Exponentially faster implementations of Select(H) for fermionic Hamiltonians

Kianna Wan[1,2]

[1]Stanford Institute for Theoretical Physics, Stanford University, Stanford, CA 94305, USA
[2]PsiQuantum, Palo Alto, CA 94304, USA

We present a simple but general framework for constructing quantum circuits that implement the multiply-controlled unitary $\text{SELECT}(H) \coloneqq \sum_\ell |\ell\rangle\langle\ell| \otimes H_\ell$, where $H = \sum_\ell H_\ell$ is the Jordan-Wigner transform of an arbitrary second-quantised fermionic Hamiltonian. $\text{SELECT}(H)$ is one of the main subroutines of several quantum algorithms, including state-of-the-art techniques for Hamiltonian simulation. If each term in the second-quantised Hamiltonian involves at most k spin-orbitals and k is a constant independent of the total number of spin-orbitals n (as is the case for the majority of quantum chemistry and condensed matter models considered in the literature, for which k is typically 2 or 4), our implementation of $\text{SELECT}(H)$ requires no ancilla qubits and uses $\mathcal{O}(n)$ Clifford+T gates, with the Clifford gates applied in $\mathcal{O}(\log^2 n)$ layers and the T gates in $O(\log n)$ layers. This achieves an exponential improvement in both Clifford- and T-depth over previous work, while maintaining linear gate count and reducing the number of ancillae to zero.

1 Introduction

Quantum computers have the potential to efficiently simulate quantum systems. A particularly promising application of both near-term and fault-tolerant architectures is solving problems in quantum chemistry and materials science. In recent years, significant advances have been made on this front; for reviews of the major algorithmic developments, we refer the reader to Refs. [1–3].

Much of the current research in quantum simulation is concerned with the estimation of Hamiltonian spectra and preparation of energy eigenstates, which can provide insight into various properties of molecules and materials. As shown by Ref. [4], the quantum phase estimation algorithm [5, 6] can be used to perform projective energy measurements, collapsing the system into a desired eigenstate with high probability if the initial state has appreciable overlap with that eigenstate. Even in the absence of a suitable initial approximation, such measurements may be applied to prepare an eigenstate by exploiting the quantum Zeno effect [7, 8]. Alternatively, approximate eigenstates may be obtained via adiabatic state preparation, given sufficient information about the gap(s) in the spectrum of the interpolating Hamiltonian [9, 10].

Several techniques are useful for realising these schemes on a gate-based quantum computer. For instance, the qubitisation procedure of Refs. [11, 12] can implement the time-evolution operator $\exp(-iHt)$ or, more directly, a walk operator corresponding to $\exp[-i\arccos(Ht)]$, for a time-independent Hamiltonian H and some $t \in \mathbb{R}$. Either of these operators can be used as the unitary input to phase estimation for the purpose of approximating eigenvalues and eigenstates of H [4, 8, 13, 14]. Adiabatic evolution can be digitally simulated by applying the truncated Dyson series algorithm of Refs. [15, 16] for time-dependent Hamiltonian simulation, or by using the method of Ref. [17], which is based on quasi-adiabatic continuation [18]. Approximate ground states can also be prepared using the

methods of Ref. [19, 20]. All of these techniques are formulated in terms of queries to unitary oracles that encode the relevant Hamiltonian(s) in some form. One such encoding is the "linear combination of unitaries" (LCU) query model, motivated by the algorithms of Refs. [21, 22]. In this model, the input Hamiltonian is decomposed as

$$H = \sum_{\ell=0}^{L-1} \alpha_\ell H_\ell,$$

where each H_ℓ is a time-independent unitary and the (possibly time-dependent) coefficients α_ℓ are real and nonnegative. Information about the Hamiltonian is accessed via two oracles: $\mathrm{SELECT}(H)$ and $\mathrm{PREPARE}(\alpha)$, which respectively encode the unitaries H_ℓ and the coefficients α_ℓ. Specifically,

$$\mathrm{SELECT}(H) := \sum_{\ell=0}^{L-1} |\ell\rangle\langle\ell| \otimes H_\ell \tag{1}$$

is a multiply-controlled operation that applies the unitary H_ℓ to the target register conditioned on the control register being in the state $|\ell\rangle$, and $\mathrm{PREPARE}(\alpha)$ is some unitary that transforms the all-zeros state of the control register as

$$\mathrm{PREPARE}(\alpha) : |0\rangle \mapsto \sum_{\ell=0}^{L-1} \sqrt{\frac{\alpha_\ell}{\alpha}} |\ell\rangle,$$

where $\alpha := \sum_{\ell=0}^{L-1} \alpha_\ell$. (In the case where the coefficients are time-dependent, $\mathrm{PREPARE}(\alpha)$ may be controlled on an additional register that encodes time [16].)

While any operator can in principle be written as a linear combination of unitaries, some Hamiltonians are more naturally expressed in this framework. Since the complexities of the aforementioned algorithms are typically dominated by that of $\mathrm{SELECT}(H)$ and $\mathrm{PREPARE}(\alpha)$,[1] it is important to design time- and space-efficient circuits for these oracles. The purpose of this paper is to provide an efficient construction for $\mathrm{SELECT}(H)$ in the case where H is obtained from a fermionic Hamiltonian via the Jordan-Wigner transformation [23], so that each H_ℓ is a tensor product of Pauli operators. Although our method is applicable to arbitrary fermionic Hamiltonians, it is worth noting that in many of the models considered in practice, each site interacts with only a small number of other sites. More precisely, for a fermionic Hamiltonian given in its second-quantised representation, let k denote the maximum number of distinct spin-orbitals that appear in each term. For most Hamiltonians of physical interest, such as the commonly studied molecular electronic structure Hamiltonian and the Fermi-Hubbard model, k does not scale with the system size. Our contribution can be stated as follows.

Main result: For any fermionic Hamiltonian for which k is a constant independent of the number of spin-orbitals n, we can construct a circuit for $\mathrm{SELECT}(H)$ using zero ancilla qubits and $\mathcal{O}(n)$ Clifford and T gates, with the Clifford gates performed in $\mathcal{O}(\log^2 n)$ layers and the T gates in $\mathcal{O}(\log n)$ layers.

This constitutes an exponential reduction in Clifford- and T-depth compared to existing methods. The approach of Ref. [24] can be applied to arbitrary LCU inputs but requires $\mathcal{O}(L)$ Clifford and T gates and $\mathcal{O}(L)$ Clifford- and T-depth, and in general $L \in \mathcal{O}(n^k)$ for the type of Hamiltonians considered here. Ref. [14] improves the gate count and depth (for both Clifford and T gates) to $\mathcal{O}(n)$ for two specific $k = 2$ Hamiltonians. Like Ref. [14], we obtain a speedup by exploiting the structure of the Jordan-Wigner encoding.[2] However, our circuits are completely different in structure from those in

---

[1]Strictly speaking, these algorithms make calls to controlled versions of $\mathrm{SELECT}(H)$ and $\mathrm{PREPARE}(\alpha)$. In our implementation, a constant number of controls can be added to $\mathrm{SELECT}(H)$ with constant *additive* overhead in the gate complexity, as will become clear in Section 2.

[2]Although we focus on the Jordan-Wigner transformation in this paper, the same ideas can be used to efficiently

Ref. [14], which cannot be parallelised to sublinear-depth in any straightforward way. Moreover, our implementation uses no ancilla qubits, in contrast to the $\sim \log n$ required by Refs. [14] and [24].

Our construction can be directly applied to asymptotically improve the circuit depth of existing fermionic simulation algorithms that are bottlenecked by SELECT(H). In Ref. [14], for example, the complexity of simulating the planar Fermi-Hubbard model is dominated by that of SELECT(H), while PREPARE(α) is extremely easy to implement as there are only three unique coefficients in the Hamiltonian. By using our circuit for SELECT(H), the overall circuit depth of estimating energies via phase estimation to absolute error at most ϵ can be immediately reduced from $\widetilde{\mathcal{O}}(\alpha n/\epsilon)$ to $\widetilde{\mathcal{O}}(\alpha/\epsilon)$ in Theorem 2 of Ref. [14] [cf. Eq. (27) therein], where $\widetilde{\mathcal{O}}$ hides logarithmic factors. Similarly, in Ref. [30], the overall depth of approximating the time-evolution operator e^{-iHt} for the $k = 4$ Sachdev-Ye-Kitaev model with n Majorana modes can be reduced from $\widetilde{\mathcal{O}}(n^{3.5}t)$ to $\widetilde{\mathcal{O}}(n^{2.5}t)$.

In addition to the exponentially reduced circuit depth and minimal space overhead, an advantage of our construction lies in its simplicity and broad applicability. The circuits consist of very few different components, and take exactly the same form for all Hamiltonians with the same k (though straightforward optimisations can be made if the class of input Hamiltonians is further restricted). The bulk of the gate complexity is due to a single gadget, composed entirely of controlled-SWAP and CNOT gates. Thus, while the use of circuit depth as a complexity measure is mainly justified by long-term considerations (of prospective architectures in which many fault-tolerant gates can be executed in parallel), the simple structure of our circuits potentially makes them amenable to near-term implementation.

2 Circuit construction

In this section, we prove our main result. We begin in subsection 2.1 by developing the circuit for SELECT(H) for a particular class of fermionic Hamiltonians, to illustrate the main idea. It will then become obvious how circuits for arbitrary fermionic Hamiltonians can be built, as we discuss in subsection 2.3, and that these circuits have linear gate count and polylogarithmic depth provided that $k \in \mathcal{O}(1)$. We also describe, in subsection 2.2, a simple way to substantially reduce the constant factors in the scaling of the T-count and T-depth.

Conventions. Unsurprisingly, circuit diagrams are an essential part of this paper. We will use the following convention for representing operators that are controlled in a nontrivial manner on one or more qubits. Such an operator will be depicted by drawing a small solid square on the control register, connected to a box on the target register that contains the name of the operator or an abbreviation thereof. For instance, SELECT(H) will be represented by
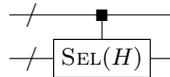
To clearly distinguish different registers, we will often label a control register using a computational basis state, and the target register using an arbitrary state $|\psi\rangle$. As an example, since ℓ is used to index the computational basis states of the control register of SELECT(H) in Eq. (1), we may add the labels $|\ell\rangle$ and $|\psi\rangle$ to the above circuit representation of SELECT(H):

implement SELECT(H) for other second-quantised fermion-to-qubit mappings that have sufficient structure. In particular, our construction can be generalised to the class of mappings defined in Ref. [25], which includes the Jordan-Wigner and Bravyi-Kitaev [26–28] transformations as special cases. This would give ancilla-free circuits with the same asymptotic gate count and depth, though with larger constant factors in general [29].

(Note that if a circuit identity holds for any computational state on the control register and any arbitrary state on the target register, it holds for all input states.) We will refer to the control register of SELECT(H) as the "selection register" and the target register as the "system register."

2.1 Main idea

Our method is most easily explained by first considering quadratic fermionic Hamiltonians that consist only of terms involving two distinct spin-orbitals. The most general form of such a Hamiltonian in a second-quantised basis is

$$\hat{H} = \sum_{p<q} \left(t_{pq} a_p^\dagger a_q + t_{pq}^* a_q^\dagger a_p + \Delta_{pq} a_p^\dagger a_q^\dagger + \Delta_{pq}^* a_q a_p \right),$$

where $a_p^\dagger$ and a_p are fermionic creation and annihilation operators associated with spin-orbital p, and $t_{pq}, \Delta_{pq} \in \mathbb{C}$. For a system of n spin-orbitals, we label the spin-orbitals from 0 to $n-1$ in accordance with the canonical ordering chosen for the Jordan-Wigner transformation. Under this transformation, the fermionic operators are mapped to Pauli operators on n qubits as $a_p \mapsto (\prod_{j=0}^{p-1} Z_j)(X_p + iY_p)/2$ and $a_p^\dagger \mapsto (\prod_{j=0}^{p-1} Z_j)(X_p - iY_p)/2$, so for $p < q$,

$$t_{pq} a_p^\dagger a_q + t_{pq}^* a_q^\dagger a_p \mapsto \frac{1}{2} \vec{Z}_{p,q} \left[\mathrm{Re}(t_{pq})(X_p X_q + Y_p Y_q) + \mathrm{Im}(t_{pq})(-X_p Y_q + Y_p X_q) \right] \qquad (2)$$

$$\Delta_{pq} a_p^\dagger a_q^\dagger + \Delta_{pq}^* a_q a_p \mapsto \frac{1}{2} \vec{Z}_{p,q} \left[\mathrm{Re}(\Delta_{pq})(X_p X_q - Y_p Y_q) + \mathrm{Im}(\Delta_{pq})(X_p Y_q + Y_p X_q) \right]. \qquad (3)$$

Here and throughout the paper, Z_p denotes the n-qubit operator that acts as Z on qubit p and as the identity on the rest of the qubits, and similarly for X and Y, while $\vec{Z}_{p,q} := \prod_{j=p+1}^{q-1} Z_j$ denotes a string of Z operators on all of the qubits between p and q (exclusive). Thus, the Jordan-Wigner transform H of $\hat{H}$ is a linear combination with real coefficients of operators that all have the form $(P_1)_p \vec{Z}_{p,q} (P_2)_q$ with $P_1, P_2 \in \{X, Y\}$. Absorbing the signs of the coefficients into P_1, we can write

$$H = \sum_{p,q=0}^{n-1} \sum_{P_1 \in \{\pm X, \pm Y\}} \sum_{P_2 \in \{X,Y\}} \alpha_{p,q,P_1,P_2} (P_1)_p \vec{Z}_{p,q} (P_2)_q,$$

where the coefficients α_{p,q,P_1,P_2} are all nonnegative and $\alpha_{p,q,P_1,P_2} = 0$ for $p \geq q$.

Clearly, H is a linear combination of unitaries, and each of the unitaries is completely specified by the two spin-orbitals p and q and the Pauli operators P_1 and P_2. Accordingly, we allocate $2\lceil \log n \rceil + 3$ qubits to the selection register, and encode each computational basis state $|\ell\rangle$ of the selection register as $|\ell\rangle \equiv |p\rangle|q\rangle|P_1\rangle|P_2\rangle$. The first two subregisters each contain $\lceil \log n \rceil$ qubits and store the binary representations of $p, q \in \{0, \ldots, n-1\}$. The third and fourth subregisters, which have two qubits and one qubit, respectively, specify $P_1 \in \{\pm X, \pm Y\}$ and $P_2 \in \{X, Y\}$. By Eq. (1), SELECT(H) can then be defined by its action on computational basis states in the selection register (and an arbitrary state $|\psi\rangle$ in the system register) as

$$\text{SELECT}(H) : |p\rangle|q\rangle|P_1\rangle|P_2\rangle \otimes |\psi\rangle \mapsto |p\rangle|q\rangle|P_1\rangle|P_2\rangle \otimes (P_1)_p \vec{Z}_{p,q} (P_2)_q |\psi\rangle \qquad (4)$$

for $p, q \in \{0, \ldots, n-1\}$. (The action of SELECT($H$) on basis states for which p and/or q are out of range is unimportant, as PREPARE(α)$|0\rangle$ has no support on such states.)

To construct a circuit that implements Eq. (4), our starting point is the following circuit identity:

$$(5)$$

which is an immediate consequence of the elementary identities

and the fact that CNOT is self-inverse. The analogue of Eq. (5) for an arbitrary number of qubits and with the two Z operators on a different pair of qubits is obvious. Letting Q_{P_1} denote the Pauli operator such that $Q_{P_1}Z = iP_1$ for $P_1 \in \{\pm X, \pm Y\}$ (i.e., $Q_{\pm X} = \pm Y$ and $Q_{\pm Y} = \mp X$), it follows that

$$(6)$$

Therefore, if the n qubits in the system register are ordered such that the qubit corresponding to spin-orbital 0 is on the top wire and the qubit corresponding to spin-orbital $n-1$ is on the bottom, the circuit on the left-hand side would implement the term $(P_1)_p \vec{Z}_{p,q}(P_2)_q|\psi\rangle$ for a particular p, q, P_1, P_2.

From here, we would obtain a circuit for SELECT(H) if we were to control the Q_{P_1}, P_2, and Z operators in the circuit of Eq. (6) on the selection register such that

(1) the states $|P_1\rangle$ and $|P_2\rangle$ of the third and fourth selection subregisters determine which Pauli operators Q_{P_1} and P_2 represent, and

(2) conditioned on the first two selection subregisters being in the state $|p\rangle|q\rangle$, Q_{P_1} and one of the Z operators are applied to qubit p of the system register, while P_2 and the other Z operator are applied to qubit q.

Condition (1) can be straightforwardly satisfied by constructing SELECT(Q) and SELECT(P) operators that choose the appropriate Q_{P_1} and P_2 according to the states $|P_1\rangle$ and $|P_2\rangle$. For concreteness, suppose that $|P_1\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$ for $P_1 = X, -X, Y, -Y$, respectively, and $|P_2\rangle = |0\rangle, |1\rangle$ for $P_2 = X, Y$, respectively. Then, SELECT(Q) and SELECT(P) can be implemented as

$$(7)$$

Condition (2) requires the ability to target a particular qubit in the system register depending on the states of the selection subregisters that encode $|p\rangle$ and $|q\rangle$. For this purpose, we define for

any single-qubit unitary U a $(\lceil \log n \rceil + n)$-qubit operator $\textsc{Inject}(U)$. When applied to $|x\rangle|\psi\rangle$, where $x \in \{0, \ldots, n-1\}$ (encoded in binary) and $|\psi\rangle$ is an arbitrary n-qubit state, $\textsc{Inject}(U)$ implements U on qubit x of $|\psi\rangle$ and acts as the identity on the other qubits, i.e.,

$$\textsc{Inject}(U) : |x\rangle|\psi\rangle \mapsto |x\rangle \left[\left(I^{\otimes x} \otimes U \otimes I^{\otimes(n-x-1)} \right) |\psi\rangle \right].$$

To synthesise $\textsc{Inject}(U)$ for any U, we use a $(\lceil \log n \rceil + n)$-qubit operator $\textsc{SwapUp}$, defined as follows. For any $x \in \{0, \ldots, n-1\}$ and n-qubit product state $\bigotimes_{y=0}^{n-1} |\varphi_y\rangle_y$,

$$\textsc{SwapUp} : |x\rangle \bigotimes_{y=0}^{n-1} |\varphi_y\rangle_y \mapsto |x\rangle|\varphi_x\rangle_0 \bigotimes_{y=1}^{n-1} |\varphi_{\sigma(y)}\rangle_y, \tag{8}$$

where σ is a fixed[3] permutation of $\{0, \ldots, n-1\}$ such that $\sigma(0) = x$. In other words, conditioned on the $\lceil \log n \rceil$-qubit control register being in the state x for $x \in \{0, \ldots, n-1\}$, $\textsc{SwapUp}$ moves the state $|\varphi_x\rangle$ of the qubit indexed by x in the target register up to the first qubit of the target register, and permutes the states of the other qubits in some way. Ref. [31] shows that $\textsc{SwapUp}$ can be implemented without ancilla qubits using $\mathcal{O}(n)$ Clifford and T gates, $O(\log^2 n)$ Clifford-depth, and $O(\log n)$ T-depth. We sketch the construction in Appendix A.1. From Eq. (8), it is easy to see that $\textsc{Inject}(U) = \textsc{SwapUp}^\dagger(U \otimes I^{(n-1)})\textsc{SwapUp}$. $\textsc{SwapUp}$ permutes the qubits in the target register in such a way that the state of qubit x is moved up to the first qubit, U is then applied to the first qubit, and $\textsc{SwapUp}^\dagger$ undoes the permutation.

(9)
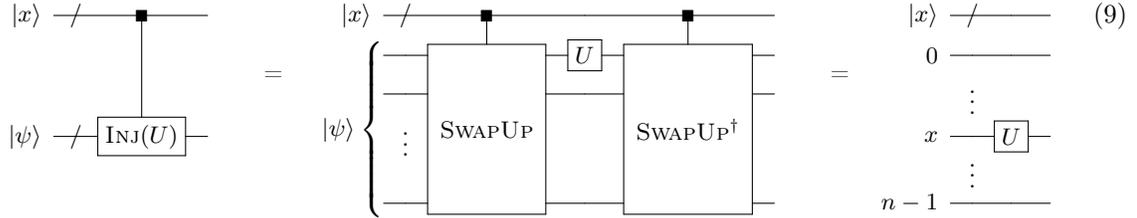
(The third circuit above illustrates the effect of $\textsc{Inject}(U)$ in the special case where the input to the control register is a computational basis state, whereas the second circuit implements $\textsc{Inject}(U)$ on arbitrary inputs.)

Hence, we can ensure that the two Z operators in Eq. (6) are applied to qubits p and q of the system register conditioned on the state of the first two selection subregisters being $|p\rangle|q\rangle$ by implementing two $\textsc{Inject}(Z)$ operators, one with $|p\rangle$ as the control register and the other with $|q\rangle$ as the control register. To correctly apply Q_{P_1} and P_2, we use the $\textsc{Select}(Q)$ and $\textsc{Select}(P)$ circuits constructed in Eq. (7) in conjunction with $\textsc{SwapUp}$ to form $\textsc{Inject-Select}(Q)$ and $\textsc{Inject-Select}(P)$. This is shown below for $\textsc{Inject-Select}(Q)$; the construction of $\textsc{Inject-Select}(P)$ is analogous.


(10)

---
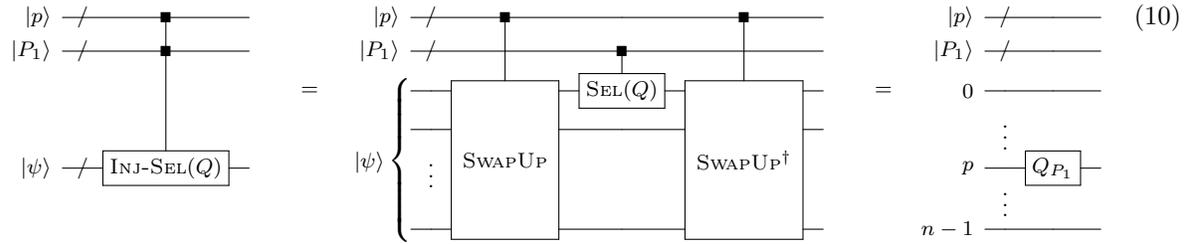
[3]In principle, any permutation σ such that $\sigma(0) = x$ would work—the action of σ on $\{1, \ldots, n-1\}$ is not relevant to the correctness of our construction. However, the implementation of $\textsc{SwapUp}$ in Appendix A.1 yields a particular permutation σ (for each n).

circuit component	T-count	T-depth	# of elementary gates to control
$\textsc{Inject}(Z)$	$28(n-1)$	$32\lceil \log n \rceil$	1
$\textsc{Inject}^*(Z)$	$8(n-1)$	$8\lceil \log n \rceil$	
$\textsc{Inject-Select}(Q)$	$28(n-1)$	$32\lceil \log n \rceil$	4
$\textsc{Inject-Select}^*(Q)$	$16(n-1)$	$16\lceil \log n \rceil$	
$\textsc{Inject-Select}(P)$	$28(n-1)$	$32\lceil \log n \rceil$	2
$\textsc{Inject-Select}^*(P)$	$16(n-1)$	$16\lceil \log n \rceil$	

Table 1: T-count and T-depth of each of the main components used to construct circuits for $\textsc{Select}(H)$ in Section 2. (The Clifford-count and Clifford-depth are $\mathcal{O}(n)$ and $\mathcal{O}(\log^2 n)$, respectively, for all of the components.) The fourth column specifies the number of one- or two-qubit (Clifford) gates to which controls need to be added in order to construct the controlled version of the operator in the first column.

With these components in hand, we can assemble the circuit for $\textsc{Select}(H)$:

(11)
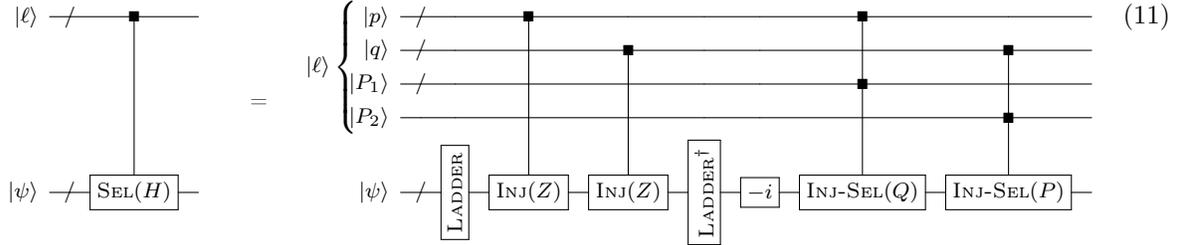
where $\textsc{Ladder}$ denotes the operator implemented by the ladder-like sequence of $n-1$ CNOT gates in Eq. (6):

(12)
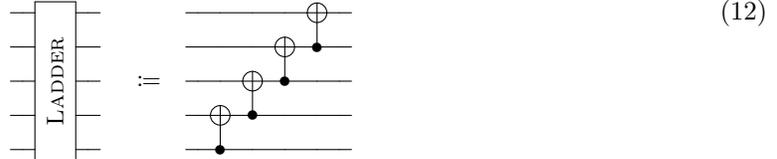
By comparing the circuit in Eq. (11) to that in Eq. (6), it can be verified that the former correctly implements $\textsc{Select}(H)$ as it is defined in Eq. (4). When the selection register is in the computational basis state $|p\rangle|q\rangle|P_1\rangle|P_2\rangle$, the two $\textsc{Inject}(Z)$ operators apply Z operators on qubits p and q in the system register, between the two sequences of CNOT gates corresponding to $\textsc{Ladder}$ and $\textsc{Ladder}^\dagger$. Then, $\textsc{Inject-Select}(Q)$ applies Q_{P_1} to qubit q and $\textsc{Inject-Select}(P)$ applies P_2 to qubit q. Thus, by Eq. (6), the circuit applies the operator $(P_1)_p \vec{Z}_{p,q}(P_2)_q$ on the system register conditioned on the state of the selection register being $|p\rangle|q\rangle|P_1\rangle|P_2\rangle$, as required by Eq. (4). This holds for all of the computational basis states, and therefore for arbitrary states of the selection register.

It is clear from Eqs. (7), (9), and (10) that the only non-Clifford gates in the circuit of Eq. (11) are the $\textsc{SwapUp}$ gadgets used to construct the "$\textsc{Inject}$" operators. As shown in Appendix A.1, $\textsc{SwapUp}$ on $\lceil \log n \rceil + n$ qubits can be implemented with $\mathcal{O}(n)$ T gates and $\mathcal{O}(\log n)$ T-depth. Each of $\textsc{Inject}(Z)$, $\textsc{Inject-Select}(Q)$, and $\textsc{Inject-Select}(P)$ uses one $\textsc{SwapUp}$ and one $\textsc{SwapUp}^\dagger$. Since the number of these operators is a constant independent of n, the total T-count of the circuit in Eq. (11) is $\mathcal{O}(n)$ and the total T-depth is $\mathcal{O}(\log n)$. The exact T costs of the components are provided by Table 1.

The Clifford complexity of $\textsc{Inject}(Z)$, $\textsc{Inject-Select}(Q)$, and $\textsc{Inject-Select}(P)$ is dominated by that of $\textsc{SwapUp}$ and its inverse, which have Clifford-count $\mathcal{O}(n)$ and Clifford-depth $\mathcal{O}(\log^2 n)$ [cf. Appendix A.1]. The circuit for the $\textsc{Ladder}$ operator given in Eq. (12) has Clifford-depth $n-1$; however, as shown in Appendix A.3, the same operator can be implemented using an ancilla-free circuit

consisting of $\mathcal{O}(n)$ CNOTs arranged in $\mathcal{O}(\log n)$ layers. It follows that the total Clifford-count of the circuit in Eq. (11) is $\mathcal{O}(n)$ and the total Clifford-depth is $\mathcal{O}(\log^2 n)$.

Although the circuit of Eq. (11) achieves $\mathcal{O}(n)$ T-count and $\mathcal{O}(\log n)$ T-depth, we can reduce the constant factors hidden under the big $\mathcal{O}$ if desired by making a few modifications, as demonstrated in the following subsection. We conclude this subsection by noting that the controlled version of SELECT(H) can be constructed by adding controls to a very small number of gates in the circuit— namely, the Z operator in each INJECT(Z) [cf. Eq. (9)], the Pauli and controlled-Pauli operators in INJECT-SELECT(Q) and INJECT-SELECT(P) [cf. Eqs. (7) and (10)], and the $(-i)$-phase gate (by implementing an $S^\dagger$ gate on the control qubit). When these operators are not applied, the circuit implements the identity since the LADDER and SwapUp operators are cancelled by their inverses. Therefore, controlling the entire circuit on any constant number of qubits incurs only constant additive gate complexity (which can be quantified using the fourth column of Table 1). This is important because algorithms that use the LCU query model generally require access to controlled-SELECT(H).

2.2 Reducing the constant factors

Before generalising the SELECT(H) circuit in subsection 2.1 to arbitrary fermionic Hamiltonians, we provide more efficient versions of the main circuit components, which can be used to reduce the T-count and T-depth by constant multiplicative factors. The Clifford-count and Clifford-depth are reduced by constant factors as well. However, we focus on the T complexity in this subsection because T gates are the bottleneck in many models of fault-tolerant quantum computation, notably those that are based on topological error correcting codes. In these settings, T gates require significantly more time and physical qubits to implement than Clifford gates [32], and even a constant-factor improvement in the T complexity may be useful.

The strategy is to replace all of the SwapUp operators by a particular phase-incorrect SwapUp operator, which is based on a phase-incorrect TOFFOLI gate introduced in Ref. [33]. As pointed out by Ref. [31], this phase-incorrect SwapUp, which we will call SwapUp*, can be implemented using $4(n-1)$ T gates that are applied in $4\lceil \log n \rceil$ layers, a considerable reduction from the $14(n-1)$ T-count and $16\lceil \log n \rceil$ T-depth of SwapUp. The circuit for SwapUp* is described in Appendix A.2. In the computational basis, SwapUp* has the same matrix elements as SwapUp up to sign, i.e., for any computational basis state $|z\rangle$,

$$\text{SwapUp}^*|z\rangle = \pm\text{SwapUp}|z\rangle = \pm|z'\rangle,$$

where $|z'\rangle = \text{SwapUp}|z\rangle$ is also a computational basis state. Hence, we can write $\text{SwapUp}^* = D \cdot \text{SwapUp}$ for some operator D that is diagonal in the computational basis, with eigenvalues ± 1. This implies that INJECT(Z) would still be implemented correctly if SwapUp and SwapUp† in the circuit of Eq. (9) were replaced with SwapUp* and SwapUp*†:

$$\text{SwapUp}^{*\dagger}Z_1\text{SwapUp}^* = (\text{SwapUp}^\dagger D^\dagger)Z_1(D \cdot \text{SwapUp})$$
$$= \text{SwapUp}^\dagger Z_1\text{SwapUp} = \text{INJECT}(Z),$$

where the second equality follows from the fact that D commutes with Z_1 and is unitary. We denote this implementation of INJECT(Z) by INJECT*(Z).[4] By the same token, SwapUp* can be used to construct INJECT(U) for any U that is diagonal in the computational basis.

On the other hand, the circuit in Eq. (10) would not correctly implement INJECT-SELECT(Q) if SwapUp* were used instead of SwapUp, since X and Y are not diagonal in the computational basis. To reduce the T cost of INJECT-SELECT(Q), we modify the circuits so that X and Y are "injected"

---

[4]We clarify that unlike in the case of SwapUp and SwapUp*, which are different operators, INJECT(Z) and INJECT*(Z) designate different circuit implementations of the *same* operator. The same goes for INJECT-SELECT(Q) and INJECT-SELECT*(Q), and INJECT-SELECT(P) and INJECT-SELECT*(P).

separately by applying $\textsc{Inject}^*(Z)$ conjugated by basis change operators, as follows:
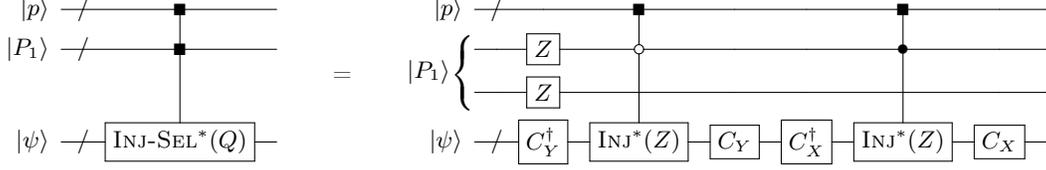
using C_X and C_Y to represent $B_X^{\otimes n}$ and $B_X^{\otimes n}$, where B_X and B_Y are Clifford gates for which $B_X Z B_X^\dagger = X$ and $B_Y Z B_Y^\dagger = Y$. Note that controlled-$\textsc{Inject}(Z)$ can be constructed by simply adding a control to the Z operator in $\textsc{Inject}^*(Z)$. The construction for $\textsc{Inject-Select}(Q)$ is similar [cf. Eq. (7)]. We denote these alternative implementations of $\textsc{Inject-Select}(Q)$ and $\textsc{Inject-Select}(P)$ by $\textsc{Inject-Select}^*(Q)$ and $\textsc{Inject-Select}^*(P)$.

The T-count and T-depth of each of these improved circuit components follow directly from the T cost of $\textsc{SwapUp}^*$, and are listed in Table 1. $\textsc{Inject}(Z)$, $\textsc{Inject-Select}(P)$, and $\textsc{Inject-Select}(Q)$ can always be replaced with their asterisked counterparts to minimise the complexity. For example, the circuit in Eq. (11), which implements $\textsc{Select}(H)$ for quadratic fermionic Hamiltonians, has T-count $112(n-1)$ and T-depth $128\lceil \log n \rceil$. Replacing the components in Eq. (11) by their improved versions would reduce the total T-count to $48(n-1)$ and the T-depth to $48\lceil \log n \rceil$.[5]

2.3 Generalising to arbitrary k

We can extend the ideas of subsections 2.1 and 2.2 to devise ancilla-free implementations of $\textsc{Select}(H)$ for the Jordan-Wigner transforms of arbitrary fermionic Hamiltonians. If at most k distinct spin-orbitals are involved in each term of the fermionic Hamiltonian, the resulting circuit has Clifford- and T-count $\mathcal{O}(kn)$, Clifford-depth $\mathcal{O}(k\log^2 n)$, and T-depth $\mathcal{O}(k\log n)$. To help illustrate the concepts by way of circuit diagrams, we will use $k=4$ Hamiltonians as a concrete example.

In its second-quantised representation, each term in a general fermionic Hamiltonian is a product of interaction operators $a_p^\dagger a_q$ or $a_p^\dagger a_q^\dagger$ (with $p < q$) and their Hermitian conjugates, and number operators $n_p := a_p^\dagger a_p$.[6] By definition of k, Hamiltonians with $k=4$ may include such terms as $a_p^\dagger a_q a_r^\dagger a_s + \text{h.c.}$, $a_p^\dagger a_q n_r n_s + \text{h.c.}$, n_p, and $n_p n_q n_r$, to list a few examples. The circuit for $\textsc{Select}(H)$ can be constructed in two main parts. Loosely speaking, one part of the circuit implements interaction operators and the other part implements number operators.

As we saw in subsection 2.1, an interaction operator involving two spin-orbitals p and q is mapped to linear combinations of Pauli "strings" of the form $(P_1)_p \vec{Z}_{p,q}(P_2)_q$ under the Jordan-Wigner transformation, with $P_1, P_2 \in \{X, Y\}$ [cf. Eqs. (2) and (3)]. More generally, any product of interaction operators is mapped to a linear combination of products of such Pauli strings. For instance, $a_p^\dagger a_q a_r^\dagger a_s + \text{h.c.}$ (for $p < q < r < s$) becomes a linear combination of $(P_1)_p \vec{Z}_{p,q}(P_2)_q(P_3)_r \vec{Z}_{r,s}(P_4)_q$, for $P_1, P_2, P_3, P_4 \in \{X, Y\}$. Hence, the circuit in Eq. (11), which implements $\textsc{Select}(H)$ in the special case that the Hamiltonian consists only of interactions between two spin orbitals, can be easily expanded to implement Hamiltonians containing arbitrary products of interaction operators. The key observation is that the identities in Eqs. (5) and (6) hold analogously for any number of pairs of Z

---

[5]As a side note, the $\textsc{Toffoli}$-count of any of the non-asterisked components can be obtained by dividing the corresponding T-count in Table 1 by 7, and the $\textsc{Toffoli}$-depth can be obtained by dividing the T-depth by 4. The asterisked operators are not constructed using $\textsc{Toffolis}$. See Appendix A.1 and A.2 for details.

[6]In theory, the Hamiltonian could contain terms that are linear, cubic, etc. in the fermionic operators. Our method can be used to implement these terms as well (basically, by exploiting the identity in Eq. (5) except with an odd number of Z operators), but since they rarely appear in Hamiltonians of interest, we omit them for simplicity.

operators, e.g., for two pairs of Z operators, we have

(13)

Consequently, by the exact same logic as that in subsection 2.1, we can "select" between Pauli operators corresponding to interaction terms using a circuit composed of the INJECT(Z), INJECT-SELECT(Q), and INJECT-SELECT(P) subroutines defined in subsection 2.1, along with a LADDER and LADDER†. This circuit would essentially be an extended version of that in Eq. (11), with two minor modifications. First, instead of absorbing the sign of the Pauli operator into P_1, as we did in subsection 2.1, we use one qubit $|\text{sgn}\rangle$ in the selection register to encode the sign (with $|\text{sgn}\rangle = |0\rangle$ corresponding to $+1$ and $|\text{sgn}\rangle = |1\rangle$ to -1). We then remove the second wire in the circuit for SELECT(Q) in Eq. (7), and modify INJECT-SELECT(Q) accordingly. Second, to account for the possibility that different terms in the Hamiltonian may be products of different numbers of interaction operators (e.g., $a_p^\dagger a_q + $ h.c. and $a_p^\dagger a_q a_r^\dagger a_s + $ h.c. may both be present in a $k = 4$ Hamiltonian), we use k of the qubits in the selection register as control qubits. We denote the states of these control qubits by $|\text{I}_p\rangle$, $|\text{I}_q\rangle$, $|\text{I}_r\rangle$, $|\text{I}_s\rangle$, etc. Note that the Pauli string $(P_3)_r \vec{Z}_{r,s}(P_4)_s$ on the right-hand side of Eq. (13) would not be implemented if the bottom two Z operators, Q_{P_3}, and P_4 are not applied on the left-hand side. It follows that by controlling the corresponding INJECT(Z), INJECT-SELECT(P), and INJECT-SELECT(Q) operators on $|\text{I}_r\rangle$ and $|\text{I}_s\rangle$, either $(P_1)_p \vec{Z}_{p,q}(P_2)_q(P_3)_r \vec{Z}_{r,s}(P_4)_q$ or $(P_1)_p \vec{Z}_{p,q}(P_2)_q$ is applied depending on the state $|\text{I}_r\rangle|\text{I}_s\rangle$. The operators associated with p and q are controlled as well in order to allow for the implementation of number operators, which do not transform into Pauli strings of the form in Eq. (13) [cf. Eq. (14) below]. For the example of $k = 4$, the (sub)circuit for interaction operators is the left part (labelled "interaction operators") of the circuit in Eq. (15).

Number operators are very straightforward to implement using INJECT(Z) gates, since

$$n_p \mapsto \frac{1}{2}(I - Z_p) \tag{14}$$

under the Jordan-Wigner transformation. Therefore, to incorporate the Hamiltonian terms that involve number operators, the state of the selection register simply needs to indicate whether Z operators should be applied on certain qubits (recalling that the overall sign is encoded in $|\text{sgn}\rangle$). It suffices to use another k of the selection register qubits as control qubits, labelling their states by $|\text{N}_p\rangle$, $|\text{N}_q\rangle$, $|\text{N}_r\rangle$, $|\text{N}_s\rangle$, etc., and control an INJECT(Z) operator on $|p\rangle$ and $|\text{N}_p\rangle$, another INJECT(Z) operator on $|q\rangle$ and $|\text{N}_q\rangle$, and so on. The full circuit for SELECT(H) for any $k = 4$ Hamiltonian is shown in Eq. (15) below. As always, each of the INJECT(Z), INJECT-SELECT(Q), and INJECT-SELECT(P) gates can be replaced by their more efficient variants constructed in subsection 2.2.

All possible terms can be encoded by appropriately choosing the correspondence between the Pauli operators in the Jordan-Wigner encoding and the computational basis states of each subregister (and constructing PREPARE(α) in a way that is consistent with this correspondence). As an explicit example, suppose that the Hamiltonian includes terms of the form $a_p^\dagger a_q n_r + $ h.c., which transform to linear combinations of $(P_1)_p \vec{Z}_{p,q}(P_2)_q$ and $(P_1)_p \vec{Z}_{p,q}(P_2)_q Z_r$, with $P_1, P_2 \in \{X, Y\}$. It can be seen from Eq. (15) that the first type of Pauli operators are applied when $|\text{N}_r\rangle = |0\rangle$ and the second type are

applied when $|\text{N}_r\rangle = |1\rangle$, with $|\text{I}_p\rangle|\text{I}_q\rangle|\text{I}_r\rangle|\text{I}_s\rangle = |1\rangle|1\rangle|0\rangle|0\rangle$ and $|\text{N}_p\rangle|\text{N}_q\rangle|\text{N}_s\rangle = |0\rangle|0\rangle|0\rangle$ for both. While the circuit in Eq. (15) implements arbitrary terms involving up to $k = 4$ spin-orbitals, it is often the case that the Hamiltonian in question only contains a few types of terms. Some of the qubits could then be removed from the selection register and the control logic could be simplified. For the molecular electronic structure Hamiltonian, which is a linear combination of $a_p^\dagger a_q + \text{h.c.}$, $a_p^\dagger a_q a_r^\dagger a_s + \text{h.c.}$, n_p, $n_p n_q$, and $a_p^\dagger a_q n_r + \text{h.c.}$ [34, 35], we would not need the qubits storing $|\text{N}_r\rangle$ and $|\text{N}_s\rangle$ and the last two INJECT(Z) gates in Eq. (15), for instance.

$$\tag{15}$$
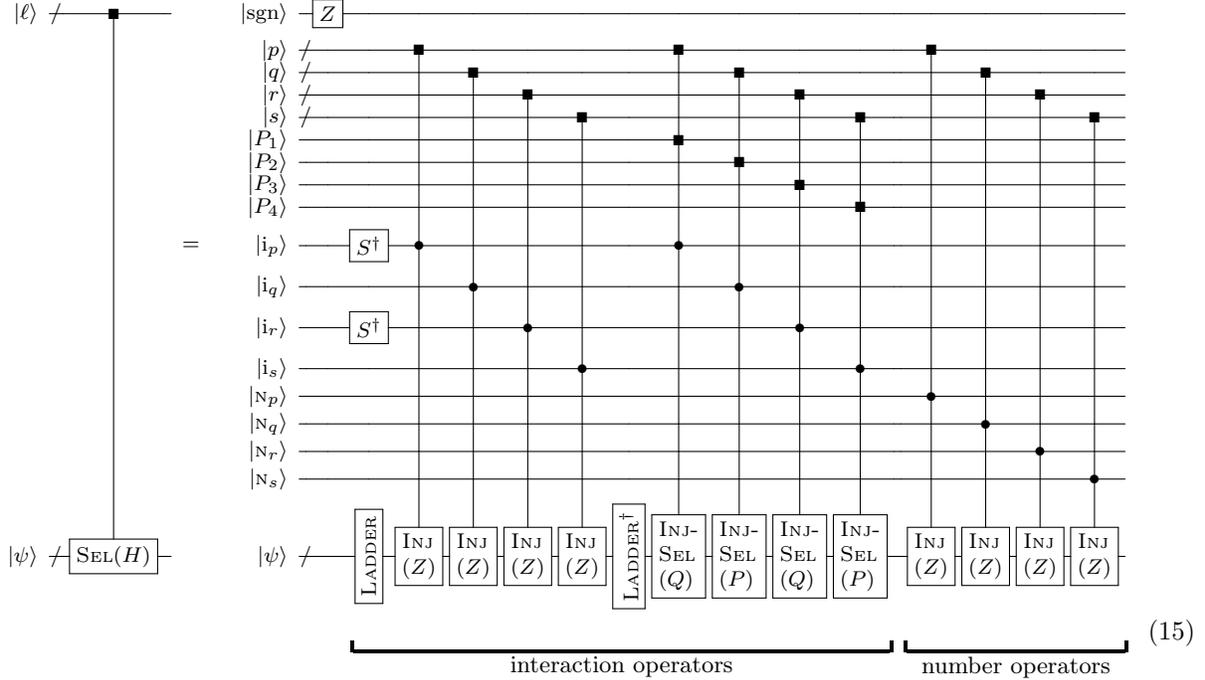
Thus, for arbitrary k, the circuit for SELECT(H) can be constructed from at most $2k$ controlled-INJECT(Z), $\lfloor k/2 \rfloor$ controlled-INJECT-SELECT(Q), and $\lfloor k/2 \rfloor$ controlled-INJECT-SELECT(P), and 2 LADDER operators. INJECT(Z), INJECT-SELECT(Q), and INJECT-SELECT(P) are implemented using $\mathcal{O}(n)$ Clifford and T gates, $\mathcal{O}(\log^2 n)$ Clifford-depth, and $\mathcal{O}(\log n)$ T-depth, while LADDER can be implemented using $\mathcal{O}(n)$ Clifford gates and $\mathcal{O}(\log n)$ Clifford-depth [cf. Appendix A.3]. The circuit therefore has Clifford- and T-count $\mathcal{O}(kn)$, Clifford-depth $\mathcal{O}(k \log^2 n)$, and T-depth $\mathcal{O}(k \log n)$ in all cases. It is clear from Eq. (15) that the selection register comprises $k\lceil \log n \rceil + \mathcal{O}(1)$ qubits, and that no ancillae are required. For practical applications, k is usually a constant independent of n, in which case the total gate count is $\mathcal{O}(n)$ and the Clifford-depth and T-depth are $\mathcal{O}(\log^2 n)$ and $\mathcal{O}(\log n)$, respectively. The exact constant factors[7] in the T complexity can be directly calculated using Table 1.

Acknowledgments

---

[7]As a comparison to previous work, the implementations of controlled-SELECT(H) in Ref. [14] for the molecular electronic structure Hamiltonian and for the planar Fermi-Hubbard Hamiltonian have T-counts of $12n + \mathcal{O}(\log n)$ and $10n + \mathcal{O}(\log n)$, respectively, and T-depth $\mathcal{O}(n)$. In our framework, controlled-SELECT(H) can be implemented for both Hamiltonians using the same circuit, and this circuit would have T-count $64n + \mathcal{O}(1)$ and T-depth $64\lceil \log n \rceil + \mathcal{O}(1)$. Hence, the exponential improvement in overall circuit depth comes at the cost of only a modest constant-factor increase in the T-count.

A Gadgets

A.1 SwapUp

Ref. [31] provides an ancilla-free, logarithmic-depth circuit for the SwapUp operator defined by Eq. (8). As the ability to implement SwapUp using only $\mathcal{O}(\log^2 n)$ Clifford-depth and $\mathcal{O}(\log n)$ T-depth is essential for proving our main result, we summarise the construction here.

First, Ref. [31] observes that for any self-inverse unitary V (on any number of qubits), the multi-target controlled-V gate $\mathrm{C}(V)_m := |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes V^{\otimes m}$ on m registers can be implemented by a circuit of the form

$$(16)$$

where $\begin{array}{c}\text{\small(controlled-}V\text{ symbol)}\end{array}$ indicates that a controlled-V gate is applied with any one of the qubits in the first register as the control qubit. (If m is even, the second wire and the topmost controlled-V gate in Eq. (16) are removed.) Thus, $\mathrm{C}(V)_m$ can be implemented without ancilla qubits using at most $2m$ controlled-V gates and $\mathcal{O}(m)$ CNOTs in such a way that the controlled-V gates are applied in 4 layers. Ref. [31] further shows that the multi-target CNOTs can be implemented in depth $\mathcal{O}(\log m)$.

Next, consider the following circuit for SwapUp, drawn for $n = 8$:

$$(17)$$

Here, $x_2 x_1 x_0$ is the binary representation of x, where x_0 is the least significant bit. The basic idea is that after the controlled-SWAP gates controlled on the qubit storing $|x_2\rangle$ are applied, the states $|\varphi_y\rangle$ with $y_2 = x_2$ are on the first four qubits of the second register (writing y in binary as $y_2 y_1 y_0$). After the controlled-SWAPs controlled on $|x_1\rangle$ are applied, the states $|\varphi_y\rangle$ with $y_2 = x_2$ and $y_1 = x_1$ are on the first two qubits, and finally the controlled-SWAP controlled on $|x_0\rangle$ ensures that $|\varphi_x\rangle$ ends up on the very first qubit. In general, if the target register has n qubits (where n is not necessarily a power of 2) and $x_{\lceil \log n \rceil - 1} \ldots x_1 x_0$ is the binary representation of x, $n - 2^{\lceil \log n \rceil - 1}$ controlled-SWAPs are controlled on $|x_{\lceil \log n - 1 \rceil}\rangle$, followed by 2^j controlled-SWAPs controlled on $|x_j\rangle$ for each $j \in \{0, \ldots, \lceil \log n \rceil - 2\}$ in descending order.

Note that for each $j \in \{0, \ldots, \lceil \log n \rceil - 1\}$, the controlled-SWAP gates controlled on $|x_j\rangle$ target disjoint pairs of qubits. Since SWAP is self-inverse, we can apply Eq. (16) with $V = \text{SWAP}$ and $m \le 2^j$ (specifically, $m = 2^j$ for $j \in \{0, \ldots, \lceil \log n \rceil - 2\}$ and $m = n - 2^{\lceil \log n \rceil - 1}$ for $j = \lceil \log n \rceil - 1$) to replace all of the controlled-SWAPs controlled on $|x_j\rangle$ in Eq. (17) with at most $2 \cdot 2^j$ controlled-SWAPs applied in 4 layers, along with $\mathcal{O}(2^j)$ CNOTs applied in $\mathcal{O}(j)$ layers. Summing over j, the total number of controlled-SWAPS is

$$2\left(n - 2^{\lceil \log n \rceil - 1}\right) + \sum_{j=0}^{\lceil \log n \rceil - 2} 2 \cdot 2^j = 2(n-1)$$

and these are applied in $4\lceil \log n \rceil$ layers. Each controlled-SWAP can be decomposed into one TOFFOLI and two CNOT gates, and each TOFFOLI has T-count 7 and T-depth 4. Therefore, SWAPUP has T-count $14(n-1)$ and T-depth $16\lceil \log n \rceil$. The total number of Clifford gates (including the multi-target CNOTs in Eq. (16) and the Cliffords in the decomposition of each controlled-SWAP) is $\mathcal{O}(n)$, and the total Clifford-depth is $\mathcal{O}(\log^2 n)$.

A.2 SWAPUP*

In subsection 2.2, we use the fact that a certain phase-incorrect version SWAPUP* of SWAPUP is less costly than SWAPUP to reduce the T-count and T-depth by constant multiplicative factors. SWAPUP* is obtained by replacing each controlled-SWAP gate in Eq. (17) with a phase-incorrect version of controlled-SWAP:
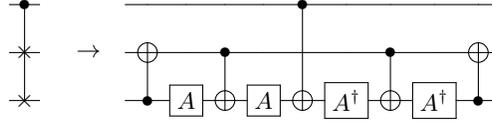
where $A \coloneqq e^{i(\pi/8)Y} = S^\dagger H T H S$ (here, H denotes the Hadamard gate). The circuit on the right-hand side implements an operator whose action on computational basis states differs from controlled-SWAP only in that $|100\rangle$ is mapped to $-|100\rangle$. Hence, since controlled-SWAP merely permutes the computational basis states, SWAPUP* acts the same as SWAPUP on computational basis states up to sign. As observed in Ref. [31], any sequence of these phase-incorrect controlled-SWAP operators that target disjoint pairs of qubits can be parallelised such that the A and $A^\dagger$ gates are applied in four layers, and the CNOTs in $\mathcal{O}(\log n)$ layers. It follows that SWAPUP* has T-count $4(n-1)$ and T-depth $4\lceil \log n \rceil$ (and Clifford-count $\mathcal{O}(n)$ and Clifford-depth $\mathcal{O}(\log^2 n)$).

A.3 LADDER

The LADDER operator, defined in Eq. (12) by a cascade of $n-1$ CNOT gates, can equivalently be implemented using a logarithmic-depth circuit, as pointed out by Gidney [36]. By Eq. (12), LADDER acts on computational basis states as

$$\text{LADDER} : |z_0\rangle|z_1\rangle \ldots |z_{n-1}\rangle \mapsto |z_0 \oplus \cdots \oplus z_{n-1}\rangle|z_1 \oplus \cdots \oplus z_{n-1}\rangle \ldots |z_{n-1}\rangle.$$

It is easily verified that the same transformation can be realised by arranging CNOT gates in a tree-like structure, shown below for $n = 8$:
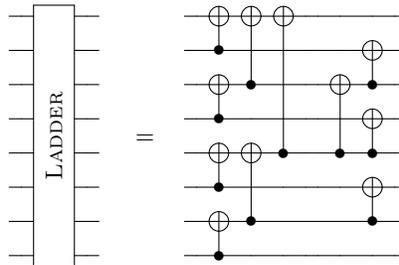
(For n that is not a power of 2, the circuit for LADDER can be obtained by starting with the circuit for the next largest power of 2, then removing all of the CNOTs supported on qubits that are out of range.) Thus, a circuit LADDER can be constructed using $\mathcal{O}(n)$ CNOTs applied in $\mathcal{O}(\log n)$ layers.

References

[1] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1), 2020. DOI: 10.1103/RevModPhys.92.015003.

[2] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, 119(19):10856–10915, 2019. DOI: 10.1021/acs.chemrev.8b00803.

[3] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020. DOI: 10.1021/acs.chemrev.9b00829.

[4] Daniel S. Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Physical Review Letters*, 83(24):5162–5165, 1999. DOI: 10.1103/PhysRevLett.83.5162.

[5] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995. URL https://arxiv.org/abs/quant-ph/9511026.

[6] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454 (1969):339–354, 1998. DOI: 10.1098/rspa.1998.0164.

[7] R. D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Physical Review Letters*, 101(13), 2008. DOI: 10.1103/PhysRevLett.101.130504.

[8] David Poulin, Alexei Kitaev, Damian S. Steiger, Matthew B. Hastings, and Matthias Troyer. Quantum algorithm for spectral measurement with a lower gate count. *Physical Review Letters*, 121(1), 2018. DOI: 10.1103/PhysRevLett.121.010501.

[9] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000. URL https://arxiv.org/abs/quant-ph/0001106.

[10] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge, 2003. URL https://arxiv.org/abs/quant-ph/0301023.

[11] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1), 2017. DOI: 10.1103/PhysRevLett.118.010501.

[12] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163. DOI: 10.22331/q-2019-07-12-163.

[13] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. Improved techniques for preparing eigenstates of fermionic Hamiltonians. *npj Quantum Information*, 4(1). DOI: 10.1038/s41534-018-0071-5.

[14] 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). DOI: 10.1103/PhysRevX.8.041015.

[15] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture, 2018. URL https://arxiv.org/abs/1805.00675.

[16] Mária Kieferová, Artur Scherer, and Dominic W. Berry. Simulating the dynamics of time-dependent Hamiltonians with a truncated Dyson series. *Physical Review A*, 99(4), 2019. DOI: 10.1103/PhysRevA.99.042314.

[17] Kianna Wan and Isaac Kim. Fast digital methods for adiabatic state preparation, 2020. URL https://arxiv.org/abs/2004.04164.

[18] Matthew B. Hastings. Lieb-Schultz-Mattis in higher dimensions. *Phys. Rev. B*, 69:104431, 2004. DOI: 10.1103/PhysRevB.69.104431.

[19] Yimin Ge, Jordi Tura, and J. Ignacio Cirac. Faster ground state preparation and high-precision ground energy estimation with fewer qubits. *Journal of Mathematical Physics*, 60(2):022202, 2019. DOI: 10.1063/1.5027484.

[20] Lin Lin and Yu Tong. Near-optimal ground state preparation, 2020. URL https://arxiv.org/abs/2002.12508.

[21] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information and Computation*, 12(11-12), 2012. DOI: 10.26421/qic12.11-12.

[22] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114 (9), 2015. DOI: 10.1103/PhysRevLett.114.090502.

[23] P. Jordan and E. Wigner. Über das Paulische Äquivalenzverbot. *Zeitschrift für Physik*, 47(9-10): 631–651, 1928. DOI: 10.1007/bf01331938.

[24] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018. DOI: 10.1073/pnas.1801723115.

[25] Vojtěch Havlíček, Matthias Troyer, and James D. Whitfield. Operator locality in the quantum simulation of fermionic models. *Physical Review A*, 95(3), 2017. DOI: 10.1103/PhysRevA.95.032332.

[26] Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002. DOI: 10.1006/aphy.2002.6254.

[27] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, 2012. DOI: 10.1063/1.4768229.

[28] Andrew Tranter, Sarah Sofia, Jake Seeley, Michael Kaicher, Jarrod McClean, Ryan Babbush, Peter V. Coveney, Florian Mintert, Frank Wilhelm, and Peter J. Love. The Bravyi-Kitaev transformation: Properties and applications. *International Journal of Quantum Chemistry*, 115(19): 1431–1441, 2015. DOI: 10.1002/qua.24969.

[29] Dan Stefan Eniceicu and Kianna Wan, unpublished.

[30] Ryan Babbush, Dominic W. Berry, and Hartmut Neven. Quantum simulation of the Sachdev-Ye-Kitaev model by asymmetric qubitization. *Physical Review A*, 99(4), 2019. DOI: 10.1103/PhysRevA.99.040301.

[31] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. Trading T-gates for dirty qubits in state preparation and unitary synthesis, 2018. URL https://arxiv.org/abs/1812.00954.

[32] Austin G. Fowler and Simon J. Devitt. A bridge to lower overhead quantum computation, 2012. URL https://arxiv.org/abs/1209.0510.

[33] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. DOI: 10.1103/PhysRevA.52.3457.

[34] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, Inc., Mineola, 1996.

[35] Trygve Helgaker, Poul Jørgensen, and Jeppe Olsen. *Molecular Electronic-Structure Theory*. John Wiley & Sons, Ltd, 2000. DOI: 10.1002/9781119019572.

[36] Craig Gidney, personal communication.