IEICE TRANS. FUNDAMENTALS, VOL.E92–A, NO.5 MAY 2009

INVITED PAPER *Special Section on Frontier of Quantum Computing*

Quantum Arithmetic Circuits: A Survey

Yasuhiro TAKAHASHI[†a)], ***Member***

SUMMARY Quantum circuits for elementary arithmetic operations are important not only for implementing Shor's factoring algorithm on a quantum computer but also for understanding the computational power of small quantum circuits, such as linear-size or logarithmic-depth quantum circuits. This paper surveys some recent approaches to constructing efficient quantum circuits for elementary arithmetic operations and their applications to Shor's factoring algorithm. It covers addition, comparison, and the quantum Fourier transform used for addition.

key words: *quantum circuits, elementary arithmetic operations, Shor's factoring algorithm*

1. Introduction

The factoring problem is one of finding a non-trivial factor of a given composite number. It is widely believed that the problem is nearly impossible to solve efficiently on a classical computer. On the other hand, Shor's algorithm proposed in 1994 solves the problem efficiently on a quantum computer [1]. This means that not only would a quantum computer be faster than a classical computer but it would also be able to break cryptosystems whose security is based on the factoring problem. An example of such a cryptosystem is RSA, which is widely used in electronic commerce protocols.

One of the key ingredients of Shor's factoring algorithm is the modular exponentiation operation, which is decomposed into elementary arithmetic operations such as addition. The important problem is how to construct efficient quantum circuits for such operations. From the practical standpoint, the solutions will contribute to implementing Shor's factoring algorithm on a quantum computer [2], [3]. Moreover, they will be useful for analyzing the relationships between the computational power of a quantum computer and the security of cryptosystems [4]–[7]. On the theoretical side, the study of the problem contributes to our understanding of the computational power of small quantum circuits [8], [9].

This paper surveys some recent approaches to constructing efficient quantum circuits for elementary arithmetic operations and their applications to Shor's factoring algorithm. It covers addition, comparison, and the quantum Fourier transform (QFT) used for addition in Shor's factoring algorithm [10]. The measures of the complexity of a quantum circuit are its size and depth and the number of qubits in it. Though they are defined exactly in Sect. 2, roughly speaking, the size, depth, and the number of qubits correspond to the computation time, the computation time when operations in the circuit are performed as parallely as possible, and the size of the memory, respectively.

When a quantum circuit is constructed, it is usually assumed that arbitrary pairs of qubits can interact. That is, the circuit is designed for an architecture in which such interaction occurs is possible. This architecture is called a general architecture. On the other hand, as discussed in [2], many proposed quantum computer architectures deal with an unidimensional array of qubits with nearest neighbor interactions only. Thus, it is important for a circuit to work on such a linear nearest neighbor (LNN) architecture. Though most of the circuits in this paper are designed for a general architecture, Sect. 5 deals with quantum circuits for the QFT on an LNN architecture.

There have been many attempts to construct efficient classical circuits for elementary arithmetic operations [11]. Are they useful for solving our problem? When the number of qubits in a quantum circuit is not a concern, it is easy to transform an efficient classical circuit into an efficient quantum one. More precisely, a depth-efficient or size-efficient classical circuit can be transformed into a depth-efficient or size-efficient quantum one by replacing a classical operation in the classical circuit with a corresponding quantum (in fact, classical reversible) operation. However, it is important to minimize the number of qubits in a quantum circuit since it seems extremely difficult to realize a quantum computer with many qubits. Thus, the number of qubits is an important consideration. In this situation, it is not obvious whether efficient classical circuits are useful for constructing efficient quantum ones. This paper mainly addresses the situation by focusing on the construction of quantum circuits with few qubits.

The rest of this paper is organized as follows. Section 2 describes the fundamental notions of quantum circuits and Shor's factoring algorithm. Section 3 describes how to construct a quantum circuit for addition with few qubits. Section 4 describes how to construct a quantum circuit for comparison that is useful for Shor's factoring algorithm with few qubits. Section 5 describes how to construct a small-size quantum circuit for the QFT. Section 6 gives open problems.

Manuscript received December 11, 2008.
Manuscript revised January 26, 2009.

[†]The author is with NTT Communication Science Laboratories, NTT Corporation, Atsugi-shi, 243-0198 Japan.
a) E-mail: takahasi@theory.brl.ntt.co.jp

DOI: 10.1587/transfun.E92.A.1276

2. Preliminaries

It is assumed that the reader is somewhat familiar with basic concepts of quantum computation. For the details, see Nielsen and Chuang's book [12].

2.1 Quantum Circuits

A quantum circuit consists of wires, quantum gates, and measurement gates for qubits, unitary operations, and quantum measurements, respectively. In a quantum circuit diagram (also called a quantum circuit), a wire is represented by a horizontal line and a quantum gate is represented by a symbol of the quantum operation on wires to which the quantum operation is applied. Information flows through the circuit from left to right. For the purposes of this survey, measurement gates are not used in quantum circuit diagrams.

To give an example of a quantum circuit, let us define the QFT on n qubits. Let b be an n-bit binary number and $b_{n-1}\cdots b_0$ be the binary representation for b. The QFT is defined as

$$\mathrm{QFT}|b_{n-1}\rangle\cdots|b_0\rangle = \frac{1}{\sqrt{2^n}}|\varphi_{n-1}(b)\rangle\cdots|\varphi_0(b)\rangle,$$

where

$$|\varphi_k(b)\rangle = |0\rangle + e^{2\pi i \sum_{j=0}^{k} \frac{b_{k-j}}{2^{j+1}}}|1\rangle$$

for $0 \le k \le n-1$. In contrast to the usual definition, the output state is not permuted since the above definition is used in Draper's addition circuit described in Sect. 3 [10]. Needless to say, the effect of the operation on a superposition state is determined by the linearity.

A quantum circuit for the QFT on three qubits is depicted in Fig. 1. It represents the following procedure. First, a Hadamard operation H is applied to the top qubit representing b_2. Next, a controlled-R_2 operation is applied to the top and middle qubits, where

$$R_k|x\rangle = e^{2\pi i x/2^k}|x\rangle$$

for $2 \le k \le 3$ and $x \in \{0, 1\}$. Then, operations, such as a controlled-R_3 operation, are applied and the output state is obtained. In a similar way, a quantum circuit for the QFT on n qubits can be constructed by using a controlled-R_k operation for $2 \le k \le n$. This is Coppersmith's construction method [13].

One-qubit and controlled-NOT (CNOT) gates are universal for quantum computation in the sense that any unitary operation can be implemented exactly by a quantum circuit using only those gates. Thus, one-qubit and two-qubit gates are universal in the same sense and, for simplicity, they are used as elementary gates. A doubly-controlled-NOT (Toffoli) gate and a triply-controlled-NOT gate are sometimes regarded as elementary gates. As described below, the measures of the complexity of a quantum circuit depend on the choice of elementary gates. However, regarding these gates as elementary gates only affects the complexity of a quantum circuit by a constant.

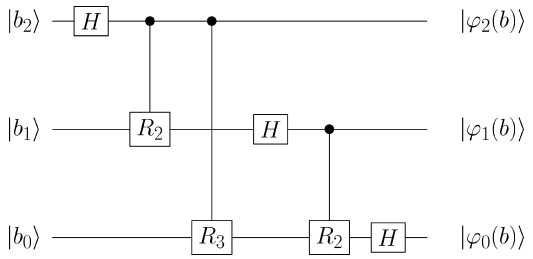

Fig. 1 A quantum circuit for the QFT on three qubits. Normalization factors of $1/\sqrt{2}$ in the output are not shown.

As mentioned earlier, the measures of the complexity of a quantum circuit are the number of qubits and its size and depth. The meaning of the number of qubits is obvious. The size of a circuit is defined as the total number of elementary gates in it. The depth of a circuit is defined as follows. Input qubits are considered to have depth 0. For each gate G, the depth of G is equal to 1 plus the maximal depth of a gate that G depends on. The depth of a circuit is equal to the maximal depth of a gate in it. Intuitively, the depth is the number of layers in the circuit, where a layer consists of gates on different qubits that can be performed simultaneously. For example, in Fig. 1, the number of qubits is 3 and the size is 6. The depth is 5 since the controlled-R_3 operation applied to the top and bottom qubits and the Hadamard operation applied to the middle qubit can be performed simultaneously. In general, Coppersmith's circuit for the QFT on n qubits uses only n qubits and its depth and size are $O(n)$ and $O(n^2)$, respectively [14]. It is known that, when R_k is ignored for all $k > m$ in the circuit, the resulting circuit computes a good approximation of the QFT and only the size decreases from $O(n^2)$ to $O(n \log n)$, where $m \in O(\log n)$ [13]. More precisely, the good approximation of the QFT achieves arbitrary inverse polynomial precision. The details of the precision can be found in [8]. A quantum circuit can use ancillary qubits, which usually start and end in the state $|0\rangle$. They are sometimes called new ancillary qubits. On the other hand, Sect. 4 deals with uninitialized ancillary qubits that start and end in a state $|x\rangle$.

2.2 Shor's Factoring Algorithm

The main part of Shor's factoring algorithm is the quantum algorithm for order-finding. Let N be an n-bit binary number to be factored and a be an n-bit binary number less than N such that a is coprime to N. The quantum algorithm for order-finding finds the order of a modulo N, where the order of a modulo N is the least number $r > 0$ such that $a^r \equiv 1 \bmod N$. Using the order, a non-trivial factor of N can be efficiently computed by using a classical computer. The key ingredients of the quantum algorithm for order-finding is the modular exponentiation operation $\mathrm{ME}(a, N)$ and the QFT. For a $2n$-bit binary number x,

$$\mathrm{ME}(a, N)|x\rangle|1\rangle = |x\rangle|a^x \bmod N\rangle,$$

where $|1\rangle$ consists of n qubits representing 1.

The quantum algorithm for order-finding uses two registers. The first register consists of $2n$ qubits that are set to $|0\rangle$ and the second one consists of n qubits that are set to $|1\rangle$. The algorithm is as follows:

1. Apply a Hadamard operation to each qubit in the first register.
2. Apply $\mathrm{ME}(a, N)$ to the first and second registers.
3. Apply the inverse of the QFT on $2n$ qubits to the first register.
4. Measure the first register.

Using the classical bits obtained in Step 4, the order can be efficiently computed by using a classical computer.

A polynomial-size quantum circuit for $\mathrm{ME}(a, N)$ can be constructed with little difficulty by using a simple classical modular exponentiation method. Thus, since we have Coppersmith's circuit for the QFT, a polynomial-size quantum circuit for order-finding can be constructed. However, the resulting circuit will use many qubits as pointed out in Sect. 1. The following sections consider a quantum circuit for order-finding with only about $2n$ qubits. An important technique for decreasing the number of qubits is the one-controlling-qubit trick [15], [16], which decreases the number of qubits in the first register described above from $2n$ to 1. When this technique is used, it suffices to consider one-qubit operations and measurements of the first register in place of the inverse of the QFT. Moreover, a quantum circuit for $\mathrm{ME}(a, N)$ is not needed. It suffices to construct a quantum circuit for the modular multiplication operation $\mathrm{MM}(a, N)$ defined as

$$\mathrm{MM}(a, N)|x\rangle = |ax \bmod N\rangle,$$

where x is an n-bit binary number. In fact, a quantum circuit for $\mathrm{MM}(a^{2^k}, N)$ with one control qubit is used for $0 \le k \le 2n-1$ sequentially. In constructing a quantum circuit for $\mathrm{MM}(a, N)$ with few qubits, an important issue is how to decompose $\mathrm{MM}(a, N)$ into elementary arithmetic operations.

2.3 Decomposition of Modular Multiplication

Quantum circuits for order-finding described in this paper use Vedral et al.'s decomposition method of $\mathrm{MM}(a, N)$ [17]. In this method, $\mathrm{MM}(a, N)$ is decomposed into the modular product-sum operation $\mathrm{MPS}(a, N)$ defined as

$$\mathrm{MPS}(a, N)|x\rangle|b\rangle = |x\rangle|ax + b \bmod N\rangle,$$

where x and b are n-bit binary numbers. Actually, $\mathrm{MM}(a, N)$ is computed by using a register consisting of n qubits representing 0 as follows:

$$\begin{aligned}|x\rangle|0\rangle &\rightarrow |x\rangle|ax \bmod N\rangle \\ &\rightarrow |ax \bmod N\rangle|x\rangle \\ &\rightarrow |ax \bmod N\rangle|(x - a^{-1}ax) \bmod N\rangle \\ &= |ax \bmod N\rangle|0\rangle.\end{aligned}$$

The first operation is $\mathrm{MPS}(a, N)$, the second is the swap operation, and the third is $\mathrm{MPS}(a^{-1}, N)^{-1}$, where a^{-1} is the multiplicative inverse of a modulo N. Note that, since a is coprime to N, a^{-1} exists and can be computed efficiently by using the extended Euclidean algorithm on a classical computer.

$\mathrm{MPS}(a, N)$ is decomposed into the modular addition operation $\mathrm{MA}(a, N)$ defined as

$$\mathrm{MA}(a, N)|b\rangle = |a + b \bmod N\rangle,$$

where b is an n-bit binary number. Actually, $\mathrm{MPS}(a, N)$ is computed by the relationship

$$\begin{aligned}&ax + b \bmod N \\ &= (2^{n-1}ax_{n-1} + (\cdots(2^1ax_1 + \\ &\quad (2^0ax_0 + b \bmod N) \bmod N)\cdots) \bmod N),\end{aligned}$$

where $x_{n-1}\cdots x_0$ is the binary representation for x. More precisely, $\mathrm{MA}(2^i a, N)$ with one control qubit is repeatedly applied to the register containing b for $0 \le i \le n-1$, where the control qubit is used to check whether $x_i = 1$ or not.

$\mathrm{MA}(a, N)$ is decomposed into addition, subtraction, and comparison operations by the relationship

$$a + b \bmod N = \begin{cases} a + b - N & \text{if } a + b \ge N, \\ a + b & \text{otherwise.} \end{cases}$$

Thus, efficient quantum circuits for these operations are useful for constructing those for order-finding.

3. Addition

3.1 Circuits with Few Qubits

Let a and b be n-bit binary numbers. Moreover, let $a_{n-1}\cdots a_0$, $b_{n-1}\cdots b_0$, and $s_n\cdots s_0$ be the binary representations for a, b, and $a + b$, respectively. Let us consider the problem of how to construct a quantum circuit with few qubits for the addition operation (sometimes called in-place addition) ADD defined as

$$\mathrm{ADD}|a\rangle|b\rangle|z\rangle = |a\rangle|a + b \bmod 2^n\rangle|z \oplus s_n\rangle,$$

where $z \in \{0, 1\}$ and $\oplus$ denotes addition modulo 2. There have been many studies on quantum circuits for ADD, most of which have focused on decreasing the number of ancillary qubits [10], [17]–[19] and have used the ripple-carry approach. To explain the approach, let us define the carry bit c_i as follows:

$$c_i = \begin{cases} 0 & i = 0, \\ \mathrm{MAJ}(a_{i-1}, b_{i-1}, c_{i-1}) & 1 \le i \le n, \end{cases}$$

where MAJ is the majority function for three bits defined as $\mathrm{MAJ}(a, b, c) = ab \oplus bc \oplus ca$. In the ripple-carry approach, the first step is to compute the carry bit c_1 by using a_0 and b_0 and c_0. Then, the next carry bit c_2 is computed by using a_1

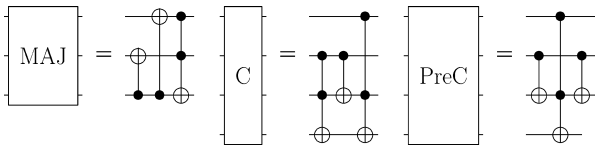

Fig. 2 The MAJ, C, and PreC gates.

and b_1 and c_1. This procedure is repeated until all carry bits are computed. After that, s_i is computed by the relationship

$$s_i = \begin{cases} a_i \oplus b_i \oplus c_i & 0 \le i \le n-1, \\ c_n & i = n. \end{cases}$$

In 1996, Vedral et al. introduced a circuit based on the ripple-carry approach [17]. The circuit uses $O(n)$ ancillary qubits and its depth and size are $O(n)$. The ancillary qubits are used to store all the carry bits. In order to decrease the number of ancillary qubits, how the carry bits are stored is important.

In 2000, Draper constructed a quantum circuit for ADD using the QFT [10]. His approach, which is different from the ripple-carry one, performs addition in the Fourier space and does not require the use of ancillary qubits. The circuit is based on the following algorithm:

1. Apply the QFT (on $n+1$ qubits) to the qubits representing b and z.
2. Apply controlled-R_k operations to appropriate pairs of qubits, where the qubits representing a are used only for control qubits.
3. Apply the inverse of the QFT as in Step 1.

If Coppersmith's circuit is used for the QFT, the circuit uses no ancillary qubits. However, the size becomes large; it is $O(n^2)$, though the depth is $O(n)$. If the approximate version of Coppersmith's circuit is used, only the size decreases, from $O(n^2)$ to $O(n \log n)$.

In 2004, Cuccaro et al. constructed an $O(n)$-size quantum circuit for ADD with only one ancillary qubit [18]. The circuit is based on the ripple-carry approach and its depth is $O(n)$. The key ingredient of the circuit is the MAJ gate depicted in Fig. 2. They used the property that the gate performs

$$|c_i\rangle|b_i\rangle|a_i\rangle \rightarrow |c_i \oplus a_i\rangle|b_i \oplus a_i\rangle|c_{i+1}\rangle$$

for $0 \le i \le n-1$. One ancillary qubit is used to store c_0 and the qubit initially storing a_i is used to store c_{i+1}. They posed a question as to whether an $O(n)$-depth quantum circuit for ADD can be constructed without ancillary qubits using the ripple-carry approach. In 2005, Takahashi et al. answered the question affirmatively [19]. Their construction is described below.

3.2 Eliminating One Ancillary Qubit

Takahashi et al.'s circuit uses no ancillary qubits and its depth and size are $O(n)$. The key ingredient is also the MAJ gate. However, they used another property that the gate performs

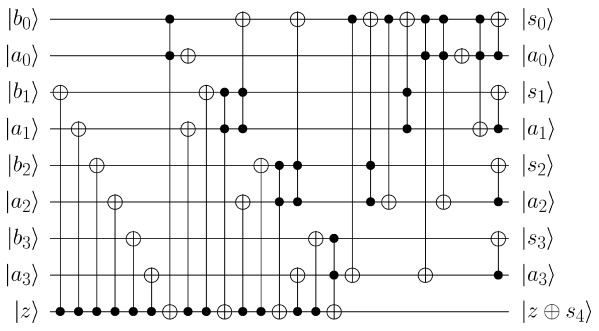

Fig. 3 The circuit for ADD for $n = 4$.

$$|z \oplus b_i\rangle|z \oplus a_i\rangle|z \oplus c_i\rangle \rightarrow |b_i \oplus c_i\rangle|a_i \oplus c_i\rangle|z \oplus c_{i+1}\rangle$$

for $1 \le i \le n-1$. The key idea is to store two sequences of all the carry bits by only using two qubits. The qubit initially storing z stores one sequence and the qubit initially storing b_0 stores the other. The latter sequence is used for the reverse computation that deletes unnecessary carry bits. The circuit consists of four stages:

1. The bit value z is added to a_i and b_i by a CNOT gate for $1 \le i \le n-1$.
2. The first carry bit c_1 is computed by a Toffoli gate. Then, the carry bit c_i is computed by the MAJ gate for $2 \le i \le n$. All the carry bits are stored in the qubit initially storing z. The carry bit c_i is also stored in the qubit initially storing b_0.
3. The value c_i in $a_i \oplus c_i$ is replaced with the value $b_0 \oplus c_1$ for $2 \le i \le n-1$ by a CNOT gate. This is done by using the carry bits stored in the qubit initially storing b_0.
4. The value $b_0 \oplus c_1$ in $a_i \oplus b_0 \oplus c_1$ is deleted by a Toffoli gate for $2 \le i \le n-1$. The desired output state is easily obtained by CNOT gates.

The circuit for $n = 4$ is depicted in Fig. 3.

3.3 Application to Shor's Factoring Algorithm

Since a quantum circuit for MA(a, N) is used in a quantum circuit for order-finding, it is not necessary to consider ADD. It suffices to consider a simpler addition operation ADD(a) (for the predetermined constant a described in Sect. 2) defined as

$$\text{ADD}(a)|b\rangle|z\rangle = |a + b \bmod 2^n\rangle|z \oplus s_n\rangle,$$

where b is an n-bit binary number and $z \in \{0, 1\}$. Other predetermined constants also have to be considered but they are omitted for simplicity. Takahashi et al.'s circuit for ADD uses the qubits representing a not only as control qubits but also for storing some other values. Unfortunately, it therefore cannot simply be transformed into a quantum circuit for ADD(a) with no ancillary qubits. On the other hand, Draper's circuit for ADD uses the qubits representing a only for control qubits, which makes the transformation possible by deleting qubits for a and replacing the controlled-R_k operations with one-qubit operations depending on a. In

this way, the use of operations depending on predetermined constants yields an efficient quantum circuit and is called the hardwiring of classical numbers [16]. Note that, though Takahashi et al.'s circuit is not useful in this situation, it is useful in similar situations such as for solving discrete logarithm problems for elliptic curves by a version of Shor's factoring algorithm, since we need to consider ADD [4].

In 2003, using the QFT-based circuit for ADD(a), Beauregard constructed a quantum circuit for order-finding with $2n + 3$ qubits [16], where n is the length of the number to be factored. The circuit's depth and size are $O(n^3)$ and $O(n^3 \log n)$, respectively, when the approximate version of Coppersmith's circuit for the QFT is used. Beauregard's circuit for MA(a, N) with $n + 2$ qubits is based on the following algorithm.

1. Add a to the initial content b. The resulting state is $|b + a\rangle$.
2. Subtract N from $b + a$. The resulting state is $|b + a - N\rangle$.
3. Write the high-order bit y of $b + a - N$ on one ancillary qubit to decide whether $b + a - N < 0$. The resulting state is $|b + a - N\rangle|y\rangle$, where y is 1 if $b + a - N < 0$ and 0 otherwise.
4. Add N to $b + a - N$ if y is 1. The resulting state is $|a + b \bmod N\rangle|y\rangle$.
5. Subtract a from $a + b \bmod N$. The resulting state is $|(a + b \bmod N) - a\rangle|y\rangle$.
6. Write the negation of the high-order bit z of $(a + b \bmod N) - a$ on the ancillary qubit to decide whether $(a + b \bmod N) - a < 0$. The resulting state is $|(a + b \bmod N) - a\rangle|y \oplus z \oplus 1\rangle$, where z is 1 if $(a + b \bmod N) - a < 0$ and 0 otherwise.
7. Add a to $(a + b \bmod N) - a$. The resulting state is $|a + b \bmod N\rangle|0\rangle$.

Note that $y \oplus z \oplus 1 = 0$ since $a + b \bmod N \geq a$ if and only if $a + b < N$. Subtraction in the above algorithm is implemented by the QFT-based circuit for ADD(a).

4. Comparison

4.1 Improved Circuit for MA(a, N)

It was not known whether efficient quantum circuits for comparison would be useful for Shor's factoring algorithm with few qubits. However, in 2006, Takahashi et al. investigated comparison and decreased the number of qubits in Beauregard's circuit for order-finding [20]. The circuit for order-finding uses $2n + 2$ qubits and its depth and size are $O(n^3)$ and $O(n^3 \log n)$, respectively, where n is the length of the number to be factored. The number of qubits is less than that in any other quantum circuit ever constructed for Shor's factoring algorithm.

Takahashi et al.'s algorithm for MA(a, N) is as follows. The main feature is that the algorithm directly compares b and $N - a$ without computing $b + a - N$.

1. Compare the initial content b to $N - a$ and write the result y on one ancillary qubit. The resulting state is $|b\rangle|y\rangle$, where y is 1 if $b < N - a$ and 0 otherwise.
2. Add a to b if y is 1 and subtract $N - a$ from b if y is 0. The resulting state is $|a + b \bmod N\rangle|y\rangle$.
3. Compare $a + b \bmod N$ to a and write the negation of the result z on the ancillary qubit. The resulting state is $|a + b \bmod N\rangle|y \oplus z \oplus 1\rangle = |a + b \bmod N\rangle|0\rangle$.

The important difference between Beauregard's and Takahashi et al.'s algorithms is the number of qubits for the register containing $a + b \bmod N$ at the end of the computation. Beauregard's algorithm deals with intermediate results that may consist of $n + 1$ qubits (e.g. $b + a - N$) and thus needs $n + 1$ qubits for the register. On the other hand, Takahashi et al.'s algorithm does not deal with such intermediate results and thus only needs n qubits for the register.

If we have circuits for addition, subtraction, and comparison that use no new ancillary qubits, the above difference implies that the number of qubits used in Takahashi et al.'s circuit for order-finding is less than that in Beauregard's by one. Takahashi et al. implemented addition and subtraction using Draper's addition circuit since it does not need new ancillary qubits. Thus, the only problem is how to construct a quantum circuit with no new ancillary qubits for the comparison operation COMP(a) (for the predetermined constant a described in Sect. 2) defined as

$$\mathrm{COMP}(a)|b\rangle|z\rangle = |b\rangle|z \oplus y\rangle,$$

where b is an n-bit binary number, $z \in \{0, 1\}$, and y is 1 if $a > b$ and 0 otherwise. Other predetermined constants also have to be considered but they are omitted for simplicity. In what follows, Takahashi et al.'s idea of using uninitialized ancillary qubits is explained along with their construction of the circuit for COMP(a).

4.2 Circuit with Uninitialized Ancillary Qubits

The idea is to introduce a circuit for computing only the high-order bit of the sum that uses not new ancillary qubits but "uninitialized" ancillary qubits. Roughly speaking, the output bit y of COMP(a) is the high-order bit of $a - b = a + (-b)$. Thus, a quantum circuit for COMP(a) can be constructed by modifying the conventional ripple-carry addition circuit in [17] if n new ancillary qubits are used. By modifying the circuit slightly, the construction can be done by using $n - 1$ new ancillary qubits. The important point is that all we need to do is compute not the sum of two binary numbers but only the high-order bit of the sum of two binary numbers. Thus, $n - 1$ "uninitialized" ancillary qubits can be used in place of $n - 1$ new (initialized) ancillary qubits. That is, ancillary qubits that are not set to $|0\rangle$ can be used if the ancillary qubits are reset to their original values at the end of the computation. Fortunately, such $n - 1$ qubits are available in the other (mostly idle) register. More precisely, when MPS(a, N) is computed, MA($2^i a, N$) with one control qubit is repeatedly applied to the register containing b for $0 \leq i \leq n - 1$. The control qubit is used to check whether

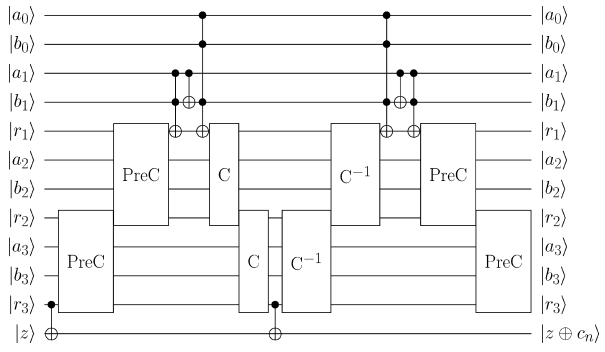

Fig. 4 The circuit for HIGH for $n = 4$.

$x_i = 1$ or not. In this situation, qubits for x_j ($j \neq i$) are "idle"; that is, they are not used during MA($2^i a, N$) and the qubits can be used as uninitialized ancillary qubits.

On the basis of this idea, a quantum circuit is first constructed for the operation HIGH defined as

$$\mathrm{HIGH}|a\rangle|b\rangle|z\rangle = |a\rangle|b\rangle|z \oplus c_n\rangle,$$

where a and b are n-bit binary numbers and $z \in \{0, 1\}$. An $(n-1)$-qubit state $|r_1\rangle \cdots |r_{n-1}\rangle$ is used as ancillary qubits, where $r_i \in \{0, 1\}$. The main components of the circuit are the C and PreC gates depicted in Fig. 2. The C gate is defined in [17] and used to compute carry bits in the circuit for ADD. When $|r_{i-1} \oplus c_i\rangle|a_i\rangle|b_i\rangle|r_i \oplus (a_i \oplus b_i) r_{i-1}\rangle$ is input to the C gate, the gate outputs $|r_{i-1} \oplus c_i\rangle|a_i\rangle|a_i \oplus b_i\rangle|r_i \oplus c_{i+1}\rangle$, where $2 \leq i \leq n-1$. Since uninitialized ancillary qubits have to be used in contrast to the circuit for ADD in [17], the PreC gate is used before the C gate. When $|r_{i-1}\rangle|a_i\rangle|b_i\rangle|r_i\rangle$ is input to the PreC gate, the gate outputs $|r_{i-1}\rangle|a_i\rangle|b_i\rangle|r_i \oplus (a_i \oplus b_i) r_{i-1}\rangle$, where $2 \leq i \leq n-1$. The circuit for HIGH for $n = 4$ is depicted in Fig. 4. The circuit can be transformed into a quantum circuit for COMP(a) easily. This is because the output bit y of COMP(a) is the high-order bit of $a + b'$, where b' is the bitwise complement of b, and because the hardwiring of a described above can be done since the circuit for HIGH uses the qubits representing a only for control qubits. The resulting circuit uses $n-1$ uninitialized ancillary qubits and its depth and size are $O(n)$.

5. The QFT

5.1 Small-Size Circuits

The QFT has been analyzed in depth and efficient quantum circuits for the QFT have been constructed [2], [3], [8], [13], [21]. In the following, let us deal with small-size quantum circuits for the exact or approximate QFT on n qubits. One such circuit is Coppersmith's circuit in 1994 described in Sect. 2 [13]. The size of the circuit for the QFT is $O(n^2)$, and it can be decreased to $O(n \log n)$ by ignoring the controlled-R_k operation for all $k > m$, though the resulting circuit is one for computing a good approximation of the QFT, where $m \in O(\log n)$. The operation performed by the circuit, which is called AQFT$_m$, maps $|b_{n-1}\rangle \cdots |b_0\rangle$ to

$$\frac{1}{\sqrt{2^n}}|\phi_{n-1}(b)\rangle|\phi_{n-2}(b)\rangle \cdots |\phi_0(b)\rangle,$$

where $|\phi_k(b)\rangle$ is $|\varphi_k(b)\rangle$ for $0 \leq k \leq m-1$ and

$$|0\rangle + e^{2\pi i \sum_{j=0}^{m-1} \frac{b_{k-j}}{2^{j+1}}}|1\rangle$$

for $m \leq k \leq n-1$.

In 2000, Cleve et al. constructed a quantum circuit smaller in size for the QFT [8]. Let F_{2^n} denote the QFT on n qubits. The key tool is the generalized recursive circuit description for F_{2^n} parameterized by $t \in \{1, \ldots, n-1\}$:

1. Apply $F_{2^{n-t}}$ to the first $n-t$ qubits.
2. For each $j \in \{1, \ldots, n-t\}$ and $k \in \{1, \ldots, t\}$, apply the controlled-$R_{n-t+k-j+1}$ operation to the j-th qubit and $(n-t+k)$-th qubit.
3. Apply F_{2^t} to the last t qubits.

When $t = n-1$, the description corresponds to Coppersmith's circuit. In general, the circuit for F_{2^n} based on the generalized recursive circuit description changes the order of the two-qubit gates in Coppersmith's circuit, though it uses exactly the same gates.

Notably, Step 2 can be performed by using a classical multiplication algorithm. To describe this more precisely, let us define the multiplication operation MULT as $\mathrm{MULT}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle|xy\rangle$, where x is an $(n-t)$-bit binary number (corresponding to the first $n-t$ qubits), y is a t-bit binary number (corresponding to the last t qubits), and $|0\rangle$ consists of n qubits representing 0. Step 2 is performed as follows:

2-1. Apply MULT to the $2n$ qubits.
2-2. For each $k \in \{1, \ldots, n\}$, apply the R_k operation to the $(n+k)$-th qubit.
2-3. Apply the inverse of MULT to the $2n$ qubits.

A quantum circuit for MULT can be constructed by using the Schönhage-Strassen algorithm. The circuit uses $O(n)$ ancillary qubits and its depth and size are $O(\log n)$ and $O(n \log n \log \log n)$, respectively. Cleve et al. construct a quantum circuit for the QFT using the circuit and the generalized recursive circuit description with $t = \lfloor n/2 \rfloor$. In their construction, the recurrence

$$S_n = S_{\lceil n/2 \rceil} + S_{\lfloor n/2 \rfloor} + O(n \log n \log \log n)$$

holds, where S_n is the size of the circuit for F_{2^n}. This implies that the size is $O(n(\log n)^2 \log \log n)$. The circuit uses $O(n)$ ancillary qubits and its depth is $O(n)$. In 2003, Ahokas et al. constructed a quantum circuit for computing a good approximation of the QFT, which is slightly different from AQFT$_m$, with size $O(n(\log \log n)^2 \log \log \log n)$ by combining Cleve et al.'s and Coppersmith's construction methods [21]. The circuit uses $O(n)$ ancillary qubits and its depth is $O(n)$.

5.2 Circuits on an LNN Architecture

On an LNN architecture, two-qubit gates have to be applied

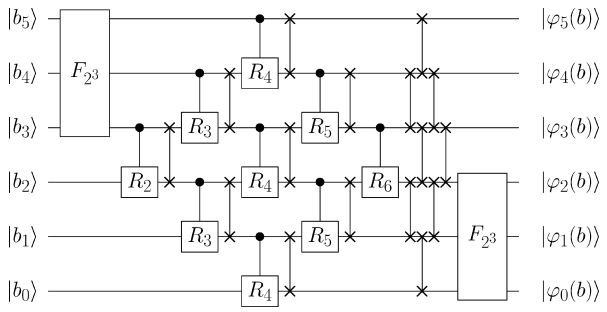

Fig. 5 The circuit for the QFT for $n = 6$ and $t = 3$. The gate adjacent to a controlled-R_k gate is a swap gate.

only to adjacent qubits. In 2004, Fowler et al. constructed a quantum circuit for AQFT$_m$ on an LNN architecture [2]. The circuit uses no ancillary qubits and its depth and size are $O(n)$ and $O(n^2)$, respectively. The construction is similar to Coppersmith's. However, in contrast to the Coppersmith's, the size is large since there are many swap gates that cannot be removed. In 2007, Takahashi et al. decreased the size to $O(n \log n)$ without increasing the depth or the number of qubits [3]. Their construction method is explained below.

A key tool is the LNN version of the generalized recursive circuit description for the QFT parameterized by t, which is based on Cleve et al.'s one. Though the details are beyond the scope of this survey, the circuit based on it for $n = 6$ and $t = 3$ is depicted in Fig. 5. The circuit does not use a fast classical multiplication algorithm since it is difficult to construct a small-size quantum circuit for the algorithm on an LNN architecture. Takahashi et al. obtain a quantum circuit for AQFT$_m$ by ignoring R_k for all $k > m$, which is called LNN-AQFT$_m$. The important point is that ignoring R_k makes it possible to remove many swap gates. For example, when R_k is ignored for all $k > 4$ of the middle part in Fig. 5, the two swap gates applied before the ignored controlled-R_6 gate are canceled out by the two applied after the ignored controlled-R_6 gate. In a similar way, the three swap gates applied before the ignored controlled-R_5 gates are canceled out. It can be shown that LNN-AQFT$_m$ uses no ancillary qubits and its depth and size are $O(n \log \log n)$ and $O(n \log n)$, respectively. Unfortunately, the depth is slightly larger than that of Fowler et al.'s, but Takahashi et al. decrease the depth to $O(n)$ without increasing the size or the number of qubits by combining LNN-AQFT$_m$ with Fowler et al.'s.

5.3 Application to Shor's Factoring Algorithm

On the basis of Beauregard's circuit for order-finding, Fowler et al. constructed an efficient one on an LNN architecture [2]. It uses $2n + 4$ qubits and its depth and size are $O(n^3)$ and $O(n^4)$, respectively, where n is the length of the number to be factored. Note that it contains $O(n^2)$ instances of the AQFT$_m$. As described above, the depth and size of Fowler et al.'s circuit for AQFT$_m$ are $O(n)$ and $O(n^2)$, respectively. On the other hand, the depth and size of Takahashi et al.'s circuit are $O(n)$ and $O(n \log n)$, respectively. Thus, when Fowler et al.'s circuit is simply replaced with Takahashi et al.'s one in Fowler et al.'s circuit for order-finding, the depth and size of the resulting circuit are $O(n^3)$ and $O(n^3 \log n)$, respectively. The number of qubits in the resulting circuit is the same as that in Fowler et al.'s original circuit for order-finding. This is because the number of qubits in Takahashi et al.'s circuit for AQFT$_m$ is the same as that in Fowler et al.'s.

6. Open Problems

Interesting challenges would be to find ways to improve the quantum circuits described above. For example, can we construct the following circuits or prove that they cannot be constructed?

- An $O(\log n)$-depth $O(n)$-size quantum circuit for ADD with $O(1)$ ancillary qubits [22], [23].
- An $O(n)$-size quantum circuit for ADD(a) with $O(1)$ ancillary qubits [19].
- A polynomial-size quantum circuit for order-finding with about n qubits [24].
- An $O(n)$-size quantum circuit for the QFT or its good approximation [8], [21].
- A sublinear-depth polynomial-size quantum circuit for the QFT [8], [14].

It would also be interesting to consider the above problems on an LNN architecture.

There are other operations that would be interesting to investigate. For instance, can we construct a quantum circuit for multiplication that is more efficient than the best known classical one [25]? Such a circuit would be closely related to an efficient quantum circuit for the QFT [8], [21]. Moreover, can we construct quantum circuits for arithmetic operations in finite fields that are more efficient than the ones in [4], [5], [7]? Such circuits would be useful for analyzing the security of Elliptic Curve Cryptography.

References

[1] P.W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," Proc. 35th IEEE Symposium on Foundations of Computer Science, pp.124–134, 1994.
[2] A.G. Fowler, S.J. Devitt, and L.C.L. Hollenberg, "Implementation of Shor's algorithm on a linear nearest neighbour qubit array," Quantum Information and Computation, vol.4, no.4, pp.237–251, 2004.
[3] Y. Takahashi, N. Kunihiro, and K. Ohta, "The quantum Fourier transform on a linear nearest neighbor architecture," Quantum Information and Computation, vol.7, no.4, pp.383–391, 2007.
[4] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," Quantum Information and Computation, vol.3, no.4, pp.317–344, 2003.
[5] P. Kaye, "Optimized quantum implementation of elliptic curve arithmetic over binary fields," Quantum Information and Computation, vol.5, no.6, pp.474–491, 2005.
[6] N. Kunihiro, "Exact analyses of computational time for factoring in quantum computers," IEICE Trans. Fundamentals, vol.E88-A, no.1, pp.105–111, Jan. 2005.

[7] D. Cheung, D. Maslov, J. Mathew, and D.K. Pradhan, "On the design and optimization of a quantum polynomial-time attack on elliptic curve cryptography," Theory of Quantum Computation, Communication, and Cryptography, Lecture Notes in Computer Science, vol.5106, pp.96–104, 2008.
[8] R. Cleve and J. Watrous, "Fast parallel circuits for the quantum Fourier transform," Proc. 41st IEEE Symposium on Foundations of Computer Science, pp.526–536, 2000.
[9] D. Bera, F. Green, and S. Homer, "Small depth quantum circuits," ACM SIGACT News, vol.38, no.2, pp.35–50, 2007.
[10] T.G. Draper, "Addition on a quantum computer," quant-ph/0008033, 2000.
[11] J. Van Leeuwen, ed., Handbook of Theoretical Computer Science, vol.A: Algorithms and Complexity, Elsevier, 1990.
[12] M.A. Nielsen and I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.
[13] D. Coppersmith, "An approximate Fourier transform useful in quantum factoring," quant-ph/0201067, 1994.
[14] C. Moore and M. Nilsson, "Parallel quantum computation and quantum codes," SIAM J. Comput., vol.31, no.3, pp.799–815, 2002.
[15] M. Mosca and A. Ekert, "The hidden subgroup problem and eigenvalue estimation on a quantum computer," Quantum Computing and Quantum Communications: First NASA International Conference, Lecture Notes in Computer Science, vol.1509, pp.174–188, 1999.
[16] S. Beauregard, "Circuit for Shor's algorithm using $2n + 3$ qubits," Quantum Information and Computation, vol.3, no.2, pp.175–185, 2003.
[17] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," Phys. Rev. A, vol.54, no.1, pp.147–153, 1996.
[18] S.A. Cuccaro, T.G. Draper, S.A. Kutin, and D.P. Moulton, "A new quantum ripple-carry addition circuit," quant-ph/0410184, 2004.
[19] Y. Takahashi and N. Kunihiro, "A linear-size quantum circuit for addition with no ancillary qubits," Quantum Information and Computation, vol.5, no.6, pp.440–448, 2005.
[20] Y. Takahashi and N. Kunihiro, "A quantum circuit for Shor's factoring algorithm using $2n + 2$ qubits," Quantum Information and Computation, vol.6, no.2, pp.184–192, 2006.
[21] G. Ahokas, R. Cleve, and L. Hales, "The complexity of quantum Fourier transforms and integer multiplication," Proc. ERATO Conference on Quantum Information Science 2003, pp.39–40, 2003.
[22] T.G. Draper, S.A. Kutin, E.M. Rains, and K.M. Svore, "A logarithmic-depth quantum carry-lookahead adder," Quantum Information and Computation, vol.6, no.4&5, pp.351–369, 2006.
[23] Y. Takahashi and N. Kunihiro, "A fast quantum circuit for addition with few qubits," Quantum Information and Computation, vol.8, no.6&7, pp.636–649, 2008.
[24] C. Zalka, "Shor's algorithm with fewer (pure) qubits," quant-ph/0601097, 2006.
[25] M. Fürer, "Faster integer multiplication," Proc. 39th ACM Symposium on Theory of Computing, pp.57–66, 2007.

Yasuhiro Takahashi received his B.S. and M.S. in Mathematics from Tohoku University in 1998 and 2000, respectively. In 2008, he received his Ph.D. in Engineering from the University of Electro-Communications. Since 2000, he has been with NTT Communication Science Laboratories. His research interests include quantum computing, complexity theory, and cryptography.