Applications of universal parity quantum computation

Michael Fellner,[1,2,*] Anette Messinger,[2] Kilian Ender,[1,2] and Wolfgang Lechner[1,2,†]

[1]*Institute for Theoretical Physics, University of Innsbruck, 6020 Innsbruck, Austria*
[2]*Parity Quantum Computing GmbH, 6020 Innsbruck, Austria*

(Received 19 May 2022; accepted 16 September 2022; published 27 October 2022)

We demonstrate the applicability of a universal gate set in the parity encoding, which is a dual to the standard gate model, by exploring several quantum gate algorithms such as the quantum Fourier transform and quantum addition. Embedding these algorithms in the parity encoding reduces the circuit depth compared to conventional gate-based implementations while keeping the multiqubit gate counts comparable. We further propose simple implementations of multiqubit gates in tailored encodings and an efficient strategy to prepare graph states.

DOI: 10.1103/PhysRevA.106.042442

I. INTRODUCTION

In recent decades, there has been an enormous effort to develop novel strategies for quantum computation [1–7], including measurement-based [8,9] or adiabatic [10] quantum computation, complementing the gate-based paradigm of quantum computation. In Ref. [11] we proposed universal quantum computation in the Lechner-Hauke-Zoller (LHZ) scheme [12] as a dual to the conventional gate model. Recent achievements in quantum hardware development on various qubit platforms [13–20] might soon allow for experimental realizations of well-known quantum algorithms for reasonable system sizes. Nevertheless, a fundamental challenge of state-of-the-art quantum devices remains the interqubit connectivity on quantum chips [21,22]. This is especially pressing because a long-range and dense (ideally all-to-all) connectivity is a crucial ingredient for many key quantum algorithms, unless algorithm-specific preprocessing steps are performed [23]. Unless efficient native implementations for long-range interactions as in, for example, the surface code [24,25] exist, the connectivity problem is often dealt with by utilizing resource-intensive SWAP gates which, apart from requiring error-prone two-qubit gates, render a parallelization of gates difficult. Although there exist several quantum routing techniques [26,27], this is in particular problematic for scalability of devices beyond the noisy intermediate-scale quantum (NISQ) era [28].

An alternative way to address the connectivity issue and, as a side effect, allow for the native implementation of higher-order interactions, was introduced with the parity encoding [12,29]. The parity encoding maps n logical qubits (with operators $\tilde{\sigma}$) to $K > n$ physical qubits (parity qubits, with operators σ), encoding the relative alignment (parity) along the z axis of two or more logical qubits such that for any state $|\psi\rangle$ in the code space

$$\tilde{\sigma}_z^{(i)}\tilde{\sigma}_z^{(j)}|\psi\rangle = \sigma_z^{(ij)}|\psi\rangle . \tag{1}$$

Here the superscripts correspond to qubit labels. In order to deal with the additional degrees of freedom in the physical Hilbert space, parity constraints of the form

$$C_l|\psi\rangle := \sigma_z^{(l_1)}\sigma_z^{(l_2)}\sigma_z^{(l_3)}\left[\sigma_z^{(l_4)}\right]|\psi\rangle = |\psi\rangle \tag{2}$$

are introduced as stabilizers of the code space, which is also referred to as the constraint-fulfilling subspace $\mathcal{H}_{\mathrm{CF}}$. The indices l_i represent pairs of logical qubits. In every constraint, each logical index must occur zero or an even number of times. The square brackets around $\sigma_z^{(l_4)}$ indicate that a constraint can contain either three or four qubits. The special case of the parity encoding involving all $n(n-1)/2$ two-body terms (parity qubits) for n logical qubits is known as the LHZ architecture [12]. It is possible to extend this by including physical qubits representing single logical qubits such that

$$\tilde{\sigma}_z^{(i)} = \sigma_z^{(i)}. \tag{3}$$

We refer to these qubits as data qubits in the following. The presence of data qubits in the parity encoding ensures that the physical qubits uniquely define the state of the logical qubits.[1] A variant of the LHZ architecture involving all data qubits has recently been shown to provide a universal gate set [11], based on the logical operators

$$\tilde{R}_x^{(i)}(\alpha) = \exp\left(-i\frac{\alpha}{2}\sigma_x^{(i)}\prod_{j<i}\sigma_x^{(ji)}\prod_{j>i}\sigma_x^{(ij)}\right), \tag{4}$$

$$\tilde{R}_z^{(i)}(\alpha) = \exp\left(-i\frac{\alpha}{2}\sigma_z^{(i)}\right) = R_z^{(i)}(\alpha), \tag{5}$$

$$\mathrm{C\tilde{P}}_\phi^{(i,j)} = R_z^{(i)}\left(\frac{\phi}{2}\right)R_z^{(ij)}\left(-\frac{\phi}{2}\right)R_z^{(j)}\left(\frac{\phi}{2}\right). \tag{6}$$

[*] m.fellner@parityqc.com
[†] wolfgang.lechner@uibk.ac.at

[1] Note that two-body parity qubits alone determine the logical state only up to a global spin flip.

Logical operators (with a tilde) act on the logical qubits defined in the constraint-fulfilling subspace and commute with all constraint operators. In contrast to that, physical operators (without a tilde) do not necessarily preserve the constraint-fulfilling subspace. The $\tilde{R}_x$ operators require chains of controlled-NOT (CNOT) gates due to the product of Pauli operators in the exponent (see, for example, Refs. [30–32] for further background on exponentiating products of Pauli matrices), while the other logical operators can be implemented with physical single-qubit operations only. The set of physical qubits that are involved in $\tilde{R}_x^{(i)}$ (i.e., the set of all physical qubits containing the logical index i) is referred to as the logical line associated with qubit (i).

In this work we study the implementation of several essential quantum algorithms in this scheme and find that, depending on the algorithm, the parity scheme can show an advantage in circuit depth or multiqubit gate count. In particular, we focus on quantum algorithms essential for Shor's factoring algorithm [2], the quantum Fourier transform (QFT) [31], and the quantum addition algorithm based on the QFT [33], as well as the implementation of Grover's diffusion operator [4]. Furthermore, we present a strategy to efficiently prepare graph states, which represent an important resource for measurement-based quantum computing [34].

II. COMMON GATES AND GATE SEQUENCES

A. Arbitrary single-qubit gates

Any single-qubit unitary U can be decomposed into rotations [31]

$$U = R_z(\alpha)R_x(\beta)R_z(\gamma), \tag{7}$$

with some angles α, β, and γ. We can thus construct any logical single-qubit gate using the operators defined in Eqs. (4) and (5) as

$$\tilde{U} = \tilde{R}_z(\alpha)\tilde{R}_x(\beta)\tilde{R}_z(\gamma). \tag{8}$$

The two $\tilde{R}_z$ rotations can be easily implemented in the LHZ scheme with physical rotations on the corresponding data qubits. The $\tilde{R}_x$ rotation requires a chain of CNOT gates along the logical line and a physical R_x rotation on one of its qubits, as discussed in Ref. [11]. If the physical R_x rotation is chosen to be performed on the respective data qubit, it can be recombined with the surrounding R_z rotations to the unitary U, now acting on the physical qubit, as shown in Fig. 1. This is possible because the CNOT gates commute with R_z gates acting on their control qubit. Hence, a single-qubit unitary on a logical qubit (i) can be implemented by performing the gate on the data qubit (i) and applying CNOT gates along the corresponding logical line.

Note that the single-qubit R_x rotation can be performed on a parity qubit instead of the data qubit of the line. In that case, the decomposed rotations cannot be recombined, but for $n > 4$, the additional R_z rotations can be performed in parallel with the CNOT chains. In combination with a partial parallelization of CNOT chains (see the Appendix of Ref. [35]), this leads to a minimal circuit depth for any single-qubit unitary of

$$d_U = 2\left\lceil \frac{n}{2} \right\rceil + 1. \tag{9}$$

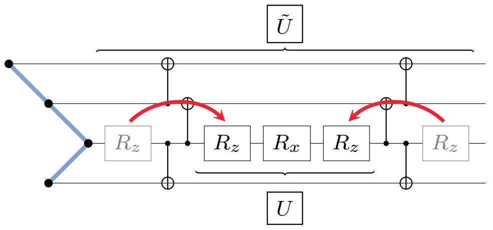

FIG. 1. Realization of a single-qubit unitary in universal parity quantum computation by exploiting Eq. (7). CNOT gates commute with R_z gates acting on their control qubit. The R_z gates on the data qubit can thus be moved through the CNOT sequence and recombined to the unitary U together with the R_x rotation. The unitary U acts locally on a physical qubit, while $\tilde{U}$ denotes a logical operation. The blue line represents the logical line corresponding to the qubit targeted by the operation.

This result can be seen from the implementation of an $\tilde{R}_x$ gate, as shown in Fig. 2 as a part of the logical CNOT gate. This implementation requires three single-qubit rotations (of which only one requires a separate step in depth) and $2(n-1)$ CNOT gates.

For products of single-qubit gates, it can be beneficial to apply the decoding sequence, perform the rotations on the respective data qubit, and encode again, which adds up to a circuit depth of $2n + 3$, as discussed in Ref. [11].

B. Two-qubit gates

While a logical $\tilde{\text{CP}}_\phi$ gate can be implemented in the parity encoding with only physical single-qubit gates according to Eq. (6), a logical CNOT gate requires two additional Hadamard gates on the target qubit,

$$\text{CNOT}^{(c,t)} = H^{(t)}\text{CZ}^{(c,t)}H^{(t)}. \tag{10}$$

In the decomposition (7), logical Hadamard gates require $\tilde{R}_x$ gates and thus CNOT chains along the logical line. An implementation of the decomposition (10) in the LHZ scheme is

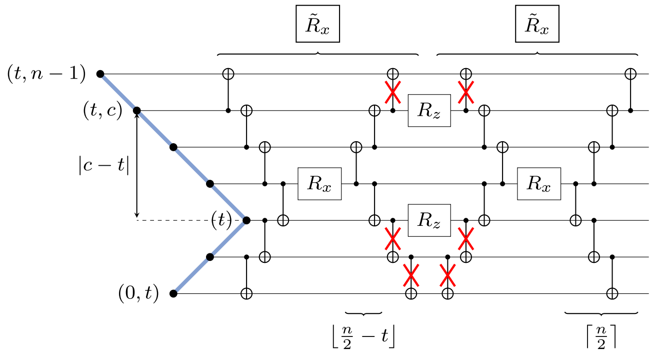

FIG. 2. Implementation of a CNOT gate in the LHZ architecture. The gate count depends on the number of logical qubits n and the distance $|c-t|$ between control (c) and target (t) qubits. CNOT gates marked with a red cross cancel; the blue line represents the logical line corresponding to the target qubit. The outer R_z rotations of the Hadamard gates are not shown; the R_z gates on qubit (t) have been merged.

depicted in Fig. 2 and results in a circuit depth of

$$d_{\text{CNOT}} = 2\left[\left\lceil \frac{n}{2}\right\rceil + \left\lfloor \max\left(\left|\frac{n}{2}-c\right|, \left|\frac{n}{2}-t\right|\right)\right\rfloor + k\right] + 3, \tag{11}$$

where c and t represent the index of the control and the target qubit, respectively, and

$$k = \begin{cases} 1 & \text{if } \left|\frac{n}{2}-c\right| = \left|\frac{n}{2}-t\right| \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The circuit depth d_{CNOT} corresponds to the depth of two $\tilde{R}_x$ gates minus the depth saved due to gate canceling, as depicted in Fig. 2. The number of required (physical) CNOT gates is $2(n-1+|c-t|)$, which is less than the CNOT count for two logical Hadamard gates, also due to canceling gates. Furthermore, seven single-qubit rotations are needed to construct a logical CNOT gate (six for the Hadamard parts and three for the CZ part, where the R_z rotations on the data qubit can be merged).

C. Intrinsic higher-order interactions

It is possible to encode the cumulative parity of multiple logical qubits (q_i) in a single parity qubit, with operator correspondences

$$\tilde{\sigma}_z^{(q_1)}\tilde{\sigma}_z^{(q_2)}\cdots\tilde{\sigma}_z^{(q_n)}\,|\psi\rangle = \sigma_z^{(q_1q_2\cdots q_n)}\,|\psi\rangle \tag{13}$$

for $|\psi\rangle \in \mathcal{H}_{\text{CF}}$. As an example, consider the three-body parity qubit (ijk). With this qubit, a logical three-body interaction of the form

$$\exp\left(i\phi\tilde{\sigma}_z^{(i)}\tilde{\sigma}_z^{(j)}\tilde{\sigma}_z^{(k)}\right) \tag{14}$$

is directly accessible via the physical single-qubit operation

$$\exp\left(i\phi\sigma_z^{(ijk)}\right). \tag{15}$$

Note that the placement of parity qubits $\sigma_z^{(q_1q_2\cdots q_n)}$ involved in more than two logical lines typically requires a tailored qubit layout [29]. This approach is in particular useful for solving combinatorial optimization problems with higher-order interactions (see Sec. III D).

D. Derived higher-order interactions

Alternatively, we can combine parity qubits with actual physical two-qubit gates to obtain logical multiqubit gates. Consider, for example, a physical controlled phase gate $\text{CP}_\phi^{(i,(jk))}$ between a parity qubit (jk) and a data qubit (i). For a computational basis state $|s_i\rangle\,|s_j\rangle\,|s_k\rangle$, $s_{\{i,j,k\}} \in \{0, 1\}$, this gate applies a phase if and only if $s_j \neq s_k$ and $s_i = 1$. Flipping the parity qubit (jk), this can be used to construct a logical 2-controlled phase gate (and in particular a CCZ gate for $\phi = \pi$) as

$$\text{CC}\tilde{\text{P}}_\phi^{(i,j,k)} = \text{C}\tilde{\text{P}}_{\phi/2}^{(i,j)}\text{C}\tilde{\text{P}}_{\phi/2}^{(i,k)}\sigma_x^{(jk)}\text{CP}_{\phi/2}^{(i,(jk))}\sigma_x^{(jk)}\text{P}_{-\phi/2}^{(i)}. \tag{16}$$

Here $\text{P}_{-\phi/2}^{(i)}$ is a single-qubit phase gate and up to a global phase equivalent to a single-qubit rotation $R_z^{(i)}(-\phi/2)$. The action of the operators in Eq. (16) on all computational basis states is given in Table I.

TABLE I. Phases acquired by application of the constituent operators for the $\text{CC}\tilde{\text{P}}_\phi^{(i,j,k)}$ gate on a computational basis state $|s_i\rangle\,|s_j\rangle\,|s_k\rangle$. For all basis states, the phases of the gate sequence sum up to the required phase. Physical operations, which are applied directly to the parity or data qubits, are written without a tilde, while a tilde denotes effective logical operations.

s_i	s_j	s_k	$\text{CC}\tilde{\text{P}}_\phi^{(i,j,k)}$	$\text{C}\tilde{\text{P}}_{\phi/2}^{(i,j)}$	$\text{C}\tilde{\text{P}}_{\phi/2}^{(i,k)}$	$\sigma_x^{(jk)}\text{CP}_{\phi/2}^{(i,(jk))}\sigma_x^{(jk)}$	$\text{P}_{-\phi/2}^{(i)}$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	0	0	$\phi/2$	$-\phi/2$
1	0	1	0	0	$\phi/2$	0	$-\phi/2$
1	1	0	0	$\phi/2$	0	0	$-\phi/2$
1	1	1	ϕ	$\phi/2$	$\phi/2$	$\phi/2$	$-\phi/2$

The Toffoli gate, an important constituent of many quantum circuits [2,36–38], can be decomposed as

$$\text{CCNOT}^{(i,j,k)} = H^{(k)}\text{CCZ}^{(i,j,k)}H^{(k)} \tag{17}$$

and, together with Eq. (16), reduced to two-qubit operators in the parity scheme. Here i and j denote the control qubits and k the target qubit. The circuit depth and the two-qubit gate count of this decomposition depend solely on the number of parity qubits in the logical line of the target qubit (i.e., the connectivity requirements of the logical qubit k). All required gates apart from the Hadamard gates on the target qubits can be implemented natively (i.e., without CNOT chains) as long as the data qubit (i) and the parity qubit (jk) are close enough on the chip to perform the physical controlled phase gate.

Similarly, a physical CP_ϕ gate between two parity qubits (ij) and (kl) effectively represents a four-qubit gate which adds a phase if and only if $s_i \neq s_j$ and $s_k \neq s_l$. In the square lattice implementation of the all-to-all connected graph, these derived multiqubit gates can be realized along the diagonal of a parity constraint. If the required connectivity is not available, the qubits can be rearranged to an algorithm-specific layout to provide the desired interactions.

E. Negative controls

For many applications [39,40], it is useful to invert control inputs of controlled multiqubit gates. That corresponds to applying a spin flip on the respective qubit before and after the controlled gate. In the parity encoding, a logical spin flip is realized by simultaneously flipping all physical spins along a logical line [11]. However, as the spin flips on parity qubits not involved in the operation cancel out, it is sufficient to only flip the data qubit corresponding to the desired negative control and the parity qubits involved in decomposition (16).

III. APPLICATIONS

A. Quantum Fourier transform

The quantum Fourier transform [31] is the quantum analog to the discrete Fourier transform and an important ingredient for many quantum algorithms such as quantum phase estimation [41] and, in further consequence, Shor's factoring

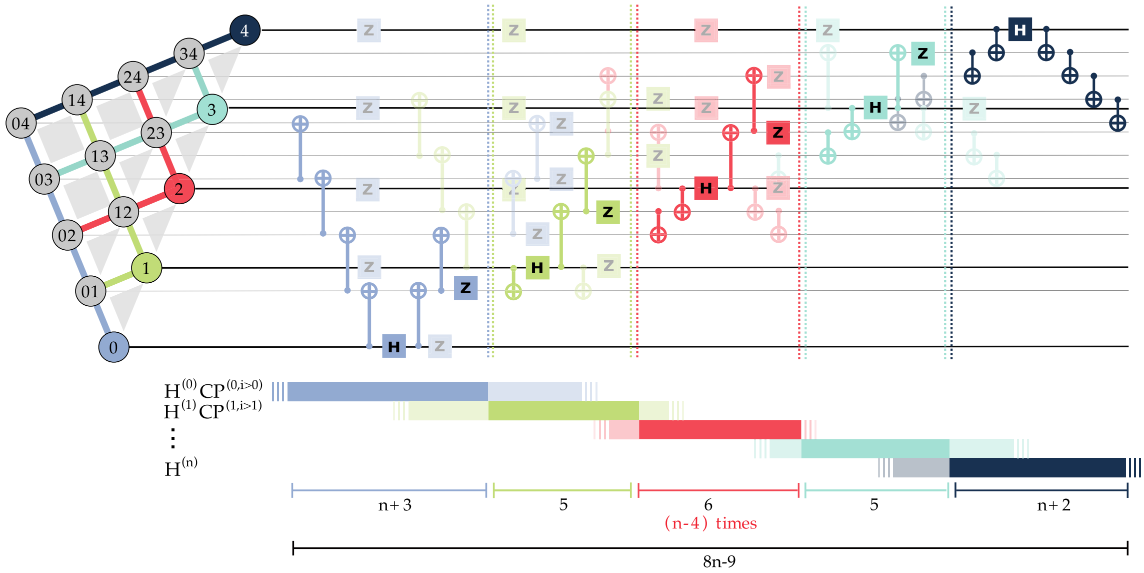

FIG. 3. Circuit for performing the QFT in the LHZ scheme. CNOT gates are required to perform the logical Hadamard gates, while single-qubit R_z rotations implement logical CPHASE gates. Gates which do not contribute to the circuit depth are faded out. For all but the first and the last line, increasing the system size would only add gates which do not contribute to the circuit depth. The diagram on the bottom visualizes the time steps occupied by each gate block of the QFT algorithm. In the layout on the left, lines depict logical lines and the gray squares and triangles represent parity constraints.

algorithm [2]. The unitary for the QFT on n qubits is given by

$$U_{\mathrm{QFT}} = \prod_{i=1}^{n} \left[H^{(i)} \prod_{j=i+1}^{n} \mathrm{C}R_{j-i+1}^{(i,j)} \right], \tag{18}$$

where the gates R_k are in this context defined as

$$R_k \equiv \mathrm{P}_{2\pi/2^k} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \tag{19}$$

and the superscripts denote the logical qubits the unitary is acting on. For an implementation of the QFT on a square lattice device in the standard gate model, SWAP gates are required to realize the controlled-phase gates. In contrast to that, in the parity architecture, these can be completely removed at the cost of requiring CNOT gates to perform the Hadamard gates according to the decomposition (7). The corresponding circuit for a possible low-depth implementation is shown in Fig. 3. The CNOT chains for implementing the logical Hadamard gates can be parallelized apart from a small contribution: For the logical qubit (0) (light blue line in the figure), the first CNOT chain along the line, the Hadamard gate on the data qubit and the second chain until qubit (0, 1) need to be performed before qubit (0, 1) is free to be used in the next line. In general, for any line i, only the gates between the crossing to the previous line [qubit $(i-1, i)$] and the crossing to the next line [qubit $(i, i+1)$] contribute to the circuit depth. The necessary logical CPHASE gates appear in the circuit as single-qubit R_z rotations (marked with Z). Note that many of them can be merged into other R_z rotations and are therefore not shown in Fig. 3. Following that procedure, the operations on the first qubit (light blue) block $n+3$ steps. The second and the $(n-1)$th ones take five steps and the last one (dark blue) $n+2$, because there is no CPHASE gate required. All other operations in between occupy six time steps. This construction results in a total circuit depth of $8n-9$.

A comparison of the required resources in the parity architecture and in the standard gate model is given in Table II. The circuit depth for the QFT in the LHZ scheme is up to a constant the same as in an all-to-all connected setup. This is remarkable in that the LHZ scheme can be implemented in current experiments and does not rely on impracticable hardware requirements.

B. Quantum addition

Another prominent problem in various quantum algorithms is quantum addition as the basic building block of algebraic manipulation of quantum registers. An efficient circuit for the addition of two quantum registers based on the QFT, proposed by Draper [33], performs the addition in the Fourier

TABLE II. Comparison of the required resources for the QFT implemented on an all-to-all connected device, a square lattice with nearest-neighbor interactions (requiring SWAP operations), and parity mapped on a square lattice with nearest-neighbor interactions. The numbers for the gate model implementations are taken from Ref. [23], while the numbers for the LHZ encoding are analytically deduced. All operations have been decomposed into single-body rotations R_x and R_z, H, and CNOT gates.

Resource	all-to-all	square lattice	LHZ
qubits	n	n	$\frac{1}{2}n(n+1)$
CNOT	$n(n-1)$	$\frac{3}{2}n(n-1)$	$2n(n-1)$
single-qubit	n^2	n^2	$\frac{1}{2}n(n+3)$
total gates	$n(2n-1)$	$\frac{1}{2}n(5n-3)$	$\frac{1}{2}n(5n-1)$
circuit depth	$8n-10$	$10n-13$	$8n-9$

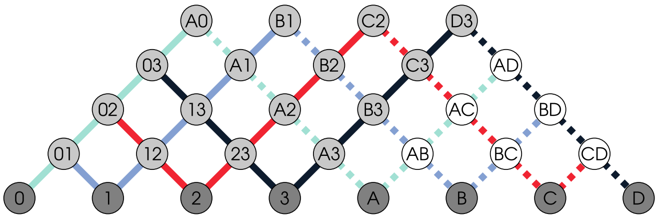

FIG. 4. Implementation proposal for the quantum addition of two registers R1 (labeled with numbers) and R2 (labeled with letters) in the LHZ scheme. The embedding consists of three standard LHZ schemes, two encoding the interactions within register R1 and R2, respectively, and one encoding the interregister interactions. The solid lines represent logical lines for register R1 and the dashed lines the logical lines for qubits in register R2. The qubits encoding interactions within register R2 (white) can be left out if no operations are necessary in register R2.

space with conditional rotation gates. Reference [42] suggests extensions of the resulting circuit that allow for computing the mean or weighted sum of a set of numbers, or even performing (modular) multiplication and exponentiation.

Quantum addition (modulo 2^n) of two n-qubit quantum states stored in registers R1 and R2 is realized by performing a QFT on one register (say, register R1), then performing controlled R_z rotations between the two registers, and finally applying the inverse QFT to the previously Fourier transformed register [33,42]. The corresponding quantum circuit is given by

$$U_{\rm QFT}^{\rm R1}\left[\prod_{i=0}^{n-1}\prod_{j=i}^{n-1} {\rm C}R_{j-i+1}^{(b_{n-j},a_{n-i})}\right]U_{\rm QFT}^{\rm R1\dagger}, \tag{20}$$

where a_i and b_i denote qubits of registers R1 and R2, respectively, and $U_{\rm QFT}^{\rm R1}$ denotes a QFT on register R1. An illustration of the circuit is provided in Ref. [33].

On a chip with all-to-all connectivity, the quantum addition algorithm can be implemented in logarithmic depth, neglecting the gates required for performing the necessary QFT steps. In the parity encoding, we can perform the logarithmic-depth part in a single time step and add the linear-depth QFT circuits, without requiring any SWAP gates.

As discussed in Sec. II, the controlled rotations can be implemented with single-qubit operations only, provided the necessary parity qubits are available. In order to fulfill that requirement, we suggest the qubit layout depicted in Fig. 4. The amount of physical qubits compared to the previously introduced architecture increases by $n(n+1)$ such that there are $3n(n+1)/2$ qubits in total.

If one of the registers does not require any computations within itself (for example, if it represents classical data), the parity qubits corresponding to interactions within that register are not necessary and the data qubits of the register can be added directly in continuation of the remaining logical lines, requiring $n(n+2)$ qubits in total. Note that either way, the extensions of logical lines hardly affect the circuit depth of the QFT part, as they do not cross any lines corresponding to the other register. Our architecture therefore allows us to perform the core quantum addition circuit in a single time step and the surrounding QFT parts with a depth linear in n, as discussed in Sec. III A. A list comparing required resources for the quantum circuit in the standard gate model and in the parity scheme is given in Table III.

TABLE III. Required resources for the core step of the QFT-based quantum addition algorithm on an all-to-all connected device and parity mapped on a square lattice with nearest-neighbor connectivity. The resources required for the involved QFT circuits are not included in this table. All operations have been decomposed into single-body rotations R_x and R_z, H, and CNOT gates.

Resource	all-to-all	LHZ
qubits	$2n$	$n(n+2)$ [a]
CNOT	$n(n+1)$	0
single-qubit	$\frac{1}{2}n(n+5)$	$n(n+2)$
total gates	$\frac{1}{2}n(3n+7)$	$n(n+2)$
circuit depth	$3\log_2(n)+1$	1

[a] If computations involving multiqubit gates are required within both registers, the number of qubits increases to $\frac{3}{2}n(n+1)$.

C. Multicontrolled gates and Grover's diffusion operator

A difficulty arising in many quantum algorithms is the implementation of multicontrolled quantum gates, as it requires a considerable number of nonlocal interactions and therefore SWAP gates. In the following, we present an implementation of the m-controlled phase gate on a parity-mapped architecture with $m+1$ logical qubits. In Grover's search algorithm [4], the multicontrolled phase gate is especially relevant for the implementation of the diffusion operator [43]. The diffusion operator on $m+1$ qubits corresponds to an m-controlled phase gate. We exploit the decomposition into 2-controlled phase and Toffoli gates [31], which are further decomposed according to Eq. (17). The decomposition is depicted in Fig. 5 and introduces $m-1$ ancilla qubits, labeled with capital letters. Here m is the number of control qubits. In the LHZ scheme, this enables us to implement an m-controlled phase gate ${\rm C}^m\tilde{\rm P}_\phi$ (and multicontrolled-NOT gates by using Hadamard gates) with gate resources scaling linearly with m and introducing $4m+3$ ancilla qubits in total. As discussed in Ref. [11], it is not necessary to use all parity qubits if the corresponding connections are not required. Figure 6 shows a possible qubit arrangement to implement the diffusion operator with the introduced decomposition for $m=4$. In this layout, the ${\rm CC}\tilde{\rm P}_\phi^{(i,j,k)}$ gates are decomposed such that the

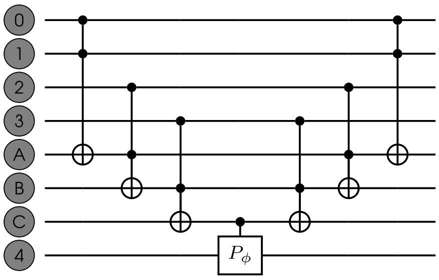

FIG. 5. Gate decomposition of a multicontrolled phase gate into Toffoli and ${\rm CP}_\phi$ gates using ancilla qubits A-C, following Ref. [31].

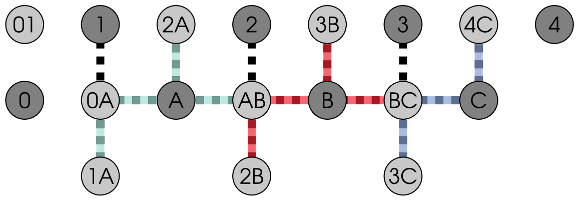

FIG. 6. Implementation layout for the n-controlled phase gate. Ancilla qubits are labeled with capital letters, parity qubits are shown in light gray, and data qubits are dark gray. Dotted lines indicate connectivity requirements between the physical qubits during computation. The parity encoding is reduced to the minimal required set of physical qubits such that only the ancilla qubits require (short) logical lines, shown by the colored lines.

physical $\mathrm{CP}_{\phi/2}^{(i,(jk))}$ gate occurring in Eq. (16) is performed between a problem qubit (i) and an ancilla qubit (jk). On the chip layout in Fig. 6, these correspond to the black dashed lines. The other CP_{ϕ} gates are performed via the protocol given in Eq. (6).

The Hadamard gates in Eq. (17), which only occur on ancilla qubits, can be implemented with eight CNOT gates each, as the logical lines (colored lines in Fig. 6) of ancilla qubits consist of only five qubits, independent of the number of control qubits. This allows for an implementation of the m-controlled-NOT gate scaling linearly in circuit depth as well as in gate count. We note that this example serves as a proof of principle for the implementation of Grover's diffusion operator in the parity encoding. While it cannot beat standard implementations using, e.g., Margolus construction of a Toffoli gate [6] in terms of quantum resources, the encoding offers additional flexibility, especially regarding chip layouts with sparse connectivity.

D. Optimization problems

Optimization problems can be formulated as energy minimization of Hamiltonians of the form

$$H_Z = \sum_i J_i\sigma_z^{(i)} + \sum_{i,j} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)} + \sum_{i,j,k} J_{ijk}\sigma_z^{(i)}\sigma_z^{(j)}\sigma_z^{(k)} + \cdots, \tag{21}$$

with the particular optimization problem being encoded in the prefactors J_i, J_{ij}, etc. Such optimization problems can be tackled on digital gate-based architectures using the quantum approximate optimization algorithm (QAOA) [44]. The QAOA variationally evolves the system with Hamiltonians H_X and H_Z as

$$|\psi\rangle = \prod_{j=1}^{p} e^{-i\beta_j H_X} e^{-i\gamma_j H_Z}, \tag{22}$$

where $H_X = \sum_i \sigma_x^{(i)}$ is a driver term to explore the search space. While the problem Hamiltonian H_Z can be implemented in the parity architecture in a single step, the driver Hamiltonian can be realized with a gate sequence of depth $7n - 8$ analogous to the circuit shown in Fig. 3 (leaving out the R_z rotations and replacing the physical Hadamard gates by parameterized R_x rotations). Such implementations and similar variants have been analyzed thoroughly in recent literature [35,45,46]. Typically, an additional Hamiltonian containing parity constraints is added to give an energy penalty to states which are not in the code space. These constraints simplify the operations required for the driver Hamiltonian. In particular, whenever all parity constraints are energetically penalized, the driver Hamiltonian reduces to a sum of single-body terms.

E. Preparation of graph states

Graph states are an important resource for measurement-based quantum computing [8,9] as well as for several error correction protocols [47,48]. Cluster states, corresponding to square grid graphs, can be prepared efficiently using nearest-neighbor interactions. The preparation of more arbitrary graph states, however, typically requires long-range interactions between many different qubits, as it involves applying a CZ gate for every edge in the graph [34]. In the parity encoding, arbitrary graph states containing n qubits can be created with a circuit depth of at most $n + 3$ and a CNOT-gate count of $2n(n - 1)$ as follows. We start by preparing n data qubits in the superposition state $|+\rangle$. We then apply the encoding sequence, in either the LHZ scheme or a reduced version of it, depending on the connectivity requirements of the respective graph, requiring a circuit of depth $n + 1$ or less (for reduced versions). Subsequently, we apply CZ gates to every pair of logical qubits to be connected in the graph, using parallel single-qubit gates on the physical qubits according to the decomposition (6). The result of this is a graph state encoded in the parity architecture. For measurement-based quantum computing, which involves measurements of the logical qubits (in arbitrary axes), an additional decoding step of depth $n + 1$ is required.

The number of parity qubits required for this procedure is in principle equal to the number of edges in the desired graph state. However, in some cases it can be useful to have additional ancillary qubits to simplify the encoding and decoding circuits.

IV. CONCLUSION AND OUTLOOK

In this work we have presented an approach to implement fundamental quantum algorithms for arbitrary system sizes, completely avoiding SWAP gates. A gate count analysis shows that our implementation has the potential to save multiqubit gates or reduce circuit depth for key constituents of Shor's algorithm. In addition, tailored parity encodings for particular algorithms can be constructed by utilizing the parity compiler [29]. On top of that, due to the redundant encoding of information, the parity encoding provides an intrinsic potential to detect and correct bit-flip errors.

One possible approach to error correction in the parity encoding is the belief propagation proposed in Ref. [49]. The effects of this fault tolerance on circuit execution results was not covered in this work.

Our proposal requires interactions between nearest neighbors only and can thus be implemented on various current NISQ devices with their natural interqubit connectivity. The proposal is also independent of the specific qubit platform. Suitable platforms are, for example, superconducting qubits [50–55], neutral atoms [13,14,56], or trapped ions [18,19,57,58]. In order to complement the bit-flip tolerance of the parity encoding, the use of noise-biased qubits [59–61] may be considered.

A combination of our findings regarding the QFT in the LHZ scheme with the achievements presented in Refs. [62,63] on programming arbitrary superposition states using quantum annealers may give rise to a QFT device without exponential gate overhead for the initial state preparation [31] and opens up a promising avenue for further research.

ACKNOWLEDGMENTS

We thank C. Ertler for valuable input on the graph state preparation and for numerous fruitful discussions. Work at the University of Innsbruck was supported by the Austrian Science Fund (FWF) through a START grant under Project No. Y1067-N27 and the Special Research Programme (SFB) “BeyondC: Quantum Information Systems Beyond Classical Capabilities”, Project No. F7108-N38. This work was supported by the Austrian Research Promotion Agency under Grant (FFG Project No. 892576, Basisprogramm).

[1] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, Proc. R. Soc. London **439**, 553 (1992).

[2] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. **26**, 1484 (1997).

[3] E. Bernstein and U. Vazirani, Quantum complexity theory, SIAM J. Comput. **26**, 1411 (1997).

[4] L. K. Grover, *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, 1996* (Association for Computing Machinery, New York, 1996), pp. 212–219

[5] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Experimental Realization of any Discrete Unitary Operator, Phys. Rev. Lett. **73**, 58 (1994).

[6] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, Phys. Rev. A **52**, 3457 (1995).

[7] M. Mosca, Quantum computer algorithms, Ph.D. thesis, University of Oxford, 1999.

[8] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, Phys. Rev. Lett. **86**, 5188 (2001).

[9] R. Raussendorf, D. E. Browne, and H. J. Briegel, Measurement-based quantum computation on cluster states, Phys. Rev. A **68**, 022312 (2003).

[10] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).

[11] M. Fellner, A. Messinger, K. Ender, and W. Lechner, Universal Parity Quantum Computing, Phys. Rev. Lett. **129**, 180503 (2022).

[12] W. Lechner, P. Hauke, and P. Zoller, A quantum annealing architecture with all-to-all connectivity from local interactions, Sci. Adv. **1**, e1500838 (2015).

[13] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum **4**, 327 (2020).

[14] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with Rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).

[15] I. Bloch, J. Dalibard, and W. Zwerger, Many-body physics with ultracold gases, Rev. Mod. Phys. **80**, 885 (2008).

[16] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner *et al.*, Probing many-body dynamics on a 51-atom quantum simulator, Nature (London) **551**, 579 (2017).

[17] J. I. Cirac and P. Zoller, Quantum Computations with Cold Trapped Ions, Phys. Rev. Lett. **74**, 4091 (1995).

[18] R. Blatt and C. F. Roos, Quantum simulations with trapped ions, Nat. Phys. **8**, 277 (2012).

[19] D. Kielpinski, C. Monroe, and D. J. Wineland, Architecture for a large-scale ion-trap quantum computer, Nature (London) **417**, 709 (2002).

[20] D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Côté, and M. D. Lukin, Fast Quantum Gates for Neutral Atoms, Phys. Rev. Lett. **85**, 2208 (2000).

[21] S. Hazra, A. Bhattacharjee, M. Chand, K. V. Salunkhe, S. Gopalakrishnan, M. P. Patankar, and R. Vijay, Ring-Resonator-Based Coupling Architecture for Enhanced Connectivity in a Superconducting Multiqubit Network, Phys. Rev. Appl. **16**, 024018 (2021).

[22] M. Tadokoro, T. Nakajima, T. Kobayashi, K. Takeda, A. Noiri, K. Tomari, J. Yoneda, S. Tarucha, and T. Kodera, Designs for a two-dimensional Si quantum dot array with spin qubit addressability, Sci. Rep. **11**, 19406 (2021).

[23] A. Holmes, S. Johri, G. G. Guerreschi, J. S. Clarke, and A. Y. Matsuura, Impact of qubit connectivity on quantum algorithm performance, Quantum Sci. Technol. **5**, 025009 (2020).

[24] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[25] M. Beverland, V. Kliuchnikov, and E. Schoute, Surface code compilation via edge-disjoint paths, PRX Quantum **3**, 020342 (2022).

[26] A. Bapat, A. M. Childs, A. V. Gorshkov, and E. Schoute, Advantages and limitations of quantum routing, arXiv:2206.01766.

[27] D. Devulapalli, E. Schoute, A. Bapat, A. M. Childs, and A. V. Gorshkov, Quantum routing with teleportation, arXiv:2204.04185.

[28] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[29] K. Ender, R. ter Hoeven, B. E. Niehoff, M. Drieb-Schön, and W. Lechner, Parity quantum optimization: Compiler, arXiv:2105.06233.

[30] A. Cowtan, S. Dilkes, R. Duncan, W. Simmons, and S. Sivarajah, in *Proceedings 16th International Conference on Quantum Physics and Logic, Orange, 2019*, edited by B. Coecke and M. Leifer (Open Publishing, Waterloo, 2019), Vol. 318, pp. 213–228.

[31] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, New York, 2011).

[32] M. Steudtner and S. Wehner, Quantum codes for quantum sim-

ulation of fermions on a square lattice of qubits, Phys. Rev. A **99**, 022308 (2019).

[33] T. G. Draper, Addition on a quantum computer, arXiv:quant-ph/0008033.

[34] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. Van den Nest, and H.-J. Briegel, in *Quantum Computers, Algorithms and Chaos, Proceedings of the International School of Physics "Enrico Fermi"*, edited by G. Casati, D. L. Shepelyansky, P. Zoller, and G. Benenti (IOS Press, Amsterdam, 2006), Vol. 162.

[35] K. Ender, A. Messinger, M. Fellner, C. Dlaska, and W. Lechner, Modular parity quantum approximate optimization, PRX Quantum **3**, 030304 (2022).

[36] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, Phys. Rev. A **54**, 147 (1996).

[37] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, and C. Monroe, Complete 3-qubit grover search on a programmable quantum computer, Nat. Commun. **8**, 1918 (2017).

[38] E. Dennis, Toward fault-tolerant quantum computation without concatenation, Phys. Rev. A **63**, 052314 (2001).

[39] M. Gado and A. Younes, Optimization of reversible circuits using Toffoli decompositions with negative controls, Symmetry **13**, 1025 (2021).

[40] M. Z. Rahman and J. E. Rice, in *Reversible Computation*, edited by S. Yamashita and S.-i. Minato (Springer, Cham, 2014), pp. 125–136.

[41] A. Kitaev, *Proceedings of the Electronic Colloquium on Computational Complexity* (Computational Complexity Foundation, Skillmann, 1996).

[42] L. Ruiz-Perez and J. C. Garcia-Escartin, Quantum arithmetic with the quantum Fourier transform, Quantum Inf. Process. **16**, 152 (2017).

[43] K. Zhang and V. E. Korepin, Depth optimization of quantum search algorithms beyond Grover's algorithm, Phys. Rev. A **101**, 032346 (2020).

[44] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[45] W. Lechner, Quantum approximate optimization with parallelizable gates, IEEE Trans. Quantum Eng. **1**, 3102206 (2020).

[46] A. Rocchetto, S. C. Benjamin, and Y. Li, Stabilizers as a design tool for new forms of the Lechner-Hauke-Zoller annealer, Sci. Adv. **2**, e1601246 (2016).

[47] R. Raussendorf and J. Harrington, Fault-Tolerant Quantum Computation with High Threshold in Two Dimensions, Phys. Rev. Lett. **98**, 190504 (2007).

[48] D. Schlingemann and R. F. Werner, Quantum error-correcting codes associated with graphs, Phys. Rev. A **65**, 012308 (2001).

[49] F. Pastawski and J. Preskill, Error correction for encoded quantum annealing, Phys. Rev. A **93**, 052325 (2016).

[50] A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, R.-S. Huang, J. Majer, S. Kumar, S. M. Girvin, and R. J. Schoelkopf, Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics, Nature (London) **431**, 162 (2004).

[51] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Charge-insensitive qubit design derived from the Cooper pair box, Phys. Rev. A **76**, 042319 (2007).

[52] A. A. Houck, J. A. Schreier, B. R. Johnson, J. M. Chow, J. Koch, J. M. Gambetta, D. I. Schuster, L. Frunzio, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Controlling the Spontaneous Emission of a Superconducting Transmon Qubit, Phys. Rev. Lett. **101**, 080502 (2008).

[53] R. Barends, J. Kelly, A. Megrant, D. Sank, E. Jeffrey, Y. Chen, Y. Yin, B. Chiaro, J. Mutus, C. Neill, P. O'Malley, P. Roushan, J. Wenner, T. C. White, A. N. Cleland, and J. M. Martinis, Coherent Josephson Qubit Suitable for Scalable Quantum Integrated Circuits, Phys. Rev. Lett. **111**, 080502 (2013).

[54] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[55] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, et al. *et al.* Strong Quantum Computational Advantage Using a Superconducting Quantum Processor, Phys. Rev. Lett. **127**, 180501 (2021).

[56] I. Cong, H. Levine, A. Keesling, D. Bluvstein, S.-T. Wang, and M. D. Lukin, Hardware-Efficient, Fault-Tolerant Quantum Computation with Rydberg Atoms, Phys. Rev. X **12**, 021049 (2022).

[57] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, Blueprint for a microwave trapped ion quantum computer, Sci. Adv. **3**, e1601540 (2017).

[58] A. C. Hughes, V. M. Schäfer, K. Thirumalai, D. P. Nadlinger, S. R. Woodrow, D. M. Lucas, and C. J. Ballance, Benchmarking a High-Fidelity Mixed-Species Entangling Gate, Phys. Rev. Lett. **125**, 080504 (2020).

[59] P. Aliferis, F. Brito, D. P. DiVincenzo, J. Preskill, M. Steffen, and B. M. Terhal, Fault-tolerant computing with biased-noise superconducting qubits: A case study, New J. Phys. **11**, 013061 (2009).

[60] S. Puri, L. St-Jean, J. A. Gross, A. Grimm, N. E. Frattini, P. S. Iyer, A. Krishna, S. Touzard, L. Jiang, A. Blais, S. T. Flammia, and S. M. Girvin, Bias-preserving gates with stabilized cat qubits, Sci. Adv. **6**, eaay5901 (2020).

[61] R. Lescanne, M. Villiers, T. Peronnin, A. Sarlette, M. Delbecq, B. Huard, T. Kontos, M. Mirrahimi, and Z. Leghtas, Exponential suppression of bit-flips in a qubit encoded in an oscillator, Nat. Phys. **16**, 509 (2020).

[62] L. M. Sieberer and W. Lechner, Programmable superpositions of Ising configurations, Phys. Rev. A **97**, 052329 (2018).

[63] C. Dlaska, L. M. Sieberer, and W. Lechner, Designing ground states of Hopfield networks for quantum state preparation, Phys. Rev. A **99**, 032342 (2019).