Some improvements to product formula circuits for Hamiltonian simulation

Andre Kornell and Peter Selinger

Dalhousie University

1 Introduction

We provide three improvements to the standard implementation of the ground state energy estimation algorithm via Trotter-Suzuki decomposition. These consist of smaller circuit templates for each Hamiltonian term, parallelization of commuting controlled rotations, and more efficient scheduling. These improvements may be regarded separately, and we anticipate that they may be combined with other improvements to the standard implementation.

Note that we are not proposing a new algorithm for ground state energy estimation, nor are we claiming that the Trotter-Suzuki product formula family of algorithms is the optimal choice for this problem. Rather, we are demonstrating the use of circuit optimization techniques to give a very efficient implementation of this particular algorithm.

2 Background

Ground state energy estimation (GSEE) is the problem of estimating the energy of the ground state of a molecule whose geometry is already known. Thus, the arrangement of the nuclei is given, and the task is to determine the minimum energy among the possible states of its electrons. This quantity has the obvious significance that it may be used to calculate the energy that is released during a chemical reaction. It plays a basic role in industrial research.

The nuclei of the molecule are assumed to be fixed, and the electrons are assumed to occupy only low-energy molecular orbitals $|\psi_0\rangle, \dots, |\psi_{m-1}\rangle$. The former assumption is known as the Born-Oppenheimer approximation, and it is conceptually justified by the fact that nuclei have far more mass than electrons. The latter assumption is conceptually justified by the fact that high-energy orbitals are negligibly occupied in the ground state of the molecule.

The state of the electrons is described by a unit vector in the fermionic Fock space, which has an orthonormal basis consisting of vectors of the form $|\psi_{k_1}\rangle \wedge \dots \wedge |\psi_{k_p}\rangle$, where $0 \leq k_1 < \dots < k_p \leq m-1$. Such a basis vector describes a state in which exactly the orbitals $|\psi_{k_1}\rangle, \dots, |\psi_{k_p}\rangle$ are occupied.

The creation operator $a_k^\dagger$ for orbital $|\psi_k\rangle$ is then defined by $a_k^\dagger|\phi\rangle = |\psi_k\rangle \wedge |\phi\rangle$, and the corresponding annihilation operator a_k is just the adjoint of $a_k^\dagger$.

The Hamiltonian of this system accounts for the attractive Coulomb force between each nucleus and each electron and the repulsive Coulomb force between each pair of electrons. The molecular orbitals are typically chosen to be real-valued wave functions, and consequently the Hamiltonian is of the form

$$H = \sum_{p,q} h_{pq}(a_p^\dagger a_q + a_q^\dagger a_p) + \sum_{p,q,r,s} h_{pqrs}(a_p^\dagger a_q^\dagger a_r a_s + a_s^\dagger a_r^\dagger a_q a_p),$$

where the coefficients h_{pq} and h_{pqrs} are real numbers and the indices p, q, r, s range over $\{0, \ldots, m-1\}$ in a way that makes the terms linearly independent.

The Jordan-Wigner transform yields an implementation of this Fock space on a quantum computer. Formally, it is a unitary operator $\mathcal{H} \to \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$, where $\mathcal{H}$ is the Fock space and there are m tensor factors of $\mathbb{C}^2$. The Jordan-Wigner transform maps each basis vector $|\psi_{k_1}\rangle \wedge \cdots \wedge |\psi_{k_p}\rangle$ to the basis vector $|b_0\rangle \otimes \cdots \otimes |b_{m-1}\rangle$, where $b_k = 1$ if the orbital $|\psi_k\rangle$ is occupied and otherwise $b_k = 0$. Under this transform, the annihilation operator a_k becomes the matrix $a_k = Z \otimes \cdots \otimes Z \otimes A \otimes I \otimes \cdots \otimes I$, where $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, and $A = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$.

The Jordan-Wigner transform depends nontrivially on the numbering of the molecular orbitals. Let U be the Jordan-Wigner transform for some initial ordering of the orbitals, and let V be the Jordan-Wigner transform after some permutation π. Then, the unitary matrix $VU^\dagger$ maps each standard basis vector $|b_0\rangle \otimes \cdots \otimes |b_{m-1}\rangle$ to $\pm|b_{\pi(0)}\rangle \otimes \cdots \otimes |b_{\pi(m-1)}\rangle$, where the sign depends on the parity of the order permutation of the occupied orbitals.

The time evolution of the electron system is implemented approximately via the fourth-order Trotter-Suzuki decomposition, which is a special case of the Lie product formula. If the Hamiltonian is the sum of two matrices, $H = H_1 + H_2$, then we approximate

$$e^{-iHt} \approx \prod_{n=1}^{N} S_2(\alpha/n) S_2(\beta/n) S_2(\alpha/n),$$

where $S_2(x) := e^{-iH_1tx/2} e^{-iH_2tx} e^{-iH_1tx/2}$ and the real constants α and β solve $2\alpha + \beta = 1$ and $2\alpha^3 + \beta^3 = 0$. In our case, the Hamiltonian H is the sum of $\mathcal{O}(m^4)$ terms, most of which are of the form $h_{pqrs}(a_p^\dagger a_q^\dagger a_r a_s + a_s^\dagger a_r^\dagger a_q a_p)$. The fourth-order Trotter-Suzuki decomposition has a straightforward generalization to this case.

This Hamiltonian simulation is used in combination with the quantum Fourier transform to estimate the ground state energy of the molecule. Intuitively, instead of simulating the evolution of the system for a single time quantity, we simulate the system for a superposition of different time quantities. Thus, we introduce an additional quantum system with some observable T, and the composite system evolves as $|\psi\rangle \otimes |T{=}t\rangle \mapsto e^{-iHt}|\psi\rangle \otimes |T{=}t\rangle$. This is implemented by introducing "precision qubits," which store the binary digits of t.

Over the course of the computation, the precision qubits and the orbital qubits become entangled. If the precision qubits are initialized in a superposition of eigenstates of T with equal weight and the orbital qubits are initialized in a state close to the ground state, then the reduced state of the precision qubits is likely to be $2^{-p/2}\sum_{k=1}^{2^p} e^{-iE_0 t_k}|T = t_k\rangle$, where E_0 is the ground state energy and p is the number of precision qubits. Furthermore, if $t_k = kt_1$, then the inverse Fourier transform of the reduced state will peak at $|\check{T} = s\rangle$, where s is the binary representation of E_0 after the radix point for a unit of energy that depends on t_1.

Thus, implementing the Hamiltonian simulation and the quantum Fourier transform and then measuring precision qubits yields a segment of the binary representation of E_0 that is determined by the number of precision qubits p and the smallest evolution time t_1.

3 Improved circuits for Hamiltonian simulation

3.1 The starting point

The straightforward implementation of this ground state energy estimation algorithm is inefficient in a number of respects. We highlight three of them and explain our improvements.

First, the terms of the Hamiltonian, such as $h_{pqrs}(a_p^\dagger a_q^\dagger a_r a_s + a_s^\dagger a_r^\dagger a_q a_p)$, are traditionally decomposed as linear combinations of Pauli operators [5]. This is because a matrix of the form $e^{i\theta\sigma}$ can be simply implemented for each angle θ and each Pauli operator σ. This decomposition increases the number of terms and hence the run time by a factor of eight. We avoid this factor of eight by avoiding this linear decomposition.

Second, in previous implementations of this algorithm, the Trotter-Suzuki decomposition was implemented in series, as they appear in the decomposition. However, many of the factors in the Trotter-Suzuki decomposition commute because they refer to disjoint sets of molecular orbitals. Furthermore, each factor of the decomposition naively corresponds to many controlled rotations when there is more than a single precision qubit, and these controlled rotations also commute. We implement these commuting rotations in a parallel, decreasing the run time by a factor of $O(m)$.

Third, in order to implement a commuting set of factors in the Trotter-Suzuki decomposition in parallel, we reorder the orbital qubits so as to group the qubits that correspond to each factor. It is necessary to reorder the qubits in this way because, while Hamiltonian terms that refer to disjoint sets of orbitals do commute, their natural circuit implementations may overlap as an artifact of the Jordan-Wigner transform. Naively, we would reorder the orbital qubits according to an arbitrary sequence of partitions into singletons, pairs, triples, and quadruples. Instead, we sequence the partitions by combining the circle method for round-robin tournaments with a pairing method suggested by Nazarov and Speyer [3].

3.2 Better circuit templates

In many situations, it is useful for the Hamiltonian to be given as a linear combination of Pauli operators. This leads to a natural but inefficient circuit representation. For example, consider the term

$$H_{0132} = h_{0132}(a_0^\dagger a_1^\dagger a_3 a_2 + a_2^\dagger a_3^\dagger a_1 a_0). \tag{1}$$

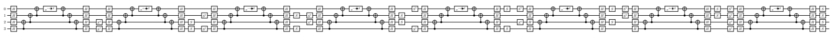
After applying the Jordan-Wigner transform, this becomes

$$H_{0132} = h_{0132}(A^\dagger \otimes A^\dagger \otimes A \otimes A + A \otimes A \otimes A^\dagger \otimes A^\dagger). \tag{2}$$

Note that A and $A^\dagger$ are not Pauli operators, but can be decomposed into Paulis as $A = \frac{1}{2}(X + iY)$ and $A^\dagger = \frac{1}{2}(X - iY)$. Substituting this into (2) and simplifying, we obtain the expression

$$H_{0132} = \frac{1}{8} h_{0132} \begin{pmatrix} X \otimes X \otimes X \otimes X & - X \otimes X \otimes Y \otimes Y \\ + X \otimes Y \otimes X \otimes Y & + X \otimes Y \otimes Y \otimes X \\ + Y \otimes X \otimes X \otimes Y & + Y \otimes X \otimes Y \otimes X \\ - Y \otimes Y \otimes X \otimes X & + Y \otimes Y \otimes Y \otimes Y \end{pmatrix}, \tag{3}$$

which is a linear combination of eight Paulis. Since the eight Pauli operators commute, the matrix exponential $e^{-iH_{0132}t}$ can be exactly written as a product of eight terms $e^{-\frac{i}{8}h_{0132}tX\otimes X\otimes X\otimes X} \ldots e^{-\frac{i}{8}h_{0132}tY\otimes Y\otimes Y\otimes Y}$. Each of these eight factors can then be easily written as a quantum circuit via suitable basis changes, resulting in the following circuit:

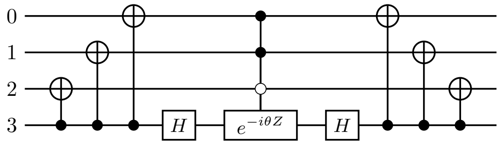


(4)

We can find an equivalent circuit that is eight times smaller by bypassing the Pauli decomposition and working directly from (2). Note that $A = |0\rangle\langle 1|$ and $A^\dagger = |1\rangle\langle 0|$, and therefore

$$H_{0132} = h_{0132}(|1100\rangle\langle 0011| + |0011\rangle\langle 1100|). \tag{5}$$

The key observation is that the matrix (5) is of rank 2, and has a much more compact circuit decomposition than (3), which is a sum of eight matrices of rank 16. Specifically, the operation $e^{-i\theta(|1100\rangle\langle 0011| + |0011\rangle\langle 1100|)}$ can be represented by the following circuit:

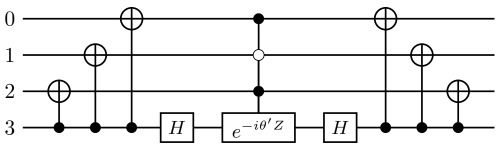


(6)

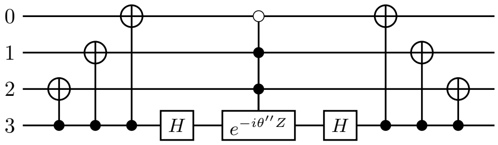
Compared to (4), the circuit (6) is much smaller. On the other hand, it contains a triply-controlled rotation, rather than an uncontrolled rotation, which a priori requires more gates. However, it turns out that the latter makes no difference:

in the context of quantum phase estimation, all of the rotations need to be controlled anyway, and we have efficient ways of implementing multiply-controlled rotations (see Section 3.3 below).

Another notable feature of (5) and (6) is that the related Hamiltonian terms H_{0231} and H_{0321} can be implemented by almost identical circuits. Indeed, these correspond, respectively, to the operations $e^{-i\theta'(|1010\rangle\langle 0101|+|0101\rangle\langle 1010|)}$ and $e^{-i\theta''(|1001\rangle\langle 0110|+|0110\rangle\langle 1001|)}$, and can be implemented by the following respective circuits:

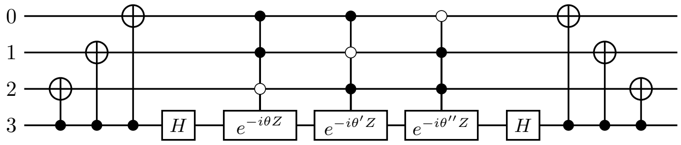


(7)



(8)

In fact, all three operations can be implemented with a single common basis change:



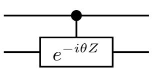
(9)

Moreover, the three controlled rotations in the center are diagonal operators and can be performed in parallel.

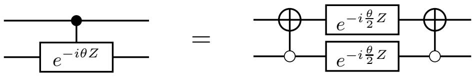
3.3 Parallel controlled rotations

Consider a number of z-rotations that are controlled by various qubits. Since all controlled z-rotations are diagonal gates in the computational basis, they all commute with each other, so in principle, they can all be performed in parallel. Here, we consider advantageous ways to actually perform them in parallel in a fault-tolerant regime.

We start with the simplest case of a z-rotation controlled by a single qubit.

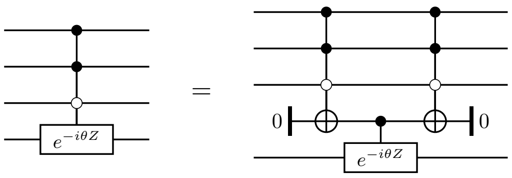


A good way to compile this is to decompose it into two uncontrolled rotations, as follows:

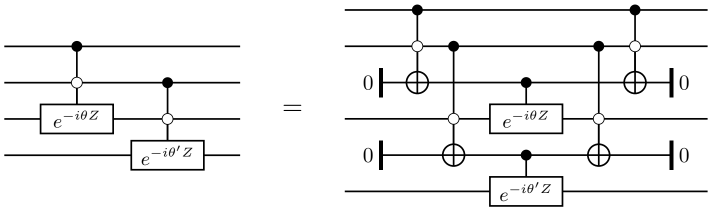


Each uncontrolled rotation can then be fault-tolerantly implemented, for example by the Ross-Selinger approximate synthesis algorithm [4] or by a fallback method [1].

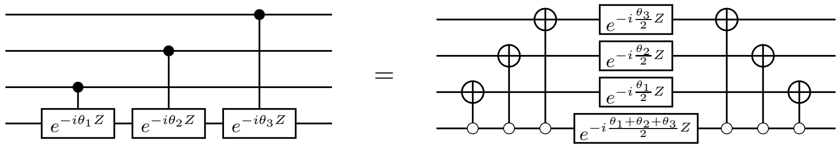
When we have more than one control, we can use Toffoli gates and an ancilla to first consolidate the multiple controls into a single one. The Toffoli gates require a number of T-gates, but that number is small compared to the number of T-gates required to implement the rotations themselves.



When we have several rotations targeted at different qubits, we can perform them in parallel, even if they share controls:



An interesting optimization is possible when we have n rotations that are targeted at the same qubit, possibly using different rotation angles, but controlled by different qubits. In this case, we only need $n + 1$ uncontrolled rotations:



$e^{-i\theta_1 Z}$
$e^{-i\theta_2 Z}$
$e^{-i\theta_3 Z}$
=
$e^{-i\frac{\theta_3}{2}Z}$
$e^{-i\frac{\theta_2}{2}Z}$
$e^{-i\frac{\theta_1}{2}Z}$
$e^{-i\frac{\theta_1+\theta_2+\theta_3}{2}Z}$

Combining the above methods allows us to perform any number of controlled z-rotations in parallel. Moreover, in the lattice surgery setting, fanout can be performed in a single time step, allowing ancillas to be copied instantaneously. Therefore, we can perform any number of controlled z-rotations in constant time (i.e., the time depends only on the approximation accuracy ϵ, but not on the number of parallel rotations).

3.4 Ski lift parallelization

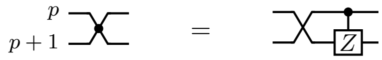
The Jordan-Wigner transform maps each annihilation operator a_p to the matrix $Z \otimes \cdots \otimes Z \otimes A \otimes I \otimes \cdots \otimes I$, where the matrix A is in position p. Recall that $A = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$. Consequently, a Hamiltonian term of the form $h_{pq}(a_p^\dagger a_q + a_q^\dagger a_p)$ for $p < q$ is mapped to a matrix that acts nontrivially on all orbital qubits r for

$p < r < q$. Thus, if $p_1 < p_2 < q_1 < q_2$, then the circuits implementing the time evolution for Hamiltonian terms $h_{p_1 q_1}(a^\dagger_{p_1} a_{q_1} + a^\dagger_{q_1} a_{p_1})$ and $h_{p_2 q_2}(a^\dagger_{p_2} a_{q_2} + a^\dagger_{q_2} a_{p_2})$ overlap. Overlapping circuits cannot be executed in parallel. However, if $p_1 < q_1 < p_2 < q_2$, then the circuits implementing the time evolution for these two Hamiltonian terms do not overlap, and hence they can be executed in parallel. We use fermionic swap operators to reduce the first case to the second.

For molecular orbitals $p \neq q$, we can define the swap operator $s_{p,q}$ on the electronic Fock space by $s_{p,q}|\psi_p\rangle = |\psi_q\rangle$, $s_{p,q}|\psi_q\rangle = |\psi_p\rangle$, and $s_{p,q}|\psi_r\rangle = |\psi_r\rangle$ for $r \neq p, q$. Of course, $s_{p,q} a_p s_{p,q} = a_q$. Thus, we can use these swap operators to implement a change of basis in which the circuits implementing the time evolution for Hamiltonian terms $h_{p_1 q_1}(a^\dagger_{p_1} a_{q_1} + a^\dagger_{q_1} a_{p_1})$ and $h_{p_2 q_2}(a^\dagger_{p_2} a_{q_2} + a^\dagger_{q_2} a_{p_2})$ do not overlap. If $p_1 < p_2 < q_1 < q_2$, then such a change of basis is clearly achieved by s_{p_2,q_1}. The same obstacle occurs for Hamiltonian terms involving three or four distinct qubits, and the same solution applies. We refer to such a change of basis as a fermionic permutation.

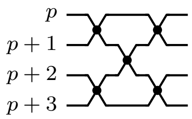
The Jordan-Wigner transform of the swap operator $s_{p,q}$ has a simple form. It is a self-adjoint unitary operator that commutes with a_r for $r \neq p, q$ and satisfies $s_{p,q} a_p s_{p,q} = a_q$. These two properties imply that $s_{p,p+1}$ has the following circuit implementation after the Jordan-Wigner transform:

$s_{p,p+1} \quad =$



Since the fermionic transposition operators $s_{p,p+1}$ satisfy the braid relations, we can define the more general $s_{p,q}$ in terms of them. For example, $s_{p,p+3}$ can be defined as follows:

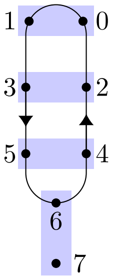
$s_{p,p+3} \quad =$



Every fermionic permutation is a composition of fermionic transpositions $s_{p,p+1}$. To decrease the run time, we should minimize the depth of these permutation circuits.

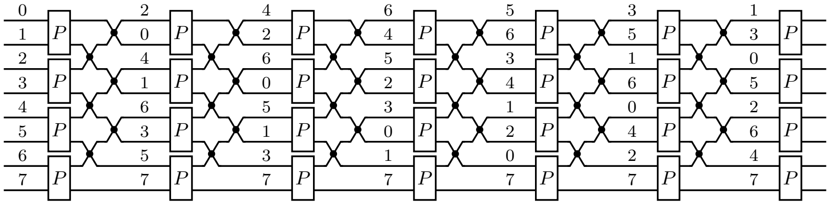
The terms of the Hamiltonian can be classified according to the number of distinct orbitals that they involve. That number may be $1, 2, 3, 4$. We refer to these terms as singleton, pair, triple, and quad terms, respectively. We can implement all the singleton terms in parallel. For pair terms, it is natural to use Kirkman's circle method for round-robin tournaments [2]. This method ensures that we implement these terms in a maximally time-efficient way and that we minimize the time spent on swapping the orbitals to arrange each new pairing.

Kirkman's circle method schedules a round-robin tournament for an even number of competitors by fixing one of those competitors and cycling the remaining competitors. We may imagine the circle method as a ski lift with one skier, who is not on the ski lift, at the bottom of the hill and the other skiers seated on the ski lift as it cycles. At each stage of the operation, we pair skiers seated across from each other, and we pair the skier seated at the bottom with the stationary skier:



Thus, if we have $2n$ skiers, the ski lift goes through $2n-1$ stages of operation, and we see n pairs at each stage. If we have $2n-1$ skiers, then we simply omit the stationary skier, and whoever is at the bottom of the ski lift will not be paired in that round.

Here is an example showing the 7 stages of the circle method for 8 orbitals:



We solve the scheduling problem for triple terms by a shrewd application of Möbius transformations over finite fields. We find the smallest prime $p \geq m-1$. Recall that $\mathbb{F}_p$ is the field of integers modulo p, and that a Möbius transformation is a permutation of $\mathbb{F}_p \cup \{\infty\}$ of the form $z \mapsto (az+b)/(cz+d)$ for parameters $a, b, c, d \in \mathbb{F}_p$. Each Möbius transformation of order three partitions $\mathbb{F}_p \cup \{\infty\}$ into orbits, at most two of which are singletons. Furthermore, every subset of cardinality three is an orbit of exactly two such Möbius transformations, which are each other's inverses. Thus, if $m = p+1$, then we obtain a sequence of partitions of our orbital qubits into triples such that every triple is in exactly one partition. We obtain maximal parallelization for the implementation of these Hamiltonian terms. If $m \neq p+1$, then we simply ignore partition blocks that contain elements greater than $m-1$.

We solve the scheduling problem for quad terms by combining the ski lift method with Möbius transformations. The intuition is that each seat of the ski lift will now hold two skiers instead of just one. A single cycle thus implements all possible quadruples of orbital qubits that may be obtained from a single partition of the orbital qubits into pairs. We use Möbius transformations to find a small set of such partitions that implements every quadruple, applying an idea of Nazarov and Speyer [3]. Specifically, we use Möbius transformations of order two to obtain partitions of $\mathbb{F}_p \cup \{\infty\}$ into pairs. Each quadruple is implemented by partitions arising from three such Möbius transformations. By considering the composition of these three Möbius transformations, we conclude that it is sufficient to use only those Möbius transformations whose determinant $ad - bc$ is a quadratic residue modulo p. As in the case of triples, we ignore partition blocks that contain elements greater than $m - 1$ in the common case that $m \neq p + 1$.

4 Putting everything together

Figure 1 shows a singleton stage, pair stage, triple stage, and quad stage in the case of 8 orbitals and 2 precision bits. Figure 2 shows a ski-lift schedule for a single Trotter-Suzuki step for 8 orbitals, including the fermionic permutations. We did not show any precision bits in this schedule; each of the boxes marking singleton, pair, triple, and quad circuits must be expanded along the lines of Figure 1, and the precision bits enter the picture at that point.

Comparing our current implementation of a Trotter-Suzuki step for GSEE with our previous implementation from December 2022, for $m = 120$ orbitals and $b = 1$ precision bit, we find a decrease in circuit depth by a factor of 720. Of this, a factor of 8 is due to improved Hamiltonian circuits (Section 3.2), and a factor of 90 is due to ski-lift parallelization and parallel controlled rotations (Sections 3.3 and 3.4). The width of the circuit increases by a factor of approximately 2. This is due to the fact that our parallel scheduling makes use of qubits that were previously idle. When increasing the number of precision bits, the December circuit's length increased linearly with b while its width remained essentially constant, whereas in the current implementation, the width increases linearly with b while the length stays constant (assuming that sets of gates that can be done in constant time in lattice surgery are counted as constant depth).

See Section 4.1 of the Peraton Labs Month 15 Milestone Report for a more detailed accounting of time and space resources, which also takes improvements in lattice surgery compiling and hardware into account. It is worth noting that the "area increase" by a factor of 28 reported there in connection with the ski lift method is not primarily due to an increased number of qubits in the logical circuit, but to an increase in the number of magic state factories, which are required to support the faster consumption of magic states caused by the circuit's increased parallelism.

Acknowledgements

This research was supported under DARPA Research Contract HR001122C0066.

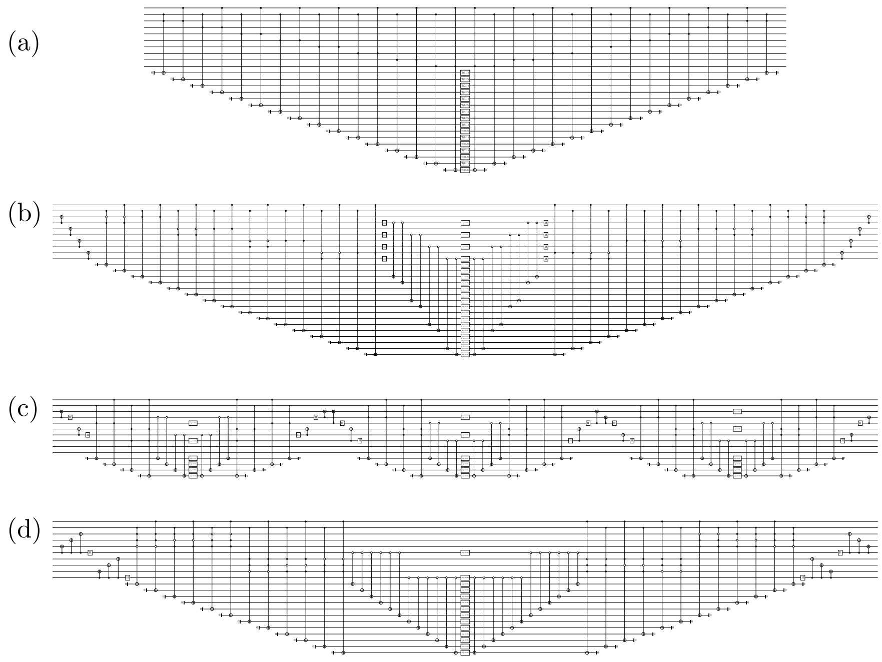


Figure 1: A typical singleton, pair, triple, and quad stage with $b = 2$ precision qubits and $m = 8$ orbital qubits. The topmost 2 inputs are the precision qubits and the bottommost 8 inputs are the orbital qubits. In (a), (b), and (d), note that all single-qubit rotations are done in parallel, using the method of Section 3.3 to decompose controlled rotations. The circuit (c) requires 3 rounds of rotations, because for each triple of orbitals (p, q, r), there are Hamiltonian terms of the forms $a_p^\dagger a_q^\dagger a_q a_r + a_r^\dagger a_q^\dagger a_q a_p$, $a_q^\dagger a_p^\dagger a_p a_r + a_r^\dagger a_p^\dagger a_p a_q$, and $a_p^\dagger a_r^\dagger a_r a_q + a_q^\dagger a_r^\dagger a_r a_p$. Since these do not commute with each other, they cannot be performed in parallel. By contrast, the circuit (d) requires only one round of rotations, because Hamiltonian quad terms on (p, q, r, s) are of the forms $a_p^\dagger a_q^\dagger a_r a_s + a_s^\dagger a_r^\dagger a_q a_p$, $a_p^\dagger a_r^\dagger a_q a_s + a_s^\dagger a_q^\dagger a_r a_p$, and $a_p^\dagger a_s^\dagger a_r a_q + a_q^\dagger a_r^\dagger a_s a_p$. These terms do commute with each other, and therefore can be performed in parallel. In (a)–(d), the multiply-controlled not gates that are used to prepare and uncompute the various ancillas can all be realized in constant time in the lattice surgery framework, i.e., in time that is independent of both b and m.

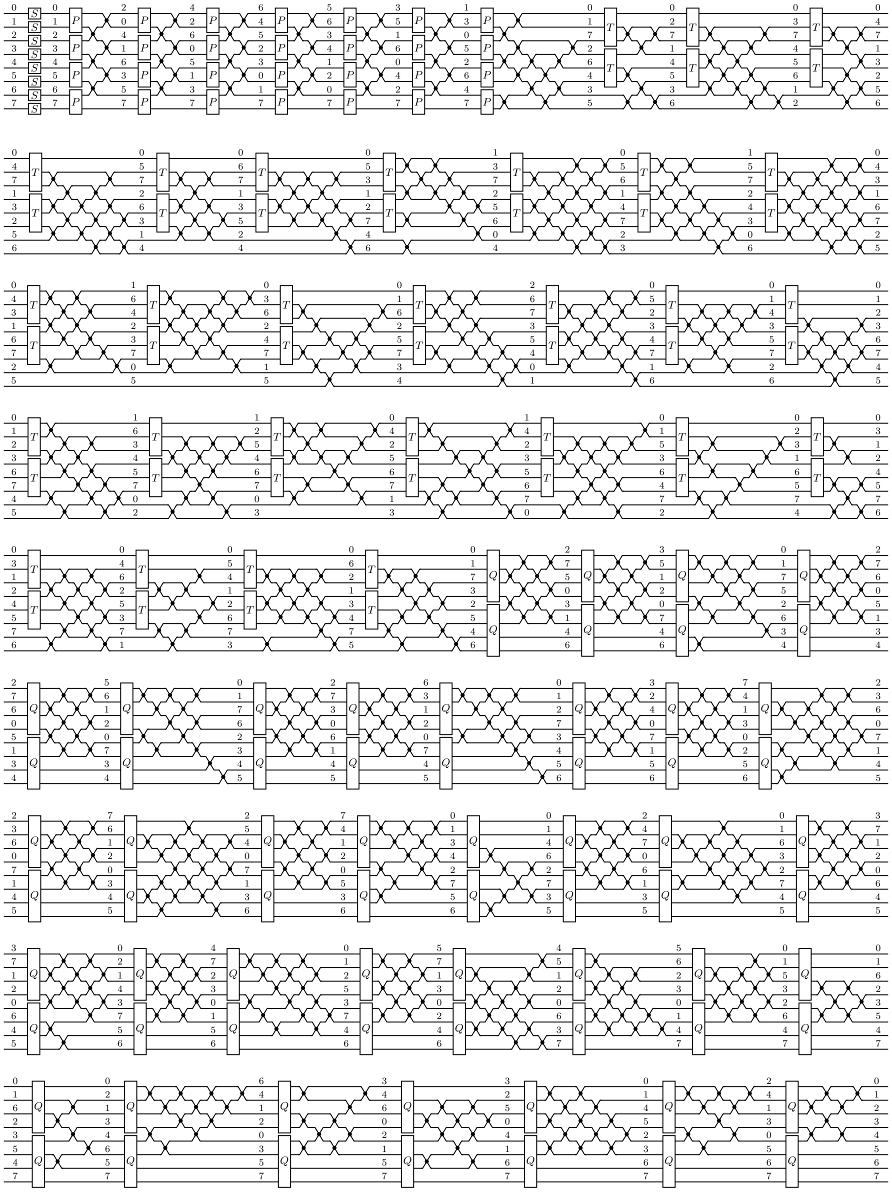


Figure 2: An example ski-lift schedule for 8 orbitals. This circuit should be read from top left to bottom right like a musical score. Singletons, pairs, triples, and quads are denotes S, P, T, and Q, respectively. Note that as the number of orbitals increases, quad stages will dominate the schedule. Also note that the transition from one quad stage to the next typically has a permutation depth of 4, but occasionally has a greater permutation depth. These greater-depth cases happen only $O(1/m)$ of the time, so the average permutation depth per stage is asymptotically constant.

References

[1] A. Bocharov, M. Roetteler, and K. M. Svore. Efficient synthesis of probabilistic quantum circuits with fallback. *Phys. Rev. A*, 91:052317, May 2015. Also available from arXiv:1409.3552.

[2] T. P. Kirkman. On a problem in combinations. *The Cambridge and Dublin Mathematical Journal*, 2:191–204, 1847.

[3] F. Nazarov and D. Speyer. Minimum number of pairings that make all quadruples. MathOverflow, January 2023. URL: mathoverflow.net/q/439372.

[4] N. J. Ross and P. Selinger. Optimal ancilla-free Clifford+T approximation of z-rotations. *Quantum Information and Computation*, 16(11–12):901–953, 2016. Also available from arXiv:1403.2975.

[5] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011. Also available from arXiv:1001.3855.