International Journal of Modern Physics B
Vol. 27 (2013) 1350191 (15 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0217979213501919

World Scientific
www.worldscientific.com

A CLASS OF EFFICIENT QUANTUM INCREMENTER GATES FOR QUANTUM CIRCUIT SYNTHESIS

XIAOYU LI[*,†,‡], GUOWU YANG[*,§], CARLOS MANUEL TORRES JR.[†],
DESHENG ZHENG[*] and KANG L. WANG[†]

[*]*College of Computer Science and Engineering,*
University of Electronic Science and Technology of China,
Chengdu, Sichuan 610054, P. R. China
[†]*Department of Electrical Engineering, University of California at Los Angeles*
Los Angeles, CA 90095, USA
[‡]*erin.xiaoyu.li@gmail.com*
[§]*guowu@uestc.edu.cn*

Received 1 April 2013
Revised 16 September 2013
Accepted 23 September 2013
Published 17 October 2013

The quantum incrementer is one of the simplest quantum operators, which exhibits basic arithmetic operations such as addition, the propagation of carry qubits and the resetting of carry qubits. In this paper, three quantum incrementer gate circuit topologies are derived and compared based upon their total number of gates, the complexity of the circuits, the types of gates used and the number of carry or ancilla qubits implemented. The first case is a generalized n-qubit quantum incrementer gate with the notation of $(n : 0)$. Two other quantum incrementer topologies are proposed with the notations of $(n : n-1 : \mathrm{RE})$ and $(n : n-1 : \mathrm{RD})$. A general method is derived to decompose complicated quantum circuits into simpler quantum circuits which are easier to manage and physically implement. Due to the cancelation of intermediate unitary gates, it is shown that adding ancilla qubits slightly increases the complexity of a given circuit by the order of $3n$, which pales in comparison to the complexity of the original circuit of the order n^2 without reduction. Finally, a simple application of the generalized n-qubit quantum incrementer gate is introduced, which is related to quantum walks.

Keywords: Quantum circuit synthesis; quantum incrementer gate; geometric quantum computation; quantum walks; ancilla qubits.

PACS numbers: 03.67.Ac, 03.67.Lx, 03.65.Sq

1. Introduction

Quantum logic synthesis[1–12] has been independently observed and studied in various aspects for many years. Remarkably, an essential question left to be answered is how to construct and generalize a single quantum gate to act as a basic element unit for logic synthesis. There are many well known efficient basic quantum gates for generic quantum computation, such as the Hadamard gate,[13] Fredkin gate[14] and Toffoli gate,[15] etc.

The quantum incrementer is one of the simplest quantum operators, which exhibits basic arithmetic operations such as addition, the propagation of carry qubits, and the resetting of carry qubits. Further investigation of the class of quantum incrementer gates can lead to much more efficient and reliable implementation of this type of basic quantum gate. Several advantages are gained by using such a type of quantum gate.[11] One enticing feature is the use of less qubits. Since the ancilla qubits could be reused at the end of a quantum circuit (see the special case of $(n : n-1 : \mathrm{RE})$), the quantum incrementer gate's output qubits reside in a pure state.[16] Furthermore, the reduction of unnecessary qubits to accomplish the same task could allocate more storage space for future quantum computers. Recently, Sakthikumaran *et al.*[17] pointed out that if a novel incrementer circuit was added to the classical Carry Select Adder (CSA) circuits, it could profoundly improve the speed of signal transmission and require less power. Similarly, as early as 1976[18] and 1979,[19] researchers have proposed the concept of an incrementer circuit, which has been successfully applied to the program counter circuit. Using this as an inspiration, the quantum incrementer gate offers the advantage of reducing propagation delay characteristics in quantum logic synthesis.

The implementation of the quantum incrementer gate has wide-ranging potential applications in quantum circuit synthesis,[16,20] quantum algorithm implementation,[21,22] realization of entangling quantum-logic gate,[23,24] and geometric quantum computation,[25–27] etc. It is well known that quantum entanglement is a key resource in quantum mechanics.[28] Quantum entanglement is an essential property for the implementation of a quantum computer.[29] Moreover, entanglement shows up as a necessary aspect in some basic quantum logic gates. Both quantum gate operation and the entangled state share a close relation: on one hand, a quantum gate can generate or increase entanglement,[23,24,30] while on the other, the entangled state can be used to construct a quantum gate operation.[31] In Sec. 3, we briefly discuss the ability of our generalized quantum incrementer gate $(n : 0)$ to generate quantum entanglement. Furthermore, the physical implementation of our generalized quantum incrementer gate $(n : 0)$ could benefit from studying its quantum geometrical phase. In this respect, geometric quantum computation,[26,27,32] a scheme that is widely employed in fault-tolerant quantum gates by implementing geometric phase shifts, could serve as an indispensable tool for investigating the geometric properties of our class of quantum incrementer gates. In this paper, the quantum incrementer gate is used in one simple example related to a "discrete-time" quantum walk.

The paper is structured as follows. In Sec. 2, we describe the mathematical formalism for basic quantum circuit and logic synthesis. The basic definitions of ancilla qubits and complexity analysis are presented. Section 3 provides the main result of our work, which includes three types of quantum incrementer gates: the generalized quantum incrementer $(n : 0)$ and numerical calculations for the two distinct topologies of this quantum incrementer which we refer to as $(n : n-1 : \mathrm{RE})$ and $(n : n-1 : \mathrm{RD})$. Especially, some analysis for quantum incrementer gate $(n : 0)$ with its matrix forms are listed. Meanwhile, we also give out a short insight about the quantum incrementer gates' entangling properties. Furthermore, we analyze and compare this class of quantum incrementer gates. In Sec. 4, we present a simple application of this quantum incrementer gate for quantum walk. Finally, Sec. 5 contains our conclusions.

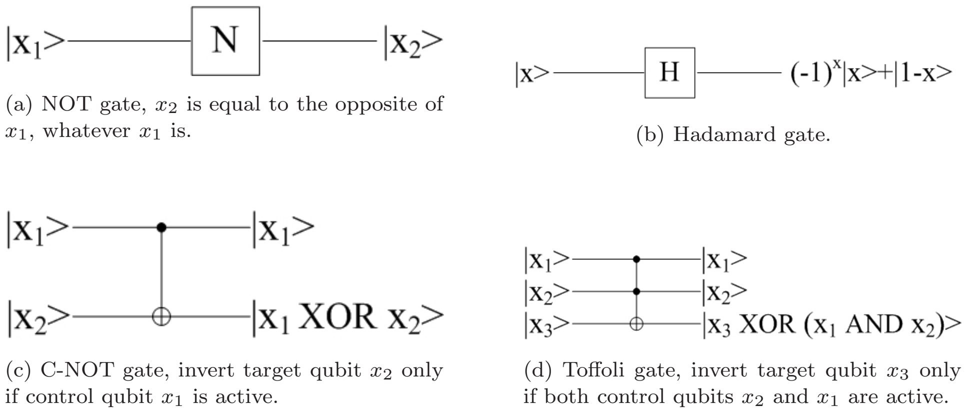
2. Preliminaries

The fundamental logic element in quantum theory is the qubit.[6] The qubit can be in a quantum state of $|0\rangle$, $|1\rangle$ or a complex linear superposition of the two orthonormal computational basis states: $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$. Qubits can be prepared in superposition states allowing for massively parallel quantum information processing and quantum logic synthesis.

2.1. *Quantum logic gate*

A quantum logic gate is a logic unit which performs a fixed unitary operation upon some subset of qubits in a finite amount of time. The most common quantum gates are the NOT gate, the Hadamard gate, the CNOT gate and the Toffoli gate[1] (see Fig. 1).

In certain scenarios, it is more convenient to express the quantum logic gates in terms of their matrix forms. Here we list some general quantum logic gates' matrix forms (see Table 1). We also present the matrix form of our generalized quantum incrementer gate $(n : 0)$ in order to better describe its entangled properties, which is very important for quantum computation and quantum information.



(a) NOT gate, x_2 is equal to the opposite of x_1, whatever x_1 is.

(b) Hadamard gate.

(c) C-NOT gate, invert target qubit x_2 only if control qubit x_1 is active.

(d) Toffoli gate, invert target qubit x_3 only if both control qubits x_2 and x_1 are active.

Fig. 1. Some basic quantum gates.

Table 1. The matrix form of some elementary quantum logic gates.

Gate	Hadamard gate	CNOT gate	Swap gate	Toffoli gate
Matrix	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

2.2. *Ancilla qubits*

Ancilla qubits[11] are initialized to $|0\rangle$ at the beginning and are usually reset to $|0\rangle$ at the end of a quantum circuit, so that the output qubits are not affected and the ancilla qubits could be reused for some special purposes. When synthesizing the quantum circuits, we always use ancilla qubits to reduce the complexity and the depth of the circuits. There exists a wide variety of applications for ancilla qubits[8–10] in the literature.

2.3. *Quantum circuits and complexity analysis*

A quantum circuit[16] is a sequence of quantum logic gates ordered into layers based on the evolution of time. The gates are consecutively applied in accordance with the order of the layers. Gates in one layer can be applied in parallel. Each layer represents a finite state of the entire system. The depth of a quantum circuit is the number of layers and the size is the amount of quantum logic gates. Since unitary evolution of a quantum logic gate is reversible, reversibility is a basic property of quantum circuits. A circuit can solve problems of a finite size, so we define families of circuits consisting of one circuit for each input size. Complexity theory is concerned with the inherent cost of computation in terms of memory usage, some designated elementary operations, or the circuit's size. A good example is the quantum Fourier transform (QFT) defined by the computational basis as below:

$$|y\rangle \mapsto 2^{-n/2} \sum_{x} e^{i\frac{2\pi}{2^n}yx}|x\rangle \tag{1}$$

A general unitary evolution of an n-qubit QFT circuit can be constructed with one or two qubits operated upon by different quantum gates. Furthermore, the QFT can also be implemented with three qubits or four qubits with the Hadamard gates (the circuit in this case is depicted in Fig. 2). The general case of n-qubits requires a trivial extension of the aforementioned QFT circuit following the same sequence of Hadamard gates and B gates. The QFT circuit operating on n-qubits contains n Hadamard gates and $n(n-1)/2$ phase shift B gates; in total $n(n+1)/2$ elementary gates. The evolution process shows us the original analysis of complexity. In essence,

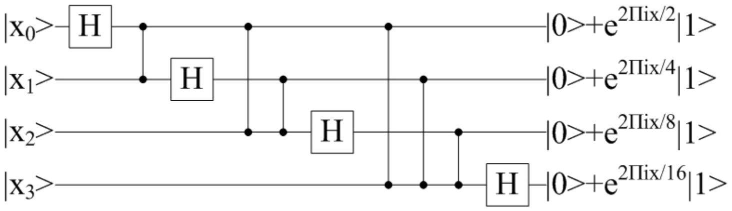


Fig. 2. A general four-qubit QFT circuit.

complexity requires a uniform and polynomial-size circuit family ($O(n^d)$ for some constant d).[33] The highest degree of the polynomial is the complexity of this circuit. Thus, the complexity of the QFT is $O(n^2)$.

3. A Family of Generalized *n*-Qubit Quantum Incrementer Gates

Once we construct an elementary family of quantum incrementer gates, they will become the general elements used in quantum circuits which also perform the same functionality. Recall that the Toffoli gate and the CNOT gate are both basic elements for quantum computation. However, their combination also performs a universal function known as the quantum adder (shown in Fig. 3). As a result, the quantum adder is also treated as a basic element in quantum computation. Similarly, the quantum incrementer is a class of generalized CNOT gates which can operate on n qubits producing cyclic permutation in the 2^n bit-string states,[20] shown in Fig. 4.

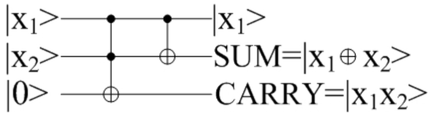


Fig. 3. Quantum adder.

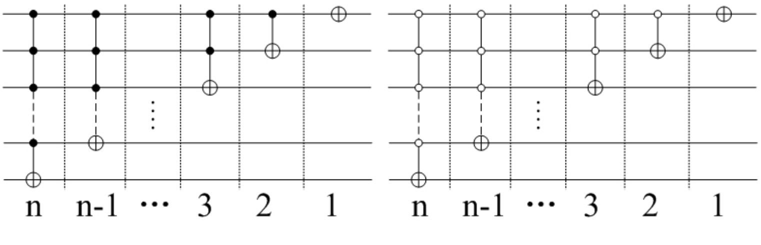


Fig. 4. n-qubit quantum incrementer and decrementer gates.

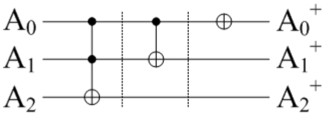


Fig. 5. Three-qubit incrementer gate.

3.1. *Basic case of quantum incrementer gate* $(n : 0)$

The function of the quantum incrementer is to add one to the current value (or state) of the input register and send the result (e.g., next state) to the output register. In Fig. 4, the solid black circle of the incrementer gate as the control qubit indicates that when it is 1, the value of the target qubit will be reversed. Conversely, the hollow circle of the decrementer gate as the control qubit means that when it is 0, the target qubit will be reversed.

As a simple example, a three-qubit incrementer gate without any carry (e.g., ancilla) qubits is shown in Fig. 5. We represent the current state of the three-qubit register by: $A_2A_1A_0$ and its corresponding next state by: ${A_2}^+{A_1}^+{A_0}^+$. The current state is the three-qubit number located at the input register before incrementation occurs and the next state is the resulting number located at the output register after incrementation is performed. The dashed vertical lines show there are three stages involved in the process. The logical function expressions are obtained easily as the following:

$$\begin{aligned} {A_2}^+ &= A_2 + A_1 * A_0\,, \\ {A_1}^+ &= A_1 + A_0\,, \\ {A_0}^+ &= /A_0\,. \end{aligned} \tag{2}$$

Here, "+", "*" and "/" represent Boolean logic XOR, AND and NOT gates, respectively. Let $A_2A_1A_0 = 011$ be the current state residing in the input register. The various gate operations performed on this current state value are as follows:

$$\begin{aligned} {A_2}^+ &= A_2 + A_1 * A_0 = 0 + 1 * 1 = 0 + 1 = 1\,, \\ {A_1}^+ &= A_1 + A_0 = 1 + 1 = 0\,, \\ {A_0}^+ &= /A_0 = /1 = 0\,. \end{aligned} \tag{3}$$

The result ${A_2}^+{A_1}^+{A_0}^+ = 100$ is the next state which resides in the output register. This example provides a concise description of the basic operation of the quantum incrementer operator. The generalized n-qubit quantum incrementer gate is the basic case in the incrementer gate family. We use the notation $(n : 0)$ to represent it, where $n \geq 3$, $n \in N^+$.

As for the n-qubit case, if the input state is $|x_n\rangle$ (binary value), then by virtue of the $(n : 0)$ gate's operation, the output should be $(|x_n + 1\rangle)$ (binary value). According to this, suppose that the $(n : 0)$ quantum incrementer gate's matrix

Table 2. Truth table for the (3 : 0) quantum incrementer gate.

Input	$\|000\rangle$	$\|001\rangle$	$\|010\rangle$	$\|011\rangle$	$\|100\rangle$	$\|101\rangle$	$\|110\rangle$	$\|111\rangle$
Output	$\|001\rangle$	$\|010\rangle$	$\|011\rangle$	$\|100\rangle$	$\|101\rangle$	$\|110\rangle$	$\|111\rangle$	$\|000\rangle$

form is A. Then, we can derive the linear equations and result as listed below:

$$A|x_n\rangle = |x_n + 1\rangle\,, \quad n \geq 3\,, \quad n \in N^+\,. \tag{4}$$

The matrix A is the mathematical expression of the generalized $(n : 0)$ quantum incrementer gate. It is known that $|0\rangle$ is the Ket vector $(1\ \ 0)^T$, $|1\rangle$ is the Ket vector $(0\ \ 1)^T$. Thus, for n-qubit input, the initial state is a Ket vector with the size of $2^n * 1$ by n times tensor product. Sequentially, the size of matrix A is $2^n * 2^n$. Since the size of this generalized $(n : 0)$ quantum incrementer gate's matrix is huge, we shall instead focus on a simple example with finite matrix size, such as three-qubits, which could be extended to the n-qubit case without loss of generality. For the three-qubit quantum incrementer gate $(3 : 0)$, its quantum truth table is shown in Table 2. Its corresponding matrix is A_{8*8} as shown below. From Eq. (4) and Table 2, we can easily verify that A_{8*8} is correct.

$$A_{8*8} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Furthermore, it is important to discuss the entangled properties of quantum logic gates, which has been an area of intense research activity in the past decades.[23,24] It is known that some quantum logic gates can produce entanglement and vice versa. As a canonical example, if the CNOT gate operates on an initial state of $(|0\rangle + |1\rangle)|0\rangle/\sqrt{2}$, then we result with the maximally entangled state of the two qubits. In contrast, if we operate on an initial state $|0\rangle|0\rangle$, then after the CNOT gate's operation the output state remains a separable state. Thus, quantum logic gates can produce entanglement, however, it greatly depends upon the choice of the initial state. Similarly, the quantum incrementer gate $(n : 0)$ can generate entanglement. Since it operates on different initial states, it will produce various results. Taking the three-qubit quantum incrementer gate $(3 : 0)$ as a simple example, let it operate on the initial states $|0\rangle|0\rangle|1\rangle$ and $|0\rangle(|1\rangle + |0\rangle)|1\rangle/\sqrt{2}$ separately. Then, we result with the output states $|0\rangle|1\rangle|0\rangle$ and $(|10\rangle + |01\rangle)|0\rangle/\sqrt{2}$. Whereas the first state is still separable, the second state is partially entangled. From this simple three-qubit example, it is evident that the generalized quantum incrementer gate $(n : 0)$ is capable of generating quantum entangled states.

In addition to the general n-qubit quantum incrementer operator $(n : 0)$, the operator's topologies are also derived. Two special cases of quantum incrementer gates of $(n : n-1 : \mathrm{RE})$ and $(n : n-1 : \mathrm{RD})$ with full ancilla qubits are discussed in the next section. Meanwhile, the notation (N : M : RE/RD) is used in this paper to distinguish the various topologies of quantum incrementer circuits following the format:

(N-number of qubits: M-number of carry qubits: *R*eset carry qubits *E*n/*D*isabled)

3.2. *The first topology of the quantum incrementer gate* $(n : n-1 : \mathbf{RE})$

As shown in Fig. 4, the generalized n-qubit quantum incrementer gate contains n different gates: $T(n), T(n-1), \ldots, T(3)$, Toffoli gate, CNOT gate and NOT gate without performing any circuit reduction. Even though the $(n : 0)$ quantum incrementer consists of fewer quantum gates, the complexity of the gates used are very high. An interesting approach would be to reduce these complex gates into their simpler constituent elementary gates, such as the CNOT gate, the Toffoli gate and the Hadamard gate.

The $(n : n-1 : \mathrm{RE})$ circuit is an n-qubit quantum incrementer gate with $n-1$ carry (e.g. ancilla) qubits which are reset upon reaching the next state in the output register. Figure 4 shows us the generalized n-qubit quantum incrementer gate $(n : 0)$. Now we shall focus on how to reduce this generalized n-qubit quantum incrementer gate $(n : 0)$ into one of its topologies called the $(n : n-1 : \mathrm{RE})$, which features full ancilla qubits and full reset qubits. The so-called "full ancilla qubits" means that we add one qubit between every two control qubits. We use the following three principles to support the circuit's reduction from $(n : 0)$ to $(n : n-1 : \mathrm{RE})$.

- Principle 1: the truth-table technology. As a basic mathematical tool, truth-table is used in logic, particularly for Boolean functions and binary logic.
- Principle 2: Unitary operator decomposition and reduction. Unitary transformations satisfy the condition:

$$UU^{-1} = U^{-1}U = I\,. \tag{5}$$

- Principle 3: "Mimicking effect".

A simple example is shown in Fig. 6. We can prove the "mimicking effect" for two equivalent circuits by their truth-table (see Tables 3 and 4). It is evident that Tables 3 and 4 are equivalent. Based on this example, the complex circuits can be decomposed into simpler circuits. By deleting certain auxiliary qubits, the "mimicking effect" can be used for the circuit's reduction. In order to get the optimized $(n : n-1 : \mathrm{RE})$ circuit, we can decompose the original complex circuit by using the "mimicking effect". Figure 7 shows the circuit of decomposed generalized

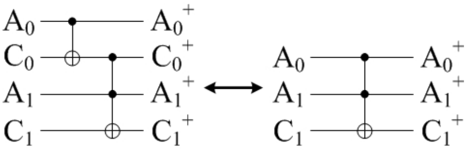


Fig. 6. “Mimicking effect” for two equivalent circuits.

Table 3. Truth table of left circuit in Fig. 6.

C_1	A_1	C_0	A_0	C_1^+	A_1^+	C_0^+	A_0^+
0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	1	0	0	1	1
0	1	0	1	1	1	1	1

Table 4. Truth table of right circuit in Fig. 6.

C_1	A_1	A_0	C_1^+	A_1^+	A_0^+
0	0	0	0	0	0
0	1	0	0	1	0
0	0	1	0	0	1
0	1	1	1	1	1

n-qubit quantum incrementer gate with full ancilla qubits and full reset qubits in terms of Toffoli, CNOT and NOT gates. There are n stages in total, where stage n to 1 correspond to $T(n)$, $T(n-1), \ldots, T(3)$, Toffoli gate, CNOT gate and NOT gate of the original circuit (refer to Fig. 4). In this circuit topology, there are $n-1$ ancilla qubits for n-qubits in the quantum register. By using $(n-1)$ ancilla qubits, the simple computation of the gate's number is n^2. So the complexity of the circuit is of the order n^2, while it contains three types of simpler and elementary quantum gates: the Toffoli gate, the CNOT gate and the NOT gate.

This particular circuit topology ($n : n-1 :$ RE) requires that the carry bit C_i ($i = 0, 1, n-2$) be reset upon its final state. Since the last operation is to determine the most-significant bit, A_{n-1}, we can now undo the carry propagations that led up to determining the next state of A_{n-1} and thus reset the carry bits by performing reverse (e.g., unitary) operations with each corresponding gate. This is possible because all of these quantum gates are unitary in nature. Thus, after having determined the most-significant qubit value which is sent to the output register, we reset the carry propagations by proceeding in reverse order and mirror-imaging each previous gate. This is the so-called “reset qubit enabled” option in this ($n : n-1 :$ RE) topology. Because of the gates' unitary property (according to Principle 2), it is possible to annihilate the adjacent (e.g., nearest-neighbor)

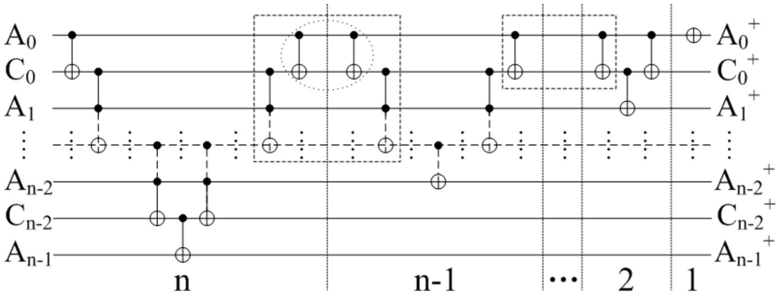


Fig. 7. Generalization of n-qubit quantum incrementer gate of topology $(n : n-1 : \mathrm{RE})$.

mirror-image unitary gates:

$$\begin{aligned} \mathrm{NOT} * \mathrm{NOT}^{-1} &= I\,, \\ \mathrm{CNOT} * \mathrm{CNOT}^{-1} &= I\,, \\ \mathrm{Toffoli} * \mathrm{Toffoli}^{-1} &= I\,. \end{aligned} \tag{6}$$

Figure 7 shows the $(n : n{-}1 : \mathrm{RE})$ circuit with the specific adjacent unitary gates (two reverse CNOT gates and $T(n)$ gates are highlighted with black gridlines) to be annihilated. This reduced circuit (shown in Fig. 8) exhibits fewer gates and results in a simpler and possibly faster circuit. Due to the annihilation of the adjacent mirror-image unitary gates, between every two stages there are $2(n-2)$ gates reduced. Summing up all reduced gates is $2[(n-2)+(n-3)+\cdots+1] = n^2-3n+2$. The total amount of gates in Fig. 8 is $3n-2$. It is evident that using ancilla qubits increases the complexity of the initial circuit shown in Fig. 4 by an order of n. However, the increase in complexity by using carry qubits is not too costly. Recall that the generalized n-qubit quantum incrementer gate needs n different gates to be synthesized, whereas the $(n : n-1 : \mathrm{RE})$ circuit topology only requires three universal gates. Furthermore, by simple reduction of the circuit in Fig. 7, we can achieve a complexity for the reduced circuit to $O(n)$ instead of $O(n^2)$. This general

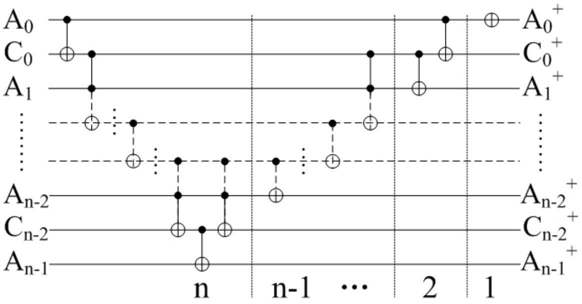


Fig. 8. Circuit layout for the $(n : n-1 : \mathrm{RE})$ after annihilation of $n-2$ specified sets of "mirror-imaged" unitary gates.

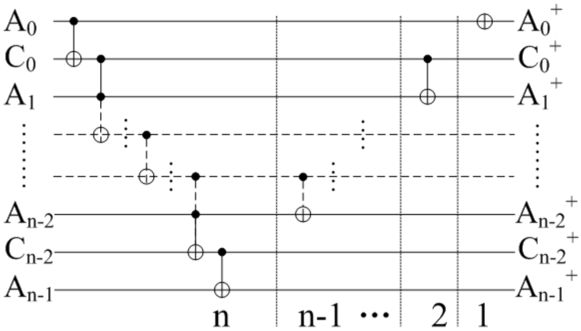


Fig. 9. Circuit layout for the $(n : n-1 : \text{RD})$ when the carry qubits are not reset after incrementation.

method is derived to decompose complicated circuits into simpler circuits which are easier to manage and physically implement.

3.3. *The other topology of the quantum incrementer gate* $(n : n-1 : \textbf{RD})$

Finally, the $(n : n-1 : \text{RD})$ circuit is an n-qubit quantum incrementer gate with $n-1$ carry bits which are not required to be reset upon reaching the final state. The circuit layout for $(n : n-1 : \text{RD})$ after annihilation also outlines a few implicit cancellations of Toffoli gates and CNOT gates. Recall that there are $n-1$ ancilla qubits as we proceed towards determining the most-significant bit, A_{n-1}. As we reset the carry qubits by proceeding backwards, we pass another $n-1$ ancilla qubits. If for some reason, we do not care about the carry qubit, then we do not have to reset these ancilla qubits. At the end of every carry propagation process of the ancilla qubit C_i, the reset enabled gate is Toffoli gate of CNOT gate, just as the gate in every dash box in Fig. 8. However, now we can delete $(n-1)$ gates in total since we will not reset the carry qubits for the case of $(n : n-1 : \text{RD})$. As a result, the amount of the gates used in the synthesis of the $(n : n-1 : \text{RD})$ circuit is $2n-1$, shown in Fig. 9. Similar to the case of $(n : n-1 : \text{RE})$, there are three basic gates: the Toffoli gate, the CNOT gate and the NOT gate.

3.4. *Analysis and discussion*

We introduced three different topologies for the quantum incrementer gate. Table 5 shows that the generalized quantum incrementer circuit $(n : 0)$ and its two special case of $(n : n-1 : \text{RE})$ and $(n : n-1 : \text{RD})$ differ in the number of carry qubits, the number of gates, the types of gates, and their complexity. "NA" indicates infinite complexity, because $T(n)$ complexity tends to infinity. Due to the cancellation of intermediate unitary gates, it is shown that adding ancilla qubits slightly increases the complexity of a given circuit by the order of $3n$, which pales in comparison

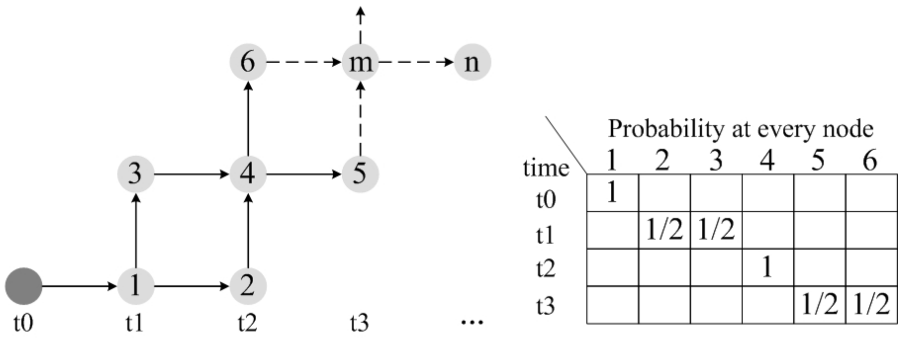
Table 5. Comparison among the topological quantum incrementer gates.

Type of circuits	$(n:0)$	$(n:n-1:\mathrm{RE})$		$(n:n-1:\mathrm{RD})$
Amount of gates	n	n^2	$3n-2$	$2n-1$
Number of carry qubits	0	$n-1$	$n-1$	$n-1$
Number of distinct gates used	n	3	3	3
Type of gates	NOT Gate CNOT Gate Toffoli Gate $T(n), \ldots, T(3)$ Gate	NOT Gate CNOT Gate Toffoli Gate	NOT Gate CNOT Gate Toffoli Gate	NOT Gate CNOT Gate Toffoli Gate
Complexity	NA	$O(n^2)$	$O(n)$	$O(n)$

to the complexity of the original generalized n-qubit quantum incrementer circuit $(n:0)$ of the order n^2 without any circuit reduction. The optimized circuit type is $(n:n-1:\mathrm{RD})$ which uses the fewest amount of gates, although it features the same complexity as the reduced circuit $(n:n-1:\mathrm{RE})$.

4. Application of the Generalized n-Qubit Quantum Incrementer Gate $(n:0)$ for Quantum Walk

In this section, we briefly introduce a potential application for the generalized case of quantum incrementer gate $(n:0)$ based on the quantum walks. These circuits are not optimized and serve only to provide an insight into quantum parallel computation circuits. Quantum walks are the analog of classical random walks with the caveat that the quantum walker can be in a superposition of positions. This superposition property of quantum mechanics can provide polynomial speed up, for quantum algorithms compared to classical algorithms. Whereas a classical random walk simulates the random movement of a particle around a graph, the principle of the quantum walker's current state is described by a probability distribution over a superposition of positions. As shown in Fig. 10, it is evident that in the classical random walks, the walker can reach state "5" or "6" by taking path "2" or "3" with equal probabilities.



time	1	2	3	4	5	6
t0	1					
t1		1/2	1/2			
t2				1		
t3					1/2	1/2

Fig. 10. A classical random walk on a graph.

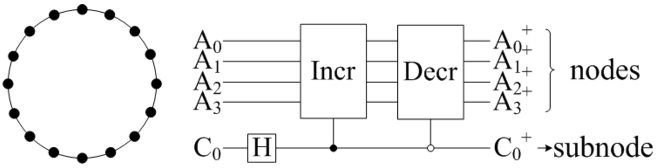


Fig. 11. A quantum walk implemented along a 16-node cycle.

A quantum walk in discrete-time is specified by a coin and shifting operator, which are applied repeatedly and are described by a unitary operator $U = SC$. The operation of S and C can be illustrated as follows,[34] in which a coin operator can result in $|L\rangle$ or $|R\rangle$, and $|P\rangle$ is a position register.

A coin operator C: $|L\rangle \to |L\rangle + i|R\rangle$ or $|R\rangle \to i|L\rangle + |R\rangle$.

A shifting operator S: $|L\rangle|P\rangle \to |L\rangle|P-1\rangle$ or $|R\rangle|P\rangle \to |R\rangle|P+1\rangle$.

These are both unitary operations, and hence their linear combination is also unitary. The generalized quantum incrementer gate consists of generalized CNOT gates. For the basic case of n-qubit, the number of distinct elementary gates required is limited to $O(n)$. As mentioned in the previous section, the generalized n-qubit quantum incrementer gate $(n : 0)$ could produce cyclic permutations depending on the specific implementation used. The same principle holds for the decrement circuit shown in Fig. 4. Although the complexity of n generalized CNOT gates is not low, their use can ensure that the gates required to perform a quantum walk grows logarithmically with the size of the state space. The shifting operator can be represented as $S = (\text{Incr.} \otimes |1\rangle + \text{Decr.} \otimes |0\rangle)$. The coin operator can be implemented by a single Hadamard gate acting on the ancilla qubit, where one ancilla qubit represents two subnodes in the quantum walk. Correspondingly, each state in the shifting operation is mapped to an adjacent state of the nodes.

Figure 11 shows a simple example for a 16-node cycle in which a quantum walk is implemented by the use of “increment” and “decrement” circuits. In this cycle each node has two adjacent edges, and hence two subnodes. For a cycle of order $2^n = 16$, it requires 4 qubits to encode the 16 nodes, and an additional qubit to encode the 2 subnodes. For the generalized n-qubit case, implementation of a quantum walk along a cycle of size 2^n requires $n + 1$ qubits in total. Thus, the complexity of this synthesis is limited to $O(n)$. Also, there exists more complex scenarios that can be efficiently implemented such as composites of highly symmetric graphs.[21,35] These studies have investigated the computational complexity of such searches using quantum walks which resulted in a quadratic speedup over classical search algorithms. These circuits are not optimal, however they give a good insight of quantum parallel computation circuits.

5. Conclusions

In summary, this paper investigated a class of n-qubit quantum incrementer gates. The generalized n-qubit case with the notation of $(n : 0)$ was first derived. Then, the two topological cases of $(n : n-1 : \text{RE})$ and $(n : n-1 : \text{RD})$ with full ancilla qubits were analyzed. The three main quantum incrementer gates were compared based upon the number of quantum gates, the number of carry qubits and the types of gates implemented. Afterwards, a general method was derived to decompose complicated circuits into simpler circuits which are easier to manage and physically implement. Any one of these three quantum incrementer circuit topologies $((n : 0), (n : n-1 : \text{RE}), (n : n-1 : \text{RD}))$ can be used for circuit synthesis depending on the particular design purpose. Implementing certain quantum algorithms, such as Shor's algorithm, with quantum circuits involves many complex operations such as addition and modular exponentiation. The variety and complexity of the quantum gates involved can be daunting. The quantum incrementer gate is the simplest nontrivial quantum circuit which exhibits basic arithmetic operations such as addition, carry propagation and the reset of carry, or auxiliary, qubits. These operations are ubiquitous in quantum computing; thus, an investigation of this simplest case, the quantum incrementer gate, offers fundamental insight which can be used to build more complicated circuits. Here, we briefly introduced the quantum walk as a simple application of the basic generalized n-qubit quantum incrementer gate $(n : 0)$. Further studies of the use of the two special cases of $(n : n-1 : \text{RE})$ and $(n : n-1 : \text{RD})$ are left as future work. How to map the ancilla qubits to the quantum walk's node or subnode is still an open question. Our research into quantum incrementer gates has provided a glimpse of their potential applications to quantum circuit synthesis and the quantum algorithm optimization process.

Acknowledgments

We would like to thank Selman Hershfield and Fengbo Ren for stimulating discussions. This work is supported by the Fundamental Research Funds for the Central Universities under Grant No. ZYGX2011YB022, and in part by the National Science Foundation of China under Grant Nos. 61272175 and 60773205.

References

1. A. Barenco *et al.*, *Phys. Rev. A* **52**, 3457 (1995).
2. N. D. Mermin, Notes for physicists on the theory of quantum computation, *Informal Notes for Three Lectures at LASSP Autumn School of Quantum Computation, Cornell* (1999).
3. N. D. Mermin, *Quantum Computer Science: An Introduction*, Vol. 38 (Cambridge University Press, UK, 2007).
4. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, UK, 2010).
5. B. Hladky, G. Drobny and V. Buzek, *Phys. Rev. A* **61**(2), 022102 (2000).
6. A. N. Al-Rabadi, *Reversible Logic Synthesis: From Fundamentals to Quantum Computing* (Springer, Berlin, 2004).
7. V. V. Shende, S. S. Bullock and I. L. Markov, *IEEE Trans. on CADICS* **25**(6), 1000 (2006).
8. G. W. Yang *et al.*, *The Comput. J.* **51**(2), 207 (2008).

9. G. W. Yang *et al.*, *J. Phys. A. Math. Gen.* **38**, 9689 (2005).
10. X. Y. Li, G. W. Yang and D. S. Zheng, *The WCCI 2010 IEEE World Congress on Computational Intelligence* (Barcelona, Spain, 2010), pp. 4239–4242.
11. C. Moore and M. Nilsson, *SIAM J. Comput.* **31**(3), 799 (2002).
12. M. McGettrick and B. Murphy, Simulation of the CCC-Not quantum gate, *Technical Report NUIG-IT-061002*, Department of Information Technology (NUI, Galway, 2002).
13. C. Y. Lu *et al.*, *Nat. Phys.* **3**(2) 91 (2007).
14. J. A. Smolin and D. P. DiVincenzo, *Phys. Rev. A* **53**(4), 2855 (1996).
15. D. Maslov and G. W. Dueck, *Electron. Lett.* **39**(25), 1790 (2003).
16. P. Hoyer and R. Spalek, *Lect. Notes Comput. Sci.* **2607**, 234 (2003).
17. S. Sakthikumaran *et al.*, *Electron. Comput. Technol.* (*ICECT*)*, 2011 3rd Int. Conf. IEEE* **1**, 273 (2011).
18. T. Kihara, Binary incrementer circuit, *U.S. Patent* 3,989,940 (1976).
19. E. Kudou, Incrementer circuit, *U.S. Patent* 4,153,939 (1979).
20. B. L. Douglas and J. B. Wang, *Phys. Rev. A* **79**(5), 052335 (2009).
21. N. Shenvi, J. Kempe and K. B. Whaley, *Phys. Rev. A* **67**(5), 052307, (2003).
22. D. Browne, E. Kashefi and S. Perdrix, *Theory Quantum Comput. Commun. Cryptogr.* **35**, (2011).
23. F. Schmidt-Kaler *et al.*, *Nature* **422**, 408 (2003).
24. O. Gazzano *et al.*, *Phys. Rev. Lett.* **110**, 250501 (2013).
25. Z. S. Wang *et al.*, *Phys. Rev. A* **76**, 044303, (2007).
26. Z. S. Wang, *Phys. Rev. A* **79**, 024304 (2009).
27. Z. S. Wang, G. Q. Liu and Y. H. Ji, *Phys. Rev. A* **79**, 054301 (2009).
28. A. K. Ekert, *Phys. Rev. Lett.* **67**(6), 661 (1991).
29. D. P. DiVincenzo, *Fortschr. Phys.* **48**, 771 (2000).
30. P. Zanardi, C. Zalka and L. Faoro, *Phys. Rev. A* **62**, 030301 (2000).
31. M. Y. Ye, Y. S. Zhang and G. C. Guo, *Phys. Rev. A* **69**, 022310 (2004).
32. L. B. Shao, Z. D. Wang and D. Y. Xing, *Phys. Rev. A* **75**, 014301 (2007).
33. C. H. Papadimitriou, *Computational Complexity* (John Wiley and Sons Ltd, US, 2003).
34. A. M. Childs *et al.*, Exponential algorithmic speedup by quantum walk, *Proc. 35th ACM Symposium on Theory of Computing* (2003), pp. 59–68.
35. D. Reitzner *et al.*, *Phys. Rev. A* **79**, 012323 (2009).