SCIENCE CHINA
Physics, Mechanics & Astronomy

• Article •

August 2020 Vol. 63 No. 8: 280311
https://doi.org/10.1007/s11433-020-1582-8

Efficient quantum arithmetic operation circuits for quantum image processing

Hai-Sheng Li[1*], Ping Fan[2], Haiying Xia[1], Huiling Peng[1], and Gui-Lu Long[3,4,5*]

[1]*College of Electronic Engineering, Guangxi Normal University, Guilin 541004, China;*
[2]*College of Information Engineering, East China JiaoTong University, Nanchang 330013, China;*
[3]*Department of Physics, Tsinghua University, Beijing 100084, China;*
[4]*Beijing Academy of Quantum Information Sciences, Beijing 100193, China;*
[5]*Frontier Science Center for Quantum Information, Beijing 100084, China*

Received February 10, 2020; accepted May 20, 2020; published online June 4, 2020

Efficient quantum circuits for arithmetic operations are vital for quantum algorithms. A fault-tolerant circuit is required for a robust quantum computing in the presence of noise. Quantum circuits based on Clifford+T gates are easily rendered fault-tolerant. Therefore, reducing the T-depth and T-Count without increasing the qubit number represents vital optimization goals for quantum circuits. In this study, we propose the fault-tolerant implementations for TR and Peres gates with optimized T-depth and T-Count. Next, we design fault-tolerant circuits for quantum arithmetic operations using the TR and Peres gates. Then, we implement cyclic and complete translations of quantum images using quantum arithmetic operations, and the scalar matrix multiplication. Comparative analysis and simulation results reveal that the proposed arithmetic and image operations are efficient. For instance, cyclic translations of a quantum image produce 50% T-depth reduction relative to the previous best-known cyclic translation.

quantum arithmetic operation, quantum fault tolerant circuit, quantum computation, quantum image processing

PACS number(s): 03.67.Lx, 42.30.Va, 03.67.Pp

Citation: H.-S. Li, P. Fan, H. Xia, H. Peng, and G.-L. Long, Efficient quantum arithmetic operation circuits for quantum image processing, Sci. China-Phys. Mech. Astron. **63**, 280311 (2020), https://doi.org/10.1007/s11433-020-1582-8

1 Introduction

Based on the principles of quantum mechanics, quantum computation utilizes quantum states [1, 2] to efficiently solve mathematical problems that are commonly insurmountable using classical computers [3, 4]. Also, quantum principles facilitate communication, thus providing an approach absolutely guarantees communication security [5, 6]. Quantum communication includes areas such as quantum secure direct communication [7-10], quantum key distribution [11], and quantum private query [12].

Several models exist for quantum computation, including blind quantum computation [13], Turing machine, and quantum circuit [14]. These models promote efficient implementation of quantum algorithms, such as the quantum search algorithm [4], discrete transform [15], wavelet transform [16, 17], Fourier transform [18], geometric transformation [19, 20], edge detection [21], and machine learning [22].

Since a unitary operation is reversible, quantum arithmetic operations, such as addition and multiplication, must be built from reversible logical circuits. The performance indicators of quantum circuits include the cost, delay, and ancilla num-

*Corresponding authors (Hai-Sheng Li, email: lhscqdx@163.com; Gui-Lu Long, email: gllong@tsinghua.edu.cn)

ber, with the optimization possible using synthesis methods [23-25]. The plain adder was implemented using Toffoli gates [26], involving $15n + 3$ quantum delay for n-bit addition. To decrease quantum delay, Draper et al. [27] designed a fast adder with $O(\log n)$ delay, and Takahashi et al. [28, 29] proposed two adders with $O(\log n)$ delay. To reduce ancillary qubits, Cuccaro et al. [30] designed a circuit with an ancillary qubit. In fact, quantum modular addition and subtraction are implemented with one ancillary qubit using the Fredkin gates [31]. Using the Toffoli, Peres [32], and TR gates [33], Thapliyal et al. [34] proposed an adder without ancilla. Circuits on controlled addition [35], quantum integer multiplier [35], and quantum comparator [27, 30, 36] have also been reported.

Fault-tolerant implementation of quantum gates is crucial for a robust quantum computing in the presence of noise [37]. Clifford and T circuits are widely accepted solutions for fault-tolerant implementation [38, 39], and Amy et al. [40, 41] proposed optimal depth implementations for the Toffoli, Fredkin, and Peres gates. Gosset et al. [42] introduced optimal T-count circuits for the Toffoli and Fredkin gates, with the Toffoli gate T-count reduction possible using ancillas [40,41,43]. To reduce circuit costs, Jones employed the Toffoli gates and measurement methods to implement a quantum addition [44]. Also, Thapliyal et al. [45, 46] designed fault-tolerant circuits for integer division using the Toffoli gates, whereas Munoz-Coreas and Thapliyal used these gates to create a controlled adder [47]. Based on the controlled adder, they introduced a T-count optimized integer multiplier without garbage, only requiring $4n + 1$ qubits [47].

Another possible quantum computation application area is in quantum image processing. In the past decade, quantum image processing was proposed [48, 49] and actively studied [50-54], with the advantages associated with the principle highlighted. For instance, the flexible representation of quantum images (FRQI) [50], normal arbitrary superposition state (NASS) [51], and novel enhanced quantum representation of digital images (NEQR) [52] store a $2^n \times 2^n$ grayscale image using $2n + 1$, $2n$, and $2n + 8$ qubits, respectively.

Without quantum arithmetic operations, quantum image translation applied to 2^n-sized images based on NASS [20] exhibited the complexity represented as $O(n^2 2^n)$. By contrast, quantum image translations were implemented using quantum arithmetic circuits with $O(n)$ complexity [55, 56], which demonstrates that efficient arithmetic operation circuits are crucial for quantum image processing.

Through the previously stated analysis, we study efficient fault-tolerant implementations of arithmetic operations for quantum image processing. We design fault-tolerant implementations of the TR gate, Peres gate, and variants, with better performance than those from the Toffoli and Fredkin gates (see Table 1). Therefore, based on the fault-tolerant implementations of the TR and Peres gates, we implemented fault-tolerant circuits for quantum arithmetic operations. Then, we created efficient circuits for cyclic and complete translations of quantum images using the quantum arithmetic operations.

2 Background

In this section, we briefly examine the general NEQR (GNEQR) proposed in ref. [54], and introduce the approach to obtain the vector forms of quantum image representations and matrix forms of quantum gates using MATLAB. Our study is based on the GNEQR.

2.1 Brief description of the GNEQR

The computational basis states $|0\rangle$, $|1\rangle$, and their dual states $\langle 0|$, $\langle 1|$ can be expressed in the row and column vectors as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \langle 0| = \begin{bmatrix} 1 & 0 \end{bmatrix}, \langle 1| = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

Considering that $|k\rangle$ is a basis state in a 2^n-dimensional Hilbert space for $k = 0, 1, \cdots, 2^n - 1$, the binary forms of $|k\rangle$ and $\langle k|$ are given as:

$$\begin{cases} |k\rangle = |k_{n-1}\rangle \otimes |k_{n-2}\rangle \otimes \ldots \otimes |k_0\rangle \\ \qquad = |k_{n-1}\rangle\, |k_{n-2}\rangle \cdots |k_0\rangle \\ \qquad = |k_{n-1} k_{n-2} \ldots k_0\rangle, \\ \langle k| = \langle k_{n-1}| \otimes \langle k_{n-2}| \otimes \ldots \otimes \langle k_0| \\ \qquad = \langle k_{n-1}|\langle k_{n-2}| \cdots \langle k_0| \\ \qquad = \langle k_{n-1} k_{n-2} \ldots k_0|, \end{cases} \tag{1}$$

where the decimal representation of k is $k = \sum_{j=0}^{n-1} k_j \times 2^j$, $k_0, k_1, \cdots, k_{n-1} \in \{0, 1\}$, and $\otimes$ is the symbol of tensor product.

An integer set is defined as follows:

$$C_m = \{0, 1, \cdots, 2^m - 1\}, \tag{2}$$

where m is an integer.

To store a 1D image of $2^n \times 1$, i.e., a column vector, the GNEQR is defined as follows:

$$\left|\Psi_n^m\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |f(j)\rangle\, |j\rangle, \tag{3}$$

where $|j\rangle = |j_{n-1} \cdots j_1 j_0\rangle$ and $f(j)$ denote the j-th location of a vector and the corresponding element value, respectively. $f(j) \in C_m$, therefore, 2^m is the number of possible values satisfying $f(j)$, for instance, $m = 1$ for binary images, and $m = 8$ for grayscale images.

To store a $2^{n-k} \times 2^k$ image, the GNEQR is expressed as:

$$\left|\Psi^m_{n-k,k}\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |f(x,y)\rangle |x\rangle |y\rangle, \tag{4}$$

where $|x\rangle = |j_{n-1} \cdots j_k\rangle$ and $|y\rangle = |j_{k-1} \cdots j_0\rangle$ are the X-axis and Y-axis of an image, and $j_0, \cdots j_k, \cdots, j_{n-1} \in \{0,1\}$. Here, $|f(x,y)\rangle$ denotes the color of the pixel on the coordinate (x,y), $f(x,y) \in C_m$.

For instance, a binary column vector $\begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$ can be stored in $\left|\Psi^1_2\right\rangle = \frac{1}{2}(|0\rangle|00\rangle + |1\rangle|01\rangle + |1\rangle|10\rangle + |0\rangle|11\rangle)$, where $[\ ]^{\mathrm{T}}$ is the transpose of matrix $[\ ]$. $\left|\Psi^1_{1,1}\right\rangle = \frac{1}{2}(|0\rangle|0\rangle|0\rangle + |1\rangle|0\rangle|1\rangle + |1\rangle|1\rangle|0\rangle + |0\rangle|1\rangle|1\rangle)$ can store a binary image of 2×2 presented in Figure 1.

2.2 The quantum gate matrix forms

The matrix of the outer product $|k\rangle\langle k|$ defines the product of $|k\rangle$ and $\langle k|$, for instance,

$$|0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Suppose U is the matrix of an n-qubit gate, two controlled-U gates are presented in Figure 2, with matrices that can be expressed as:

$$\begin{cases} C_{\mathrm{up}}(U) = |1\rangle\langle 1| \otimes U + |0\rangle\langle 0| \otimes I^{\otimes n}, \\ C_{\mathrm{un}}(U) = U \otimes |1\rangle\langle 1| + I^{\otimes n} \otimes |0\rangle\langle 0|, \end{cases} \tag{5}$$

where I is the identity matrix, and $I^{\otimes n}$ is the n fold tensor product $I \otimes I \otimes \cdots \otimes I$. For instance, unitary matrices of 1-qubit X and V are

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, V = \frac{1+i}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}, V^\dagger = \frac{1-i}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix},$$

and the corresponding controlled gates are presented in the dashed box in Figure 2, where i is an imaginary unit.

2.3 Fault-tolerant circuits

A specific instruction set for Clifford and T circuits [40, 41], comprising a few fault-tolerant gates, is given by X, XOR gates in Figure 2, and

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}, T^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{bmatrix}.$$

Using the specific instruction set, Amy et al. [40] give the optimal implementations of Toffoli gate, and Peres gate (see Figure 3(a) and (b)).

By modifying the Toffoli gate circuit in Figure 3(a), we obtain the optimal T-depth implementations for another Toffoli gate and Fredkin gate (see Figure 3(c) and (d)).

3 The optimal T-depth implementations for the TR gate, Peres gate, and their variants

In this section, we design the optimal T-depth implementations for the TR gate, Peres gate, and their variants.

3.1 TR gate and its variant

Inspired by the method in ref. [34], we use two symbols in Figure 4 to label the TR gate as $TR1$ and its variant $TR2$.

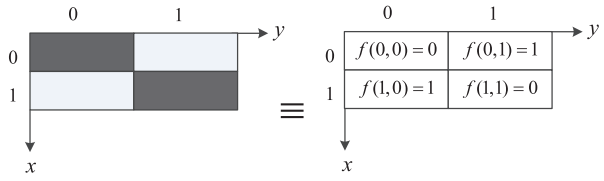

Figure 1 A 2×2 binary image.

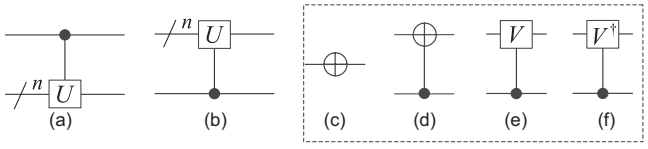

Figure 2 Representations of some quantum gates, including (a) $C_{\mathrm{up}}(U)$, (b) $C_{\mathrm{un}}(U)$, (c) the NOT gate (i.e., X gate), (d) the controlled-NOT gate (i.e., XOR gate), (e) the controlled-V gate, and (f) the controlled-$V^\dagger$ gate.

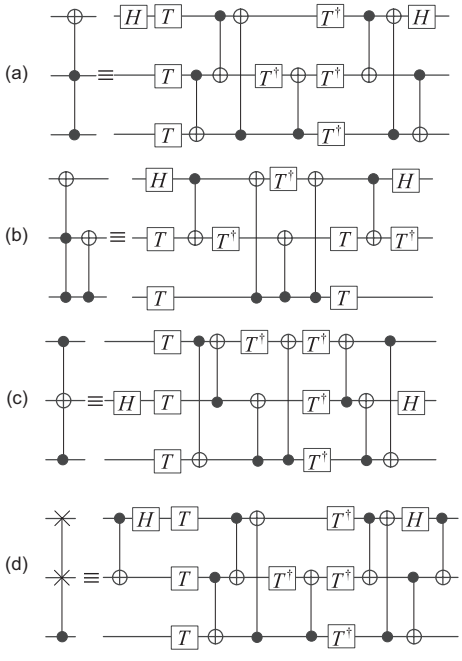

Figure 3 Optimal implementation circuits for (a) the Toffoli gate (T-depth 3, total depth 9), (b) the Peres gate (T-depth 4, total depth 8), (c) another Toffoli gate (T-depth 3, total depth 9), and (d) the Fredkin gate (T-depth 3, total depth 11).

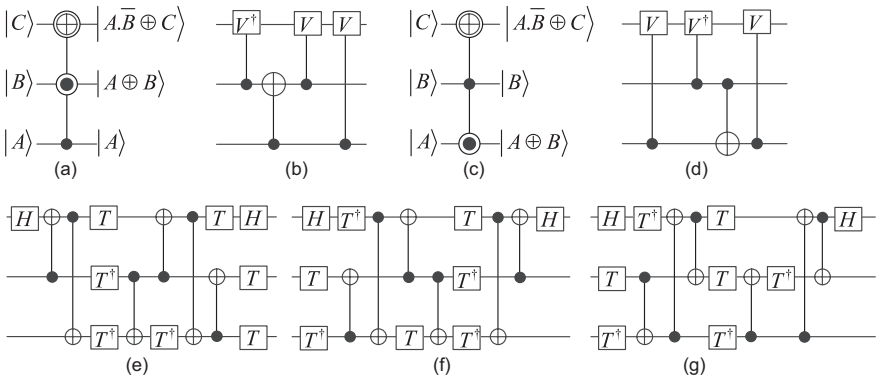

Figure 4 Illustration of the TR gate and its variant showing (a) the quantum symbol of $TR1$, (b) the implementation of $TR1$, (c) the quantum symbol of $TR2$, (d) the implementation of $TR2$, (e) the optimal T-depth implementation of $TR1$, (f) another optimal T-depth implementation of $TR1$, and (g) the optimal T-depth implementation of $TR2$.

We implement the TR gate and its variant using four elementary quantum gates in Figure 2, respectively. Their T-depths and total depths are 3 and 9, respectively.

$TR1$ and $TR2$ are implemented as:

$$TR1 : |C\rangle|B\rangle|A\rangle \rightarrow \left|A.\overline{B} \oplus C\right\rangle |A \oplus B\rangle |A\rangle,$$

and

$$TR2 : |C\rangle|B\rangle|A\rangle \rightarrow \left|A.\overline{B} \oplus C\right\rangle |B\rangle |A \oplus B\rangle,$$

where the symbols "." and " $\oplus$ " are multiplication and exclusive-or operators, respectively. $\overline{B}$ is equal to $(1 - B)$.

Using eq. (1), we can calculate the vector form of a state, such as:

$$\begin{aligned} |0\rangle &= \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]^{\mathrm{T}}, \\ |1\rangle &= \left[\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]^{\mathrm{T}}, \\ &\vdots \\ |7\rangle &= \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]^{\mathrm{T}}, \end{aligned} \tag{6}$$

where $|k\rangle$ $(k = 0, 1, \cdots, 7)$ comprises the tensor products of 3 computational basis states.

For simulation verification convenience, we calculate the matrix form of the gate $TR1$ as follows:

$$TR1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

From eq. (6), the matrix $TR1$ can be expressed as:

$$TR1 = [|0\rangle, |7\rangle, |2\rangle, |1\rangle, |4\rangle, |3\rangle, |6\rangle, |5\rangle]. \tag{7}$$

The matrix form of the gate $TR2$ is given by

$$TR2 = [|0\rangle, |5\rangle, |3\rangle, |2\rangle, |4\rangle, |1\rangle, |7\rangle, |6\rangle]. \tag{8}$$

3.2 The Peres gate and its variant

We designed the optimal T-depth implementation circuits for the Peres gate and its variant presented in Figures 5 and 6. Their T-depths and total depths were also 3 and 9, respectively. Compared with the circuit in Figure 3(b), the proposed Peres gate circuits exhibit lower T-depths.

$PG1$, $PG2$, and $PG3$ in Figures 5 and 6 implement

$$\begin{cases} PG1 : |C\rangle|B\rangle|A\rangle \rightarrow |A.B \oplus C\rangle|A \oplus B\rangle|A\rangle, \\ PG2 : |C\rangle|B\rangle|A\rangle \rightarrow |A.B \oplus C\rangle|B\rangle|A \oplus B\rangle, \\ PG3 : |B\rangle|C\rangle|A\rangle \rightarrow |B\rangle|A.B \oplus C\rangle|A \oplus B\rangle. \end{cases}$$

Their matrix forms are given as:

$$\begin{cases} PG1 = [|0\rangle, |3\rangle, |2\rangle, |5\rangle, |4\rangle, |7\rangle, |6\rangle, |1\rangle], \\ PG2 = [|0\rangle, |1\rangle, |3\rangle, |6\rangle, |4\rangle, |5\rangle, |7\rangle, |2\rangle], \\ PG3 = [|0\rangle, |1\rangle, |2\rangle, |3\rangle, |5\rangle, |6\rangle, |7\rangle, |4\rangle]. \end{cases} \tag{9}$$

Using eqs. (7)-(9), we obtain the following:

$$PG1 * TR1 = TR1 * PG1 = I_8, \tag{10}$$

and

$$\begin{aligned} TR2 * PG1 &= PG2 * TR1 \\ &= [|0\rangle, |2\rangle, |3\rangle, |1\rangle, |4\rangle, |6\rangle, |7\rangle, |5\rangle], \end{aligned} \tag{11}$$

where I_8 is an 8×8 identity matrix. The effects of the eqs. (10) and (11) are presented in Figure 7.

3.3 Quantum gate implementation circuit indicators

The quantum cost and delay are two key indicators of non-

fault tolerant circuits. According to refs. [23, 24], the quantum cost and delay of the elementary gates in the dashed box in Figure 2 are identical. For fault-tolerant circuits, the four key indicators are the T-depth, T-count, total depth, and total count. The implementation circuit indicators of these gates in Figures 3-6 are presented in Table 1. The data show

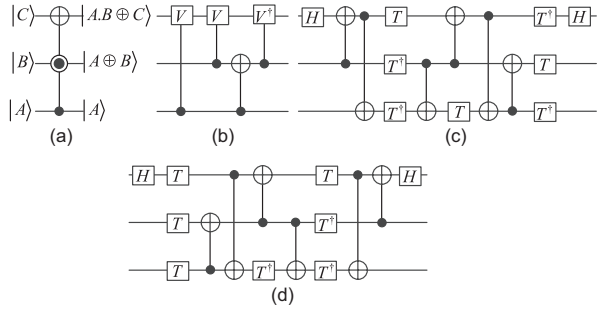

Figure 5 Illustration of a Peres gate $PG1$ showing (a) the quantum symbol of $PG1$, (b) the implementation of $PG1$, (c) the optimal T-depth implementation of $PG1$, and (d) another optimal T-depth implementation of $PG1$.

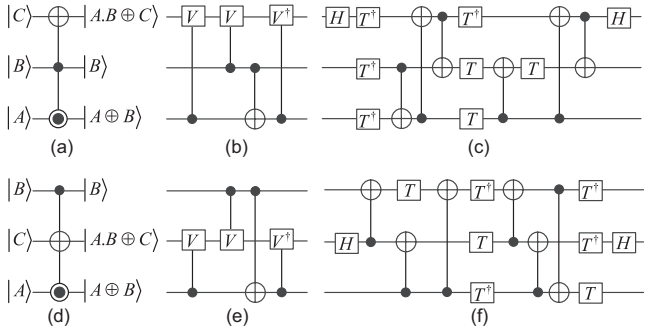

Figure 6 Variants of the Peres gate $PG2$ and $PG3$ showing (a) the quantum symbol of $PG2$, (b) the implementation of $PG2$, (c) the optimal T-depth implementation of $PG2$, (d) the quantum symbol of $PG3$, (e) the implementation of $PG3$, (f) and the optimal T-depth implementation of $PG3$.

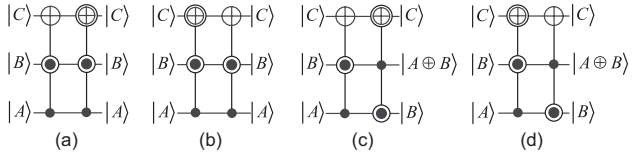

Figure 7 The relationships between the Peres and TR gates shown by (a) $TR1 * PG1$, (b) $PG1 * TR1$, (c) $TR2 * PG1$, and (d) $PG2 * TR1$.

Table 1 Performance indicators of the gates in Figures 3-6

Indicators	Toffoli	Fredkin	$TR1$	$TR2$	$PG1$	$PG2$	$PG3$
Cost	5	7	4	4	4	4	4
Delay	5	7	4	4	4	4	4
T-depth	3	3	3	3	3	3	3
Total depth	9	11	9	9	9	9	9
T-count	7	7	7	7	7	7	7
Total count	16	18	15	15	15	15	15

that $TR1$, $TR2$, $PG1$, $PG2$, and $PG3$ exhibit better performance than the Toffoli and Fredkin gates.

4 Quantum arithmetic operation circuit design

4.1 Quantum adder

A quantum adder implemented using the $PG1$ and $TR2$ gates is presented in Figure 8.

Suppose that $|b\rangle = |b_{n-1}b_{n-2}\cdots b_0\rangle$, $|a\rangle = |a_{n-1}a_{n-2}\cdots a_0\rangle$, and $|s\rangle = |s_n s_{n-1}\cdots s_0\rangle$, then, the quantum adder implemented the operation $s = a + b$. The executing processes are described as follows.

(1) Step 1. The circuit in step 1 involves $2n - 2$ XOR gates and n $PG1$ gates. When the fault-tolerant circuit in Figure 5(c) is used for implementing the Peres gate $PG1$, step 1 shows a T-depth of $3n$ and a total depth of $8n+2$. However, if we use the circuit in ref. [40] (see Figure 3(b)) to implement the Peres gate, the circuit in step 1 shows a T-depth of $4n$ and a total depth of $9n$. After step 1, the input state is transformed to the following:

$$|s_n\rangle\left(\bigotimes_{i=n-1}^{1}|b_i \oplus c_i\rangle\,|a_i \oplus c_i\rangle\right)|s_0\rangle\,|a_0\rangle\,,$$

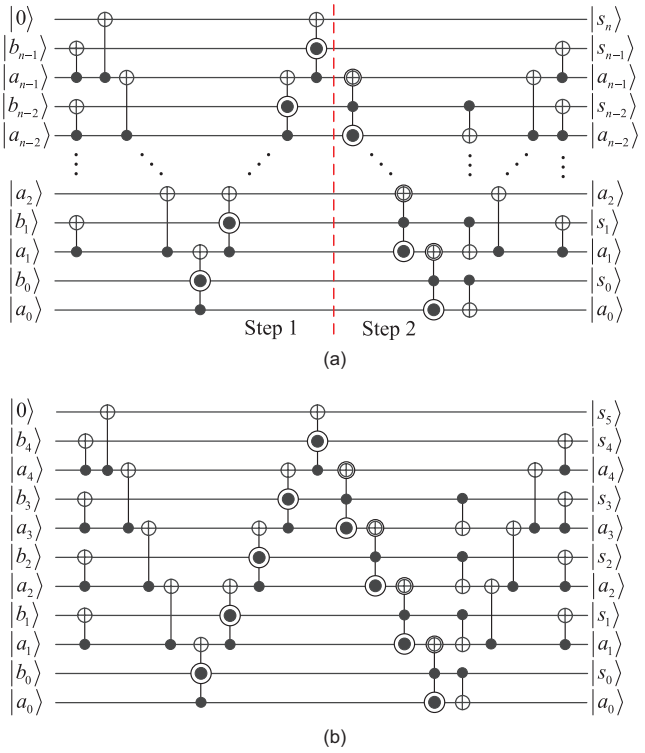

Figure 8 (Color online) Diagram showing (a) the implementation circuit of the quantum adder, and (b) a 5-bit quantum adder.

where

$$\begin{cases} c_i = a_{i-1}b_{i-1} \oplus b_{i-1}c_{i-1} \oplus a_{i-1}c_{i-1}, 2 \le i \le n, \\ c_1 = a_0 b_0. \end{cases} \tag{12}$$

(2) Step 2. The circuit in step 2 involves $n-1$ $TR2$ gates and $3n-4$ XOR gates with a T-depth of $3n-3$ and a total depth of $8n-5$. After step 2, the output is as follows: $|s_n\rangle \bigotimes_{i=n-1}^{0} (|s_i\rangle |a_i\rangle)$.

Thus, the proposed quantum adder has a T-depth of $6n-3$ and a total depth of $16n-3$. An example of the quantum adder is presented in Figure 8(b).

4.2 Quantum modular adder

We can modify the circuit of the quantum adder to obtain a quantum modular adder. The quantum modular adder implemented the operation $s = (a+b) \ mod \ 2^n$, where $|s\rangle = |s_{n-1} \cdots s_0\rangle$. An example of a quantum modular adder is presented in Figure 9.

The quantum modular adder comprises $n-1$ $PG1$ gates, $n-2$ $TR2$ gates, and $5n-10$ XOR gates. Thus, the proposed quantum modular adder has a T-depth of $6n-9$ and a total depth of $16n-19$.

4.3 Quantum modular subtractor

By substituting $TR1$ and $PG2$ for $PG1$ and $TR2$ in the circuit of the quantum modular adder, we obtain the circuit of the quantum modular subtractor. The quantum modular subtractor implemented the operation $d = (b-a) \ mod \ 2^n$, where $|d\rangle = |d_{n-1} \cdots d_0\rangle$. An example of the quantum modular subtractor is presented in Figure 10.

The quantum modular subtractor contains the $n-1$ $TR1$ gates, $n-2$ $PG2$ gates, and $5n-10$ XOR gates, with a T-depth of $6n-9$ and a total depth of $16n-19$.

4.4 Quantum comparator and quantum carry circuit

The quantum comparator implemented is presented in Figure 11(a) using the $TR1$ and $PG1$ gates. c shows the result of the comparison of two numbers, i.e., if $b \ge a$, $c = 0$; otherwise, $c = 1$. The executing processes are described as follows:

(1) Step 1. The circuit in step 1 contains the $2n-1$ XOR gates and n $TR1$ gates. When the fault-tolerant circuit in Figure 4(e) is used to implement the $TR1$ gate, step 1 shows a T-depth of $3n$ and a total depth of $8n+3$. After step 1, the input state is transformed to the following:

$$|c\rangle |b_{n-1} \oplus a_{n-1}\rangle \left(\bigotimes_{i=n-1}^{2} |a_i \oplus e_i\rangle |b_{i-1} \oplus e_{i-1}\rangle \right) |a_1 \oplus e_1\rangle |d_0\rangle |a_0\rangle,$$

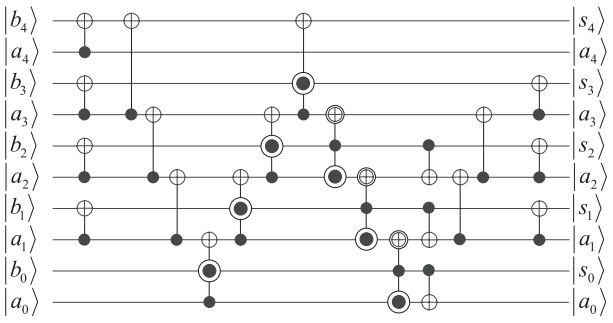

Figure 9 A 5-bit quantum modular adder.

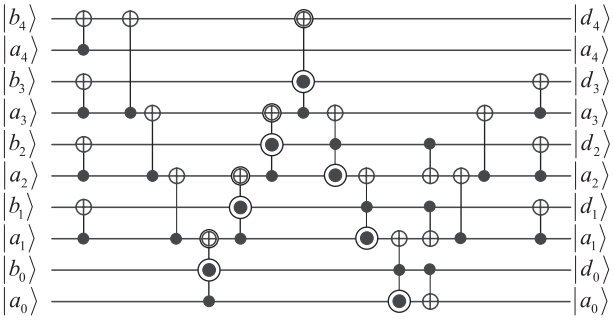

Figure 10 Illustration of a 5-bit quantum modular subtractor.

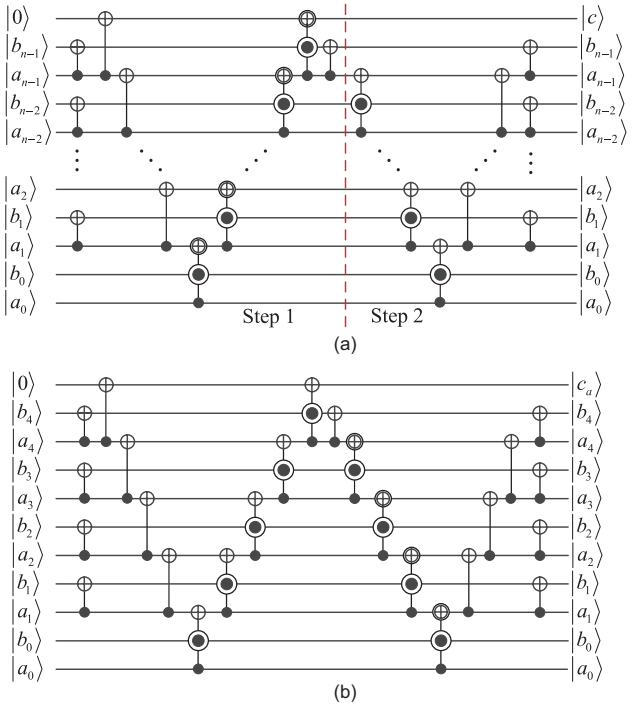

Figure 11 (Color online) Diagram showing (a) quantum comparator, and (b) quantum carry circuit.

where

$$\begin{cases} e_i = a_{i-1}\overline{b_{i-1}} \oplus \overline{b_{i-1}}e_{i-1} \oplus a_{i-1}e_{i-1}, 2 \le i \le n, \\ e_1 = a_0 \overline{b_0}. \end{cases} \tag{13}$$

(2) Step 2. The circuit in step 2 involves the $2n-3$ XOR gates and $n-1$ $PG1$ gates. When the fault-tolerant circuit in Figure 5(d) is used to implement the $PG1$ gate, step 2 shows a T-depth of $3n-3$ and a total depth of $8n-6$. After step 2, the output is as follows: $|c\rangle \overset{0}{\underset{i=n-1}{\otimes}} |b_i\rangle |a_i\rangle$.

The proposed comparator's T-depth is $6n-3$ and the total depth is $16n-3$.

Swapping $TR1$ and $PG2$ in the circuit of the comparator, we obtain the quantum carry circuit presented in Figure 11(b). The carry circuit implemented is expressed as:

$$|0\rangle \overset{0}{\underset{i=n-1}{\otimes}} |b_i\rangle |a_i\rangle \rightarrow |c_a\rangle \overset{0}{\underset{i=n-1}{\otimes}} |b_i\rangle |a_i\rangle,$$

where $c_a = 1$ for $a+b \geq 2^n$, and $c_a = 0$ for $a+b < 2^n$.

When the fault tolerant circuits in Figures 5(c) and 4(f) are used to implement the $PG1$ and $TR1$ gates, respectively, the proposed carry circuit also shows a T-depth of $6n-3$ and a total depth of $16n-3$.

4.5 Controlled quantum adder

The controlled quantum adder implemented the operation $t = da + b$, where $|t\rangle = |t_n t_{n-1} \cdots t_0\rangle$, $|a\rangle = |a_{n-1} a_{n-2} \ldots a_0\rangle$, and $d \in \{0,1\}$. For $n = 1$, we designed a gate named CA in Figure 12, which implemented the following:

$$|d\rangle |0\rangle |b_0\rangle |a_0\rangle \rightarrow |d\rangle |t_1\rangle |t_0\rangle |a_0\rangle.$$

The optimal T-depth circuit of CA is presented in Figure 12(c), with a T-depth of 9 and a total depth of 26.

When $n > 1$, the circuit of the controlled quantum adder is exhibited in Figure 13, and the executing processes are described as follows.

(1) Step 1. The circuit in step 1 contains the $2n-3$ XOR gates, Toffoli gate, CA gate, and $n-1$ $PG1$ gates. When the fault-tolerant circuit in Figure 5(c) is used to implement the Peres gate $PG1$, step 1 yields a T-depth of $3n+9$ and a total depth of $8n+29$. After step 1, the input state is transformed to the following:

$$|d\rangle |t_n\rangle |t_{n-1} \oplus a_{n-1}\rangle \left(\overset{1}{\underset{i=n-2}{\otimes}} |b_i \oplus c_i\rangle |a_i \oplus c_i\rangle \right) |a_0 \oplus b_0\rangle |a_0\rangle,$$

where c_i is presented in eq. (12).

(2) Step 2. The circuit in step 2 involves the $n-1$ $TR1$ gates, $n-1$ Toffoli gates, and $2n-3$ XOR gates with a T-depth of $6n-6$ and a total depth of $19n-19$. After step 2, the output is expressed as $|d\rangle |t_n\rangle \overset{0}{\underset{i=n-1}{\otimes}} (|t_i\rangle |a_i\rangle)$.

The proposed controlled adder shows a T-depth of $9n+3$ and a total depth of $27n+10$.

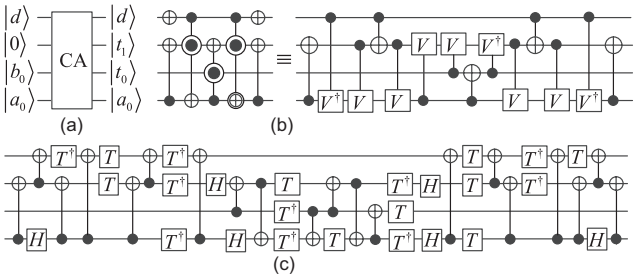

Figure 12 Representation of the gate CA showing (a) the quantum symbol, (b) the implementation of CA, and (c) the optimal T-depth implementation.

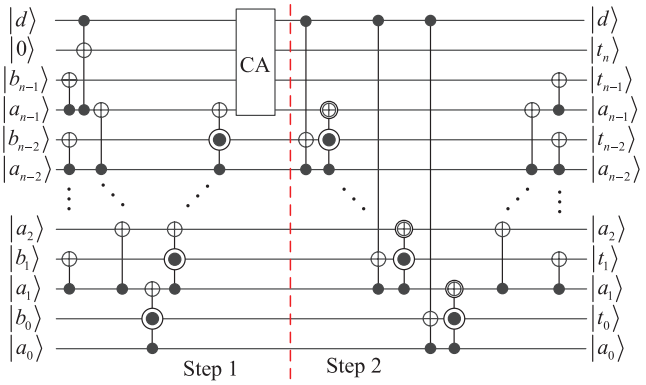

Figure 13 (Color online) Quantum controlled adder for $n > 1$.

4.6 Controlled quantum modular subtractor

The controlled quantum modular subtractor implemented the operation $l = (b - da) \bmod 2^n$, where $|l\rangle = |l_{n-1} \cdots l_1 l_0\rangle$, and $d \in \{0,1\}$. The circuit of the controlled quantum modular subtractor is presented in Figure 14. The executing processes are described as follows.

(1) Step 1. The circuit in step 1 comprises the $2n-6$ XOR gates, $n-1$ $TR1$ gate, and a $PG3$ gate, with a T-depth of $3n$ and a total depth of $8n$. After step 1, the input state is transformed to the following:

$$|d\rangle |l_{n-1}\rangle |a_{n-1} \oplus e_{n-1} \oplus d\rangle \left(\overset{1}{\underset{i=n-2}{\otimes}} |b_i \oplus e_i\rangle |a_i \oplus e_i\rangle \right) \times |b_0 \oplus a_0\rangle |a_0\rangle,$$

where e_i is presented in ref. (13).

(2) Step 2. The circuit in step 2 contains the $n-1$ $TR1$ gates, $n-1$ Toffoli gates, and $2n-3$ XOR gates with a T-depth of $6n-6$ and a total depth of $19n-19$. After step 2, the output is expressed as $|d\rangle |t_n\rangle \overset{0}{\underset{i=n-1}{\otimes}} (|t_i\rangle |a_i\rangle)$.

The proposed controlled modular subtractor shows a T-depth of $9n-6$ and a total depth of $27n-19$.

4.7 Controlled quantum modular adder

The controlled quantum modular adder implemented the operation $t = (da + b) \bmod 2^n$, where $|t\rangle = |t_{n-1} \cdots t_1 t_0\rangle$, and

$d \in \{0, 1\}$. By exchanging the order of the $TR1$ and $PG1$ gates in Figure 14, we obtain the circuit of a controlled quantum modular adder. An example of a controlled quantum modular addition is presented in Figure 15.

The proposed controlled modular adder possesses a T-depth of $9n - 6$ and a total depth of $27n - 19$.

4.8 Quantum multiplier

For convenience, the notations for the quantum arithmetic operations are listed in Figure 16. Since $a \times b = \sum_{i=0}^{m-1} a_i \times b \times 2^i$, an m-bit multiplication can be implemented by n-controlled additions. To implement the multiplication, we designed a quantum multiplier circuit presented in Figure 17(a) using controlled quantum adders. The multiplier implemented the following:

$$|a\rangle|0\rangle^{\otimes 2n}|b\rangle \rightarrow |a\rangle|p\rangle|b\rangle,$$

where $p = a \times b$, $a = a_{m-1} \cdots a_1 a_0$, $b = b_{n-1} \cdots b_1 b_0$, and $p = p_{m+n-1} \cdots p_1 p_0$.

For $n = m$, the proposed multiplier shows a T-depth of $9n^2 - 3n - 3$ and a total depth of $27n^2 - 34n + 19$.

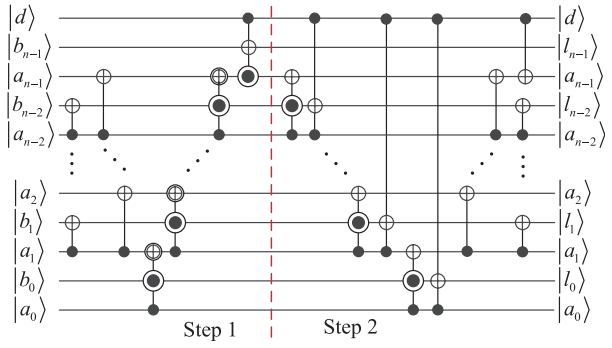

Figure 14 (Color online) The controlled quantum modular subtractor.

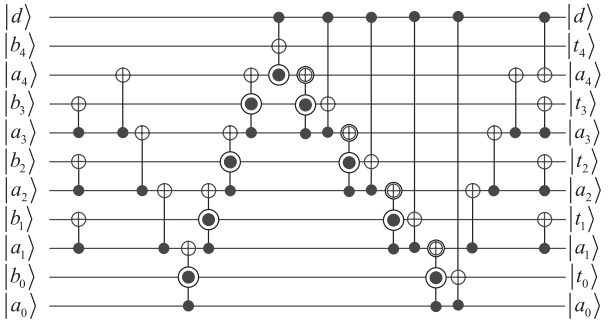

Figure 15 A 5-bit controlled quantum modular adder.

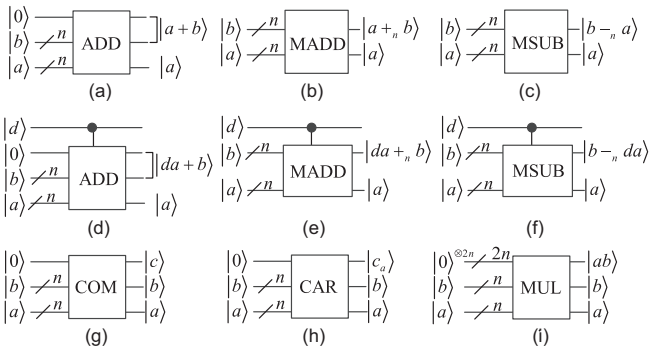

Figure 16 Notations for the quantum arithmetic operation circuits, including (a) quantum adder, (b) quantum modular adder, (c) quantum modular subtractor, (d) controlled quantum adder, (e) controlled quantum modular adder, (f) controlled quantum modular subtractor, (g) quantum comparator, (h) quantum carry circuit, and (i) quantum multiplier. $+_n$ and $-_n$ are the symbols of the modular 2^n addition and subtraction, respectively.

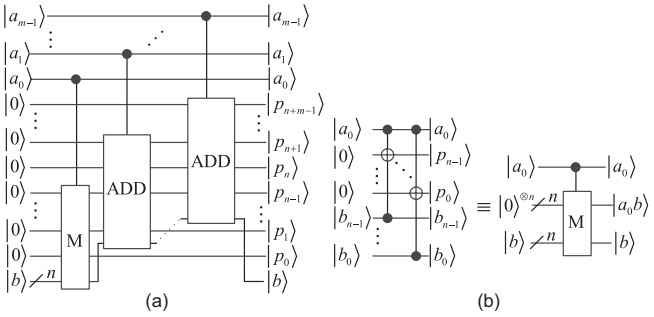

Figure 17 Illustration of (a) the circuit of the quantum multiplier and (b) the circuit of the quantum multiplier for $m = 1$.

4.9 Comparative analysis of the quantum arithmetic operation circuits

The quantum cost, delay, ancilla, and width are the key performance indicators of non-fault tolerant circuits. Meanwhile, the T-depth, total depth, T-count, total count, and ancilla are the principal performance indicators of fault-tolerant circuits. Using the fault-tolerant circuits in Figure 3 to implement the Toffoli and Fredkin gates, we can obtain performance indicators for the quantum arithmetic operations presented in Tables 2 and 3.

Since $(b - a) \bmod 2^n = \overline{(\bar{b} + a) \bmod 2^n}$, where $\bar{b}$ denotes bitwise implementation, the modular subtractors in refs. [31,45] add to the $2n$ NOT gates to produce the modular subtractor. From Table 2, we observe that the proposed arithmetic operations are superior to the corresponding operations in refs. [26, 30, 31, 34-36, 45, 47].

Three adders and a comparator in refs. [27-29] display logarithmic T-depths. For simplicity, we consider only the Toffoli gates in the four circuits previously stated, and the Peres and TR gates in the circuits. Their comparisons in Table 3 demonstrate that the adders and the comparator in

refs. [27-29] show smaller T-depths than that proposed in this study for high n. However, these require many ancillary qubits and enormous T-counts. Figure 18 shows that the T-depth of the proposed adder is lower than those stated in refs. [27-29] for $n < 16$, $n < 50$, and $n < 90$, respectively. The circuits for the adders and the comparator ($n = 10$, $n = 8$, or $n = 7$) are provided in refs. [27-29], with their comparison data presented in Table 4. We can see from Tables 3 and 4, and Figure 18 that the proposed adder and comparator are superior to the others when n is a low number. Since grayscale and RGB color images use 8 and 24 bits to represent a color, respectively, the operation numbers are not high for image processing. Furthermore, few ancillary qubits are also essential for efficient quantum image

Table 2 Comparisons of quantum arithmetic operations

Quantum arithmetic operations		Cost	Delay	T-depth	Total depth	T-count	Total count	Ancilla	Width
Adder	proposed	$13n-10$	$10n-4$	$6n-3$	$16n-3$	$14n-7$	$35n-21$	0	$2n+1$
	[26][a)]	$24n-10$	$24n-10$	$12n-6$	$40n-18$	$28n-14$	$68n-32$	$n+1$	$3n+1$
	[30][b)]	$17n-12$	$10n$	$6n-3$	$18n-4$	$14n-7$	$39n-23$	1	$2n+2$
	[34][c)]	$13n-10$	$11n-6$	$6n-3$	$18n-8$	$14n-7$	$35n-21$	0	$2n+1$
Modular adder	proposed	$13n-22$	$10n-14$	$6n-9$	$16n-19$	$14n-21$	$35n-55$	0	$2n$
	[30][b)]	$17n-28$	$10n-10$	$6n-9$	$18n-22$	$14n-21$	$39n-61$	1	$2n+1$
	[31][d)]	$16n-14$	$15n-13$	$6n-6$	$23n-21$	$14n-14$	$38n-36$	1	$2n+1$
Modular subtractor	proposed	$13n-22$	$10n-14$	$6n-9$	$16n-19$	$14n-21$	$35n-55$	0	$2n$
	[30][b)]	$19n-28$	$10n-8$	$6n-9$	$18n-2$	$14n-21$	$41n-61$	1	$2n+1$
	[31][d)]	$18n-14$	$15n-11$	$6n-6$	$23n-19$	$14n-14$	$40n-36$	1	$2n+1$
Comparator	proposed	$12n-8$	$10n-4$	$6n-3$	$16n-3$	$14n-7$	$34n-19$	0	$2n+1$
	[30][b)]	$18n-8$	$10n+1$	$6n-3$	$18n-3$	$14n-7$	$40n-19$	1	$2n+2$
	[36][e)]	$14n+1$	$14n+1$	$6n$	$22n+1$	$14n$	$36n+1$	1	$2n+2$
Controlled adder	proposed	$17n$	$15n+4$	$9n+3$	$27n+10$	$21n+5$	$50n+9$	0	$2n+2$
	[35][f)]	$24n+5$	$19n+10$	$9n+6$	$40n+18$	$28n+7$	$68n+16$	1	$2n+3$
	[47][g)]	$19n+4$	$17n+7$	$9n+6$	$29n+15$	$21n+14$	$52n+26$	1	$2n+3$
Controlled modular adder	proposed	$17n-18$	$15n-12$	$9n-6$	$27n-19$	$21n-14$	$50n-50$	0	$2n+1$
	[31][d)]	$24n-14$	$19n-9$	$9n-3$	$31n-13$	$28n-14$	$68n-36$	1	$2n+3$
	[45][h)]	$19n-16$	$17n-12$	$9n-6$	$29n-20$	$21n-14$	$52n-48$	0	$2n+1$
Controlled modular subtractor	proposed	$17n-18$	$15n-12$	$9n-6$	$27n-19$	$21n-14$	$50n-50$	0	$2n+1$
	[31][d)]	$26n-14$	$19n-7$	$9n-3$	$31n-11$	$28n-14$	$70n-36$	1	$2n+3$
	[45][h)]	$21n-16$	$17n-10$	$9n-6$	$29n-18$	$21n-14$	$54n-48$	0	$2n+1$
Multiplier	proposed	$17n^2-12n$	$15n^2-6n+4$	$9n^2-3n-3$	$27n^2-34n+19$	$21n^2-9n-5$	$50n^2-84n+50$	0	$4n$
	[35][f)]	$30n^2-4n+3$	$19n^2+16n-6$	$9n^2+6n$	$40n^2+24n-6$	$28n^2+7n$	$74n^2+7n+3$	1	$4n+1$
	[47][g)]	$19n^2-10n-4$	$17n^2+19n+7$	$9n^2-6$	$29n^2-5n-15$	$21n^2-14$	$52n^2-10n-26$	1	$4n+1$

a) is the design by Vedral et al. [26] in 1996; b) is the design by Cuccaro et al. [30] in 2004; c) is the design by Thapliyal and Ranganathan [34] in 2013; d) is the design by Thomsen et al. [31] in 2013; e) is the design by Xia et al. [36] in 2018; f) is the design by Jayashree et al. [35] in 2016; g) is the design by Munoz-Coreas and Thapliyal [47] in 2019; h) is the design by Thapliyal et al. [45] in 2016.

Table 3 Comparisons of adder and comparator for $n > 16$

Adder and comparator		Cost	Delay	T-depth	Total depth	T-count	Total count	Ancilla	Width
Adder	proposed	$8n-4$	$8n-4$	$6n-3$	$14n-3$	$14n-7$	$30n-15$	0	$2n+1$
	[27][a)] ($\approx$)	$50n$	$20\log n+40$	$12\log n+24$	$36\log n+72$	$70n$	$160n$	$2n$	$4n$
	[28][b)] ($\approx$)	$145n$	$150\log n$	$90\log n$	$270\log n$	$203n$	$464n$	$3n/\log n$	$2n+3n/\log n$
	[29][c)] ($\approx$)	$70n$	$90\log n$	$54\log n$	$162\log n$	$98n$	$224n$	$3n/\log n$	$2n+3n/\log n$
Comparator	proposed	$8n-4$	$8n-4$	$6n-3$	$14n-3$	$14n-7$	$30n-15$	0	$2n+1$
	[28][b)] ($\approx$)	$30n$	$10\log n+25$	$6\log n+15$	$18\log n+45$	$42n$	$96n$	$2n$	$4n$

a) is the design by Draper et al. [27] in 2004; b) is the design by Takahashi and Kunihiro [28] in 2008; c) is the design by Takahashi et al. [29] in 2009.

Table 4 Comparisons of examples of adder and comparator

Adder and comparator		Cost	Delay	T-depth	Total depth	T-count	Total count	Ancilla	Width
Adder	proposed ($n = 10$)	120	96	57	157	133	329	0	21
	[27][a)] ($n = 10$)	359	105	60	185	441	1052	14	35
	proposed ($n = 8$)	104	76	45	125	105	259	0	17
	[28][b)] ($n = 8$)	322	107	60	197	350	872	7	24
	[29][c)] ($n = 8$)	356	134	69	226	399	983	7	24
Comparator	proposed ($n = 7$)	76	66	39	109	91	219	0	15
	[28][b)] ($n = 7$)	172	59	33	103	203	491	9	24

a) is the design by Draper et al. [27] in 2004; b) is the design by Takahashi and Kunihiro [28] in 2008; c) is the design by Takahashi et al. [29] in 2009.

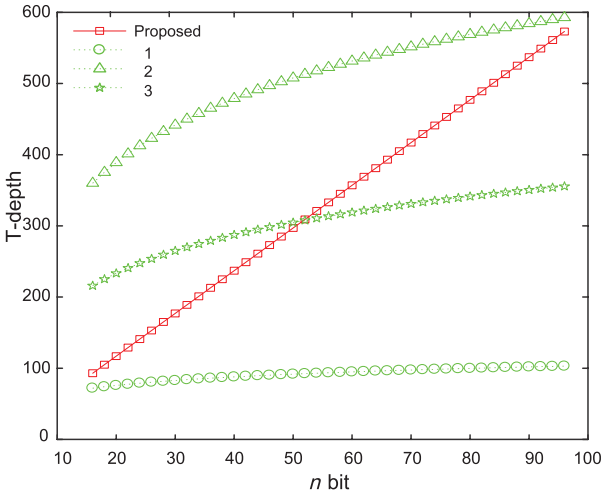

Figure 18 (Color online) T-depth of the proposed adder and three that already exist. Note: 1 is the design by Draper et al. [27] in 2004; 2 was designed by Takahashi and Kunihiro [28] in 2008; and 3 was introduced by Takahashi et al. [29] in 2009.

processing algorithms. Therefore, the proposed arithmetic operations are more suitable for quantum image processing.

5 Quantum image operators

In this section, considering a $2^{n-k} \times 2^k$ image is stored in the GNEQR state $\left|\Psi^m_{n-k,k}\right\rangle$ in eq. (4), we describe the implementation of quantum image operators using elementary quantum arithmetic operations. We simulate the image operations with a classical computer running in MATLAB R2017a, and Windows 7, with 64 GB RAM.

5.1 Cyclic translation of a quantum image

Using the modular adder and the modular subtractor in Figure 16, we designed the implementation circuits of the cyclic translation of the quantum image in Figure 19. After the cyclic right translation, the output is expressed as:

$$\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-k}-1}\sum_{y=0}^{2^k-1}|f(x,y)\rangle|x\rangle\left|(y+a) \bmod 2^k\right\rangle\right)|a\rangle.$$

Similarly, after the cyclic left translation, the output is as follows:

$$\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-k}-1}\sum_{y=0}^{2^k-1}|f(x,y)\rangle|x\rangle\left|(y-a) \bmod 2^k\right\rangle\right)|a\rangle.$$

When $a = 7$ and a binary image of 32×32 are the input, simulation results are presented in Figure 20.

5.2 The local cyclic translation of quantum image

Using the controlled modular adder and the controlled modular subtractor in Figure 16, we designed the implementation circuits for the local cyclic translation of a quantum image in Figure 21.

The circuits in Figure 21 transform the input $\left|\Psi^m_{n-k,k}\right\rangle|a\rangle$ into the following:

$$\frac{1}{\sqrt{2^n}}\left(\sum_{x=0}^{2^{n-k}-1}\sum_{y=0}^{2^{k-1}-1}|f(x,y)\rangle|x\rangle\left|z_1 \bmod 2^{k-1}\right\rangle\right)|a\rangle,$$

$$\frac{1}{\sqrt{2^n}}\left(\sum_{x=0}^{2^{n-k}-1}\sum_{y=0}^{2^{k-1}-1}|f(x,y)\rangle|x\rangle\left|z_2 \bmod 2^{k-1}\right\rangle\right)|a\rangle,$$

and

$$\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-k}-1}\sum_{y=0}^{2^{k-1}-1}|f(x,y)\rangle|x\rangle\left|z_3 \bmod 2^{k-1}\right\rangle\right)|a\rangle,$$

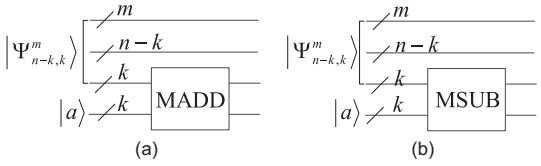

Figure 19 Cyclic translation of a quantum image showing (a) the cyclic right translation and (b) cyclic left translation of images.

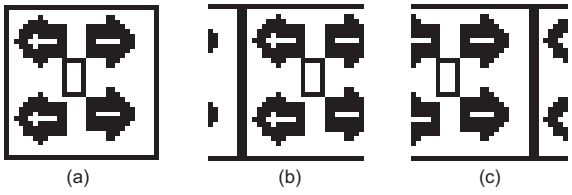

Figure 20 Simulation results of cyclic translation of a quantum image for (a) an original binary image of 32×32, (b) the cyclic right translation, and (c) the cyclic left translation.

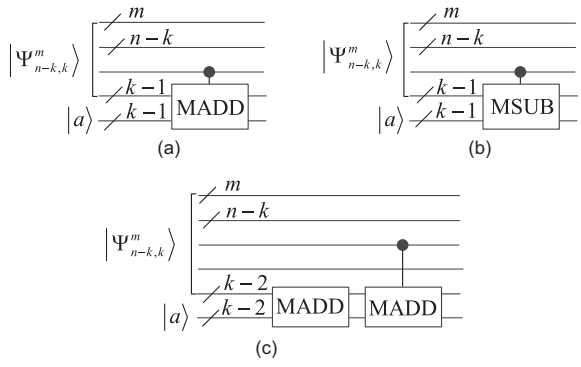

Figure 21 Local cyclic translation of a quantum image for (a) the local cyclic right translations of a quantum image, (b) the local cyclic right translations of a quantum image, and (c) combination translation.

where $z_1 = y_1 + j_k.a$, $z_2 = y_1 - j_k.a$, $z_3 = y_1 + a + j_k.a$, $|y\rangle = |j_k\rangle|y_1\rangle$, and $j_k \in \{0, 1\}$.

Let $a = 5$ and a binary image of 32×32 in Figure 22(a) be the input. The results after shifting the right side of the original image right and left by 5 pixels, respectively, are presented in Figure 22(b) and (c). Figure 22(d) shows that complex cyclic translations can be implemented by combining quantum arithmetic operations.

5.3 Complete quantum image translation

By substituting $TR1$ and $PG2$ for $PG1$ and $TR2$ in the circuit of the quantum adder, we obtain the inverse circuit of the quantum adder, which is named IADD. The IADD implemented the modular subtraction and the borrow operation of the most significant qubit. Using the carry circuit, the adder, the comparator in Figure 16, and the IADD, we designed the implementation circuits for the complete translation of a quantum image in Figure 23. The circuits in Figure 23 transform the input $|0\rangle^{\otimes m}\left|\Psi^m_{n-k,k}\right\rangle|a\rangle|0\rangle$ into the following:

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-k}-1}\left(\sum_{y=0}^{2^k-a-1}|f(x,y)\rangle|f(x,y)\rangle|x\rangle|y+a\rangle + \sum_{y=2^k-a}^{2^k-1}|0\rangle^{\otimes m}|f(x,y)\rangle|x\rangle\left|(y+a)\bmod 2^k\right\rangle\right)|a\rangle|0\rangle, \quad (14)$$

and

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-k}-1}\left(\sum_{y=0}^{a-1}|0\rangle^{\otimes m}|f(x,y)\rangle|x\rangle\left|(y-a)\bmod 2^k\right\rangle + \sum_{y=a}^{2^k-1}|f(x,y)\rangle|f(x,y)\rangle|x\rangle|y-a\rangle\right)|a\rangle|0\rangle. \quad (15)$$

Eqs. (14) and (15) show that the translated region is filled with zeroes. Next, we use the measurement method in ref. [54] to read out transformed images. When $a = 3$ and a binary image of 16×16 are the input, the simulation results are presented in Figure 24.

5.4 Scalar multiplication of a column vector

We can see from eq. (3) that $\left|\Psi^m_n\right\rangle$ can store a $2^n \times 1$ column vector. Therefore, using the multiplier in Figure 16, we designed the implementation circuit for the scalar multiplication of the column vector in Figure 25(a), which implemented the following:

$$|a\rangle|0\rangle^{m+h}\left|\Psi^m_n\right\rangle \rightarrow |a\rangle \otimes \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|af(j)\rangle|f(j)\rangle|j\rangle,$$

where a is an integer of h bits.

For $a = 2$, $h = 2$, $m = 4$, and $n = 3$, an example of scalar multiplication is presented in Figure 25(b).

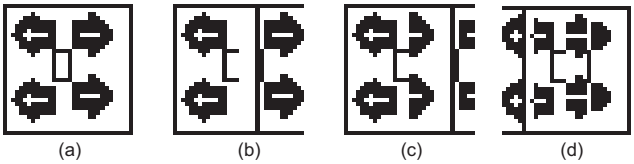

Figure 22 Simulation results for local cyclic translation of a quantum image showing (a) an original binary image of 32×32, (b) the local cyclic right translation, (c) the local cyclic left translation, and (d) the combination translation.

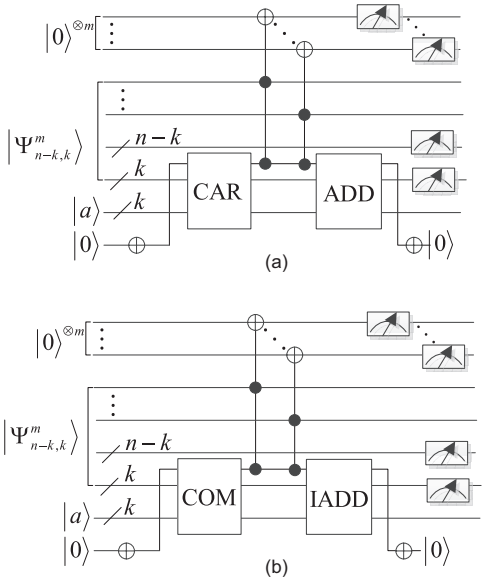

Figure 23 Complete translation of a quantum image showing (a) the complete right translation of a quantum image and (b) the complete left translation of a quantum image.

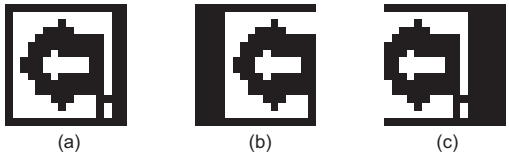

Figure 24 Simulation results of the complete translation of a quantum image for (a) an original binary image, (b) the complete right translation, and (c) the complete left translation.

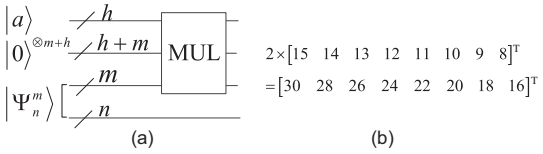

Figure 25 The scalar multiplication of a column vector for (a) the circuit of the scalar multiplication and (b) the simulation of the scalar multiplication.

Table 5 Comparisons of quantum image operations

Quantum image operations		Cost	Delay	T-depth	Total depth	T-count	Total count	Ancilla	Width
Cyclic translation	proposed	$13n-22$	$10n-14$	$6n-9$	$16n-19$	$14n-21$	$35n-55$	0	$3n+m$
	[20][a)]	$O(n^2 2^n)$	$O(n^2 2^n)$	$O(n^2 2^n)$	$O(n^2 2^n)$	$O(n^2 2^n)$	$O(n^2 2^n)$	0	$2n$
	[55][b)]	$24n-10$	$24n-10$	$12n-6$	$40n-18$	$28n-14$	$68n-32$	$n+1$	$4n+m+1$
	[56][c)]	$24n-10$	$24n-10$	$12n-6$	$40n-18$	$28n-14$	$68n-32$	$n+1$	$4n+2$
Local cyclic translation	proposed	$17n-35$	$15n-27$	$9n-15$	$27n-46$	$21n-35$	$50n-100$	0	$3n+m-1$
Entire right translation	proposed	$25n+5m-18$	$20n+5m-8$	$12n+3m-6$	$32n+9m-6$	$28n+7m-14$	$69n+16m-40$	1	$3n+2m+1$
	[55][b)]	$24n+10m-10$	$24n+10m-10$	$12n+6m-6$	$40n+18m-18$	$28n+14m-14$	$68n+32m-32$	$n+1$	$4n+2m+1$
Entire left translation	proposed	$25n+5m-18$	$20n+5m-8$	$12n+3m-6$	$32n+9m-6$	$28n+7m-14$	$69n+16m-40$	1	$3n+2m+1$
	[55][b)]	$O(n^2)+10m$	$O(n^2)+10m$	$O(n^2)+6m$	$O(n^2)+18m$	$O(n^2)+14m$	$O(n^2)+32m$	$3n+1$	$6n+2m+1$
Scalar multiplication	proposed	$17m^2-12m$	$15m^2-6m+4$	$9m^2-3m-3$	$27m^2-34m+19$	$21m^2-9m-5$	$50m^2-84m+50$	0	$4m+n$

a) is the design by Fan et al. [20] in 2016; b) is the design by Wang et al. [55] in 2015; c) is the design by Zhou et al. [56] in 2017.

5.5 Comparison analysis of quantum image operators

For comparison convenience, a $2^n \times 2^n$ image in 2^m gray levels is the input. Meanwhile, a $2^n \times 1$ vector is the input for scalar multiplication, where each vector element contains m bits. The performance indicators of the circuits for the proposed image operations are presented in Table 5. Meanwhile, comparisons of the translations of a quantum image [20, 55, 56] and the proposed translations are also presented in Table 5.

Table 5 contains the data showing that the proposed image operations are superior to the corresponding operations in refs. [20, 55, 56]. Complete translations in ref. [56] are unable to implement the reset of the carry and borrow bits, i.e., exiting a garbage output. To avoid this drawback, we used the quantum carry circuit and comparator to preserve auxiliary quantum state $|0\rangle$ presented in Figure 23.

6 Conclusions

In this study, we designed optimal T-depth implementations for the TR gate, Peres gate, and variants, which exhibited better performance than the Toffoli and Fredkin gates. Next, we produced efficient fault-tolerant circuits for quantum arithmetic operations. Comparative analysis revealed that the proposed arithmetic operations were superior to the existing corresponding operations. Meanwhile, the proposed adder and comparator exhibited many advantages for ancillary qubits, with circuit widths better than those of the existing corresponding operations. Furthermore, the proposed arithmetic operations were more suitable for quantum image processing. We implemented quantum image processing operations using the quantum arithmetic operations. These quantum image operations included cyclic and complete translations. Performance analysis revealed that the cyclic and complete translations involved a maximum T-depth of $12n+3m-6$. The proposed cyclic translations produced 50% T-depth reduction relative to existing best-known cyclic translation. We demonstrated that the proposed image operations are efficient. As future work, the results in this study will be extended and applied in practical image processing, such as coding and denoising.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61762012, and 61763014), the Science and Technology Project of Guangxi (Grant No. 2018JJA170083), the National Key Research and Development Plan (Grant Nos. 2018YFC1200200, and 2018YFC1200205), the Fund for Distinguished Young Scholars of Jiangxi Province (Grant No. 2018ACB2101), the Natural Science Foundation of Jiangxi Province of China (Grant No. 20192BAB207014), and the Science and Technology Research Project of Jiangxi Provincial Education Department (Grant No. GJJ190297).

1 P. Benioff, J. Stat. Phys. **22**, 563 (1980).
2 D. Deutsch, Proc. R. Soc. Lond. A **400**, 97 (1985).
3 P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
4 L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997), arXiv: quant-ph/9706033.
5 S. S. Chen, L. Zhou, W. Zhong, and Y. B. Sheng, Sci. China-Phys. Mech. Astron. **61**, 090312 (2018).
6 Z. X. Cui, W. Zhong, L. Zhou, and Y. B. Sheng, Sci. China-Phys. Mech. Astron. **62**, 110311 (2019).
7 F. Gao, S. J. Qin, W. Huang, and Q. Y. Wen, Sci. China-Phys. Mech. Astron. **62**, 070301 (2019).
8 Z. R. Zhou, Y. B. Sheng, P. H. Niu, L. G. Yin, G. L. Long, and L. Hanzo, Sci. China-Phys. Mech. Astron. **63**, 230362 (2020), arXiv: 1805.07228.
9 R. He, J. G. Ma, and J. Wu, Europhys. Lett. **127**, 50006 (2019).
10 Z. Gao, T. Li, and Z. Li, Europhys. Lett. **125**, 40004 (2019).
11 L. Zhou, Y. B. Sheng, and G. L. Long, Sci. Bull. **65**, 12 (2020).
12 J. W. Wu, Z. S. Lin, L. G. Yin, and G. L. Long, Quantum Eng. **1**, e26 (2019).
13 Y. B. Sheng, and L. Zhou, Phys. Rev. A **98**, 052343 (2018).
14 M. A. Nielsen, and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000), p.129.
15 L. Hao, and G. L. Long, Sci. China-Phys. Mech. Astron. **54**, 936 (2011).
16 H. S. Li, P. Fan, H. Xia, S. Song, and X. He, Sci. Rep. **8**, 13884 (2018).
17 H. S. Li, P. Fan, H. Xia, and S. Song, Inf. Sci. **504**, 113 (2019).
18 H. S. Li, P. Fan, H. Xia, S. Song, and X. He, Quantum Inf. Process. **17**, 333 (2018).
19 P. Q. Le, A. M. Iliyasu, F. Dong, and K. Hirota, Int. J. Appl. Marth. **40**, 113 (2010).
20 P. Fan, R. G. Zhou, N. Jing, and H. S. Li, Inf. Sci. **340-341**, 191 (2016).
21 Y. Zhang, K. Lu, and Y. H. Gao, Sci. China Inf. Sci. **58**, 1 (2015).
22 Y. B. Sheng, and L. Zhou, Sci. Bull. **62**, 1025 (2017).
23 W. N. N. Hung, X. Y. Song, G. W. Yang, J. Yang, and M. Perkowski,

IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **25**, 1652 (2006).
24 A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995), arXiv: quant-ph/9503016.
25 Y. Liu, G. L. Long, and Y. Sun, Int. J. Quantum Inform. **06**, 447 (2008).
26 V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996), arXiv: quant-ph/9511018.
27 T. G. Draper, S. A. Kutin, E. M.Rains, and K. M. Svore, arXiv: quant-ph/0406142.
28 Y. Takahashi, and N. Kunihiro, Quantum Inform. Comput. **8**, 636 (2008).
29 Y. Takahashi, S. Tani, and N. Kunihiro, arXiv: 0910.2530.
30 S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, arXiv: quant-ph/0410184.
31 M. K. Thomsen, R. Glück, and H. B. Axelsen, J. Phys. A-Math. Theor. **43**, 382002 (2013).
32 A. Peres, Phys. Rev. A **32**, 3266 (1985).
33 H. Thapliyal, and N. Ranganathan, IEEE computer society Annual symposium on VLSI (IEEE, Tampa, 2009), pp. 229-234.
34 H. Thapliyal, and N. Ranganathan, J. Emerg. Technol. Comput. Syst. **9**, 17 (2013).
35 H. V. Jayashree, H. Thapliyal, H. R. Arabnia, and V. K. Agrawal, J. Supercomput. **72**, 1477 (2016).
36 H. Xia, H. Li, H. Zhang, Y. Liang, and J. Xin, Int. J. Theor. Phys. **57**, 3727 (2018).
37 X. Zhou, D. W. Leung, and I. L. Chuang, Phys. Rev. A **62**, 052316 (2000), arXiv: quant-ph/0002039.
38 B. Giles, and P. Selinger, Phys. Rev. A **87**, 032332 (2013), arXiv: 1212.0506.
39 V. Kliuchnikov, D. Maslov, and M. Mosca, Phys. Rev. Lett. **110**, 190502 (2013), arXiv: 1212.0822.
40 M. Amy, D. Maslov, M. Mosca, and M. Roetteler, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **32**, 818 (2013).
41 M. Amy, D. Maslov, and M. Mosca, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **33**, 1476 (2014).
42 D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, arXiv: 1308.4134.
43 C. Gidney, arXiv: 1212.5069v1.
44 C. Jones, arXiv: 1709.06648v3.
45 H. Thapliyal, T. S. S. Varun, and E. Munoz-Coreas, arXiv: 1609.01241.
46 H. Thapliyal, E. Munoz-Coreas, T. S. S. Varun, and T. Humble, IEEE Trans. Emerg. Top. Comput. **52**, 1 (2020).
47 E. Munoz-Coreas, and H. Thapliyal, IEEE Trans. Comput. **68**, 729 (2019).
48 G. Beach, C. Lomont, and C. Cohen, in *32nd Applied Imagery Pattern Recognition Workshop* (IEEE, Washington, 2003), pp. 39-44.
49 S. E. Venegas-Andraca, and S. Bose, in *Proc. SPIE Conference Quantum Information and Computation* (SPIE, Orlando, 2003), pp. 137-147.
50 P. Q. Le, F. Dong, and K. Hirota, Quantum Inf. Process. **10**, 63 (2011).
51 H. S. Li, Q. Zhu, R. G. Zhou, M. C. Li, I. Song, and H. Ian, Inf. Sci. **273**, 212 (2014).
52 Y. Zhang, K. Lu, Y. Gao, and M. Wang, Quantum Inf. Process. **12**, 2833 (2013).
53 F. Yan, A. M. Iliyasu, Y. Guo, and H. Yang, Theor. Comput. Sci. **752**, 71 (2018).
54 H. S. Li, P. Fan, H. Y. Xia, H. Peng, and S. Song, IEEE Trans. Circuits Syst. I: Reg. Papers **66**, 341 (2018).
55 J. Wang, N. Jiang, and L. Wang, Quantum Inf. Process. **14**, 1589 (2015).
56 R. G. Zhou, C. Tan, and H. Ian, Int. J. Theor. Phys. **56**, 1382 (2017).