity, we found a reduction of up to 42.8% in sequence lengths across all connectivities, which could yield considerable improvements in gate fidelity if taken into account when designing hardware. We introduced the new abstraction of gates up-to-spin-permutations, achieving a free reduction in average sequence length when the compiler can track spin locations.

However, as we saw in the QEC case study, the length of individual pulse sequences does not tell the whole story. Device-level parallelism restrictions can dramatically change the tradeoff between different connectivities, and relatively minor differences in pulse sequences can translate to critical differences in resource cost. The exponential error suppression of QEC greatly rewards any minor improvement in physical performance and punishes any inefficiency. Pulse sequences are only one part of the operation of a quantum computer but may be a

key part of the make-or-break situation typical of QEC.

We envision several areas of future work to further explore pulse-aware layout design and to improve pulse sequences themselves. Our QEC study involved four hand-picked layouts, but a more systematic automated search of the quantum dot array design space is needed, paying more attention to potential restrictions arising from large internal readout components and fabrication constraints. Additionally, incorporating gate fidelity directly into the optimization, rather than relying on pulse count as a proxy, may lead to pulse sequences with better on-chip performance.

This work is the first step towards pulse-informed, QEC-informed hardware design for EO systems. The design space for quantum dot arrays is extensive, giving the opportunity to tailor a layout and schedule to the specific QEC scheme and hardware constraints at hand. We believe that similar co-design considerations will be critical to enable highly scalable, highly performant quantum computing on exchange-only qubits in silicon.

DATA AVAILABILITY

The complete library of pulse sequences is publicly available at [65].

AUTHOR CONTRIBUTIONS

JDC generated the pulse library, performed simulations, and compiled the manuscript. GGG developed preliminary pulse sequence simulation code and assisted with the theory of EO qubit gates. JDC, FL, and MTM integrated the pulse library into the device control software. FL and MTM performed device experiments. FAM assisted with understanding noise in silicon spin qubits. PP and JSC suggested dot topologies to consider. PP suggested the CXSWAP operation and envisioned uses for the CXSWAP. JDC, AL, and SP integrated the pulse library into the Intel Quantum Compiler. ATS, GGG, NCB, and JSC coordinated the project and provided guidance on direction. All authors revised the manuscript.

-
- [1] D. Litinski, *Quantum* **3**, 128 (2019), [arXiv:1808.02892 \[cond-mat, physics:quant-ph\]](#).
- [2] C. Gidney and M. Ekerå, *Quantum* **5**, 433 (2021).
- [3] C. Chamberland and E. T. Campbell, *PRX Quantum* **3**, 010331 (2022).
- [4] I. H. Kim, Y.-H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, *Physical Review Research* **4**, 023019 (2022).
- [5] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoefler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vaschillo, Assessing requirements to scale to practical quantum advantage (2022), [arXiv:2211.07629 \[quant-ph\]](#).
- [6] T. Leblond, C. Dean, G. Watkins, and R. Bennink, *ACM Transactions on Quantum Computing* **5**, 1 (2024).
- [7] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, *Reviews of Modern Physics* **95**, 025003 (2023).
- [8] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, and M. Veldhorst, *npj Quantum Information* **3**, 1 (2017).
- [9] L. Petit, H. G. J. Eenink, M. Russ, W. I. L. Lawrie, N. W. Hendrickx, S. G. J. Philips, J. S. Clarke, L. M. K. Vandersypen, and M. Veldhorst, *Nature* **580**, 355 (2020).
- [10] C. H. Yang, R. C. C. Leon, J. C. C. Hwang, A. Saraiva, T. Tanttu, W. Huang, J. Camirand Lemyre, K. W. Chan, K. Y. Tan, F. E. Hudson, K. M. Itoh, A. Morello, M. Pioro-Ladrière, A. Laucht, and A. S. Dzurak, *Nature* **580**, 350 (2020).
- [11] L. C. Camenzind, S. Geyer, A. Fuhrer, R. J. Warburton, D. M. Zumbühl, and A. V. Kuhlmann, *Nature Electronics* **5**, 178 (2022).
- [12] J. Y. Huang, R. Y. Su, W. H. Lim, M. Feng, B. van Straaten, B. Severin, W. Gilbert, N. Dumoulin Stuyck, T. Tanttu, S. Serrano, J. D. Cifuentes, I. Hansen, A. E. Seedhouse, E. Vahapoglu, R. C. C. Leon, N. V. Abrosimov, H.-J. Pohl, M. L. W. Thewalt, F. E. Hudson, C. C. Escott, N. Ares, S. D. Bartlett, A. Morello, A. Saraiva, A. Laucht, A. S. Dzurak, and C. H. Yang, *Nature* **627**, 772 (2024).
- [13] D. Loss and D. P. DiVincenzo, *Physical Review A* **57**, 120 (1998).
- [14] 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, *Nature Communications* **7**, 13575 (2016).
- [15] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak, *Nature Communications* **8**, 1766 (2017).
- [16] A. M. J. Zwerver, T. Krähenmann, T. F. Watson, L. Lampert, H. C. George, R. Pillarisetty, S. A. Bojarski, P. Amin, S. V. Amitonov, J. M. Boter, R. Caudillo, D. Correas-Serrano, J. P. Dehollain, G. Droulers, E. M. Henry, R. Kotlyar, M. Lodari, F. Lüthi, D. J. Michalak, B. K. Mueller, S. Neyens, J. Roberts, N. Samkharadze, G. Zheng, O. K. Zietz, G. Scappucci, M. Veldhorst, L. M. K. Vandersypen, and J. S. Clarke, *Nature Electronics* **5**, 184 (2022).
- [17] S. Neyens, O. K. Zietz, T. F. Watson, F. Luthi, A. Nethewala, H. C. George, E. Henry, M. Islam, A. J. Wagner, F. Borjans, E. J. Connors, J. Corrigan, M. J. Curry, D. Keith, R. Kotlyar, L. F. Lampert, M. T. Mađzik, K. Millard, F. A. Mohiyaddin, S. Pellerano, R. Pil-

- larisetty, M. Ramsey, R. Savytsky, S. Schaal, G. Zheng, J. Ziegler, N. C. Bishop, S. Bojarski, J. Roberts, and J. S. Clarke, *Nature* **629**, 80 (2024).
- [18] H. C. George, M. T. Mađzik, E. M. Henry, A. J. Wagner, M. M. Islam, F. Borjans, E. J. Connors, J. Corrigan, M. Curry, M. K. Harper, D. Keith, L. Lampert, F. Luthi, F. A. Mohiyaddin, S. Murcia, R. Nair, R. Nahm, A. Nethewala, S. Neyens, R. D. Raharjo, C. Rogan, R. Savytsky, T. F. Watson, J. Ziegler, O. K. Zietz, R. Pillarisetty, N. C. Bishop, S. A. Bojarski, J. Roberts, and J. S. Clarke, 12-spin-qubit arrays fabricated on a 300 mm semiconductor manufacturing line (2024), [arXiv:2410.16583 \[cond-mat\]](https://arxiv.org/abs/2410.16583).
- [19] 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, *Nature Nanotechnology* **9**, 981 (2014).
- [20] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi, N. Usami, K. M. Itoh, and S. Tarucha, *Nature Nanotechnology* **13**, 102 (2018).
- [21] A. West, B. Hensen, A. Jouan, T. Tantt, C.-H. Yang, A. Rossi, M. F. Gonzalez-Zalba, F. Hudson, A. Morello, D. J. Reilly, and A. S. Dzurak, *Nature Nanotechnology* **14**, 437 (2019).
- [22] M. Urdampilleta, D. J. Niegemann, E. Chanrion, B. Jadot, C. Spence, P.-A. Mortemousque, C. Bäuerle, L. Hutin, B. Bertrand, S. Barraud, R. Maurand, M. Sanquer, X. Jehl, S. De Franceschi, M. Vinet, and T. Meunier, *Nature Nanotechnology* **14**, 737 (2019).
- [23] G. Zheng, N. Samkharadze, M. L. Noordam, N. Kalhor, D. Brousse, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, *Nature Nanotechnology* **14**, 742 (2019).
- [24] A. Noiri, K. Takeda, T. Nakajima, T. Kobayashi, A. Sammak, G. Scappucci, and S. Tarucha, *Nature* **601**, 338 (2022).
- [25] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, *Nature* **601**, 343 (2022).
- [26] A. R. Mills, C. R. Guinn, M. J. Gullans, A. J. Sigillito, M. M. Feldman, E. Nielsen, and J. R. Petta, *Science Advances* **8**, eabn5130 (2022).
- [27] J. Z. Blumoff, A. S. Pan, T. E. Keating, R. W. Andrews, D. W. Barnes, T. L. Brecht, E. T. Croke, L. E. Euliss, J. A. Fast, C. A. Jackson, A. M. Jones, J. Kerckhoff, R. K. Lanza, K. Raach, B. J. Thomas, R. Velunta, A. J. Weinstein, T. D. Ladd, K. Eng, M. G. Borselli, A. T. Hunter, and M. T. Rakher, *PRX Quantum* **3**, 010352 (2022).
- [28] M. T. Mađzik, S. Asaad, A. Youssry, B. Joecker, K. M. Rudinger, E. Nielsen, K. C. Young, T. J. Proctor, A. D. Baczewski, A. Laucht, V. Schmitt, F. E. Hudson, K. M. Itoh, A. M. Jakob, B. C. Johnson, D. N. Jamieson, A. S. Dzurak, C. Ferrie, R. Blume-Kohout, and A. Morello, *Nature* **601**, 348 (2022).
- [29] P. Steinacker, N. D. Stuyck, W. H. Lim, T. Tantt, M. Feng, A. Nick, S. Serrano, M. Candido, J. D. Cifuentes, F. E. Hudson, K. W. Chan, S. Kubicek, J. Jusot, Y. Canvel, S. Beyne, Y. Shimura, R. Loo, C. Godfrin, B. Raes, S. Baudot, D. Wan, A. Laucht, C. H. Yang, A. Saraiva, C. C. Escott, K. D. Greve, and A. S. Dzurak, *A 300 mm foundry silicon spin qubit unit cell exceeding 99% fidelity in all operations* (2024), [arXiv:2410.15590](https://arxiv.org/abs/2410.15590).
- [30] T. Tantt, W. H. Lim, J. Y. Huang, N. Dumoulin Stuyck, W. Gilbert, R. Y. Su, M. Feng, J. D. Cifuentes, A. E. Seedhouse, S. K. Seritan, C. I. Ostrove, K. M. Rudinger, R. C. C. Leon, W. Huang, C. C. Escott, K. M. Itoh, N. V. Abrosimov, H.-J. Pohl, M. L. W. Thewalt, F. E. Hudson, R. Blume-Kohout, S. D. Bartlett, A. Morello, A. Laucht, C. H. Yang, A. Saraiva, and A. S. Dzurak, *Nature Physics* , 1 (2024).
- [31] S. G. J. Philips, M. T. Mađzik, S. V. Amitonov, S. L. de Snoo, M. Russ, N. Kalhor, C. Volk, W. I. L. Lawrie, D. Brousse, L. Tryputen, B. P. Wuetz, A. Sammak, M. Veldhorst, G. Scappucci, and L. M. K. Vandersypen, *Nature* **609**, 919 (2022).
- [32] A. J. Weinstein, M. D. Reed, A. M. Jones, R. W. Andrews, D. Barnes, J. Z. Blumoff, L. E. Euliss, K. Eng, B. H. Fong, S. D. Ha, D. R. Hulbert, C. A. C. Jackson, M. Jura, T. E. Keating, J. Kerckhoff, A. A. Kiselev, J. Matten, G. Sabbir, A. Smith, J. Wright, M. T. Rakher, T. D. Ladd, and M. G. Borselli, *Nature* **615**, 817 (2023).
- [33] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley, *Nature* **408**, 339 (2000).
- [34] D. Bacon, J. Kempe, D. A. Lidar, and K. B. Whaley, *Physical Review Letters* **85**, 1758 (2000).
- [35] B. Sun, T. Brecht, B. H. Fong, M. Akmal, J. Z. Blumoff, T. A. Cain, F. W. Carter, D. H. Finestone, M. N. Fireman, W. Ha, A. T. Hatke, R. M. Hickey, C. A. C. Jackson, I. Jenkins, A. M. Jones, A. Pan, D. R. Ward, A. J. Weinstein, S. J. Whiteley, P. Williams, M. G. Borselli, M. T. Rakher, and T. D. Ladd, *PRX Quantum* **5**, 020356 (2024).
- [36] F. Borsoi, N. W. Hendrickx, V. John, M. Meyer, S. Motz, F. van Riggelen, A. Sammak, S. L. de Snoo, G. Scappucci, and M. Veldhorst, *Nature Nanotechnology* **19**, 21 (2024).
- [37] K. Raach, in *Bulletin of the American Physical Society* (American Physical Society, 2024).
- [38] 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, *Physical Review X* **13**, 011023 (2023).
- [39] R. Li, L. Petit, D. P. Franke, J. P. Dehollain, J. Helsen, M. Steudtner, N. K. Thomas, Z. R. Yoscovits, K. J. Singh, S. Wehner, L. M. K. Vandersypen, J. S. Clarke, and M. Veldhorst, *Science Advances* **4**, eaar3960 (2018).
- [40] D. P. Franke, J. S. Clarke, L. M. K. Vandersypen, and M. Veldhorst, *Microprocessors and Microsystems* **67**, 1 (2019).
- [41] J. M. Boter, J. P. Dehollain, J. P. G. van Dijk, T. Hensgens, R. Versluis, J. S. Clarke, M. Veldhorst, F. Sebastiano, and L. M. K. Vandersypen, in *2019 IEEE International Electron Devices Meeting (IEDM)* (2019) pp. 31.4.1–31.4.4.
- [42] X. Xue, B. Patra, J. P. G. van Dijk, N. Samkharadze, S. Subramanian, A. Corna, B. Paquelet Wuetz, C. Jeon, F. Sheikh, E. Juarez-Hernandez, B. P. Esparza, H. Ramapurawala, B. Carlton, S. Ravikumar, C. Nieva, S. Kim, H.-J. Lee, A. Sammak, G. Scappucci, M. Veldhorst, F. Sebastiano, M. Babaie, S. Pellerano, E. Charbon, and L. M. K. Vandersypen, *Nature* **593**, 205 (2021).
- [43] Z. Shi, C. B. Simmons, J. R. Prance, J. K. Gamble, T. S. Koh, Y.-P. Shim, X. Hu, D. E. Savage, M. G. Lagally, M. A. Eriksson, M. Friesen, and S. N. Coppersmith,

- Physical Review Letters **108**, 140503 (2012).
- [44] F. Setiawan, H.-Y. Hui, J. P. Kestner, X. Wang, and S. D. Sarma, *Physical Review B* **89**, 085314 (2014).
- [45] B. H. Fong and S. M. Wandzura, *Quantum Information & Computation* **11**, 1003 (2011).
- [46] R. W. Andrews, C. Jones, M. D. Reed, A. M. Jones, S. D. Ha, M. P. Jura, J. Kerckhoff, M. Levendorf, S. Meenehan, S. T. Merkel, A. Smith, B. Sun, A. J. Weinstein, M. T. Rakher, T. D. Ladd, and M. G. Borselli, *Nature Nanotechnology* **14**, 747 (2019).
- [47] J. Kempe, D. Bacon, D. A. Lidar, and K. B. Whaley, *Physical Review A* **63**, 042307 (2001).
- [48] D. Zeuch and N. E. Bonesteel, *Physical Review A* **93**, 010303 (2016).
- [49] D. Zeuch and N. E. Bonesteel, *Physical Review B* **102**, 075311 (2020).
- [50] E. Knill, R. Laflamme, and W. H. Zurek, *Science* **279**, 342 (1998).
- [51] A. Yu. Kitaev, *Annals of Physics* **303**, 2 (2003).
- [52] D. Aharonov and M. Ben-Or, *SIAM Journal on Computing* **38**, 1207 (2008).
- [53] A. Y. Kitaev, *Russian Mathematical Surveys* **52**, 1191 (1997).
- [54] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, *Journal of Mathematical Physics* **43**, 4452 (2002).
- [55] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, *Nature Communications* **12**, 2172 (2021).
- [56] M. McEwen, D. Bacon, and C. Gidney, *Quantum* **7**, 1172 (2023).
- [57] Z. Cai, A. Siegel, and S. Benjamin, *PRX Quantum* **4**, 020345 (2023).
- [58] A. Strikis and L. Berent, *PRX Quantum* **4**, 020321 (2023).
- [59] A. Siegel, A. Strikis, and M. Fogarty, Towards early fault tolerance on a $2 \times N$ array of qubits equipped with shuttling (2024), [arXiv:2402.12599](https://arxiv.org/abs/2402.12599) [quant-ph].
- [60] B. Hetényi and J. R. Wootton, *Physical Review A* **109**, 032433 (2024).
- [61] G. Li, Y. Ding, and Y. Xie, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, Providence RI USA, 2019) pp. 1001–1014.
- [62] G. Nannicini, L. S. Bishop, O. Gunluk, and P. Jurcevic, Optimal qubit assignment and routing via integer programming (2021), [arXiv:2106.06446](https://arxiv.org/abs/2106.06446).
- [63] M. J. Reagor, T. C. Bohdanowicz, D. R. Perez, E. A. Sete, and W. J. Zeng, Hardware optimized parity check gates for superconducting surface codes (2022), [arXiv:2211.06382](https://arxiv.org/abs/2211.06382) [quant-ph].
- [64] R. Chao and B. W. Reichardt, *PRX Quantum* **1**, 010302 (2020).
- [65] J. Chadwick, G. G. Guerreschi, F. Lüthi, M. Madzik, F. Mohiyaddin, P. Prabhu, A. Schmitz, A. Litteken, S. Premaratne, N. Bishop, and J. Clarke, Data for "Short two-qubit pulse sequences for exchange-only spin qubits in 2D layouts" (2024).
- [66] Y. He, S. K. Gorman, D. Keith, L. Kranz, J. G. Keizer, and M. Y. Simmons, *Nature* **571**, 371 (2019).
- [67] A. J. Sigillito, M. J. Gullans, L. F. Edge, M. Borselli, and J. R. Petta, *npj Quantum Information* **5**, 1 (2019).
- [68] T. A. Baart, M. Shafiei, T. Fujita, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen, *Nature Nanotechnology* **11**, 330 (2016).
- [69] A. R. Mills, D. M. Zajac, M. J. Gullans, F. J. Schupp, T. M. Hazard, and J. R. Petta, *Nature Communications* **10**, 1063 (2019).
- [70] C. Volk, A. M. J. Zwerver, U. Mukhopadhyay, P. T. Eendebak, C. J. van Diepen, J. P. Dehollain, T. Hensgens, T. Fujita, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen, *npj Quantum Information* **5**, 1 (2019).
- [71] S. Che, S. W. Oh, H. Qin, Y. Liu, A. Sigillito, and G. Li, Fast Virtual Gate Extraction For Silicon Quantum Dot Devices (2024), [arXiv:2409.15181](https://arxiv.org/abs/2409.15181) [cond-mat].
- [72] M. McEwen, D. Bacon, and C. Gidney, Data for "Relaxing Hardware Requirements for Surface Code Circuits using Time-dynamics" (2023).

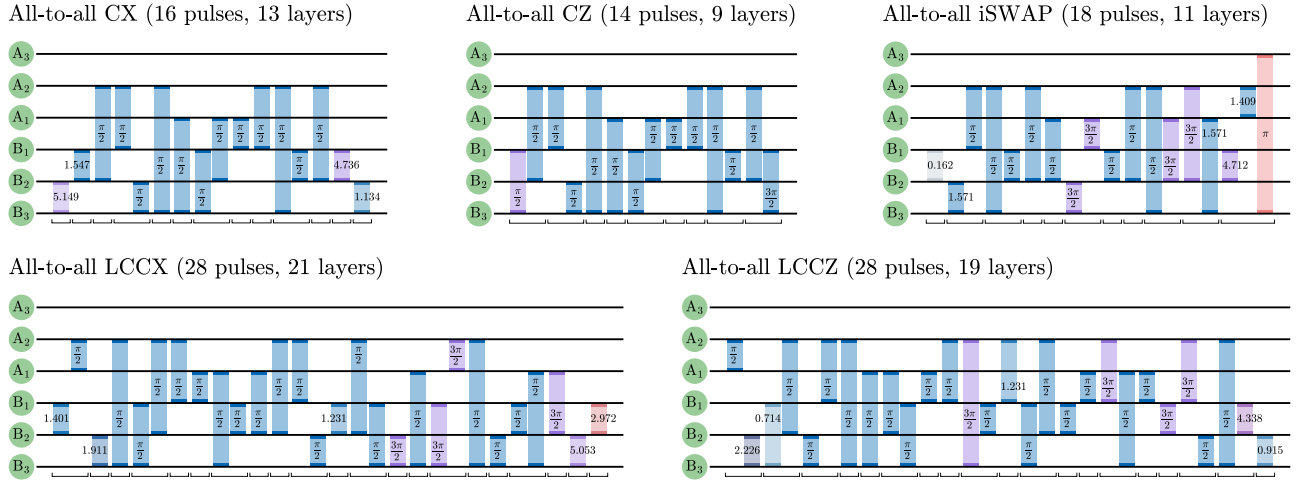


FIG. 12. All-to-all sequences used as reference sequences in the optimization algorithm.

Appendix: All-to-all pulse sequences

In Figure 12, we show the all-to-all pulse sequences for CX, CZ, iSWAP, LCCX, and LCCZ that are used in the optimization to obtain the pulse sequences presented in the main text.

The all-to-all CX pulse sequence was obtained by adding local corrections (the four pulses with non- $\pi/2$ angles) to the sequence from Ref. [44] to yield a full CX

operation. The CZ was obtained similarly with different local corrections. The iSWAP operation is locally equivalent to CX + SWAP + single-qubit corrections, so we took the all-to-all CX pulse, added an all-to-all SWAP (three π pulses) and the appropriate single-qubit corrections, and then removed/fused exchange pulses to reduce the final sequence length. The all-to-all LCCX and LCCZ were obtained by removing/fusing exchange pulses starting from the sequences presented in [32].