SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2016, Vol. 59 042406:1–042406:8
doi: 10.1007/s11432-015-5411-x

Improved quantum ripple-carry addition circuit

Feng WANG[1], Mingxing LUO[2], Huiran LI[2], Zhiguo QU[3*] & Xiaojun WANG[4]

[1]*College of mathematical sciences, Dezhou University, Dezhou* 253023*, China;*
[2]*Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu* 610031*, China;*
[3]*Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing* 210044*, China;*
[4]*School of Electronic Engineering, Dublin City University, Dublin 9, Ireland*

Received June 23, 2015; accepted July 9, 2015; published online February 19, 2016

Abstract A serious obstacle to large-scale quantum algorithms is the large number of elementary gates, such as the controlled-NOT gate or Toffoli gate. Herein, we present an improved linear-depth ripple-carry quantum addition circuit, which is an elementary circuit used for quantum computations. Compared with previous addition circuits costing at least two Toffoli gates for each bit of output, the proposed adder uses only a single Toffoli gate. Moreover, our circuit may be used to construct reversible circuits for modular multiplication, Cx mod M with $x < M$, arising as components of Shor's algorithm. Our modular-multiplication circuits are simpler than previous constructions, and may be used as primitive circuits for quantum computations.

Keywords quantum addition circuit, modular multiplication, CNOT gate, Toffoli gate, circuit complexity

Citation Wang F, Luo M X, Li H R, et al. Improved quantum ripple-carry addition circuit. Sci China Inf Sci, 2016, 59(4): 042406, doi: 10.1007/s11432-015-5411-x

1 Introduction

Classical computation algorithms also consist of a finite sequence of instructions for solving certain problems using a classical computer. A quantum computation algorithm runs on a realistic physical model such as an atom, a photon, or iron [1–4]. A quantum computation algorithm is also a step-by-step procedure based on a well-defined quantum computer or quantum computation model. In general, all classical computation algorithms can be implemented on a quantum computer. However, the quantum algorithm has its own inherent features beyond the classical computer. The first well-known example of this idea was provided by Peter Shor [5,6], who addressed two classical computational problems that have not been solved with classical polynomial time algorithms. One problem is the integer decomposition. The other is the discrete logarithm over a finite group. Large integer decomposition as a primitive difficult problem used for the well-known RSA cryptography [7] may be applied to ensure the authority of classical data [8–11] or image protection [12]. Based on the quantum circuit model [13,14], Peter Shor presented polynomial time quantum algorithms for these problems. The exponential-speedup has (up to now) depended on the fast implementation of a quantum Fourier transformation. Large integer decomposition

* Corresponding author (email: qzghhh@126.com)

can also be improved to address the general classical problem, i.e., the Hiden subgroup problem [14–17]. Such algorithms may be realized using hybrid photonic systems [4], hyper-parallel photonic systems [18], or an optimal approximate quantum evolution [17]. Some algorithms have also been implemented using different physical systems [19–21].

Since Shor introduced his quantum algorithms [5,6], many improvements to them have been made [22–26]. The quantum addition of two integer numbers is a key operation for constructing such quantum circuits, or other quantum algorithms. Since the importance of quantum addition in quantum computation, some efficient quantum circuits have been constructed for the addition of two n-bit binary numbers. Without use of ancillary qubits, Takahashi and Kunihiro [27] presented quantum addition circuits with depth $O(n)$. Takahashi et al. also improved their algorithm using depth $O(\log n)$ [28]. Draper et al. [29] presented an efficient addition circuit by borrowing techniques from the classical carry-lookahead arithmetic circuit. Their quantum carry-lookahead adder accepts two n-bit numbers and adds them to depth $O(\log n)$ using $O(n)$ ancillary qubits. Their schemes have reduced the cost of additions with a slight increase in the number of qubits. Recently, Takahashi et al. [30] demonstrated how to construct an $O(n)$-depth $O(n)$-sized quantum circuit for the addition of two n-bit binary numbers without ancillary qubits. The exact size is $7n-6$. Using this circuit, they also constructed an $O(d(n))$-depth $O(n)$-sized quantum circuit for an addition using $O(n/d(n))$ ancillary qubits for any $d(n)=\Omega(\log n)$. If an unbounded fan-out gate is allowed, they can construct an $O(e(n))$-depth $O(n)$-sized circuit with $o(n)$ ancillary qubits for any $e(n)=\Omega(\log n)$. Derived from the ripple-carry adder [27], Cuccaro et al. [31] presented a new quantum ripple-carry addition circuit, using $2n+O(1)$ Toffoli gates, $5n+O(1)$ CNOT gates, and $2n+O(1)$ negations. The depth is $2n+O(1)$ and only one auxiliary qubit is required. Thomsen and Axelsen [32] optimized this circuit through a parallelization scheme.

In this paper, motivated from the classical carry-lookahead arithmetic circuit, we consider quantum circuits for the addition of two binary numbers with low complexity. The complexity of a quantum circuit consists of the number of multiqubit logic gates, circuit depth, and number of qubits. Generally, the number of multiqubit logic gates and circuit depth correspond to the physical implementation complexity and the computation time respectively, whereas the number of qubits corresponds to the memory size. We regard the number of multiqubit logic gates as the primary consideration because it is difficult in a faithful quantum implementation. Herein, we present an improved quantum addition circuit. The circuit is based on the ripple-carry approach [31]. The key ingredient of the new adder is a circuit computing the majority of three bits. Differing from previous quantum additions [19–32], the register locations of the output are different from these of the input, i.e., the input bits will be swapped after the addition. Based on this difference, our circuit uses only n Toffoli gates and $5n+O(1)$ CNOT gates, where the depth is $3n+O(1)$ and the number of the auxiliary qubit is $n+O(1)$. The proposed circuit is described in more detail in Section 2. We then describe the use of multiplication circuits in Section 3, which improves the results in [32].

2 The quantum addition circuit

2.1 Basic addition circuit

Our goal is to compute the sum of two n-bit numbers, $\boldsymbol{a}$ and $\boldsymbol{b}$. In a quantum application, several different inputs $\boldsymbol{a}$ or $\boldsymbol{b}$ may be computed simultaneously, such as the Shor's algorithm. However, because all operations performed are permutation operations, thus it is unnecessary to worry about the superposition states. Equivalently, we compute

$$|\boldsymbol{a}\rangle|\boldsymbol{b}\rangle|\boldsymbol{0}\rangle_C \mapsto |\boldsymbol{a}\rangle|\boldsymbol{b}\rangle|\boldsymbol{a}+\boldsymbol{b}\rangle, \tag{1}$$

where $|\boldsymbol{0}\rangle_C$ is an auxiliary qubit system. Let $\boldsymbol{a}=a_{n-1}\cdots a_0$, where a_0 is the lowest-order bit. Similarly, denote $\boldsymbol{b}=b_{n-1}\cdots b_0$. Here, A_j and B_j denote the memory locations of input bits a_j and b_j. Compared with previous schemes [27–32], the memory locations of the output system will be changed in our scheme,

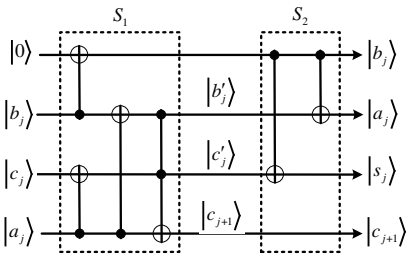

Figure 1 Elementary quantum circuit. $|0\rangle, b_j$, c_j, and a_j are input bits in order. b'_j, c'_j, c_{j+1} are output bits of sub-circuit S_1. b_j, a_j, s_j, and c_{j+1} are output bits of sub-circuit S_2.

Table 1 The values of $\mathrm{MAJ}(a_j, b_j, c_j)$ dependent on input bits a_j, b_j, and c_j

$a_j b_j c_j$	000	001	010	011	100	101	110	111
$\mathrm{MAJ}(a_j, b_j, c_j)$	0	0	0	1	0	1	1	1

as shown in Figure 1. Therefore, the labels of the memory locations in the output system are not presented in Eq. (1).

The quantum addition algorithm is based on the classical ripple-carry addition algorithm [31]. For the input of two n-bit numbers, $\boldsymbol{a}$ and $\boldsymbol{b}$, $n+1$ bits of auxiliary quantum registers are required as temporary storage registers, and are denoted as $\boldsymbol{c}$. The carry string for two input bits, $\boldsymbol{a}$ and $\boldsymbol{b}$, are defined recursively. In detail, let $c_0 = 0$, and $c_{j+1} = \mathrm{MAJ}(a_j, b_j, c_j) = a_j b_j \oplus a_j c_j \oplus b_j c_j$ for $j > 0$. Note that $\mathrm{MAJ}(a_j, b_j, c_j)$ are symmetric in terms of a_j, b_j, and c_j. It then easily follows that $s_j = a_j \oplus b_j \oplus c_j$, which are also symmetric in terms of a_j, b_j, and c_j for all $j < n$, and $s_n = c_n$. Thus, the additional output of $\boldsymbol{a}$ and $\boldsymbol{b}$ is $s_n s_{n-1} \cdots s_2 s_1$. In a classical ripple-carry adder, one computes each c_j from c_1 up to c_n. We then have to erase the carry bits [27,28]. However, we found by redesigning the circuit that erasing the circuit is unnecessary. The values of $\mathrm{MAJ}(a_j, b_j, c_j)$ are shown in Table 1. Through the values in this table, we can construct a new quantum circuit for $\mathrm{MAJ}(a_j, b_j, c_j)$ without erasing circuit, as shown in Figure 1.

To describe our circuit, the bit evolution shown in Figure 1 is as follows.

$$\begin{pmatrix} \underline{|0_j\rangle|b_j\rangle|c_j\rangle|a_j\rangle} \\ |0\rangle|0\rangle|0\rangle|0\rangle \\ |0\rangle|0\rangle|0\rangle|1\rangle \\ |0\rangle|0\rangle|1\rangle|0\rangle \\ |0\rangle|0\rangle|1\rangle|1\rangle \\ |0\rangle|1\rangle|0\rangle|0\rangle \\ |0\rangle|1\rangle|0\rangle|1\rangle \\ |0\rangle|1\rangle|1\rangle|0\rangle \\ |0\rangle|1\rangle|1\rangle|1\rangle \end{pmatrix} \xrightarrow{S_1} \begin{pmatrix} \underline{|b_j\rangle|b'_j\rangle|c'_j\rangle|c_{j+1}\rangle} \\ |0\rangle|0\rangle|0\rangle|0\rangle \\ |0\rangle|1\rangle|1\rangle|0\rangle \\ |0\rangle|0\rangle|1\rangle|0\rangle \\ |0\rangle|1\rangle|0\rangle|1\rangle \\ |1\rangle|1\rangle|0\rangle|0\rangle \\ |1\rangle|0\rangle|1\rangle|1\rangle \\ |1\rangle|1\rangle|1\rangle|1\rangle \\ |1\rangle|0\rangle|0\rangle|1\rangle \end{pmatrix} \xrightarrow{S_2} \begin{pmatrix} \underline{|b_j\rangle|a_j\rangle|s_j\rangle|c_{j+1}\rangle} \\ |0\rangle|0\rangle|0\rangle|0\rangle \\ |0\rangle|1\rangle|1\rangle|0\rangle \\ |0\rangle|0\rangle|1\rangle|0\rangle \\ |0\rangle|1\rangle|0\rangle|1\rangle \\ |1\rangle|0\rangle|1\rangle|0\rangle \\ |1\rangle|1\rangle|0\rangle|1\rangle \\ |1\rangle|0\rangle|0\rangle|1\rangle \\ |1\rangle|1\rangle|1\rangle|1\rangle \end{pmatrix}. \tag{2}$$

From this equation, the memory locations of the output system differ from those of the input system. In detail, after the circuit, the input bit b_j is redefined at the first auxiliary register $|0\rangle$, the input bit a_j is redefined at $|b_j\rangle$, and the output bit s_j is located at the second auxiliary register $|c_j\rangle$. This addition circuit costs five CNOTs and one Toffilo gate for each $j \geqslant 1$.

Using the elementary circuit shown in Figure 1, a general addition circuit was constructed as shown in Figure 2. For input bits $|0\rangle|a_j\rangle|b_j\rangle|c_j\rangle$, the second circuit S_2 may be implemented in parallel with the followed subcircuit S_1 of the input $|0\rangle|a_{j+1}\rangle|b_{j+1}\rangle|c_{j+1}\rangle$, $j = 0, 1, \ldots, n-2$. The final evolution is obtained as

$$|\boldsymbol{a}\rangle|\boldsymbol{b}\rangle|\boldsymbol{0}\rangle \mapsto |\boldsymbol{a}\rangle|\boldsymbol{b}\rangle|\boldsymbol{s}\rangle. \tag{3}$$

Note that, at the end of the circuit, the outcomes of $|b_{n-1}\rangle$ and $|c_{n-1}\rangle$ are the same after the subcircuit

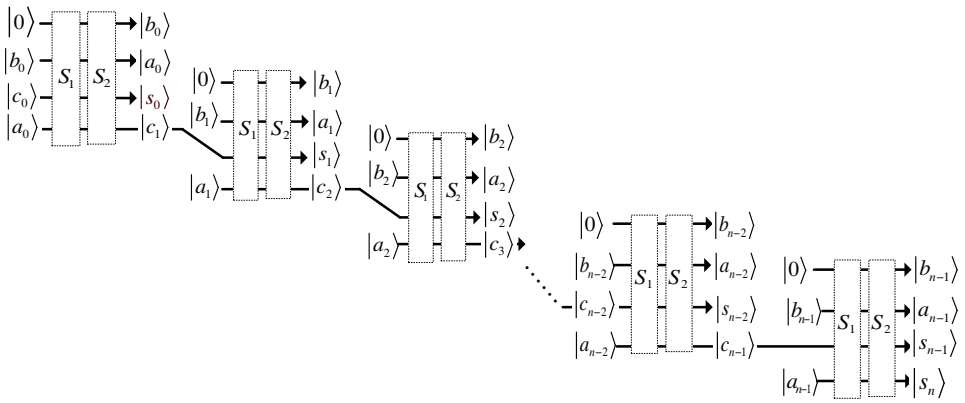

Figure 2 The ripple-carry adder for $n \geqslant 3$. S_i denote the subcircuits shown in Figure 1. a_j and b_j are input bits. $|0\rangle$ and $|c_0\rangle$ are auxiliary registers.

S_1 from Eq. (2). Moreover, $c_0 = 0$, and we do not need a MAJ gate to compute $c_1 = a_0 b_0$. Thus, our addition circuit costs $5n-2$ CNOT gates and n Toffoli gates. The circuit depth is $3n+3$ and the number of auxiliary qubits is $n+1$ (two qubits for $j = 0$ while one qubit for $j \geqslant 1$).

2.2 Extensions

In this subsection, various slightly modified versions of the ripple-carry adder will be used to consider the following two problems.

Problem 1. $\boldsymbol{a} + \boldsymbol{b}$ modulo 2^n, i.e., without the highest bit.

Problem 2. The highest bit only: We compute the highest bit, but do not overwrite the b input. This circuit can be adapted to provide a comparator.

For each case, the quantum circuit is a simple modification of the addition circuit described in Subsection 2.1. The final results are summarized in Table 2. Here, for each circuit, we present the number of the Toffoli gate, the number of the CNOT gate, and the overall depth required.

In the following, we discuss the quantum circuits for the Problems 1 and 2 in detail. Consider the Problem 1, and suppose that we wish to compute $\boldsymbol{a} + \boldsymbol{b} \mod 2^n$, i.e., the high bit c_n is omitted. The details of the algorithm are shown in Algorithm 1. This circuit has $n-2$ number of subcircuits S_1 and S_2. So, it contains $n-1$ Toffoli gates and $5n-7$ CNOT gates for $n \geqslant 3$. The depth is $3n$ and the number of auxiliary qubit is n (two qubits for $j = 0$ while one qubit for $1 \leqslant j \leqslant n-2$).

Algorithm 1

Require: $\boldsymbol{a} \in F_2^n, \boldsymbol{b} \in F_2^n, |\boldsymbol{0}\rangle, c_0 = 0$;
1: $j = 0 \to n-1$;
2: **if** $j < n-1$ **then**
3: Compute $s_j = a_j \oplus b_j \oplus c_j$;
4: Compute $c_{j+1} = \mathrm{MAJ}(a_j, b_j, c_j) = a_j b_j \oplus a_j c_j \oplus b_j c_j$;
5: **else**
6: Compute $s_j = a_j \oplus b_j \oplus c_j$;
7: **end if**
8: Output: $s_{n-1}, s_{n-2}, \ldots, s_0$.

Consider the Problem 2. The circuit shown in Subsection 2.1 may be reused. However, because only the highest bit is required, we can simplify it as follows. Note that the addition of any two n-bit strings may be represented with $n+1$ bits. Thus, we have Algorithm 2.

The evolution procedure is defined as

$$|\boldsymbol{a}\rangle|\boldsymbol{b}\rangle|\boldsymbol{0}\rangle \mapsto |\boldsymbol{a}\rangle|\boldsymbol{b}\rangle|s_n\rangle. \tag{4}$$

The circuit is easily followed from Figure 2. This circuit costs $2n-2$ CNOT gates, n Toffoli gates, and has a depth of $2n$.

Table 2 Addition circuit summary, for $n \geqslant 3$. The first column provides the function which is computed. The followed columns provide the numbers of input, output, and ancillary qubits, the numbers of Toffoli and CNOT gates, and the overall depth

Function	Number of bits in	Number of bits out	Number anc. bits	Size of Toffoli	Size of CNOT	Size of depth
$+$ in $\mathbb{Z}$	$2n$	$3n+1$	$n+1$	n	$5n-2$	$3n+3$
$+$ in $\mathbb{Z}$ [31]	$2n$	$2n+1$	1	$2n-1$	$5n-3$	$2n+4$
$+ \pmod{2^n}$	$2n$	$3n$	n	$n-1$	$5n-7$	$3n$
$+ \pmod{2^n}$ [31]	$2n$	$2n+1$	1	$2n-3$	$5n-7$	$2n+2$
Highest bit	$2n$	$2n+1$	1	n	$2n-2$	$2n$
Highest bit [31]	$2n$	$2n+1$	1	$2n-1$	$4n-3$	$2n+3$
VBE adder [30]	$2n$	$3n$	n	$4n-2$	$4n-2$	$6n-2$

Algorithm 2

Require: $\boldsymbol{a} \in F_2^n, \boldsymbol{b} \in F_2^n, |\mathbf{0}\rangle, c_0 = 0$
1: For $j = 0 \to n-1$
2: **if** $j < n-1$ **then**
3: Compute $c_{j+1} = \mathrm{MAJ}(a_j, b_j, c_j) = a_j b_j \oplus a_j c_j \oplus b_j c_j$;
4: **end if**
5: Output $s_n = c_n$.

3 Multiplication circuit

We now design several circuits for $C \cdot \boldsymbol{x} \mod M$ and related operations, using the additive building blocks described above.

3.1 Circuits for $(2^k+1)\boldsymbol{x}$

The circuits for $(2^k+1)\boldsymbol{x}$ (not modular) can be constructed using shifts and adds, but the challenge is avoiding unnecessary ancillary qubits. Our circuits are structured as follows.

Our goal is to compute

$$|\boldsymbol{x}\rangle|\mathbf{0}\rangle \mapsto |\boldsymbol{x}\rangle|(2^k+1)\boldsymbol{x}\rangle. \tag{5}$$

Case 1. $0 \leqslant k \leqslant n-1$.

For bit values $x_j (j < n)$ of $\boldsymbol{x}$, the bit values of $(2^k+1)\boldsymbol{x}$, S_j can be constructed using a k-bit shift of $\boldsymbol{x}$ followed by an $n+k$ bit add (i.e., $2^k\boldsymbol{x} + \boldsymbol{x}$). The addition can be conducted using a generic Cuccaro adder-$2^k\boldsymbol{x}$ on the main qubits, and $\boldsymbol{x}$ on the ancillary qubits; however, clearing these ancillary qubits is difficult. Another approach is to construct logical sub-expressions for the output bits. The formula in Eq. (8) gives sub-expressions for each S_j bit [33].

$$S_j = \begin{cases} x_j, & 0 \leqslant j \leqslant k-1; \\ x_i \oplus x_{j-k} \oplus c_j, & k \leqslant j \leqslant n-1; \\ x_{j-k} \oplus c_j, & n \leqslant j \leqslant n+k-1; \\ x_{j-k-1} c_{j-1}, & j = n+k, \end{cases} \tag{6}$$

$$c_j = \begin{cases} 0, & 0 \leqslant j \leqslant k; \\ x_{j-1}x_{j-k-1} \oplus x_{j-1}c_{j-1} \oplus x_{j-k-1}c_{j-1}, & k+1 \leqslant j \leqslant n; \\ x_{j-k-1} \oplus c_{j-1}, & n+1 \leqslant j \leqslant n+k-1. \end{cases}$$

However, their circuit complexity is also high. In what follows, we make use of the scheme in Section 2.

First, a k-bit shift of $\boldsymbol{x}$ is constructed at $n+k$ auxiliary qubits, as shown in Figure 3. We obtain

$$|\boldsymbol{x}\rangle|\mathbf{0}\rangle \mapsto |\boldsymbol{x}\rangle|\tilde{\boldsymbol{x}}\rangle, \tag{7}$$

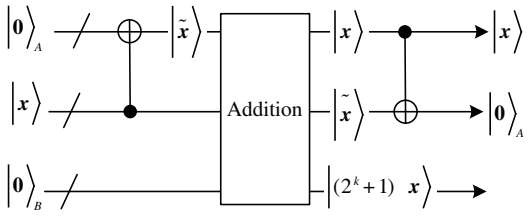

Figure 3 Circuits for $(2^k+1)\boldsymbol{x}$. The subcircuit addition denotes the addition presented in Section 2. A and B are auxiliary qubit systems.

Table 3 Multiplication circuit summary, for $n \geqslant 3$. The first column gives the function which is computed. The followed columns provide the numbers of input bits, output bits, and ancillary qubits, Toffoli gate and CNOT gate. n_1 denotes as non-zero bits of input bits

Function	Number of bits in	Number of bits out	Number of anc. bits	Size of Toffoli	Size of CNOT
$\|(2^k+1)\boldsymbol{x}\rangle$	n	$n+k+1$	$n+k+1$	n	$5n-2$
$\|(2^k+1)\boldsymbol{x}\rangle$ [33]	n	$n+k+1$	$k+1$	$6n$	$3n$
$\|\boldsymbol{x} \mod M\rangle$	n	$2n$	n	$2n$	$10n$
$\|\boldsymbol{x} \mod M\rangle$ [33]	n	n	$k+1$	$5n$	$n+5$

where $|\tilde{\boldsymbol{x}}\rangle$ is a k-bit shift of $|\boldsymbol{x}\rangle$, and the auxiliary qubit system $|\boldsymbol{0}\rangle = |0\rangle^{\otimes n+k}$. The total cost is n CNOT gates.

Second, from our scheme in Section 2, we can obtain

$$|\boldsymbol{x}\rangle|\tilde{\boldsymbol{x}}\rangle|\boldsymbol{0}\rangle_B \mapsto |\boldsymbol{x}\rangle|\tilde{\boldsymbol{x}}\rangle|\boldsymbol{x}+2^k\boldsymbol{x}\rangle_B, \tag{8}$$

where the auxiliary qubit system $|\boldsymbol{0}\rangle_B = |0\rangle^{\otimes n+k+1}$. The total costs are $5n-2$ CNOT gates and n Toffoli gates. Finally, we need to clear out the qubit information in quibt system C using n CNOT gates. The circuit for $(2^k+1)\boldsymbol{x}$ requires n Toffoli gates and $7n$ CNOT gates, which are less than $6n$ Toffoli gates and $3n$ CNOT gates [33].

Case 2. $k \geqslant n$.

In this case, only $2n$ CNOT gates are required to compute $2^k\boldsymbol{x}$, and another $2n$ CNOT gates are required to compute $2^k\boldsymbol{x}+\boldsymbol{x}$. Here, one only needs to copy the first n bits of $\boldsymbol{x}$ into the first n bits of $2^k\boldsymbol{x}$, as shown in Eq. (8). Thus, the total cost is $4n$ CNOT gates, see Table 3.

3.2 Circuits for $\boldsymbol{x}(\mathbf{mod}\ M)$ for $\boldsymbol{x} \leqslant 2M$

The circuits for $\boldsymbol{x}(\mathrm{mod}\ M)$ for odd $\boldsymbol{x} \leqslant 2M$ (modular-reduction circuit) can be applied with one comparator $\boldsymbol{x} > M$ or $\boldsymbol{x} < M$, and one conditional subtraction $\boldsymbol{x} - M$ if $\boldsymbol{x} > M$. Here, from the addition circuit shown in Section 2, the subtraction can be evaluated using a bitwise negation as $(\boldsymbol{x}-M) = (\boldsymbol{x}'+M)'$. The cost is the same as that of the addition circuit shown in Table 2. The comparator is similar to a subtractor-one subtracts $\boldsymbol{x}-M$ and checks $\boldsymbol{x}-M<0$. For our highest bit, only adders can be modified to perform a comparison, leaving their data inputs unchanged and producing a one-bit result as the most significant carry-bit of the subtraction. Although the resulting circuit operates correctly for only $\boldsymbol{x} \leqslant 2M$, $\boldsymbol{x} \leqslant 2M$ can be guaranteed in Shor's algorithm. The cost is no more than that of the addition circuit shown in the Table 2. The total circuit requires at most $2n$ number of Toffoli gate, and $10n$ number of CNOT gate.

To construct general quantum circuits for $2^k\boldsymbol{x}(\mathrm{mod}\ M)$, we begin with a linear-sized circuit for $2\boldsymbol{x}$ $(\mathrm{mod}\ M)$, which can be completed using the bit shift circuit of $2\boldsymbol{x}$ and the modular-reduction circuit above. If the outcome of the first comparator is $2\boldsymbol{x} > M$, the modular-reduction circuit is used to obtain $2\boldsymbol{x}(\mathrm{mod}\ M)$, and the bit shift circuit of $2^{k-1}(2\boldsymbol{x} \mod M)$ is designed to complete the general quantum circuit. If the outcome of the first comparator is $2\boldsymbol{x} < M$, $2^2\boldsymbol{x}(\mathrm{mod}\ M)$ may be reconsidered. This procedure may be iterative, as shown in Figure 4. Although the resulting circuit in Figure 4 operates correctly for only $\boldsymbol{x} \leqslant M$, $\boldsymbol{x} \leqslant M$ can be guaranteed in Shor's algorithm.

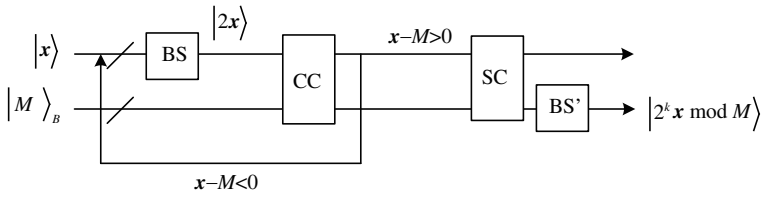

Figure 4 Circuits for $2^k\boldsymbol{x} \bmod M$. The subcircuit BS denotes the bit shift circuit, CC denotes the comparator circuit while SC denote the subtraction circuit using the addition presented in Section 2.

3.3 Circuits for $C\boldsymbol{x}(\bmod M)$ for $C < M/2$

Different from a general constant $C < M$, if $C < M/2$ is satisfied, we can obtain the following equivalent computation procedure. In detail, consider integers $0 \leqslant x < M$ and $1 < C < M/2$ with $\gcd(C, M) = 1$. Defining the integers $D = \lceil M/(2C) \rceil$ and $r = C - (\lceil M/2 \rceil \mod 2C)$, then

$$C\boldsymbol{x} \mod M = r\lfloor \boldsymbol{x}/D \rfloor (\bmod \lceil M/2 \rceil) + C(x \mod D). \tag{9}$$

Thus, one can compute $Cx \mod M$ using division of a remainder. In fact, for the integer D, it follows that $\boldsymbol{x} = D\lfloor \boldsymbol{x}/D \rfloor + (\boldsymbol{x} \mod D)$. Then, $C\boldsymbol{x} = CD\lfloor \boldsymbol{x}/D \rfloor + C(\boldsymbol{x} \mod D)$. Note that $C(x \mod D) \leqslant C(D-1) < M/2$. Moreover,

$$\begin{aligned} CD\lfloor \boldsymbol{x}/D \rfloor \mod \lceil M/2 \rceil &= (CD - \lceil M/2 \rceil)\lfloor x/D \rfloor \\ &= (CD - (CD - C + (\lceil M/2 \rceil \mod 2C)))\lfloor \boldsymbol{x}/D \rfloor \\ &= (C - (\lceil M/2 \rceil \mod 2C))\lfloor \boldsymbol{x}/D \rfloor. \end{aligned}$$

To construct reversible circuits using this result, we can use the circuit for division with remainder [31–33] to represent $\boldsymbol{x}$ through the pair $(\lfloor \boldsymbol{x}/D \rfloor, \ \boldsymbol{x} \mod D)$. A challenging part is to implement multiplications using constants $r\lfloor \boldsymbol{x}/D \rfloor (\bmod \lceil M/2 \rceil)$ and $C(\boldsymbol{x} \mod D)$ with reversible circuits.

4 Conclusion

In this paper, linear-sized circuits are proposed for several special cases of quantum addition and modular multiplication. First, a ripple-carry adder is proposed to reduce the Toffoli gate. In comparison to previous circuits with at least $2n$ Toffoli gates, our circuit costs only n Toffoli gates through the use of more auxiliary qubits [29–32]. Note that a Toffoli gate may be decomposed at least six CNOT gates [1,34] and is very difficult to achieve experimentally. Thus, our improvement is important experimentally. Moreover, our addition circuits are adapted to construct a multiplication circuit derived from Shor's algorithm [5]. The recent results in [33] require $6n$ and $5n$ Toffoli gates for $|(2^k+1)\boldsymbol{x}\rangle$ and $|\boldsymbol{x} \bmod M\rangle$, respectively. Our results require n and $2n$, respectively. In adition, our total costs are $10n + O(1)$ and $20n$, respectively, which are also less than the circuits in [32] in terms of the equivalent Toffoli gate [34]. Of course, additional auxiliary qubits have to be used in our circuits. Our results may be used to improve Shor's circuit or other quantum algorithms based on a quantum computation model [13–16].

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61303039, 61373131), Natural Science Foundation of Shandong Province (Grant No. ZR2015FL024), Fundamental Research Funds for the Central Universities (Grant No. 2682014CX095), PAPD and CICAEET Funds, Open Foundation of Jiangsu Engineering Center of Network Monitoring (Nanjing University of Information Science & Technology) (Grant No. KJR1502), and Science Foundation Ireland (SFI) under the International Strategic Cooperation Award (Grant No. SFI/13/ISCA/2845).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Nielsen M, Chuang I. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2000. 150–280
2 Zhou C, Bao W S, Fu X Q. Decoy-state quantum key distribution for the heralded pair coherent state photon source with intensity fluctuations. Sci China Inf Sci, 2011, 41: 1136–1145
3 Wu H, Wang X B, Pan J W. Quantum communication, status and prospects (in Chinese). Sci China Inf Sci, 2014, 44: 296–311
4 Luo M X, Ma S Y, Chen X B, et al. Hybrid quantum states joining and splitting assisted by quantum dots in one-side optical microcavities. Phys Rev A, 2015, 91: 042326
5 Shor P W. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Santa Fe, 1994. 124–134
6 Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput, 1997, 26: 1484–1509
7 Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public key cryptography. Comm ACM, 1978, 21: 120–126
8 Guo P, Wang J, Geng X H, et al. A variable threshold-value authentication architecture for wireless mesh networks. J Internet Tech, 2014, 15: 929–936
9 Fu Z, Sun X, Liu Q, et al. Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans Commun, 2015, 98: 190–200
10 Ren Y, Shen J, Wang J, et al. Mutual verifiable provable data auditing in public cloud storage. J Internet Tech, 2015, 16: 317–324
11 Xia Z, Wang X, Sun X, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parall Distr Syst, in press. doi:10.1109/TPDS.2015.2401003
12 Li J, Li X, Yang B, et al. Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Foren Secur, 2015, 10: 507–518
13 Feynman R P. Simulating physics computers. Inter J Theor Phys, 1982, 21: 476–487
14 Deutsch D. Quantum theory, the Church-Turing principle and the universal quantum computer. Proc Royal Society London A, 1985, 400: 97–117
15 van Dam W, Hallgren S I L. Quantum algorithms for some hidden shift problems. SIAM J Comput, 2006, 36: 763–778
16 Luo M X, Deng Y. The independence of reduced subgroup-state. Inter J Theor Phys, 2014, 53: 3124–3134
17 Luo M X, Chen X B, Yang Y X, et al. Geometry of quantum computation with qudits. Sci Rep, 2014, 4: 4044
18 Luo M X, Wang X. Parallel photonic quantum computation assisted by quantum dots in one-side optical microcavities. Sci Rep, 2014, 4: 4732
19 Martin-Lopez E, Laing A, Lawson T, et al. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. Nature Phot, 2012, 6: 773–776
20 Lucero E, Barends R, Chen Y, et al. Computing prime factors with a Josephson phase qubit quantum processor. Nature Phys, 2012, 8: 719–723
21 Hao L, Long G L. Experimental implementation of a fixed-point duality quantum search algorithm in the nuclear magnetic resonance quantum system. Sci China Phys Mech Astronomy, 2011, 54: 936–941
22 Shende V, Bullock S S, Markov I L. Synthesis of quantum-logic circuits. IEEE Tran Comput AID Design, 2006, 26: 1000–1010
23 Beauregard S. Circuit for Shor's algorithm using $2n+3$ qubits. Quantum Inform Comput, 2003, 3: 175–185
24 Proos J, Zalka C. Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Inform Comput, 2003, 3: 317–344
25 Fowler A G, Devitt S J, Hollenberg L C L. Implementation of Shor's algorithm on a linear nearest neighbour qubit array. Quantum Inform Comput, 2004, 4: 237–251
26 Martí-López E, Laing A, Lawson T, et al. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. Nature Photon, 2012, 6: 773–776
27 Takahashi Y, Kunihiro N. A linear-size quantum circuit for addition with no ancillary qubits. Quantum Inform Comput, 2005, 5: 440–448
28 Takahashi Y, Kunihiro N. A fast quantum circuit for addition with few qubits. Quantum Inform Comput, 2008, 8: 636–649
29 Draper T G, Kutin S A, Rains E M, et al. A logarithmic-depth quantum carry-lookahead adder. Quantum Inform Comput, 2006, 6: 351–369
30 Takahashi Y, Tani S, Kunihiro N. Quantum addition circuits and unbounded fan-out. Quantum Inform Comput, 2010, 10: 0872–0890
31 Cuccaro S A, Draper T G, Kutin S A, et al. A new quantum ripple-carry addition circuit, In: 8th Workshop on Quantum Information Processing, Cambridge, 2005. 1–9
32 Thomsen M K, Axelsen H B. Optimization of a reversible (Quantum) ripple-carry adder. LNCS, 2008, 5204: 228–241
33 Markov I L, Saeedi M. Constant-optimized quantum circuits for modular multiplication and exponentiation. Quantum Infor Comput, 2012, 12: 0361–0394
34 Yu N K, Duan R Y, Ying M S. Five two-qubit gates are necessary for implementing the Toffoli gate. Phys Rev A, 2013, 88: 010304(R)