Universal programmable quantum circuit schemes to emulate an operator

Anmer Daskin, Ananth Grama, Giorgos Kollias, and Sabre Kais

Universal programmable quantum circuit schemes to emulate an operator

Anmer Daskin,[1] Ananth Grama,[1] Giorgos Kollias,[1] and Sabre Kais[2,3,a]

[1]*Department of Computer Science, Purdue University, West Lafayette, Indiana 47907, USA*
[2]*Department of Chemistry, Department of Physics and Birck Nanotechnology Center, Purdue University, West Lafayette, Indiana 47907, USA*
[3]*Qatar Environment and Energy Research Institute, Doha, Qatar*

Unlike fixed designs, programmable circuit designs support an infinite number of operators. The functionality of a programmable circuit can be altered by simply changing the angle values of the rotation gates in the circuit. Here, we present a new quantum circuit design technique resulting in two general programmable circuit schemes. The circuit schemes can be used to simulate any given operator by setting the angle values in the circuit. This provides a fixed circuit design whose angles are determined from the elements of the given matrix–which can be non-unitary–in an efficient way. We also give both the classical and quantum complexity analysis for these circuits and show that the circuits require a few classical computations. For the electronic structure simulation on a quantum computer, one has to perform the following steps: prepare the initial wave function of the system; present the evolution operator $U = e^{-iHt}$ for a given atomic and molecular Hamiltonian H in terms of quantum gates array and apply the phase estimation algorithm to find the energy eigenvalues. Thus, in the circuit model of quantum computing for quantum chemistry, a crucial step is presenting the evolution operator for the atomic and molecular Hamiltonians in terms of quantum gate arrays. Since the presented circuit designs are independent from the matrix decomposition techniques and the global optimization processes used to find quantum circuits for a given operator, high accuracy simulations can be done for the unitary propagators of molecular Hamiltonians on quantum computers. As an example, we show how to build the circuit design for the hydrogen molecule. © 2012 American Institute of Physics. [http://dx.doi.org/10.1063/1.4772185]

I. INTRODUCTION

The classical logical devices can be broadly categorized as fixed and programmable devices. As we understand from their names, the circuits in a fixed logic can only support one function which is determined at the time of manufacture. This cannot be changed at a later day. On the other hand, programmable devices such as programmable logic devices and field-programmable gate arrays are able to support an infinite number of functionalities since they can be reconfigured outside of the manufacturing environment. With this feature designers and programmers can run and simulate their test designs and algorithms.[1]

Quantum computing has become a huge new interdisciplinary area by providing different approaches and protocols to various subfields including: communication, encryption, global binary optimization (see adiabatic quantum computing[2]), linear algebra, and so on;[3–5] however, programmable quantum circuits and chip designs like those in classical computers have remained an open issue.

In the circuit model of quantum computing, unitary matrix operators represent the algorithms or some part of the computations.[6] Hence, one of the fundamental issues is to have a general purpose quantum circuit or a quantum chip that can realize different types of algorithms in a fast and

an efficient way. The possibility of designing universal quantum gate arrays as a general purpose quantum computer has been discussed in Ref. 7. It is shown that a gate array can be programmed to evaluate the expectation value of a given operator.[8] For the realization of a quantum gate, a cell structured quantum circuit design based on the activation and the deactivation of the gates on different qubits is proposed: It is shown that a combination of such cells can be used to realize a given quantum gate sequence.[9] Moreover, different schemes of general programmable universal quantum circuits are shown for two[10,11] and three qubits[12–14] found by applying different decomposition schemes to a given unitary operator. Based on the general two-qubit circuit design, a two-qubit quantum processor is experimentally realized.[15] However, the realization of a general quantum processor and a full-scale quantum computer is still an obstacle which requires new theoretical and experimental improvements.[16]

It is known that the realization of quantum logical operations can be simplified by using the higher dimensional Hilbert spaces.[16,17] In this paper, using ancilla qubits, we describe a new circuit design approach which produces two programmable quantum circuit designs. These can be further improved to design general large-scale quantum chips and programmable quantum gate arrays. The circuits also support simulation of non-unitary matrices. We also show the complexity analysis for the circuits: in terms of quantum complexity, they have about the same complexity as non-programmable designs which are generated by using

---

[a]Author to whom correspondence should be addressed. Electronic mail: kais@purdue.edu.

matrix decompositions in numerical linear algebra such as QR decomposition,[18] the quantum Shannon decomposition, the cosine-sine decomposition and some others[19,20] (see Ref. [20] for the comparison and the complexities of these methods). In terms of classical complexity, since angles for our programmable circuits can be determined simply individual matrix elements, the classical complexity is much simpler than the decomposition methods.

This paper is organized as follows: After giving the general simulation idea, the details of two circuit designs implementing this idea are presented. Then the complexity of the circuits are analyzed in terms of classical and quantum complexities. Finally, we discuss the circuit designs and possible future directions. In Appendix A, more computational details related to matrices are presented.

II. THE GENERAL SIMULATION IDEA

For a given real unitary $U^{N \times N}$ with $N = 2^n$, and n is the number of qubits, the relationship between the input $|\psi\rangle = \alpha_1|0\ldots0\rangle + \cdots + \alpha_N|1\ldots1\rangle$ and the output $|\varphi\rangle$ is defined as $|\varphi\rangle = U|\psi\rangle$ generating N states

$$U|\psi\rangle = \begin{pmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & & \vdots \\ u_{N1} & \cdots & u_{NN} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix}. \quad (1)$$

Any system of higher dimension (ancilla qubits are added to the original system) can also be used to generate this output on N chosen states with some normalization. Our goal is to create a matrix V (shown in Eq. (2)) which represents the system with the ancilla. We then modify the initial input $|0\rangle|\psi\rangle$ to this extended system V (the initial state of ancilla is taken as $|0\rangle$) by using quantum operations such that the application of V to this modified input $|\tilde{\psi}\rangle$ includes the output given in Eq. (1) with a normalization constant κ

$$V|\tilde{\psi}\rangle = \begin{pmatrix} V_1 & & & \\ & V_2 & & \\ & & \ddots & \\ & & & V_X \end{pmatrix} |\tilde{\psi}\rangle = \begin{pmatrix} \kappa\beta_1 \\ \vdots \\ \kappa\beta_2 \\ \vdots \\ \kappa\beta_N \\ \vdots \end{pmatrix}, \quad (2)$$

where each V_i has some distinct rows of U as their leading rows. Adding a sufficient number of ancilla qubits to control each V_i uniformly (as shown in Fig. 1) permits us to produce the circuit equivalent of matrix V in the above equation. If we assume that the first row of V_i is (or includes) the ith row
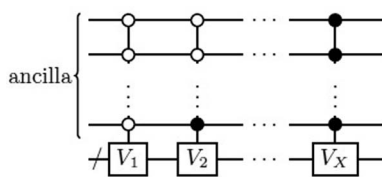
FIG. 1. The number of qubits on the ancilla determines the number of V_is and hence the size of V in Eq. (2).
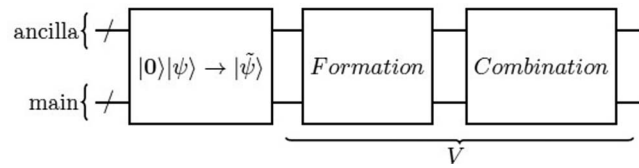
FIG. 2. Block circuit diagram to simulate U by modifying the input $|0\rangle|\psi\rangle$ to $|\tilde{\psi}\rangle$ and constructing V in two steps: the formation of the elements of U in V and bringing the same row elements in U to the first rows of V_is in V, combination. The necessary gates to form V and to also transform $|0\rangle|\psi\rangle$ to $|\tilde{\psi}\rangle$ will generate the circuit.

of U, then we need to use $(X = N)$, such V_i blocks as shown in Eq. (2).

The quantum operations to construct the matrix V and the operations to modify the input $|0\rangle \otimes |\psi\rangle$ form the circuit that simulates the given operator. That means, steps to form rows of U in V and also to transform $|0\rangle|\psi\rangle$ to $|\tilde{\psi}\rangle$ generate the general circuit design for the simulation of U. One way to formulate these steps and to build V_i matrices and the input $|\tilde{\psi}\rangle$ is as follows: First, the system is extended by adding auxiliary qubits. These ancilla qubits uniformly control different block quantum operations, V_is, on the main n qubits (in this paper, n or $(n + 1)$ number of auxiliary qubits are used). After the formation of all elements of U which we call the formation step, the same row elements of U are brought to the first row of each V_i which we call the combination step. The input is modified $(|0\rangle|\psi\rangle \rightarrow |\tilde{\psi}\rangle)$ by a small circuit such that $V|\tilde{\psi}\rangle$ produces an output which includes the normalized N states expected from the operation $U|\psi\rangle$. We call this step the input modification step. The measurement results for these N states exactly simulate $U|\psi\rangle$. The circuit design to be found with these steps can be drawn as a block circuit diagram (as shown in Fig. 2). This approach provides a new way to find circuit designs. Hence, we will describe two different programmable circuit schemes based on the block circuit in Fig. 2.

III. GENERATION OF PROGRAMMABLE CIRCUITS

A. The first circuit design

In this design, first we create all elements of U at the diagonal positions of V by using one rotation gate for each element of U, formation step. In the combination step, the elements on each ith row of U are collected in the first row of each V_i.

1. Formation step

In this step, the elements of U are tiled across the diagonal of a new higher-dimensional matrix V_f. This is a block diagonal matrix with 2×2 blocks across the diagonal. For each element of U, one rotation gate is used. The angular value for the gate is determined to form an element of U as its cosine value. Controlling such gates in a uniform binary coded fashion produces the matrix which has all elements of U on its diagonal

$$V_f = \begin{pmatrix} R_1 & & \\ & \ddots & \\ & & R_{N^2} \end{pmatrix}_{2N^2 \times 2N^2}, \quad R_j = \begin{pmatrix} c_j & s_j \\ -s_j & c_j \end{pmatrix}, \quad (3)$$

where $c_j = \cos(\arccos(u_j))$ generating the jth element of U and $s_j = \sin(\arccos(u_j))$. We use $(n+1)$ number of ancilla qubits to uniformly control each R_j, $1 \leq j \leq N^2$.

2. Combination step

To bring the same row elements of U to the first rows of the V_is, we need a quantum operation V_c which will produce the matrix $V = V_c V_f$ represented as

$$
\begin{pmatrix} K & & & \\ & K & & \\ & & \ddots & \\ & & & K \end{pmatrix}
\begin{pmatrix} c_1 & s_1 & & \\ -s_1 & c_1 & & \\ & & \ddots & \\ & & & \begin{matrix} c_{N^2} & s_{N^2} \\ -s_{N^2} & c_{N^2} \end{matrix} \end{pmatrix}
=
\begin{pmatrix} ku_{11} & \cdot & ku_{12} & \ldots & ku_{1N} \\ \vdots & \vdots & \vdots & & \vdots \\ & & & \ddots & \\ & & & & \begin{matrix} ku_{N1} & \cdot & ku_{N2} & \ldots & ku_{NN} \\ \vdots & \vdots & \vdots & & \vdots \end{matrix} \end{pmatrix},
$$

$$(4)$$

where K should have a form similar to the following matrix:

$$
K = \begin{pmatrix}
k & 0 & k & \ldots & 0 & k & 0 \\
0 & k & 0 & \ldots & k & 0 & k \\
k & 0 & k & \ldots & 0 & k & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & k & 0 & \ldots & k & 0 & k \\
k & 0 & k & \ldots & 0 & k & 0 \\
0 & k & 0 & \ldots & k & 0 & k
\end{pmatrix}_{2N \times 2N}.
$$

$$(5)$$

For a system with $(n+1)$ qubits, the single Hadamard gates on the first n qubits generate the above matrix with $k = \pm 1/\sqrt{2^n}$. Hence, V_c is the matrix form of this operation in the system with $(2n + 1)$ qubits where we apply the Hadamard gates to the $(n + 1)$st, nth, $\ldots$, 3rd, and 2nd qubits from the bottom in the circuit.

3. Input modification step

In the final matrix in Eq. (4), since the corresponding state for the rows which possess the elements of U with the normalization factor k are to be assigned as N chosen states simulating U, we should modify the input in such a way that the elements represented as "$\cdot$"s between ku_{ij} and $ku_{i(j+1)}$ are disregarded. That means the initial input should be transformed into $|\tilde{\psi}\rangle$ by a prior operation to the final matrix V so that the corresponding elements in the input to "$*$" elements are set to zero

$$
|\mathbf{0}\rangle |\psi\rangle \rightarrow |\tilde{\psi}\rangle
$$
$$
= [\alpha_1 \ \alpha_2 \ \ldots \alpha_N \ 0 \ldots 0]^T
$$
$$
\rightarrow [\kappa\alpha_1 \ 0 \ \kappa\alpha_2 \ldots 0 \ \kappa\alpha_N \ldots \ \kappa\alpha_1 \ 0 \ \kappa\alpha_2 \ \ldots \ 0 \ \kappa\alpha_N]^T,
$$

$$(6)$$

where κ is a normalization constant. It is easy to see that this modification can be succeeded by simple Hadamard gates on the first n qubits, and sequential swap operations between the $(n+1)$st and the remaining n qubits.

The equivalent circuit simulating any U is drawn in Fig. 3 for n qubit system by adding $n + 1$ ancilla qubits and replacing the block circuits in Fig. 2 with the explicit circuits found above.

At the end of this circuit, which can be decomposed into one- and two-qubit gates by using the decomposition technique discussed in Sec. IV, the following set of N states exactly simulates the given unitary U after normalization:

$$
|0\ldots000 - 0\ldots0\rangle,
$$
$$
|0\ldots010 - 0\ldots0\rangle,
$$
$$
\vdots
$$
$$
|1\ldots110 - 0\ldots0\rangle,
$$

$$(7)$$

where the dashes are used to separate the main and the ancillary.

In Appendix A, we give an example of the explicit matrices used for each step of the algorithm.

B. The second circuit design

In the first circuit design, the elements of U are initially formed on the diagonal of V by using uniformly controlled

FIG. 3. The first circuit design for a given general matrix: the initial Hadamards and the SWAPs are to modify the input, and the last Hadamards carry the elements to the first rows of the V_is (combination step). The uniformly controlled quantum gates in the middle form all elements of U on the diagonal of V (formation step).

rotation gates. Here, we take a group of elements from a row of U and create them as the leading row of small block matrices by preserving the ratios between the elements. Using a rotation gate for each two of these initial small blocks, we create larger block matrices which will have more elements of U in their first rows. This combination of steps is iteratively done until the final V_is with leading rows having the rows of U as in Eq. (2) are constructed. Since the final blocks, V_is, are $N \times N$, the matrix V is $N^2 \times N^2$; therefore, n qubits are needed for the ancilla. The input modification step follows the same idea as described for the first design.

1. Formation step

As stated above, instead of forming matrix elements at the diagonal positions by using a rotation gate for each ele-

ment of U, a group of elements is created in the first row of each block with the same ratio as those elements in the original matrix. For instance, if the initial blocks are of dimension 2×2, the first row implements two elements, u_{ij} and u_{ik}, of U. Thus, the ratio between the first element and the second element of a 2×2 block matrix is the same as u_{ij}/u_{ik} (since the block is 2×2, the elements of the block matrix are the cosine and sine values of an angle θ_x which provides the equality $\cos(\theta_x)/\sin(\theta_x) = u_{ij}/u_{ik}$). In our circuit designs, we will assume $k = j + 1$, and so the first row elements of each block implement the ratios of the elements in the same order as the original matrix. Therefore, if the first blocks are of dimension $d \times d$; the total number of initial blocks will be N/d since each block implements d number of elements. The following matrix represents the formation step for 2×2 initial blocks:

$$V_f = \begin{pmatrix} k_1^1 u_{11} & k_1^1 u_{12} \\ -k_1^1 u_{12} & k_1^1 u_{11} \\ & & \ddots \\ & & & k_{\frac{N}{2}}^1 u_{1N-1} & k_{\frac{N}{2}}^1 u_{1N} \\ & & & -k_{\frac{N}{2}}^1 u_{1N} & k_{\frac{N}{2}}^1 u_{1N-1} \\ & & & & & \ddots \\ & & & & & & k_{\frac{N}{2}}^N u_{NN-1} & k_{\frac{N}{2}}^N u_{NN} \\ & & & & & & -k_{\frac{N}{2}}^N u_{NN} & k_{\frac{N}{2}}^N u_{NN-1} \end{pmatrix}, \tag{8}$$

where k_j^is are the normalization constants, and u_{ij}s are the elements of U. The V_i block operations in Fig. 4 produce a matrix V_f with 4×4 block matrices on its diagonal.

2. Combination step

After the formation with ratios, blocks are combined using one rotation gate for each pair of two blocks so as to form new larger blocks with new normalization constants that preserve the original ratios of the elements. Each of these new blocks has twice as many elements as the former blocks. As an example, we will combine two 4×4 matrices located on the diagonal of the matrix V_8

$$V_8 = \begin{pmatrix} k_1 u_1 & \dots & k_1 u_4 & & & \\ \cdot & \cdots & \cdot & & & \\ \cdot & \cdots & \cdot & & & \\ \cdot & \cdots & \cdot & & & \\ & & & k_2 u_5 & \dots & k_2 u_8 \\ & & & \cdot & \cdots & \cdot \\ & & & \cdot & \cdots & \cdot \\ & & & \cdot & \cdots & \cdot \end{pmatrix}, \tag{9}$$

where and k_1 and k_2 are the normalization factors. The following matrix, V_{c_8}, can be used as a combination matrix to

generate an 8×8 larger block from the above pair of two 4×4 blocks:

$$V_{c_8} = \begin{pmatrix} c_x & 0 & 0 & 0 & s_x & 0 & 0 & 0 \\ 0 & c_x & 0 & 0 & 0 & s_x & 0 & 0 \\ 0 & 0 & c_x & 0 & 0 & 0 & s_x & 0 \\ 0 & 0 & 0 & c_x & 0 & 0 & 0 & s_x \\ -s_x & 0 & 0 & 0 & c_x & 0 & 0 & 0 \\ 0 & -s_x & 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & -s_x & 0 & 0 & 0 & c_x & 0 \\ 0 & 0 & 0 & -s_x & 0 & 0 & 0 & c_x \end{pmatrix}, \tag{10}$$

where $c_x = \cos(\theta_x)$, $s_x = \sin(\theta_x)$, and θ_x is an angle to achieve the required ratio. The matrix multiplication $V_{c_8} V_8$ produces a matrix with the leading row $[k_x u_1 \dots k_x u_8]$, where $k_x = \sin_x \times k_2$ and $k_x = \cos_x \times k_1$.

It is easy to see that the matrix V_{c_8} can be written as $R(2\theta_x) \otimes I \otimes I$. Hence, any such general combination matrix can be written as $R \otimes I^D$ where D is the size of the blocks to be combined by using V_c; and R is a general one qubit rotation gate. This means that for the blocks operating on c qubits, if we apply a rotation gate to the $(c + 1)$st qubit, it will be equivalent in matrix form to the operation $V_c V_{2^{c+1}}$. Hence, putting

FIG. 4. The second circuit with 4×4 initial blocks: The differently controlled quantum gates in the networks, after the V_i blocks, combine small blocks and build the $N \times N$ blocks at the end. The initial Hadamards are for the modification of the input. The V_i blocks are for the formation step.

single rotation gates on $(c+1)$st, $(c+2)$nd, ..., nth qubits generates an $N \times N$ matrix. Furthermore, by controlling each V_c operation (or equivalently single rotation gates, Rs) uniformly by the upper qubits in the circuit (see the uniformly controlled rotation gates located after the V_i block operations in Fig. 4), we can generate N such separate blocks and the following final matrix:

$$
V = \begin{pmatrix}
u_{11} & \cdots & u_{1N} & & & \\
\vdots & \vdots & \vdots & & & \\
& & & \ddots & & \\
& & & & u_{N1} & \cdots & u_{NN} \\
& & & & \vdots & \vdots & \vdots
\end{pmatrix}.
$$

(11)

Since the resulting rows in each block are unit vectors and have the same ratio as the row elements of U, they are equal to the corresponding rows of U. (The final normalization constants become equal to 1.)

For the general case, if the initial blocks are operating on the last c qubits, we need to use $N/2^c$ uniformly controlled rotation gates on each main qubit (excluding the last c qubits) in order to recursively combine small blocks. At the end, we have $N \times N$ blocks whose leading rows are the rows of U as shown in Eq. (11).

3. Input modification ($|0\rangle|\psi\rangle \rightarrow |\tilde{\psi}\rangle$)

Modification of the input $[\alpha_1 \alpha_2 \ldots \alpha_N 0 \ldots 0]^T$ as $[\kappa\alpha_1 \ldots \kappa\alpha_N | \kappa\alpha_1 \ldots \kappa\alpha_N | \ldots | \kappa\alpha_1 \ldots \kappa\alpha_N]^T$ with the normalization constant κ allows us to simulate U by using V in Eq. (11) on the chosen N states

$$
|0\ldots 000 - 0\ldots 0\rangle,
$$

$$
\vdots \qquad\qquad (12)
$$

$$
|1\ldots 111 - 0\ldots 0\rangle.
$$

This input with $\kappa = 1/\sqrt{2^N}$ can be produced by applying the Hadamard gates to all ancilla qubits at the beginning of the circuit.

Consequently, the general circuit design shown in Fig. 4 is obtained which is able to simulate any real unitary matrix. For more explicit matrix forms and illustrative details, please refer to Sec. A 2 of Appendix A.

IV. COMPLEXITY ANALYSIS OF THE CIRCUITS

In the cases of classical and quantum complexities of the circuits explained above, it is easy to see that they depend on mostly the costs of uniformly controlled networks such as the one in Fig. 5(a). Such a network controlled by k qubits can be decomposed in terms of 2^k CNOT gates and 2^k single rotation

(a)

(b)

FIG. 5. (a) A gray-coded multi-control network. (b) The decomposition of the gray-coded network in (a) into CNOT and single quantum gates.

gates.[12] For instance, the circuit as illustrated for $k = 2$ in Fig. 5(a) can be decomposed as in Fig. 5(b). The angle values in the decomposed circuit are found to be the solution of the system of the linear equation $M^k \boldsymbol{\theta} = \boldsymbol{\phi}$

$$M^k \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{2^k} \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{2^k} \end{pmatrix}, \tag{13}$$

where k is the number of control qubits in the network, and the entries of M are defined as

$$M_{ij} = (-1)^{b_{i-1} \cdot g_{j-1}}, \tag{14}$$

in which the power term is found by taking the dot product of the standard binary code of the index $i - 1$, b_{i-1}, and the binary representation of $j - 1$th gray-coded integer, g_{j-1}. Since M^k is a column permuted version of the Hadamard matrix, we see that M is unitary. Thus, $(M^k)^{-1} = 2^{-k}(M^k)^T$, and the new angle values in the decomposed circuit are the result of the mere matrix vector multiplication[12]

$$\boldsymbol{\theta} = 2^{-k}(M^k)^T \boldsymbol{\phi}. \tag{15}$$

A. The complexity of the first circuit design

1. The classical complexity

In the first circuit diagram (see Fig. 3), since there is only one such network, we need to multiply the $2^{2n} \times 2^{2n}$ matrix by the vector of dimension 2^{2n} constructed by taking the arc-cosines of every element of U. Hence, the classical complexity for the first circuit is $O(2^{4n})$. However, since M is the permuted version of the Hadamard matrix, by using the fast Hadamard transform,[21] which requires $O(N \log N)$ computations for the transform of a vector by the Hadamard matrix, this can be achieved in

$$O(2^{2n} \log(2^{2n})) = O(2n 2^{2n}). \tag{16}$$

2. The quantum complexity

The quantum complexity of the circuit is the number of gates required for the decomposition of the network, the combination of the blocks and the input modification: 2^{2n} CNOT, 2^{2n} single rotation, $2n$ Hadamard, and n SWAP gates.

B. The complexity of the second circuit

The classical and the quantum complexities for the second circuit are determined by the number of networks which are formed by putting the quantum gates in blocks controlled uniformly together as shown in Fig. 6 and by the combination steps. Since the quantum gates in different blocks with the same angles operate for every case of the control qubits, putting them together do not produce networks. Instead, they need to be applied only once such as the controlled X gates
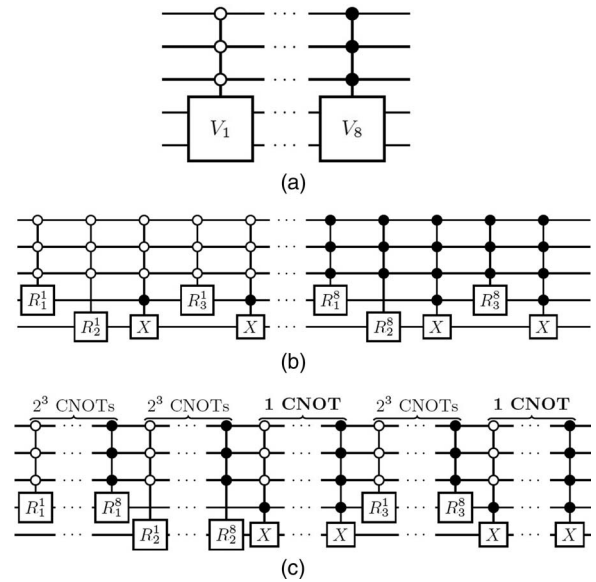
FIG. 6. The circuit in (a) with 4×4 initial blocks can be represented as in (b) by using the circuit given in Fig. (7). Without changing the order of the gates having the same control state, the gates can be moved to form uniformly controlled networks as in (c): If a gate has the same angle value for all control states such as the control X gates in the circuit, they are equal to a single gate (in the case of X gates in the circuit, only one CNOT is required).

shown in Fig. 6(c). Hence, if the initial blocks of $2^c \times 2^c$ (operating on c qubits) include m different quantum gates (the type of the gates are the same, but each requires different angles in different blocks such as R_1^1 and R_1^8 in Fig. 6), these blocks together produce m gray-coded networks controlled by 2^{2n-c} qubits.

In addition, in the combination step, we use binary coded networks on each main qubit excluding the last c qubits to produce $N \times N$ blocks. Thus, we will also have $n - c$ gray-coded networks for the combination step for which the numbers of control qubits go down by one from one combination step to another (or from one gray-coded network to another). The classical and the quantum complexity will be determined mostly by the decompositions of these $m + (n - c)$ networks.

1. Classical complexity

As mentioned above, in the formation step, the combination of decomposed block circuits together form m gray-coded networks for m different gate as represented for two-qubit blocks in Fig. 6. Hence, to find the decompositions of these networks as in Fig. 5(b) by the formula given in Eq. (15), m number of matrix-vector multiplications are needed: The dimensions of the matrices are $2^{2n-c} \times 2^{2n-c}$ and the dimensions of the vectors are 2^{2n-c}. Using the fast Hadamard transform, the complexity for this part is found to be $O_f = O(m(2n - c)(2^{2n-c}))$ instead of $O(m(2^{2n-c})^2)$ by the naïve matrix vector multiplication.

Furthermore, the combination step is the summation of the computations done for finding the angles of $(n - c)$ gray-coded networks (remember that the number of control qubits decreases by one from one network to another). This is equal

to $O((2^{2n-c-1})^2) + O((2^{2n-c-2})^2) + \ldots + O((2^{2n-c-n+c})^2)$ $= O(2^{4n-2c} - 2^{2n})$ by the naïve matrix vector multiplication. By the fast Hadamard transform, the complexity of the combination step is as follows:

$$
\begin{aligned}
O_c &= O((2n - c - 1)(2^{2n-c-1})) + O((2n - c - 2)(2^{2n-c-2})) \\
&\quad + \ldots + O(n2^n) \\
&= O(2 \times (1 - (2n - c)2^{2n-c-1} + (2n - c - 1)2^{2n-c})) \\
&\quad - O(2 \times (1 - n2^{n-1} + (n - 1)2^n)) \\
&= O((2n - c - 2)2^{2n-c} - (n - 2)2^n).
\end{aligned} \tag{17}
$$

Thus, while the total complexity by the naïve multiplication is

$$
\begin{aligned}
O(2^{4n-2c} - 2^{2n}) &+ O(m(2^{2n-c})^2) \\
&= O((m + 1)2^{4n-2c} - 2^{2n}),
\end{aligned} \tag{18}
$$

by the fast Hadamard transform, it is

$$
\begin{aligned}
O_f + O_c = O((m + 1)(2n - c)2^{2n-c} &- 2^{2n-c+1} \\
&- (n - 2)2^n)).
\end{aligned} \tag{19}
$$

2. The quantum complexity

In terms of the quantum complexity, the analysis follows the same structure: as mentioned, m different gates in the blocks on c qubits create m networks controlled by $2n - c$ qubits. The decomposition of these networks requires $m2^{2n-c}$ CNOT and the same number of single gates.

Since $n - c$ combinations ($n - c$ network) are necessary, the complexity of the combination step is the summation of $n - c$ terms: $2^{2n-c-1} + 2^{2n-c-2} + \cdots + 2^{2n-c-n+c}$ $= 2^{2n-c} - 2^n$.

Then the total CNOT complexity reads as

$$
2^{2n-c} - 2^n + m2^{2n-c} + \Phi = (m + 1)2^{2n-c} - 2^n + \Phi, \tag{20}
$$

where Φ represents the common gates in each block that needs to be run only once.

Example: As an example, the complexity of a general 4×4 block circuit can be found as follows: By using the Schmidt decomposition,[4] any 1×4 unit vector $\mathbf{u_x}$ can be decomposed as: $\mathbf{u_x} = \sum_{i=1}^{2} a_i \mathbf{v_i^1} \otimes \mathbf{v_i^2}$. Since V_1 and V_2 composed of v_i^1 and v_i^2 vectors are 2×2 unitary matrices, these matrices (with the elements $\cos_1$ and $\sin_1$ for V_1, and $\cos_2$ and $\sin_2$ for V_2) and the coefficients satisfying $|a_1|^2 + |a_2|^2 = 1$ can be considered as the rotation gates. For the coefficients, a_1 and a_2 are the cosine and the sine values of a rotation gate ($a_1 = \cos_a$ and $a_2 = \sin_a$). The resulting decomposition becomes equal to the following:

$$
\mathbf{u_x^T} = \begin{pmatrix} a_1\cos_1\cos_2 + a_2\sin_1\sin_2 \\ -a_1\cos_1\sin_2 + a_2\sin_1\cos_2 \\ -a_1\sin_1\cos_2 + a_2\cos_1\sin_2 \\ a_1\sin_1\sin_2 + a_2\cos_1\cos_2 \end{pmatrix}, \tag{21}
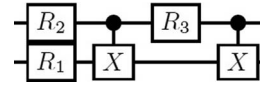$$

FIG. 7. Quantum circuit which is found by following the Schmidt decomposition and can generate any vector of dimension 4 as the first row of its matrix representation.

which requires three rotation gates in general. The circuit given in Fig. 7 forms any $\mathbf{u_x}$ as the leading row of its 4×4 matrix.

Therefore, taking this circuit to implement the blocks in Fig. 4 gives $c = \Phi = 2$, and $m = 3$; hence, the CNOT complexity of the whole circuit in Fig. 4 reads as $2^{2n} - 2^n + 2$. Also note that if the blocks in the circuit shown in Fig. 4 were of dimension 2×2, then the complexity would be $2^{2n} - 2^n$.

C. Comparison with the non-programmable circuit designs

The reported non-general circuit decompositions have the CNOT complexities ranging from $O(n^3 2^{2n})$ to the most efficient one $\frac{3}{4}2^{2n} - \frac{3}{2}2^n$. The proven lower bound for the CNOT complexity is $(2^{2n-2} - 3n/4 - 1/4)$ without using any auxiliary qubits.[20] Even though the circuit designs given in this paper are general and fixed size for any operator, their complexities are greater by roughly a factor of 2 compared to those non-programmable circuits. In addition, if we can make m less than or equal to 2^{c-2}, then we can also go below the lower bound. This is likely to happen because the common quantum gates in the blocks (as two CNOTs in 4×4 blocks) do not affect the upper bound of the complexity. Hence, by benefiting from this property, the lower bound complexity may be reduced with the use of higher Hilbert spaces.

V. DISCUSSION AND CONCLUSION

A. Programmable quantum chips

The circuit designs given here are independent of the type of operator; hence, they can be used to design general purpose quantum processors and quantum chips in which the angles are set by a preprocessing unit. They can also help in the design of possible quantum gate arrays.[7] In addition, because the circuit designs are highly dependent on the matrix elements, for the application specific circuits aimed to implement particular types of systems, any level of sparsity in the system may reduce the number of gates significantly in the general design; hence, more efficient quantum chips can be built for particular uses. For instance, if half of each of the row elements are zero in the given matrix, considering the first approach, the blocks at the end of the combination steps can be made to have the dimension $(N/2 \times N/2)$. Hence, this will lead the circuit to require fewer combination steps (the number of qubits in ancilla is reduced by one), which lowers both the classical and CNOT complexities and makes any possible fabrication easier.

B. Finding angles

In the case of finding the angle values on classical computers for a given unitary operator, the process can be parallelized conveniently to find the angles. For instance, the distribution of each row to the different cores may be one way of parallelization of the method. This can be further improved and designed in terms of more small blocks. And so the computation time to generate angles for both circuits can be very fast.

The combination procedure described for both circuit designing processes can be further improved to combine circuits for different unitary operations by considering them as initial blocks. One of the individual blocks used to generate a row of the given matrix can also be used as the state preparation circuit (for instance, Fig. 7) for an arbitrary circuit. Furthermore, the circuits generated by the first approach have high resemblance to the qubus quantum computer.[22] Similar ideas can be used to implement circuit design techniques for this type of quantum computers as well.

C. Complex cases

It is important to note that in this paper, even though real matrices are considered, it is straightforward to implement any complex case as well by considering each rotation gate as also being able to produce any complex element of a unitary matrix in the first circuit design. This may require more than one simple rotation gate, but it shall not increase the upper bound of the quantum complexity. However, the modification for the second circuit may not be as simple as for the first one: this may require additional gates during the combination and formation steps.

D. Simulation of molecular Hamiltonians

The exponential growth of computational cost with the number of atoms is a huge computational challenge for the exact quantum chemistry calculations. Even for a simple molecule like methanol, using only the 6-31G** basis for the valence electrons, there are 50 orbitals. The 18 valence electrons can be distributed in these orbitals in any way that satisfies the Pauli exclusion principle. This leads to about 10^{17} possible configurations making an exact or full configuration interaction calculation almost impossible on classical computers.[6] However, it has been shown that a quantum computer can be used to estimate the ground and excited state energies of molecules efficiently.[6, 23–32] For the electronic structure simulation on a quantum computer, one has to perform the following steps: prepare the initial wave function of the system; present the evolution operator $U = e^{-iHt}$ for a given atomic and molecular Hamiltonian H in terms of quantum gates array; and apply the phase estimation algorithm to find the energy eigenvalues.[32] Thus, for the simulation of a quantum system, it is necessary to find an equivalent quantum circuit to the unitary propagator of the Hamiltonian representing that system. The molecular electronic Hamiltonian, in the Born-Oppenheimer approximation, is described in the second

quantization form as[6, 16, 27]

$$\mathcal{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_s a_r, \qquad (22)$$

where the matrix elements h_{pq} and h_{pqrs} are the set of one- and two-electron integrals, and a_j and $a_j^\dagger$ are the spinless fermionic annihilation and creation operators. Let the set of single-particle spatial functions constitute the molecular orbitals $\{\varphi(\mathbf{r})\}_{k=1}^M$ and the set of spin orbitals $\{\chi(\mathbf{x})\}_{p=1}^{2M}$ be defined with $\chi_p = \varphi_i \sigma_i$ and the set of space-spin coordinates $\mathbf{x} = (\mathbf{r}, \omega)$ where σ_i is a spin function. The one-electron integral is defined as[6]

$$h_{pq} = \int d\mathbf{x} \chi_p^*(\mathbf{x}) \left(-\frac{1}{2}\nabla^2 - \sum_\alpha \frac{Z_\alpha}{r_{\alpha\mathbf{x}}} \right) \chi_q(\mathbf{x})$$

$$= \langle \varphi_p \mid H^{(1)} \mid \varphi_q \rangle \delta_{\sigma_p \sigma_q}, \qquad (23)$$

and the two electron integral is

$$h_{pqrs} = \int d\mathbf{x}_1 d\mathbf{x}_2 \frac{\chi_p^*(\mathbf{x}_1)\chi_q^*(\mathbf{x}_2)\chi_s(\mathbf{x}_1)\chi_r(\mathbf{x}_2)}{r_{12}}$$

$$= \langle \varphi_p \mid \langle \varphi_q \mid H^{(2)} \mid \varphi_r \rangle \mid \varphi_s \rangle \delta_{\sigma_p \sigma_q} \delta_{\sigma_r \sigma_s}, \qquad (24)$$

where $r_{\alpha\mathbf{x}}$ is the distance between the αth nucleus and the electron, r_{12} is the distance between electrons, ∇^2 is the Laplacian of the electron spatial coordinates, and $\chi_p(\mathbf{x})$ is a selected single-particle basis: $\chi_p = \varphi_p \sigma_p$, $\chi_q = \varphi_q \sigma_q$, $\chi_r = \varphi_r \sigma_r$, and $\chi_s = \varphi_s \sigma_s$.

To describe the hydrogen molecule in minimal basis which is the minimum number of spatial functions required to describe the system, one spatial function is needed per atom denoted φ_{H1} and φ_{H2}. The molecular spatial-orbitals are defined by symmetry: $\varphi_g = \varphi_{H1} + \varphi_{H2}$ and $\varphi_u = \varphi_{H1} - \varphi_{H2}$; which correspond to four spin orbitals: $|\chi_1\rangle = |\varphi_g\rangle|\alpha\rangle$, $|\chi_2\rangle = |\varphi_g\rangle|\beta\rangle$, $|\chi_3\rangle = |\varphi_u\rangle|\alpha\rangle$, and $|\chi_4\rangle = |\varphi_u\rangle|\beta\rangle$. The STO-3G basis is used to evaluate the spatial integrals of the Hamiltonian which is defined as $\mathcal{H} = H^{(1)} + H^{(2)}$, where since $h_{pqrs} = h_{pqsr}$, $H^{(1)}$ and $H^{(2)}$ are simplified as[6, 16, 27]

$$H^{(1)} = h_{11}a_1^\dagger a_1 + h_{22}a_2^\dagger a_2 + h_{33}a_3^\dagger a_3 + h_{44}a_4^\dagger a_4 \qquad (25)$$

and

$$\begin{aligned}
H^{(2)} &= h_{1221}a_1^\dagger a_2^\dagger a_2 a_1 + h_{3443}a_3^\dagger a_4^\dagger a_4 a_3 + h_{1441}a_1^\dagger a_4^\dagger a_4 a_1 \\
&+ h_{2332}a_2^\dagger a_3^\dagger a_3 a_2 + (h_{1331} - h_{1313})a_1^\dagger a_3^\dagger a_3 a_1 \\
&+ (h_{2442} - h_{2424})a_2^\dagger a_4^\dagger a_4 a_2 + (h_{1423})(a_1^\dagger a_4^\dagger a_2 a_3 \\
&+ a_3^\dagger a_2^\dagger a_4 a_1) + (h_{1243})(a_1^\dagger a_2^\dagger a_4 a_3 + a_3^\dagger a_4^\dagger a_2 a_1). \quad (26)
\end{aligned}$$

The spatial integral values evaluated for atomic distance 1.401 a.u., the Hamiltonian matrix found as a 16×16 matrix,[6] so 4 qubits are required to implement the unitary propagator of this Hamiltonian which is found from $e^{-i\mathcal{H}t}$ by setting $t = 1$ (see the note in Ref. 33).

The accuracy of the circuit design for the unitary propagator also determines the accuracy of the simulation. The generation of quantum circuits by using matrix decomposition techniques or global optimization methods[34] (as done for water and hydrogen molecules in Ref. 6) requires searching a
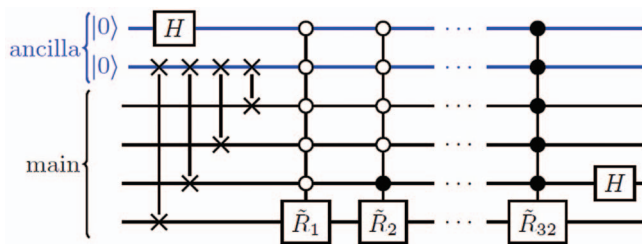
FIG. 8. The circuit for the simulation of the hydrogen molecule. The angle values for the rotation gates are determined to create the elements of $\tilde{U}_{H_2}$: There are only 19 rotation gates, the rest are X gates in order to get the right order for the elements after the combination. For diagonal elements of $\tilde{U}_{H_2}$, these rotations are only around z-axis. For nonzero-diagonal elements, rotation about z-axis followed by rotations about y-axis. The angles for these gates are given in Table I.

huge complex space and simulation of the unitary matrices of quantum systems on classical computers. For large matrices, this hinders the efficiency, and hence, the accuracy of the circuits. Since the angles for the rotation gates in our circuits are determined from the matrix elements directly (for instance, in the first design, Fig. 3), we only take the arcosine of the values, and generating these angles requires only a few computations; the accuracy and the efficiency of the circuits are always high. This helps to get very accurate circuit designs for the simulation of quantum systems. For instance, for the 16×16 unitary propagator of hydrogen molecule given in Ref. 6, nine qubits are required in the circuit scheme given in Fig. 3. Since the unitary propagator is highly sparse and has only 19 nonzero elements, most of the uniformly controlled gates in the circuit will be identity except 19 of them. Hence, in Appendix B we have shown how to reduce the number of qubits to 6 qubits, Fig. 8. We give the rotation values for the gates in Table I. Therefore, since our circuit designs

have fixed designs, using different basis sets or parameters to compute the Hamiltonian will not change the circuit design and the accuracy of it.

In summary, we present general programmable quantum circuits which can simulate any given $2^n \times 2^n$ real matrix. Because of the structure of the circuits, they can be used to fabricate specific or general purpose quantum chips and processors. Since the circuit designs are highly dependent on the matrix elements; for the application specific circuits aimed to implement particular type of systems, any level of sparsity in the system may reduce the number of gates significantly. In addition, we show that the generation of circuits with the complexity less than the lower bound is possible by making $m \leq 2^{c-2}$ and increasing Φ in the given complexity.

ACKNOWLEDGMENTS

APPENDIX A: THE EXPLICIT ILLUSTRATION OF THE STEPS

Here, we detail the implementation of the input modification, the formation (V_f), and the combination (V_c) steps. A sketch of the matrix format of the operations can be found in Eq. (A3)–for the one-qubit case in the first circuit design–and Eqs. (A8) and (A9)–for the two-qubit case in the second circuit design; here, blanks denote zeros and dots denote matrix parts of no interest for the final operation.

1. First circuit design

Starting with an arbitrary input, $|\psi\rangle = (\alpha_0, \alpha_1)^T$, and the following arbitrary unitary matrix:

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}, \tag{A1}$$

the first method requires $2n + 1 = 3$ qubits for the simulation with the input

$$|\psi_{initial}\rangle = |0\rangle \otimes |0\rangle \otimes |\psi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{A2}$$

The followings represent the formation matrix, V_f, the matrix after the combination step, V and the modified input, $|\tilde{\psi}\rangle$:

TABLE I. Parameters for the rotation gates.

State of control qubits	Matrix elements	Angle for R_z	Angle for R_y
00000	0.9788−0.2049i	−0.4127	0
00010	0.3987+0.9171i	2.3214	0
00100	0.3987+0.9171i	2.3214	0
00110	−0.2607+0.9517i	3.6763	0.3253
00111	0.1401−0.0817i	−1.0559	2.8158
01000	0.1401−0.0817i	−1.0559	2.8158
01001	−0.2607+0.9517i	3.6763	0.3253
01011	0.9354+0.3535i	0.7226	0
01101	0.3189+0.9478i	2.4925	0
01110	0.4766+0.8604i	2.1299	0.3629
01111	−0.1577+0.0874i	5.271	2.779
10000	−0.1577+0.0874i	5.271	2.779
10001	0.4766+0.8604i	2.1299	0.3629
10011	0.3130+0.9498i	2.5049	0
10101	0.3189+0.9478i	2.4925	0
10111	0.3130+0.9498i	2.5049	0
11001	0.9569+0.2410i	0.4934	0
11011	0.8889+0.4582i	0.9519	0
11101	0.8889+0.4582i	0.9519	0
11111	1	0	0

$$V_f = \begin{pmatrix} u_{00} & \cdot & & & & & \\ \cdot & \cdot & & & & & \\ & & u_{01} & \cdot & & & \\ & & \cdot & \cdot & & & \\ & & & & u_{10} & \cdot & \\ & & & & \cdot & \cdot & \\ & & & & & & u_{11} & \cdot \\ & & & & & & \cdot & \cdot \end{pmatrix}, \quad V = \frac{1}{\sqrt{2}}\begin{pmatrix} u_{00} & \cdot & u_{01} & \cdot & & & \\ \cdot & \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & \cdot & & & \\ & & & & u_{10} & \cdot & u_{11} & \cdot \\ & & & & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \quad |\tilde{\psi}\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} \alpha_0 \\ 0 \\ \alpha_1 \\ 0 \\ \alpha_0 \\ 0 \\ \alpha_1 \\ 0 \end{pmatrix}. \quad (A3)$$

For illustration purposes, below we also present full forms of some of the operators and the output vector for the same case: The full form of the resulting matrix from the formation step is as follows:

$$V_f = \begin{pmatrix} u_{00} & \sqrt{1-u_{00}^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sqrt{1-u_{00}^2} & u_{00} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_{01} & \sqrt{1-u_{01}^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{1-u_{01}^2} & u_{01} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & u_{10} & \sqrt{1-u_{10}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sqrt{1-u_{10}^2} & u_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_{11} & \sqrt{1-u_{11}^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\sqrt{1-u_{11}^2} & u_{11} \end{pmatrix}. \quad (A4)$$

The combination matrix V_C and the matrix for input modification V_M are defined as

$$V_c = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}, \quad V_m = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}. \quad (A5)$$

For the initial input $|\psi_{initial}\rangle$ as in Eq. (A2), the final output state $|\psi_{final}\rangle$ becomes

$$|\psi_{final}\rangle = V_c V_f V_m |\psi_{initial}\rangle$$

$$= \frac{1}{2}\begin{pmatrix} \alpha_0 u_{00} + \alpha_1 u_{01} \\ -\alpha_0\sqrt{1-u_{00}^2} - \alpha_1\sqrt{1-u_{01}^2} \\ \alpha_0 u_{00} - \alpha_1 u_{01} \\ -\alpha_0\sqrt{1-u_{00}^2} + \alpha_1\sqrt{1-u_{01}^2} \\ \alpha_0 u_{10} + \alpha_1 u_{11} \\ -\alpha_0\sqrt{1-u_{10}^2} - \alpha_1\sqrt{1-u_{11}^2} \\ \alpha_0 u_{10} - \alpha_1 u_{11} \\ -\alpha_0\sqrt{1-u_{10}^2} + \alpha_1\sqrt{1-u_{11}^2} \end{pmatrix}. \quad (A6)$$

Clearly, the normalized states $|00-0\rangle$ and $|10-0\rangle$ simulate the original given system.

2. Second circuit design

For the same case, since the second circuit design initially works at least a pair of matrix elements, it will create the unitary at the initial step. There will be no need for the combination step. Hence, the output will be simulated on the states $|00\rangle$ and $|10\rangle$. For two-qubit system below, the simulation goes as follows:

$$U = \begin{pmatrix} u_{00} & u_{01} & u_{02} & u_{03} \\ u_{10} & u_{11} & u_{12} & u_{13} \\ u_{20} & u_{21} & u_{22} & u_{23} \\ u_{30} & u_{31} & u_{32} & u_{33} \end{pmatrix}. \quad (A7)$$

In the formation step, if we use 4×4 blocks as shown in Fig. 4, there will be no need for the combination step since we will have already formed the rows of U at the formation step. However, if we use 2×2 initial blocks, we need to use one rotation gate for each pair of the elements, then the combination step. Thus, at the end of the formation step, we get the following matrix:

$$
V_f = \begin{pmatrix}
k_0 u_{00} & k_0 u_{01} \\
\cdot & \cdot \\
& & k_1 u_{02} & k_1 u_{03} \\
& & \cdot & \cdot \\
& & & & k_2 u_{10} & k_2 u_{11} \\
& & & & \cdot & \cdot \\
& & & & & & k_3 u_{12} & k_3 u_{13} \\
& & & & & & \cdot & \cdot \\
& & & & & & & & k_4 u_{20} & k_4 u_{21} \\
& & & & & & & & \cdot & \cdot \\
& & & & & & & & & & k_5 u_{22} & k_5 u_{23} \\
& & & & & & & & & & \cdot & \cdot \\
& & & & & & & & & & & & k_6 u_{30} & k_6 u_{31} \\
& & & & & & & & & & & & \cdot & \cdot \\
& & & & & & & & & & & & & & k_7 u_{32} & k_7 u_{33} \\
& & & & & & & & & & & & & & \cdot & \cdot
\end{pmatrix},
\tag{A8}
$$

where k_is are the normalization constants. After the sequential combination steps and the modification on the input, we get the following matrix and the modified input:

$$
V = \begin{pmatrix}
u_{00} & u_{01} & u_{02} & u_{03} \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
& & & & u_{10} & u_{11} & u_{12} & u_{13} \\
& & & & \cdot & \cdot & \cdot & \cdot \\
& & & & \cdot & \cdot & \cdot & \cdot \\
& & & & \cdot & \cdot & \cdot & \cdot \\
& & & & & & & & u_{20} & u_{21} & u_{22} & u_{23} \\
& & & & & & & & \cdot & \cdot & \cdot & \cdot \\
& & & & & & & & \cdot & \cdot & \cdot & \cdot \\
& & & & & & & & \cdot & \cdot & \cdot & \cdot \\
& & & & & & & & & & & & u_{30} & u_{31} & u_{32} & u_{33} \\
& & & & & & & & & & & & \cdot & \cdot & \cdot & \cdot \\
& & & & & & & & & & & & \cdot & \cdot & \cdot & \cdot \\
& & & & & & & & & & & & \cdot & \cdot & \cdot & \cdot
\end{pmatrix},
\quad |\tilde{\psi}\rangle = 1/2 \begin{pmatrix}
\alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3
\end{pmatrix}.
\tag{A9}
$$

The final state is equivalent to $|\psi_{final}\rangle = V|\tilde{\psi}\rangle$. In $|\psi_{final}\rangle$, the states $|0000\rangle$, $|0100\rangle$, $|1000\rangle$, and $|1100\rangle$ are the respective states that simulate the original given unitary matrix.

APPENDIX B: EXPLICIT CIRCUIT FOR THE UNITARY PROPAGATOR OF THE HYDROGEN MOLECULE

As mentioned, the unitary matrix, U_{H_2}, for the hydrogen molecule has 19 nonzero elements, 15 of them located at the diagonal. Since the unitary is 16×16 we need 4 main and 5 ancilla qubits for the first circuit design given in Fig. 3. And the uniformly controlled rotation gates in the formation steps are the R_y gates followed by R_z gates where we use identity for the zero elements. However, we can benefit from the sparsity of the matrix and reduce the number of ancilla to 2 qubits

instead of 5: The non-diagonal matrix elements are located at (13, 4), (4, 13), (7, 10), and (10, 7), where (i, j) are the row and column indices. We apply a permutation matrix, P, to reduce the bandwidth of the matrix. PU_{H_2} takes non-diagonal elements (13, 4), (4, 13), (7, 10), and (10, 7) to (5, 4), (4, 5), (7, 8), and (8, 7) which creates another unitary, $\tilde{U}_{H_2}$. $\tilde{U}_{H_2}$ is a structured matrix where all the elements are located on the (i, i), $(i, i + 1)$, or $(i - 1, i)$ positions. Hence, we can use 2 qubits for ancilla and 4 qubits for the main to create matrix V having 4×4 block matrices on its diagonal by using only one Hadamard gate in the combination step. In the formation step, the control qubits for R_y gates and R_z gates are determined to form the couple of (i, i) and $(i, i + 1)$, or $(i - 1, i)$ and (i, i) elements on the first row of these 4×4 matrices. The angle values are determined from the polar representation of each element and given in Table I. The circuit for $\tilde{U}_{H_2}$ is shown

in Fig. 8 where $\tilde{R}$ represents a combination of a R_y and a R_z gates. Please note that the circuit equivalences of the permutation matrices such as P are the combinations of multi-control CNOT gates where which elements to be switched is determined by the control qubits. And the input should be also permuted prior to the circuit. This can be done by simply switching the input for the qubits. At the end of this circuit, since the leading rows of 4×4 matrices simulate the unitary, we get the simulation result from the states $|\mathbf{0}\rangle$, $|\mathbf{4}\rangle$, $|\mathbf{8}\rangle$, $|\mathbf{12}\rangle$, ..., $|\mathbf{60}\rangle$.

[1] C. Bobda, *Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications* (Springer, 2007).

[2] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, Science **292**, 472 (2001).

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[4] P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing* (Oxford, 2007).

[5] C. P. Williams, *Explorations in Quantum Computing (Texts in Computer Science)*, 2nd ed. (Springer, London, 2011).

[6] A. Daskin and S. Kais, J. Chem. Phys. **134**, 144112 (2011).

[7] M. A. Nielsen and I. L. Chuang, Phys. Rev. Lett. **79**, 321 (1997).

[8] J. P. Paz and A. Roncaglia, Phys. Rev. A **68**, 052316 (2003).

[9] P. B. M. Sousa and R. V. Ramos, Quantum Inf. Comput. **7**, 228 (2007); e-print arXiv:quant-ph/0602174.

[10] G. Vidal and C. M. Dawson, Phys. Rev. A **69**, 010301 (2004).

[11] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, Phys. Rev. Lett. **91**, 027903 (2003).

[12] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Phys. Rev. Lett. **93**, 130502 (2004).

[13] F. Vatan and C. P. Williams, "Realization of a general three-qubit quantum gate," e-print arXiv:quant-ph/0401178.

[14] W. Hai-Rui, D. Yao-Min, and Zhang-Jie, Chin. Phys. Lett. **25**, 3107 (2008).

[15] D. Hanneke, J. P. Home, J. D. Jost, J. M. Amini, D. Leibfried, and D. J. Wineland, Nat. Phys. **6**, 13 (2009).

[16] B. P. Lanyon, M. Barbieri, M. P. Almeida, T. Jennewein, T. C. Ralph, K. J. Resch, G. J. Pryde, J. L. O'Brien, A. Gilchrist, and A. G. White, Nat. Phys. **5**, 134 (2009).

[17] T. D. Mackay, S. D. Bartlett, L. T. Stephenson, and B. C. Sanders, J. Phys. A: Math. Theor. **35**, 2745 (2002).

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (Johns Hopkins University Press, Baltimore, MD, 1996).

[19] B. Drury and P. Love, J. Phys. A: Math. Theor. **41**, 395305 (2008).

[20] V. V. Shende, S. S. Bullock, and I. L. Markov, IEEE Trans. Comput.-Aided Des. **25**, 1000 (2006).

[21] B. J. Fino and V. R. Algazi, IEEE Trans. Comput. **25**, 1142 (1976).

[22] K. L. Brown, S. De, V. M. Kendon, and W. J. Munro, New J. Phys. **13**, 095007 (2011).

[23] D. S. Abrams and S. Lloyd, Phys. Rev. Lett. **83**, 5162 (1999).

[24] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).

[25] L. Veis and J. Pittner, J. Chem. Phys. **133**, 194106 (2010).

[26] H. Wang, S. Kais, A. Aspuru-Guzik, and M. R. Hoffmann, Phys. Chem. Chem. Phys. **10**, 5388 (2008).

[27] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Mol. Phys. **109**, 735 (2011).

[28] A. Papageorgiou and C. Zhang, Quantum Inf. Process. **11**, 541 (2012).

[29] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White, Nat. Chem. **2**, 106 (2010).

[30] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, Proc. Natl. Acad. Sci. U.S.A. **105**, 18681 (2008).

[31] L. Veis, J. Višňák, T. Fleig, S. Knecht, T. Saue, L. Visscher, and J. Pittner, Phys. Rev. A **85**, 030304 (2012).

[32] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, Annu. Rev. Phys. Chem. **62**, 185 (2011).

[33] For the matrix exponentiation, the MATLAB *expm* function which uses the Pade approximation with scaling and squaring is used.[6]

[34] A. Daskin and S. Kais, Mol. Phys. **109**, 761 (2011).