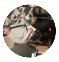
The cost of quantum gate primitives

Article *in* Journal of Multiple-valued Logic and Soft Computing · January 2006

6 authors, including:

Soonchil Lee
Korea Advanced Institute of Science and Technology



Taegon Kim
Korea Institute of Science and Technology

Jae-Seung Lee
ICE Bonds

J. of Mult.-Valued Logic & Soft Computing., Vol. 12, pp. 561–573
Reprints available directly from the publisher
Photocopying permitted by license only

The Cost of Quantum Gate Primitives

Soonchil Lee[1], Seong-Joo Lee[1], Taegon Kim[1],
Jae-Seung Lee[2], Jacob Biamonte[3] and Marek Perkowski[3]

[1]*Department of Physics, Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea*
[2]*Department of Chemistry, Kent State University, Kent, OH44242, USA*
[3]*Department of Electrical Engineering, Portland State University, Portland Oregon, 97207-0751, USA*
E-mail: sclee@mail.kaist.ac.kr

The cost of the most frequently used gates in quantum computation is estimated by counting the sequence of basic physical operations required for implementation on a quantum computer. The sequence of physical operations comprising each gate is minimized using a software algorithm based on commutation rules. Because operation costs are machine dependent, an ideal practical quantum computer is presented and used for calculations. The presented gate costs are not necessarily minimal; however at least they provide reference to the upper bound.

Keywords: quantum computation, quantum algorithms, quantum synthesis, quantum gate cost

1 INTRODUCTION

Quantum computation has been the subject of intense study during the last decade because well designed quantum algorithms can solve certain problems practically impossible in classical computation. All quantum algorithms can be represented by a unitary operation, and any unitary operation can be replaced by a combination of controlled-NOT (CNOT) and single qubit operations, or gates [1]. A quantum system is said to be a universal quantum computer if capable of implementing these elementary gates. Many efforts have been made to realize a quantum computer, and the liquid state nuclear magnetic resonance (NMR) [2, 3, 4] has been the most successful so far.

A quantum algorithm in the form of a unitary matrix is first created and then decomposed to a series of standard elementary quantum gates for computation [5]. In implementation, each gate is again converted to a sequence of physical operations that a given type of quantum computer can actually perform. This is analogous to classical computation where an algorithm is programmed in an assembly language, and then compiled to machine code. The total calculation time in quantum computation depends on the number of basic gates in the series and the number of physical operations required for a quantum system to implement each gate. Let us denote a series of physical operations as a pulse sequence distinguishing it from the series of basic gates, as the physical operations are either the time evolution of finite duration under the influence of an externally applied magnetic field, or interactions between qubits. In quantum computation, calculation time is a very precious resource due to the finite coherence time of a quantum system. Therefore, it is important to know the cost of gates for successful implementation of an algorithm, and the future design of a practical quantum computer.

In this work, the costs of the most frequently used quantum gates such as the Toffoli and Fredkin gate were estimated. It is already known how these gates are decomposed to a series of the single and two qubit gate primitives [6]. Once the pulse sequences for the single and two qubit gates are obtained, the total pulse sequence for a circuit is given by replacing each elementary gate by the corresponding pulse sequence. The pulse sequence of more complicated circuits with larger numbers of input qubits can be obtained in the same way, that is, by finding the quantum circuits composed of simpler gates and replacing each gate by the corresponding pulse sequence. The cost of a gate is determined by reducing the number of pulses in the sequence using the commutation rules of the pulse operations using our software algorithm.

The values we present here are not necessarily minimal, since we have no way of knowing if the solutions we found are local or global minima. Therefore, they may not be the true costs of gates, and we claim only to provide the upper bounds as the worst case. Since implementing the physical operations depends on the Hamiltonian of a quantum system, we defined an ideal model of a practical quantum computer. Some work assumes that the cost of the gates with the same number of input qubits are identical [7] but in practice, this is not the case, making many of the known results less practical for actual implementation. The time evolution under the influence of a single term of the Hamiltonian is a single physical operation, we optimize the circuit at the level of such operations (pulses). This is a new approach in logic synthesis of quantum circuits since previous publications [6,7,12,17] optimized the quantum circuit at higher levels of abstraction. Our software modifies an initial non-optimal design by shifting gates and applying quantum logic identities, analogously to [17], but calculating the combined cost of the operations necessary to build arbitrary quantum circuits instead of the total gate cost. Although our current software is restricted to small circuits, the principle of optimizing a quantum circuit by applying template-matching methods [17] at a level lower than in [17] is a new concept that will be further refined in forthcoming papers. It can be used not only to optimize single gates as here, but also complete quantum circuits.

2 MODEL QUANTUM COMPUTER

Our estimation of the gate cost is based on the time evolution operations under a Hamiltonian. Except for the optical quantum computer [8], most quantum computer systems work this way, but the general principle described here is applicable to any system. A quantum system should meet several requirements to be a quantum computer [9]. The component particles of the system should have at least two well defined quantum states to be used as qubits, and should interact with each other. There must be a way for external devices to perform single qubit operations and read the qubit states. Long coherence time and scalability of the system is desirable for a good quantum computer. From the practical point of view, an ideal quantum computer needs to have more detailed specifications in addition to these requirements. We define a model quantum computer as a system that meets the following specifications:

(i) The Hamiltonian of the system is given by,

$$H = \sum_{i,\alpha} a_{i\alpha}\sigma_{i\alpha} + \sum_{i,j} J_{ij}\sigma_{iz}\sigma_{jz} \tag{1}$$

where α = x, y or z and symbol σ represent one of the Pauli operators. This is the most familiar form of the spin Hamiltonian where spins are interacting with each other in an external magnetic field. However, this Hamiltonian is not particular to spin systems but is general, as similar forms are relevant for any quantum computer. It is common to refer to the first term of the Hamiltonian in Equation 1 as the Zeeman term, and the second term as the interaction term.

$$\sum_{i,\alpha} a_{i\alpha}\sigma_{i\alpha} \qquad (2) – \text{Zeeman term}$$

The Zeeman term is necessary to produce all the single qubit gates. As its name implies the second term defines interaction between qubits, such as those that are essential to make a CNOT gate. In the standard form, the Ising model is characterized by the interaction of only Z-components of spins. The interaction form should not necessarily be of the Ising type, although it is advantageous because the Ising type interaction can generate the indispensable CNOT gate, while it is not quite clear if general interactions can do the same. Another interaction model, the XY type, has a notable advantage over the Ising type; it can generate a SWAP gate as a single pulse. A SWAP gate is realized by 3 CNOT gates and each CNOT gate requires 5 pulses with the Ising type Hamiltonian. Therefore, the XY type is more effective in generating this operation; nevertheless SWAP gates cannot be made to replace CNOT gates.

(ii) Any term in the Hamiltonian can be turned on and off independently. This statement implies that each qubit is individually addressable, and that the evolution of interactions is independent of each other as well as the single qubit operations. This natural requirement for computation cannot be easily implemented in practice. One of the reasons that NMR has been the most successful quantum computer is that the interactions between qubits are very well defined, both mathematically and physically. From a practical point of view the most serious disadvantage of NMR is that we cannot artificially turn off the interactions of the nuclear spins. In systems having more than two qubits, there are unwanted interactions all the time; a strong short pulse is used to implement single qubit operations so that those interactions can be neglected, and a refocusing scheme is used to turn off all the interactions except the one necessary to implement double qubit operations such as CNOT. Although Refocusing is a useful trick in NMR, allowing one to remove the coupled evolution between spins, or all evolution entirely, it has the disadvantage of making a pulse sequence several times longer [10,15]. Similar approximations and separation tricks should be used in other systems such as the Josephson device quantum computer where the external field evokes not only interactions but also single qubit operations [11]. The capability of independently turning on and off each term in a Hamiltonian is important to the practical quantum computer, unless the evolution operations of mixed terms were defined and used as basic gates instead of the ones familiar to us now.

(iii) The time evolution under the influence of a single term of the Hamiltonian is a single physical operation. The evolution with the Zeeman term rotates qubits about the direction of the field. In NMR, usually the hardware for the rotation about the x- and y-axes is physically implemented with the omission of the z-axis. The local field in the nucleus of a molecule is different from the external field, due in part to the shielding effect of the electrons. The difference between the local and external field is called the chemical shift. The rotation about the z-axis is replaced by either the composite rotations about the x- and y-axes using the Euler rotation, or the use of the chemical shift. The angular speed of spin recession about the direction of the field is proportional to the strength of the field. Since the shift at each nucleus is different, it can play the role of the addressable single qubit rotation about the z-axis. The rotation is effectively done by just changing the reference of the rotating frame of each qubit. If a composite pulse is used, this operation requires more than one pulse, and if the change of the reference frame is adopted, no pulse is required. Irrespective of the way this operation is implemented, we count this operation as one pulse. It is often possible to perform several single qubit gates with one pulse simultaneously, that is, several qubits can be addressed concurrently by one pulse referred to as a hard pulse in spectroscopy. A series of rotations about different axes can also be performed by one pulse using the Euler rotations even though the angles are different. Since the tricks mentioned here are machine dependent, we don't consider them in our estimation of gate costs. The physical pulses one is able to implement by the Hamiltonian from Equation 1 are,

$$\begin{aligned} R_{i\alpha}(\phi) &= e^{-i\frac{\phi}{2}\sigma_{i\alpha}} \\ J_{ij}(\phi) &= e^{-i\frac{\phi}{2}\sigma_{iz}\sigma_{jz}} \end{aligned} \tag{3}$$

We define the cost of a gate as the number corresponding to its minimal pulse implementation. Our algorithm performs full reduction using the commutation rules on the sequence of pulses representing the gates circuit. The commutation rules used in the reduction algorithm are;

(1) $$[R_{i\alpha}, R_{j\alpha'}] = 0 \text{ for } i \neq j$$

(2)
$$\begin{aligned} R_{ix}(\pm\pi)R_{iy}(\phi) &= R_{iy}(-\phi)R_{ix}(\pm\pi) \\ R_{ix}(\phi)R_{iy}(\pm\pi) &= R_{iy}(\pm\pi)R_{ix}(-\phi) \\ R_{ix}(\pm\tfrac{\pi}{2})R_{iy}(\phi) &= R_{iz}(\pm\phi)R_{ix}(\pm\tfrac{\pi}{2}) \\ R_{ix}(\pm\tfrac{\pi}{2})R_{iz}(\phi) &= R_{iy}(\mp\phi)R_{ix}(\pm\tfrac{\pi}{2}) \\ R_{ix}(\phi)R_{iy}(\pm\tfrac{\pi}{2}) &= R_{iy}(\pm\tfrac{\pi}{2})R_{iz}(\pm\phi) \\ R_{ix}(\phi)R_{iz}(\pm\tfrac{\pi}{2}) &= R_{iz}(\pm\tfrac{\pi}{2})R_{iy}(\mp\phi) \end{aligned}$$

and the relations generated by the cyclic permutation of x $\to$ y $\to$ z.

(3) $$[J_{ij}, J_{i'j'}] = 0$$

(4) $$[J_{ij}, R_{i'z}] = 0$$

To explain the commutation rules and spinors, it is helpful to think of the group of all rotations in 3-dimensional Euclidean space. The rotation group is actually a real Lie group typically denoted as SO(3). The well known group of orthogonal matrices that rotates a unit vector about the standard coordinate axes through an angle ϕ are given as 3×3 matricies and maps to U(2), the unitary group we use to represent single qubits, and a qubit is a two-component spinor. An interesting property of spin ½ particles is that they must be rotated by an angle of 4π in order to return to their original configuration. Mathematically we represent this due to the fact that U(2) and SO(3) are not globally isomorphic, even though their infinitesimal generators U(2) and SO(3) are isomorphic. U(2) is actually a "double cover" of SO(3). This means that $R_{i\alpha}(\phi)$ is not equal to $R_{i\alpha}(\phi - 2\pi)$ but equal to $R_{i\alpha}(\phi - 4\pi)$ and that the commutation relations have a period of 2π[16], and is the reason for the angles being divided by 2 in the well known equations representing rotations of a single qubit.

3 COST ESTIMATION

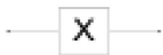
3.1 Single Qubit Gates

The most frequently used single qubit gates in quantum algorithms are the NOT(N) *(also known as Pauli-X, or X [15])*, Hadamard(H), and phase(P) *(also known as S [15])* gates in the vector space spanned by,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \; |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4}$$

These gates are the special cases of the single qubit rotation operations and are implemented by the rotation pulses as

$$N = iR_x(\pi) = i\begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & -i\sin\left(\frac{\pi}{2}\right) \\ -i\sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} =$$



$$H = iR_y\left(\frac{\pi}{2}\right)R_z(\pi) = i\begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) \end{bmatrix}\begin{bmatrix} e^{-i\frac{\pi}{2}} & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}$$

$$= \left(\frac{1}{\sqrt{2}}\right)\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} =$$

$$P = e^{i\frac{\phi}{2}}R_z(\phi) = e^{i\frac{\phi}{2}}\begin{bmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} =$$

Therefore, the costs of Gates N and P are said to be 1, and that of H is 2, from the definition of our model quantum computer. It is worthwhile to note that gates with the same number of input qubits can have different costs in practice. The pulse sequence of a gate is not unique in general. It is

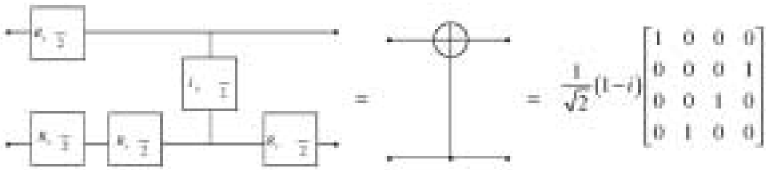


FIGURE 1
Representation of the CNOT Gate.

also worthwhile to note the fact that the N, H and P gates are implemented up to overall phase, we illustrate an example of this for the N gate below.

Let us denote a NOT gate such that it is correct to overall phase, doing so we have,

$$N = R_x(\pi) = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & -i\sin\left(\frac{\pi}{2}\right) \\ -i\sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix} = (-i)\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

If we assume $|\Psi\rangle$ to be in an arbitrary state, such that $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, having a density matrix that takes the form,

$$\rho = |\Psi\rangle\langle\Psi| = \begin{bmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{bmatrix},$$

now allowing N to act on arbitrary ρ we have

$$N(|\Psi\rangle\langle\Psi|)N^\dagger = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix}\begin{bmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{bmatrix}\begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} = \begin{bmatrix} |\beta|^2 & \alpha^*\beta \\ \alpha\beta^* & |\alpha|^2 \end{bmatrix}$$

From this one can clearly deduce that a NOT gate can be created to serve the same function, up to a overall phase.

3.2 Two Qubit Gates

The most frequently used two qubit gates are the CNOT and SWAP gates. A possible pulse sequences for the CNOT gate is,

$$CNOT_{ij} = R_{iz}\left(\frac{\pi}{2}\right)R_{jx}\left(\frac{\pi}{2}\right)R_{jy}\left(\frac{\pi}{2}\right)J_{ij}\left(-\frac{\pi}{2}\right)R_{jy}\left(\frac{-\pi}{2}\right) \quad (5)$$

Pulse sequence for CNOT gate (accurate to phase, where i is the target bit) where the qubit upper and lower qubit are the control and target, respectively. As shown by Equation 5, the cost of a CNOT gate is 5. Another frequently used controlled gate is the controlled-V where V^2 is

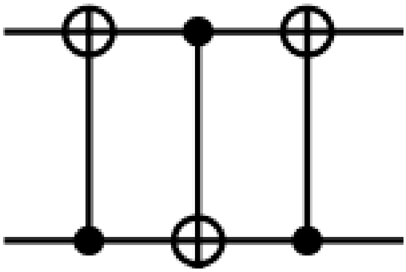


FIGURE 2
SWAP Gate comprised of 3 CNOT gates.

equivalent to a NOT gate. The cost of this gate is also 5 because it can be implemented by

$$Controlled - V = R_{2y}\left(\frac{\pi}{2}\right)R_{1z}\left(\frac{\pi}{4}\right)R_{2z}\left(\frac{\pi}{4}\right)J_{12}\left(\frac{-\pi}{4}\right)R_{2y}\left(\frac{-\pi}{2}\right) \quad (6)$$

Pulse sequence for Controlled=V gate (accurate to phase, where i is the target bit).

Once the pulse sequences of the CNOT, controlled-V, and single qubit gates are known, the pulse sequence for the other multi-qubit gates can be obtained if the gate is decomposed to a series of these basic gates.

The SWAP gate is decomposed of three CNOT gates as shown in Figure 2. The pulse sequence of the SWAP gate obtained by replacing each CNOT gate by the sequence in Equation 5 is,

$$\begin{aligned} SWAP = {} & R_{2y}\left(\frac{\pi}{2}\right)R_{1z}\left(\frac{-\pi}{2}\right)R_{2z}\left(\frac{-\pi}{2}\right)J_{12}\left(\frac{\pi}{2}\right)R_{2y}\left(\frac{-\pi}{2}\right) \\ & \times R_{1y}\left(\frac{\pi}{2}\right)R_{2z}\left(\frac{-\pi}{2}\right)R_{1z}\left(\frac{-\pi}{2}\right)J_{12}\left(\frac{\pi}{2}\right)R_{1y}\left(\frac{-\pi}{2}\right) \\ & \times R_{2y}\left(\frac{\pi}{2}\right)R_{1z}\left(\frac{-\pi}{2}\right)R_{2z}\left(\frac{-\pi}{2}\right)J_{12}\left(\frac{\pi}{2}\right)R_{2y}\left(\frac{-\pi}{2}\right) \end{aligned} \quad (7)$$

Using our algorithm we showed that Equation 7 can be reduced to Equation 8, and from Equation 8, the cost of the SWAP gate is shown to be 11. This is not necessarily either a unique or minimum expression.

$$\begin{aligned} SWAP = {} & R_{2y}\left(\frac{\pi}{2}\right)R_{2z}\left(\frac{-3\pi}{2}\right)R_{1z}\left(\frac{-3\pi}{2}\right) \\ & \times J_{12}\left(\frac{\pi}{2}\right)R_{2y}\left(\frac{\pi}{2}\right)R_{1y}\left(\frac{-\pi}{2}\right)J_{12}\left(\frac{\pi}{2}\right)R_{1x}\left(\frac{\pi}{2}\right) \\ & \times R_{2x}\left(\frac{-\pi}{2}\right)J_{12}\left(\frac{\pi}{2}\right)R_{2y}\left(\frac{-\pi}{2}\right) \end{aligned} \quad (8)$$

$$R_x^1(\phi)=\begin{bmatrix} \cos\frac{\phi}{2} & 0 & -i\sin\frac{\phi}{2} & 0 \\ 0 & \cos\frac{\phi}{2} & 0 & -i\sin\frac{\phi}{2} \\ -i\sin\frac{\phi}{2} & 0 & \cos\frac{\phi}{2} & 0 \\ 0 & -i\sin\frac{\phi}{2} & 0 & \cos\frac{\phi}{2} \end{bmatrix} \qquad R_x^2(\phi)=\begin{bmatrix} \cos\frac{\phi}{2} & -i\sin\frac{\phi}{2} & 0 & 0 \\ -i\sin\frac{\phi}{2} & \cos\frac{\phi}{2} & 0 & 0 \\ 0 & 0 & \cos\frac{\phi}{2} & -i\sin\frac{\phi}{2} \\ 0 & 0 & -i\sin\frac{\phi}{2} & \cos\frac{\phi}{2} \end{bmatrix}$$

$$R_y^1(\phi)=\begin{bmatrix} \cos\frac{\phi}{2} & 0 & -\sin\frac{\phi}{2} & 0 \\ 0 & \cos\frac{\phi}{2} & 0 & -\sin\frac{\phi}{2} \\ \sin\frac{\phi}{2} & 0 & \cos\frac{\phi}{2} & 0 \\ 0 & \sin\frac{\phi}{2} & 0 & \cos\frac{\phi}{2} \end{bmatrix} \qquad R_y^2(\phi)=\begin{bmatrix} \cos\frac{\phi}{2} & -\sin\frac{\phi}{2} & 0 & 0 \\ \sin\frac{\phi}{2} & \cos\frac{\phi}{2} & 0 & 0 \\ 0 & 0 & \cos\frac{\phi}{2} & -\sin\frac{\phi}{2} \\ 0 & 0 & \sin\frac{\phi}{2} & \cos\frac{\phi}{2} \end{bmatrix}$$

$$R_z^1(\phi)= e^{-i\frac{\phi}{2}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\phi} & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{bmatrix} \qquad R_z^2(\phi) = e^{-i\frac{\phi}{2}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{bmatrix}$$

$$J_{12}(\phi) = e^{-i\frac{\phi}{2}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi} & 0 & 0 \\ 0 & 0 & e^{i\phi} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Two Qubit Rotation Operations

3.3 Three Qubit Gates

We calculated the cost of the most frequently used three qubit gates (Toffoli and Fredkin), as well as the Miller gate [12] and the Peres gate [13]. The circuit diagrams of these four gates are shown in Figure 3. The Peres gate is interesting because it was the cheapest found among those familiar in the universal set of reversible logic gates. It is just like a Toffoli gate but without the last CNOT gate as shown in Figure 3(a). The costs and the pulse sequences of the gates obtained by using the reduction program are summarized in Table 1 except for the case of the Toffoli gate.

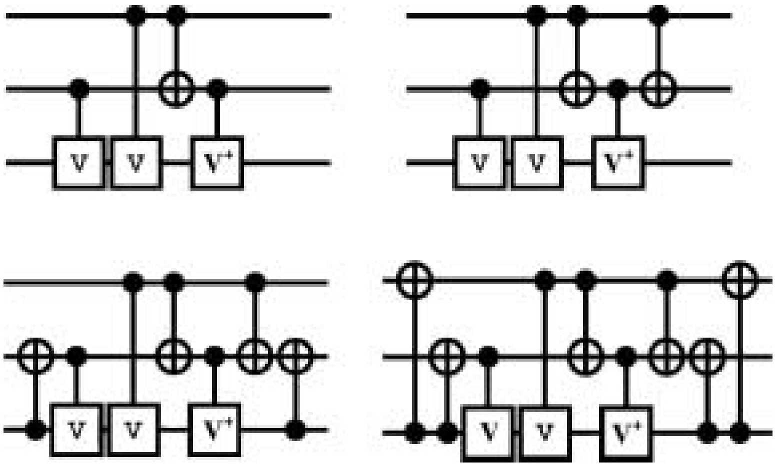


FIGURE 3
Peres(a), Toffoli (b), Fredkin(c), and Miller(d).

The pulse sequence of the Toffoli gate reduced from the circuit in Figure 3(b) is composed of 15 pulses and contains 5 interaction terms. However, the equivalent sequence of this gate analyzed by the geometric algebra method presented in [2] is composed of 13 pulses and contains 6 interaction terms. The sequence we listed in Table 1 for the Toffoli gate is the one with lower cost. This case indicates that there is at least one quantum circuit for the Toffoli gate more efficient than shown in Figure 3(b), a possibility also exists such that the sequences listed in the table can be reduced further. Although the cost of the Toffoli gate given in Table 1 is lower than the gate shown in Figure 3(b), the gate from Figure 3(b) is practically cheaper on NMR where the refocusing scheme requires several extra pulses to make a single interaction effective. It is also possible that equivalent sequences can have a different number of interaction terms because, $R_{iz}(\pi)R_{jz}(\pi)J_{ij}(\pi)$ is equal to the identity operation.

In previous work, the cost of the Fredkin gate was counted to be the same as that of the Toffoli gate assuming that the costs of the gates with the same number of input qubits are the same [6]. From Table 1 it is clearly seen that the costs are different in practice even though the number of qubits are the same. The Toffoli gate operation corresponds to the selective excitation between only two states among a possibility of eight Eigenstates for three qubit systems, while the Fredkin gate corresponds to the double or zero quantum transition depending on the definition of the $|0\rangle$ and $|1\rangle$ state. The single quantum transition is simpler to implement than the zero and double quantum transition making the Toffoli gate cheaper than the Fredkin gate, on NMR. The Peres gate has a cost of 12, which is minimal

TABLE 1
Cost of gate primitives.

Gate	Pulse Sequence	Cost
NOT	$i\,R_x(\pi)$	1
Phase	$e^{i\frac{\phi}{2}}R_z(\phi)$	1
Hadamard	$i\,R_y(\frac{\pi}{2})R_z(\pi)$	2
CNOT	$e^{-i\frac{\pi}{4}}R_{2y}(\frac{\pi}{2})R_{1z}(-\frac{\pi}{2})R_{2z}(-\frac{\pi}{2})J_{12}(\frac{\pi}{2})R_{2y}(-\frac{\pi}{2})$	5
SWAP	$e^{-i\frac{3\pi}{4}}R_{2y}(\frac{\pi}{2})R_{2z}(-\frac{3\pi}{2})R_{1z}(-\frac{3\pi}{2})J_{12}(\frac{\pi}{2})R_{2y}(\frac{\pi}{2})$ $R_{1y}(-\frac{\pi}{2})J_{12}(\frac{\pi}{2})R_{1x}(\frac{\pi}{2})R_{2x}(-\frac{\pi}{2})J_{12}(\frac{\pi}{2})$ $R_{2y}(-\frac{\pi}{2})$	11
Peres	$e^{-i\frac{\pi}{8}}R_{3y}(\frac{\pi}{2})R_{3z}(-\frac{\pi}{4})R_{2z}(-\frac{3\pi}{4})R_{1z}(-\frac{\pi}{4})J_{23}(\frac{\pi}{4})$ $R_{2x}(-\frac{\pi}{2})J_{12}(\frac{\pi}{2})R_{2y}(\frac{\pi}{2})R_{2z}(\frac{\pi}{4})J_{31}(-\frac{\pi}{4})$ $J_{23}(-\frac{\pi}{4})R_{3y}(-\frac{\pi}{2})$	12
Toffoli	$e^{-i\frac{\pi}{8}}R_{1z}(-\frac{\pi}{4})R_{2z}(-\frac{\pi}{4})J_{12}(\frac{\pi}{4})R_{3y}(-\frac{\pi}{2})$ $R_{3z}(\frac{\pi}{4})J_{31}(-\frac{\pi}{4})J_{23}(-\frac{\pi}{4})R_{3y}(\frac{\pi}{2})J_{31}(\frac{\pi}{2})$ $R_{3x}(\frac{\pi}{2})J_{23}(-\frac{\pi}{4})R_{3x}(-\frac{\pi}{2})J_{31}J_{31}(-\frac{\pi}{2})$	13
Fredkin	$e^{-i\frac{7\pi}{8}}R_{2y}(-\frac{\pi}{2})R_{1z}(-\frac{3\pi}{4})J_{23}(\frac{\pi}{2})R_{3x}(\frac{\pi}{)}2R_{3z}(-\frac{3\pi}{4})$ $J_{12}(\frac{\pi}{2})R_{2y}(\frac{\pi}{2})R_{2z}(-\frac{7\pi}{4})J_{23}(\frac{\pi}{4})R_{2x}(-\frac{\pi}{2})$ $J_{12}(\frac{\pi}{2})R_{2y}(\frac{\pi}{2})R_{2z}(-\frac{\pi}{4})J_{31}(-\frac{\pi}{4})J_{23}(-\frac{\pi}{4})$ $R_{3x}(\frac{\pi}{2})R_{2x}(-\frac{\pi}{2})J_{23}(\frac{\pi}{2})R_{2y}(-\frac{\pi}{2})$	19
Miller	$e^{-i\frac{7\pi}{8}}R_{2y}(-\frac{\pi}{2})J_{12}(\frac{\pi}{3})R_{2x}(-\frac{\pi}{2})R_{2z}(-\frac{5\pi}{)}4R_{1x}(\frac{\pi}{2})$ $J_{31}(\frac{\pi}{2})R_{1y}(\frac{\pi}{2})R_{1z}(-\frac{7\pi}{4})R_{3x}(\frac{\pi}{2})R_{3z}(-\frac{5\pi}{4})$ $J_{23}(\frac{\pi}{4})R_{2x}(-\frac{\pi}{2})J_{12}(\frac{\pi}{2})R_{2y}(-\frac{\pi}{2})R_{2z}(-\frac{\pi}{4})$ $J_{31}(-\frac{\pi}{4})J_{23}(-\frac{\pi}{4})R_{3y}(\frac{\pi}{2})R_{2x}(-\frac{\pi}{2})R_{1x}(-\frac{\pi}{2})$ $J_{23}(\frac{\pi}{2})R_{2y}(-\frac{\pi}{2})R_{31}(\frac{\pi}{2})R_{1y}(-\frac{\pi}{2})$	24

among the three qubit universal gates as expected. The sequences of the gates listed in the table are correct with respect to the overall phase.

4 SOFTWARE ALGORITHM

A greedy algorithm has been implemented using the commutation rules for sequences too complicated to be calculated by hand. The program gathers pulses operating on the same spins together using the commutation rules and then reduces the number of pulses by combining the same kind of operations or canceling each other. This process is repeated until complete. The cost of any gate can be estimated by this program once its quantum circuit composed of gates with known pulse sequences is given. Irrespective of the order in which the pulses are combined or cancelled, the program always gives the same answer. The minimum number of pulses depends only on the initial quantum circuit. Therefore, finding an efficient quantum circuit is the important first step in determining the cost of a gate. Because the current version of the program searches only for a local minimum for a given circuit, the values obtained by the program might not be the global minimum cost. The cost of the circuit found depends therefore much on the cost of the initial circuit. It is well known in reversible circuits [17] that using only the cost-reducing transformations *can not* find the global minimum; one has to use also some transformations that lead to intermediate increase of the cost function. The same is true for quantum circuits discussed here, since reversible (permutative) circuits are a subset of them. The future work should analyze the cost-increasing transformations and various tree searching algorithms potentially capable of finding the global minimum.

5 CONCLUSION

The method to optimize quantum circuits based on commutation rules can be used with the template matching method from [17] to optimize arbitrary quantum circuits with realistic costs. In practice, both methods are constrained to small circuits because of their computational inefficiency. Future research will be on efficient implementation of quantum circuit optimization algorithms. In this work, the costs of the primitive gates most frequently used in quantum computation were estimated for future reference. Since the cost is machine dependent, we assumed an ideal quantum computer with practical specifications. Contrary to much work done that assumes the same cost for all gates with similar input, we found this assumption to be generally not satisfied in practice. Having realistic gate costs will allow us to write software that is better realizable on quantum computers and stimulate the development of less expensive gates such the Peres gate. Since the minimal number of pulses of a gate depends on the initial quantum circuit, the values presented here are not necessarily the true cost but they provide the upper bounds. The cost can be decreased by finding more efficient quantum circuits for the gates. Deeper understanding of the relation between the logic and physical operations would aid in finding more efficient circuits.

REFERENCES

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

[2] Cory, D.G., Fahmy, A.F. and Havel, T.F. (1997). Nuclear magnetic resonance spectroscopy: an experimentally accessible paradigm for quantum computing, in *Proc. of the 4th Workshop on Physics and Computation* (Complex Systems Institute, Boston, New England) 1996 Science *275,* 350.

[3] Jones, J.A. and Mosca, M. (1998). Implementation of a Quantum Algorithm on a Nuclear Magnetic Resonance Quantum Computer, *J. Chem. Phys*., *109,* 1648.

[4] Gershenfeld, N.A. and Chuang, I.L. (1997). Bulk Spin-Resonance Quantum Computation, *Science 275,* 350.

[5] Kim, J., Lee, J.-S. and Lee, S. (2000). Implementing unitary operators in quantum computation *Phys. Rev*. A *61,* 032312.

[6] Smolin, J. and DiVincenzo, D.P. (1996). “Five two-qubit gates are sufficient to implement the quantum Fredkin gate.” *Physical Review* A, *53,* 4, 2855–2856.

[7] Shende, V.V., Prasad, A.K., Markov, I.L. and Hayes, J.P. (2003). Synthesis of Reversible Logic Circuits, *IEEE Trans. on CAD 22,* 710.

[8] Kwiat, P.G., Mitchell, J.R., Schwindt, P.D.D. and White, A.G. (2000). Grover’s search algorithm: an optical approach, *J. Mod. Optics 47,* 257.

[9] DiVincenzo, D.P. (2000). in Mesoscopic Electron Transport, eds. Sohn, Kowenhoven, Schoen (Kluwer 1997), p. 657, cond-mat/9612126; “The Physical Implementation of Quantum Computation,” *Fort. der Physik 48,* 771 quant-ph/0002077.

[10] Linden, N., Herve, B., Carbajo, R.J., Freeman, R. and Newton, I. (1999). Pulse sequences for NMR quantum computers: how to manipulate nuclear spins while freezing the motion of coupled neighbours Inst., *Chemical Physics Letters, 305,* 1–2, 28–34.

[11] Makhlin, Y., Schon, G. and Shnirman, A. (2000). Quantum state engineering with Josephson-junction devices arXiv:cond-mat/0011269.

[12] Miller, D.M. (2002). Spectral and Two-Place Decomposition Techniques in Reversible Logic, *IEEE Midwest Symposium on Circuits and Systems,* proceedings on CD-ROM, Tulsa, OK.

[13] Peres, A. (1985). Reversible logic and quantum computers, *Phys. Rev*. A *32,* 3266.

[14] Price, M.D., Somaroo, S.S., Dunlop, A.E., Havel, T.F. and Cory, D.G. (1999). Generalized (controlled)n-NOT quantum logic gates, *Phys. Rev.* A 60:2777–2780.

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

[16] Cartan, E. (1966). The Theory of Spinors, *Dover Press*.

[17] Miller, D.M., Maslov, D. and Dueck, G.W. (2003). “A Transformation Based Algorithm for Reversible Logic Synthesis”, *Design Automation Conference,* Anaheim.