IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 13 August 2018; revised 15 March 2019; accepted 22 March 2019.
Date of publication 15 April 2019; date of current version 4 June 2021.

Digital Object Identifier 10.1109/TETC.2019.2910870

Quantum Circuit Designs of Integer Division Optimizing T-count and T-depth

HIMANSHU THAPLIYAL, EDGARD MUÑOZ-COREAS, T. S. S. VARUN, AND TRAVIS S. HUMBLE

H. Thapliyal, E. Muñoz-Coreas, and T.S.S. Varun are with the Department of Electrical and Computer Engineering,
University of Kentucky, Lexington KY 40506, USA
T.S. Humble is with the Quantum Computing Institute, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
CORRESPONDING AUTHOR: H. THAPLIYAL (hthapliyal@uky.edu)

ABSTRACT Quantum circuits for mathematical functions such as division are necessary to use quantum computers for scientific computing. Quantum circuits based on Clifford+T gates can easily be made fault-tolerant but the T gate is very costly to implement. The small number of qubits available in existing quantum computers adds another constraint on quantum circuits. As a result, reducing T-count and qubit cost have become important optimization goals. The design of quantum circuits for integer division has caught the attention of researchers and designs have been proposed in the literature. However, these designs suffer from excessive T gate and qubit costs. Many of these designs also produce significant garbage output resulting in additional qubit and T gate costs to eliminate these outputs. In this work, we propose two quantum integer division circuits. The first proposed quantum integer division circuit is based on the restoring division algorithm and the second proposed design implements the non-restoring division algorithm. Both proposed designs are optimized in terms of T-count, T-depth and qubits. Both proposed quantum circuit designs are based on (i) a quantum subtractor, (ii) a quantum adder-subtractor circuit, and (iii) a novel quantum conditional addition circuit. Our proposed restoring division circuit achieves average T-count savings from 79.03 to 91.69 percent compared to the existing works. Our proposed non-restoring division circuit achieves average T-count savings from 49.22 to 90.03 percent compared to the existing works. Further, both our proposed designs have linear T-depth. We also illustrate the application of the proposed quantum division circuits in quantum image processing with a case study of quantum bilinear interpolation.

INDEX TERMS Quantum computing, quantum circuits, integer division, restoring division, non-restoring division, clifford+T gates, quantum arithmetic

I. INTRODUCTION

Among the emerging computing paradigms, quantum computing appears promising due to its applications in number theory, encryption, search and scientific computation [1], [2], [3], [4], [5], [6], [7], [8]. Quantum circuits for integer arithmetic operations such as addition, subtraction, multiplication and division are required in the quantum circuit implementations of several quantum algorithms such as the class number algorithm, the triangle finding algorithm, Shor's integer factoring algorithm and solving exponential congruences [1], [9], [10]. Quantum arithmetic circuits for division can be used in the circuit implementation of quantum algorithms such as those for computing shifted quadratic character problems, principal ideal problems and hidden shift problems [2], [3], [4]. Quantum division circuits are also used in the circuit implementations of higher level functions such as the greatest common divisor, the square root or the reciprocal [11], [12]. Quantum circuits for the square root and reciprocal are used in quantum algorithms such as those for computing roots of polynomials, solving linear systems of equations and the principal ideal problem [5], [6], [7]. Further, a quantum division circuit is needed to realize quantum algorithms in scientific and image processing applications. For instance, division can be used to realize fractional powers, the natural logarithm or bilinear interpolation [6], [8], [13]. Thus, researchers have included dedicated libraries of basic quantum integer arithmetic functions such as division in quantum programming languages such as Quipper and $LIQUi|\rangle$ and in quantum computing design tools [1], [14].

Quantum computation can be performed on quantum circuits built from quantum gates. Any constant inputs in the quantum circuit are called ancillae. Garbage outputs are any outputs which exist in the quantum circuit to preserve one-to-one mapping but are neither one of the primary inputs nor a useful output. The inputs regenerated at the outputs are not considered garbage outputs [15]. Ancillae and garbage outputs are circuit overhead that need to be minimized.

The fault-tolerant implementation of quantum circuits is gaining the attention of researchers because physical quantum computers are prone to noise errors [16], [17], [18]. Fault-tolerant implementations of quantum gates and quantum error correcting codes can be used to overcome the limits imposed by noise errors in implementing quantum computing [17], [18], [19]. Recently, researchers have implemented quantum logic gates such as the Toffoli gate, Fredkin gate and quantum full adder with fault-tolerant implementations of the Clifford +T gates due to their demonstrated tolerance to noise errors [18], [19], [20]. However, the increased tolerance to noise errors comes with the increased implementation overhead associated with the quantum T gate [18], [19], [20]. The Clifford gate set alone is not universal [16], [17], [20], [21]. Thus, an additional non-Clifford gate such as the quantum T gate must be used to obtain a universal gate set. However, the fault tolerant implementations of the T gate are significantly more costly relative to the fault tolerant implementations of the Clifford gates [17], [20], [21]. Because of the increased cost to realize the T gate, T-count and T-depth are important performance measures for fault-tolerant quantum circuit design.

The design of circuits for integer division on quantum computers is an active area of research. Works such as [22], [23], [24], [25], [26], [27], [28] and [29] have been proposed in the literature to design division circuits based on reversible gates. While interesting designs, the division circuits in [25], [26], [27], [28] and [29] cannot be implemented in quantum hardware because they depend on reversible registers with feedback or reversible latches with feedback. In contrast, designs such as those proposed in [22], [23] and [24] present dividers that can be used in quantum computation. The design in [22] implements the restoring division algorithm and the designs in [23] are based on the non-restoring division algorithm. The design presented in the recent work in [24] uses a novel division algorithm. The design in [22] has a significant overhead in terms of T gates because it depends on quantum gates that cannot be exactly constructed using Clifford+T gates. The Clifford+T gate approximations for these gates are costly in terms of T-count [30]. Further, the T-count increases as the accuracy of the approximation of these gates is improved [30]. In contrast the dividers in [23] depend on quantum gates that can be exactly realized with Clifford+T gates. At most 7 T gates are required to implement each logic gate in [23]. Thus, the designs in [23] require significantly fewer T gates than the design presented in [22]. However, the designs in [23] produces significant garbage output. Thus, the dividers in [23] will have additional ancillae and T gate overhead from removing these garbage outputs. Designs such as those by [26], [23] and [22] are based on the restoring division algorithm or non-restoring division algorithm. The restoring division algorithm and non-restoring division algorithm are iterative algorithms that can be implemented on quantum computers. The restoring division algorithm is shown in Algorithm 1 (see Figure 3) and the non-restoring algorithm is shown in Algorithm 2 (see Figure 6). A difference between them is that one bit of the remainder is produced per iteration of Algorithm 1 and one bit of the quotient is produced for each iteration of Algorithm 2. Also, the restoring division algorithm will restore the partial quotient if the produced remainder bit is 0. On the other hand,the non-restoring division algorithm will perform addition if the produced quotient bit is 0 and subtraction otherwise. A recent design presented in [24] also depends on quantum gates that can be exactly realized with Clifford+T gates. However, the design methodology presented in [24] generates a quantum circuit that suffers from significant T gate and qubit cost overhead. In addition, the design in [24] produces significant garbage output. Thus, the divider in [24] will have additional ancillae and T gate overhead from removing these garbage outputs.

To address the shortcomings of the existing work this paper presents two designs for quantum circuit integer division based on Clifford+T gates. The first proposed quantum circuit is based on the restoring division algorithm and the second proposed quantum circuit is based on the non-restoring division algorithm. Both proposed quantum integer division circuits are based on (i) a quantum subtractor, (ii) a quantum adder-subtractor circuit, and (iii) a novel quantum conditional addition circuit. The preliminary version of this work is available in [31], [32]. The proposed quantum restoring division circuit has quadratic T-count, linear T-depth and requires $3 \cdot n$ qubits (where n is the size of the inputs). The proposed non-restoring division circuit has quadratic T-count, linear T-depth and requires $3 \cdot n - 1$ qubits (where n is the size of the inputs).

This paper is organized as follows: Section II discusses the Clifford+T gate set, background on resource cost measures, the quantum subtractor, quantum adder-subtractor circuit and the novel quantum conditional addition circuit. The proposed quantum restoring integer division circuit is presented in Section III while the comparison with the existing works is presented in Section IV. In Section V, the design of the proposed quantum non-restoring integer division circuit is discussed and Section VI illustrates comparison with the existing works. Section VII illustrates the application of the proposed quantum division circuits in quantum image processing with a case study of quantum bilinear interpolation.

II. BACKGROUND

A. QUANTUM GATES

Fault-tolerant implementation of quantum circuits is gaining the attention of researchers because physical quantum computers are prone to noise errors [16], [17], [18]. Recently, researchers have implemented quantum logic gates and circuits with fault-tolerant implementations of the Clifford+T gate set due to its demonstrated tolerance to noise errors [18], [19], [20]. Figure 1 presents the gates that make up the Clifford+T quantum gate family. The T gate is needed because the Clifford gate set is not universal alone [16], [17], [20], [21]. Evaluating quantum circuit performance in terms of T-count and T-depth is of interest to researchers because the

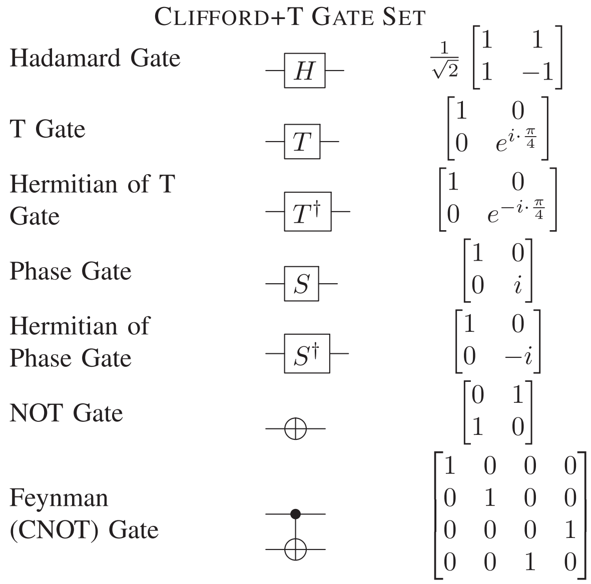

CLIFFORD+T GATE SET
Hadamard Gate
H
$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
T Gate
T
$\begin{bmatrix}1 & 0\\ 0 & e^{i\cdot\frac{\pi}{4}}\end{bmatrix}$
Hermitian of T Gate
$T^\dagger$
$\begin{bmatrix}1 & 0\\ 0 & e^{-i\cdot\frac{\pi}{4}}\end{bmatrix}$
Phase Gate
S
$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
Hermitian of Phase Gate
$S^\dagger$
$\begin{bmatrix}1 & 0\\ 0 & -i\end{bmatrix}$
NOT Gate
$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Feynman (CNOT) Gate
$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$

FIGURE 1. **The quantum gate set used in this work.**

fault-tolerant implementation of the T gate is significantly more costly than the fault-tolerant implementation costs of the other Clifford+T gates [17], [20], [21]. The number of qubits in a quantum circuit is a resource measure of interest because of the limited number of qubits available on existing quantum computers [33]. We now define the T-count, T-depth and qubit cost resource measures.

- T-count: T-count is the total number of T gates used in the quantum circuit.
- T-depth: T-depth is the number of T gate layers in the circuit, where a layer consists of quantum operations that can be performed simultaneously.
- Qubit cost: Qubit cost is the total number of qubits required to design the quantum circuit.

B. DESIGN OF QUANTUM SUBTRACTOR

The subtractor circuit takes two n bit, 2's complement binary inputs a and b. At the end of computation, the input a emerges unchanged and the input b is transformed to the difference of b from a. The quantum subtractor calculates $(\overline{\overline{b}+a})$ which is equivalent to $b-a$ [34]. A quantum ripple carry adder is used to realize the quantum subtractor circuit. We use the quantum ripple carry adder proposed in [35] for developing the quantum subtractor circuit. We chose the ripple carry adder proposed in [35] because this adder has been shown in the literature to be efficient in terms of gates and has a T-depth that is constant and independent of the circuit size n. We determined that this quantum subtractor will have a T-count of $14\cdot n-14$ and a T-depth of 10. Thus, the quantum subtraction circuit used in our proposed dividers will have a T-count of order $\mathcal{O}(n)$ and a T-depth of order $\mathcal{O}(1)$.

C. DESIGN OF QUANTUM ADDER-SUBTRACTOR (Ctrl-Addsub) CIRCUIT

The quantum adder-subtractor (*Ctrl-AddSub*) circuit takes two n bit, 2's complement binary inputs a and b and a single one bit input *ctrl*. Operation of the quantum *Ctrl-AddSub* circuit is conditioned on the value of *ctrl*. When *ctrl* is high, the circuit calculates $b-a$. When the *ctrl* input is low, the circuit calculates $b+a$. The quantum *Ctrl-AddSub* calculates $(\overline{\overline{b}+a})$ when *ctrl* is high. The expression $(\overline{\overline{b}+a})$ is equivalent to $b-a$. The quantum *Ctrl-AddSub* circuit is based on the design presented in [34] and uses the ripple carry adder in [35]. We determined that this quantum *Ctrl-AddSub* circuit will have a T-count of $14\cdot n-14$ and a T-depth of 10. Thus, the quantum *Ctrl-AddSub* circuit used in our proposed dividers will have a T-count of order $\mathcal{O}(n)$ and a T-depth of order $\mathcal{O}(1)$.

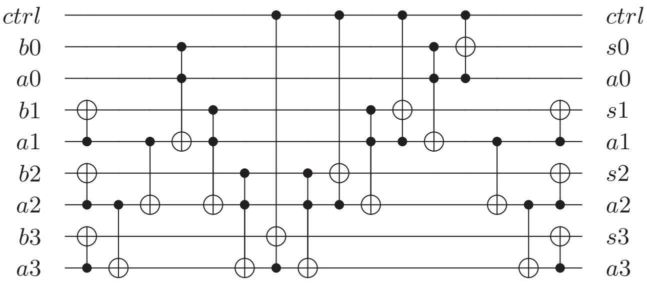


FIGURE 2. **Circuit design of the quantum conditional addition (*Ctrl-AddNOP*) circuit used in this work.**

D. DESIGN OF QUANTUM CONDITIONAL ADDITION (Ctrl-AddNOP) CIRCUIT

The quantum Ctrl-AddNOP circuit in this work is a modified version of the Ctrl-AddNOP circuit proposed in [36]. Operation of the quantum *Ctrl-AddNOP* circuit is conditioned on the value of *ctrl*. When *ctrl* is high, the circuit calculates $b+a$. When *ctrl* is low, the circuit performs no computation. An illustrative example is shown in Figure 2 for two 4 qubit operands $a_0...a_3$ and $b_0...b_3$. We are able to reduce the amount of qubits and quantum gates required because we do not need the carry out qubit in the proposed integer dividers.

We determined that our quantum *Ctrl-AddNOP* circuit will have a T-count of $21\cdot n-14$ and a T-depth of $2\cdot n$. Thus, the quantum *Ctrl-AddNOP* circuit used in our proposed dividers will have a T-count of order $\mathcal{O}(n)$ and a T-depth of order $\mathcal{O}(n)$.

III. DESIGN OF THE PROPOSED RESTORING QUANTUM INTEGER DIVISION CIRCUIT

We now present our proposed restoring quantum integer division circuit. The proposed design produces no garbage output and has lower T-count and qubit costs compared to the existing works. The quantum circuits used in our proposed quantum restoring division circuit are (i) the quantum subtractor and (ii) the novel quantum *Ctrl-AddNOP* circuit. Our proposed quantum restoring divider saves T gates by avoiding computation in the Quantum Fourier Transform (QFT) domain. More details on QFT can be found in [37], [38]. We also base our design on the T gate efficient quantum subtractor and the novel quantum *Ctrl-AddNOP* circuit presented in Section II. The modules used in our quantum circuit do not produce garbage outputs and restore inputs to their original values. Thus, we are able to save qubits and T gates by placing these quantum

Algorithm 1: Restoring Division Algorithm

```
Function Restoring($(a, b)$)
Requirements: $a$ and $b$ are positive and 2's complement.
    //Takes 2 n bit values $a$ and $b$ as input.
    //Returns the quotient as an $n$ bit number $Q$ and
    //the remainder from the division as an $n$ bit
    //number $R$.

1   $R = 0^n$; // Where $0^n$ are $n$ zeros.
2   $Q = a_{n-1}a_{n-2}\cdots a_1a_0$
3       // $a_{n-1}$ is the most significant bit of $a$.
4
5   For $i = 0$ to $n-2$
6       $Y = R_{n-2-i}R_{n-3-i}\cdots R_1R_0Q_{n-1}\cdots Q_{n-1-i}$
7       // Where $R_{n-2-i}$ is the most significant bit of $Y$.
8       $Y = Y - b$
9       If $(Y < 0)$
10          $Y = Y + b$
11      End
12      $R_{n-1-i} = \overline{R_{n-2-i}}$
13  End
14
15  $Q = Q - b$
16  If $(Q < 0)$
17      $Q = Q + b$;
18  End
19  $R_0 = \overline{Q_{n-1}}$
20
21  Return $Q, R$
```

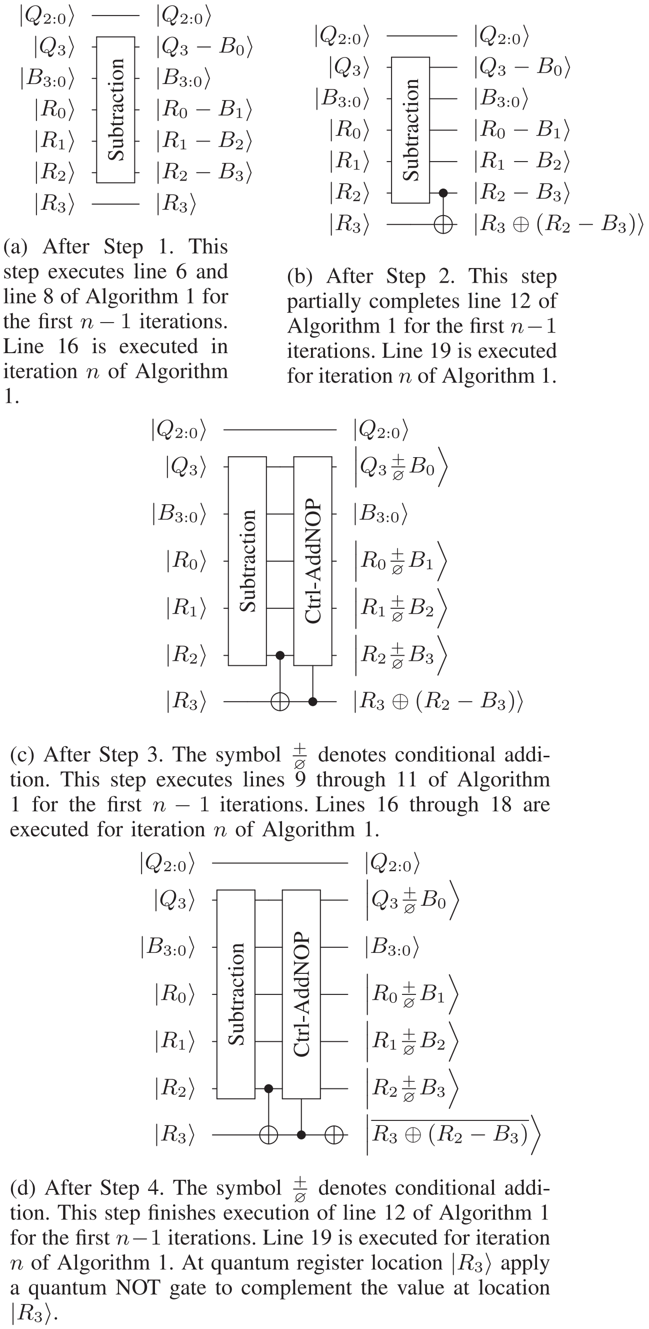
FIGURE 3. The restoring division algorithm.

circuits such that our proposed quantum restoring division circuit will produce no garbage outputs.

This proposed quantum integer division circuit calculates division by implementing the restoring division algorithm. The restoring division algorithm is illustrated by Algorithm 1 shown in Figure 3. The restoring division algorithm is a well established technique to perform division [39], [40].

Consider the division of two n bit 2's complement positive binary numbers a and b. Let $|B\rangle$ be a n bit quantum register that is initialized to the value b, let $|Q\rangle$ be a n bit quantum register that is initialized with the value a and let $|R\rangle$ be a n bit quantum register initialized to 0. At the end of computation, the quantum register $|B\rangle$ will be restored to the value b while the quantum register $|R\rangle$ will have the remainder of the division of a by b. At the end of computation, the quantum register $|Q\rangle$ will have the quotient of the division of a by b.

The proposed methodology is generic in nature and can design a quantum restoring integer division circuit of any size. An illustrative example of the proposed quantum restoring integer division circuit for the division of two 4 bit numbers $a_0 \ldots a_3$ and $b_0 \ldots b_3$ is shown in Figure 5. A quantum circuit is generated for each step of the design. The proposed methodology is repeated n times to produce the complete division circuit. The steps of the proposed design methodology are now shown along with an illustrative example of the generation of a single iteration i in Figure 4:


$|Q_{2:0}\rangle$
$|Q_3\rangle$
$|B_{3:0}\rangle$
$|R_0\rangle$
$|R_1\rangle$
$|R_2\rangle$
$|R_3\rangle$
Subtraction
Ctrl-AddNOP
$|Q_3 - B_0\rangle$
$|R_0 - B_1\rangle$
$|R_1 - B_2\rangle$
$|R_2 - B_3\rangle$
$|R_3 \oplus (R_2 - B_3)\rangle$
$|Q_3 \overset{+}{\underset{\varnothing}{}} B_0\rangle$
$|R_0 \overset{+}{\underset{\varnothing}{}} B_1\rangle$
$|R_1 \overset{+}{\underset{\varnothing}{}} B_2\rangle$
$|R_2 \overset{+}{\underset{\varnothing}{}} B_3\rangle$
$|\overline{R_3 \oplus (R_2 - B_3)}\rangle$

(a) After Step 1. This step executes line 6 and line 8 of Algorithm 1 for the first $n-1$ iterations. Line 16 is executed in iteration n of Algorithm 1.

(b) After Step 2. This step partially completes line 12 of Algorithm 1 for the first $n-1$ iterations. Line 19 is executed for iteration n of Algorithm 1.

(c) After Step 3. The symbol $\overset{+}{\underset{\varnothing}{}}$ denotes conditional addition. This step executes lines 9 through 11 of Algorithm 1 for the first $n-1$ iterations. Lines 16 through 18 are executed for iteration n of Algorithm 1.

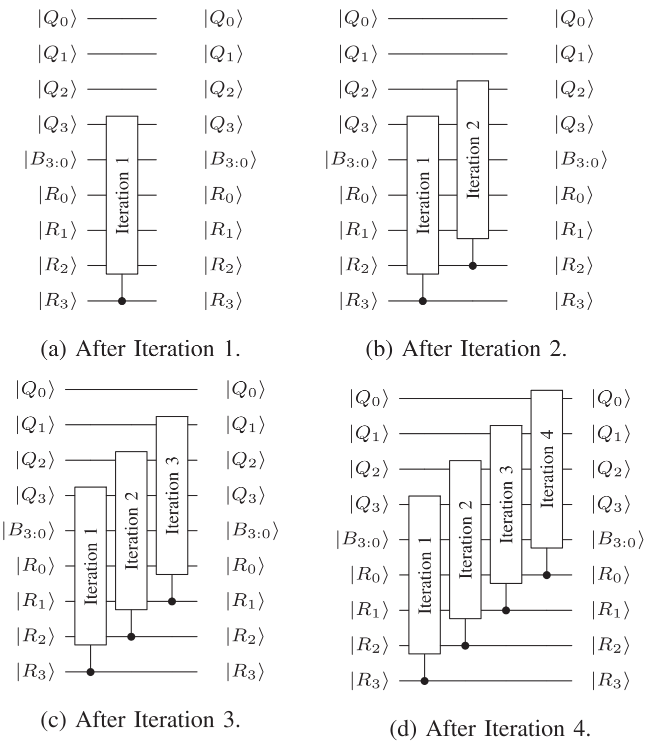
(d) After Step 4. The symbol $\overset{+}{\underset{\varnothing}{}}$ denotes conditional addition. This step finishes execution of line 12 of Algorithm 1 for the first $n-1$ iterations. Line 19 is executed for iteration n of Algorithm 1. At quantum register location $|R_3\rangle$ apply a quantum NOT gate to complement the value at location $|R_3\rangle$.

FIGURE 4. Circuit generation of iteration i (where $0 \le i \le n-1$) of the proposed quantum restoring division circuit for the case of $i = 0$: Steps 1-4. Register values after each step are shown. The symbol $\overset{+}{\underset{\varnothing}{}}$ denotes conditional addition.

The following steps of the proposed methodology are repeated n times. Starting with the first $n-1$ iterations.

For $i = 0 : 1 : n-2$:

- Step 1: This step executes line 6 and line 8 of Algorithm 1 in quantum hardware. Step 1 is shown for a 4 bit restoring divider in Figure 4(a). This step has the following two sub-steps:
    - Sub-step 1: Treat the locations $|R_{n-2-i}\rangle$ through $|R_0\rangle$ of quantum register $|R\rangle$ and locations $|Q_{n-1}\rangle$



(a) After Iteration 1.

(b) After Iteration 2.

(c) After Iteration 3.

(d) After Iteration 4.

FIGURE 5. Generation of the proposed quantum restoring integer divider circuit: Iterations 1-4. After Iteration 4, $|Q\rangle$ will have the quotient, $|R\rangle$ will contain the remainder and $|B\rangle$ will be unchanged.

through $|Q_{n-1-i}\rangle$ of quantum register $|Q\rangle$ as one n qubit combined quantum register $|Y\rangle$ such that the values at locations $|Q_{n-1}\rangle$ through $|Q_{n-1-i}\rangle$ will occupy locations $|Y_{i-1}\rangle$ through $|Y_0\rangle$ and locations $|R_{n-2-i}\rangle$ through $|R_0\rangle$ will occupy locations $|Y_{n-1}\rangle$ through $|Y_i\rangle$. Location $|R_{n-2-i}\rangle$ now contains the inverted remainder bit $\overline{r_{n-1-i}}$ of the division of a by b.

  - Sub-step 2: Apply quantum register $|B\rangle$ and quantum register $|Y\rangle$ to the quantum subtraction circuit so that, at the end of computation, $|B\rangle$ is unchanged and $|Y\rangle$ has the result of computation.

- Step 2: This step prepares register location $|R_{n-1-i}\rangle$ for use in subsequent steps. Step 2 is shown for a 4 bit restoring divider in Figure 4(b). At quantum register locations $|R_{n-1-i}\rangle$ and $|Y_{n-1}\rangle$, apply a CNOT gate such that quantum register location $|Y_{n-1}\rangle$ is unchanged and quantum register location $|R_{n-1-i}\rangle$ is transformed to $|R_{n-1-i} \oplus Y_{n-1}\rangle$. Location $|R_{n-1-i}\rangle$ now contains the inverted remainder bit $\overline{r_{n-1-i}}$.
- Step 3: Step 3 implements lines 9 through 11 of Algorithm 1 in quantum hardware. Step 3 is shown for a 4 bit restoring divider in Figure 4(c). This step has the following two sub-steps:
  - Sub-step 1: Apply the quantum registers $|B\rangle$ and $|Y\rangle$ to a quantum *Ctrl-AddNOP* circuit such that the quantum register $|B\rangle$ will be unchanged and quantum register $|Y\rangle$ will contain the result of computation.
  - Sub-step 2: Apply quantum register location $|R_{n-1-i}\rangle$ to the quantum *Ctrl-AddNOP* circuit such that the operation of the *Ctrl-AddNOP* circuit will be conditioned on the value at location $|R_{n-1-i}\rangle$. Quantum register location $|R_{n-1-i}\rangle$ is unchanged.
- Step 4: This step executes line 12 of Algorithm 1 by calculating $R_{n-1-i} = \overline{R_{n-2-i}}$. Step 4 is shown for a 4 bit restoring divider in Figure 4(d). At quantum register location $|R_{n-1-i}\rangle$ apply a quantum NOT gate to complement the value at location $|R_{n-1-i}\rangle$. Quantum register location $|R_{n-1-i}\rangle$ now has the remainder bit r_{n-1-i} of the division of a by b. The value at location $|R_{n-2-i}\rangle$ was copied to quantum register location $|R_{n-1-i}\rangle$ in Step 2.

The final iteration has the following steps:

- Step 1: This step executes line 15 of Algorithm 1 in quantum hardware. Step 1 is shown for a 4 bit restoring divider in Figure 4(a). Apply quantum register $|B\rangle$ and quantum register $|Q\rangle$ to the quantum subtraction circuit so that, at the end of computation, $|B\rangle$ is unchanged and $|Q\rangle$ has the result of computation. Location $|Q_{n-1}\rangle$ now contains the inverted remainder bit $\overline{r_0}$.
- Step 2: This step prepares register location $|R_0\rangle$ for use in subsequent steps and partially completes the execution of line 19 of Algorithm 1. Step 2 is shown for a 4 bit restoring divider in Figure 4(b). At quantum register locations $|R_0\rangle$ and $|Q_{n-1}\rangle$, apply a CNOT gate such that quantum register location $|Q_{n-1}\rangle$ is unchanged and quantum register location $|R_0\rangle$ shall be transformed to the value $|R_0 \oplus Q_{n-1}\rangle$. Location $|R_0\rangle$ now contains the inverted remainder bit $\overline{r_0}$.
- Step 3: Step 3 implements lines 16 through 18 of Algorithm 1 in quantum hardware. Step 3 is shown for a 4 bit restoring divider in Figure 4(c). Step 3 has the following two sub-steps:
  - Sub-step 1: Apply the quantum registers $|B\rangle$ and $|Q\rangle$ to a quantum *Ctrl-AddNOP* circuit such that the quantum register $|B\rangle$ will be unchanged and quantum register $|Q\rangle$ will contain the result of computation.
  - Sub-step 2: Apply quantum register location $|R_0\rangle$ to the quantum *Ctrl-AddNOP* circuit such that the operation of the *Ctrl-AddNOP* circuit will be conditioned on the value at location $|R_0\rangle$. Quantum register location $|R_0\rangle$ is unchanged.

After this step, quantum register $|Q\rangle$ will have the quotient of the division of a and b.

- Step 4: This step executes line 19 of Algorithm 1 by calculating $R_0 = \overline{Q_{n-1}}$. Step 4 is shown for a 4 bit restoring divider in Figure 4(d). At quantum register location $|R_0\rangle$ apply a quantum NOT gate to complement the value at location $|R_0\rangle$. Quantum register location $|R_0\rangle$ now has the remainder bit r_0 of the division of a by b. After this step, quantum register $|R\rangle$ now has the remainder of the division of a and b. The value at location $|Q_{n-1}\rangle$ was copied to quantum register location $|R_0\rangle$ in Step 2.

TABLE 1. Comparison of resource count between proposed and existing work in terms of Toffoli gates.

	Khosropour *et al.* [22]	Dibbo *et al.* [24]†	Proposed
Total Toffoli gates	0	$\approx n^3$	$5 \cdot n^2 - 4 \cdot n$
2-control AND*	0	$\approx n^2$	0
CCNOT**	0	$\approx n^3 - n^2$	$5 \cdot n^2 - 4 \cdot n$

† *The design by Dibbo et al. [24] is modified to remove garbage output.*
* *2-control AND is a Toffoli gate that calculates $A \cdot B \oplus 0$.*
** *CCNOT is a Toffoli gate that calculates $A \cdot B \oplus C$.*

TABLE 2. Comparison of resource count between proposed and existing work in terms of T gates and qubits.

	Khosropour *et al.* [22]	Dibbo *et al.* [24]†	Proposed
T-count	$\approx 400 \cdot n^2$	$\approx 9 \cdot n^3$	$35 \cdot n^2 - 28 \cdot n$
T-depth	$130 \cdot n$	NA	$23 \cdot n$
Qubits	$4 \cdot n$	$\approx \frac{1}{2} n^3 + 4 \cdot n$	$3 \cdot n$

† *The design by Dibbo et al. [24] is modified to remove garbage output. Table entries are marked NA where a closed-form expression is not available for the design by Dibbo et al.*

IV. COST ANALYSIS OF THE PROPOSED RESTORING DIVISION CIRCUIT

A. T-COUNT ANALYSIS

The T-count of the proposed quantum integer division circuit is illustrated shortly for each step of the proposed design methodology. The steps are iterated n times.

- The T-count for Step 1 is $14 \cdot n - 14$. We use a quantum subtraction circuit of T-count $14 \cdot n - 14$ in this step.
- Step 2 does not require T gates.
- The T-count for Step 3 is $21 \cdot n - 14$. We use a quantum *Ctrl-AddNOP* circuit of T-count $21 \cdot n - 14$ in this step.
- Step 4 does not require T gates.

We determine the T-count for a single iteration of the proposed design by summing the T-count for each step in the methodology as shown below:

$$14 \cdot n - 14 + 21 \cdot n - 14. \tag{1}$$

This expression can be simplified to the following:

$$35 \cdot n - 28. \tag{2}$$

The steps in the proposed methodology are iterated n times. Thus, the T-count for the proposed restoring division circuit is $n \cdot (35 \cdot n - 28)$ which simplifies to the expression shown below:

$$35 \cdot n^2 - 28 \cdot n. \tag{3}$$

B. T-DEPTH ANALYSIS

Our proposed design is based on T-depth efficient designs of quantum subtraction circuits and quantum *Ctrl-AddNOP* circuits. We determined that garbageless and T gate optimized quantum subtraction circuits in the literature such as the design in [34] have a T-depth that is constant and independent of the circuit size n. Thus, these subtraction circuits have T-depth of order $\mathcal{O}(1)$. We determined as well that *Ctrl-AddNOP* circuits in the literature such as the design in [36] scale as a function of circuit size n. Thus, these *Ctrl-AddNOP* circuits have a T-depth of order $\mathcal{O}(n)$.

To calculate T-depth, we must consider the T-depth seen by each quantum register $|B\rangle$, $|Q\rangle$ and $|R\rangle$ during each step. The quantum register that sees the most T gate layers will determine the T-depth of the quantum division circuit. Figure 5 shows that register $|B\rangle$ along with a location in $|R\rangle$ see the most functional block layers. By considering how each register is applied to the functional blocks it encounters, we could determine that quantum register $|B\rangle$ sees the most T gate layers. Therefore, we only consider the T-depth of quantum register $|B\rangle$ for the total T-depth calculation. The T-depth of the proposed quantum integer division circuit is illustrated shortly for each step of the proposed design methodology. The steps are iterated n times.

- Step 1 has a constant T-depth of 10. This T-depth is seen at locations $|B_1\rangle$ through $|B_{n-2}\rangle$ of quantum register $|B\rangle$.
- Step 2 does not require T gates.
- In Step 3, quantum register $|B\rangle$ sees a constant T-depth of 13 at locations $|B_1\rangle$ through $|B_{n-2}\rangle$.
- Step 4 does not require T gates.

We determine total T-depth for the proposed quantum restoring integer division circuit by summing the T-depth for each step of the methodology. The T-depth for one iteration is given as $10 + 13 = 23$. After n iterations the T-depth for the proposed quantum restoring integer division circuit will be $23 \cdot n$.

C. COST COMPARISON

Comparison of the proposed design with the existing quantum restoring division circuit by Khosropour *et al.* [22] are presented in Tables 1 and 2. We also compare against the recently proposed alternative design methodology in Dibbo *et al.* [24] as shown in Tables 1 and 2. We use Bennett's garbage removal scheme to remove the garbage outputs from the design by Dibbo *et al.* [41].

1) COST COMPARISON IN TERMS OF NUMBER OF TOFFOLI GATES

Table 1 shows that the design by Khosropour *et al.* is based on controlled phase gates and does not use Toffoli gates. Table 1 shows that our proposed design and the design by Dibbo *et al.* have Toffoli gate counts of order $\mathcal{O}(n^2)$ and of order $\mathcal{O}(n^3)$ respectively.

To perform the comparison we determined the number of Toffoli gates used in each design. To determine how many Toffoli gates operate as CCNOT gates and operate as 2-control AND gates, we examined the inputs and outputs of each Toffoli gate. The divider by Khosropour *et al.* is based on controlled phase gates. Therefore, the number of Toffoli gates in the design is 0.

2) COST COMPARISON IN TERMS OF T-COUNT

Table 2 shows that our proposed design and the design by Khosropour *et al.* have T-count costs of order $\mathcal{O}(n^2)$ while the T-count for the design by Dibbo *et al.* is of order $\mathcal{O}(n^3)$. We calculated that our proposed design methodology achieves improvement ratios ranging from 91.26 to 93.00 percent and 22.22 to 99.24 percent compared to the design by Khosropour *et al.* and the design by Dibbo *et al.* in terms of T-count.

To perform the comparison we implemented each design with Clifford+T gates. To realize quantum gates such as the Toffoli gate, we use the Clifford+T implementations presented in [19]. To implement the controlled phase gates in the design by Khosropour *et al.* with Clifford+T gates we must use the Clifford+T approximations presented in works such as [30]. To have a fair comparison, we select the approximations for the controlled phase gates with the poorest accuracy in our calculations because they are shown in [30] to have the lowest T-count. Thus, we compare the proposed work against the best case T-count of the design by Khosropour *et al.*

3) COST COMPARISON IN TERMS OF QUBITS

Table 2 shows that our proposed design and the design by Khosropour *et al.* have qubit costs of order $\mathcal{O}(n)$ while the qubit cost for the design by Dibbo *et al.* is of order $\mathcal{O}(n^3)$. We calculated that our proposed design methodology achieves an improvement ratio of 25.00 percent compared to the design by Khosropour *et al.* We determined that our proposed design achieves improvement ratios ranging from 75.00 to 99.99 percent compared to the design by Dibbo *et al.* We determined the qubit cost for each design by summing the qubits required for the quotient, remainder, garbage outputs, and primary inputs.

4) COST COMPARISON IN TERMS OF T-DEPTH

The T-depth cost of the proposed design and design by Khosropour *et al.* are $\mathcal{O}(n)$. A closed-form T-depth expression is not available for the design by Dibbo *et al.* We calculated that the design by Khosropour *et al.* has a T-depth that is 5.6 times higher than the T-depth of the proposed work. To perform the comparison we implemented each designs with Clifford+T gates. To have a fair comparison, we select the approximations for the controlled phase gates with the poorest accuracy in our calculations because they are shown in [30] to have the lowest T-count. Thus, we compare the proposed work against the best case T-count of the design by Khosropour *et al.*

V. DESIGN OF THE PROPOSED NON-RESTORING QUANTUM INTEGER DIVISION CIRCUIT

We now present our proposed non-restoring quantum integer division circuit. The proposed design produces no garbage output and has lower T-count and qubit costs compared to the existing work. The quantum circuits required to build our proposed quantum non-restoring division circuit are (i) the quantum subtractor, (ii) the quantum *Ctrl-AddSub* circuit and (iii) the novel quantum *Ctrl-AddNOP* circuit. Our proposed quantum non-restoring divider saves T gates by not doing computation in the QFT domain. We also base our design on the T gate efficient quantum subtractor, the quantum *Ctrl-AddSub* circuit and the novel quantum *Ctrl-AddNOP* circuit presented in Section II. The modules used in our proposed quantum circuit do not produce garbage outputs and restore inputs to their original values. Thus, we are able to save qubits and T gates by placing these quantum circuits such that our proposed quantum non-restoring division circuit will produce no garbage outputs. We are able to save additional qubits and T gates because the remainder will be at most $n-1$ bits wide when we divide two n bit numbers with our proposed divider.

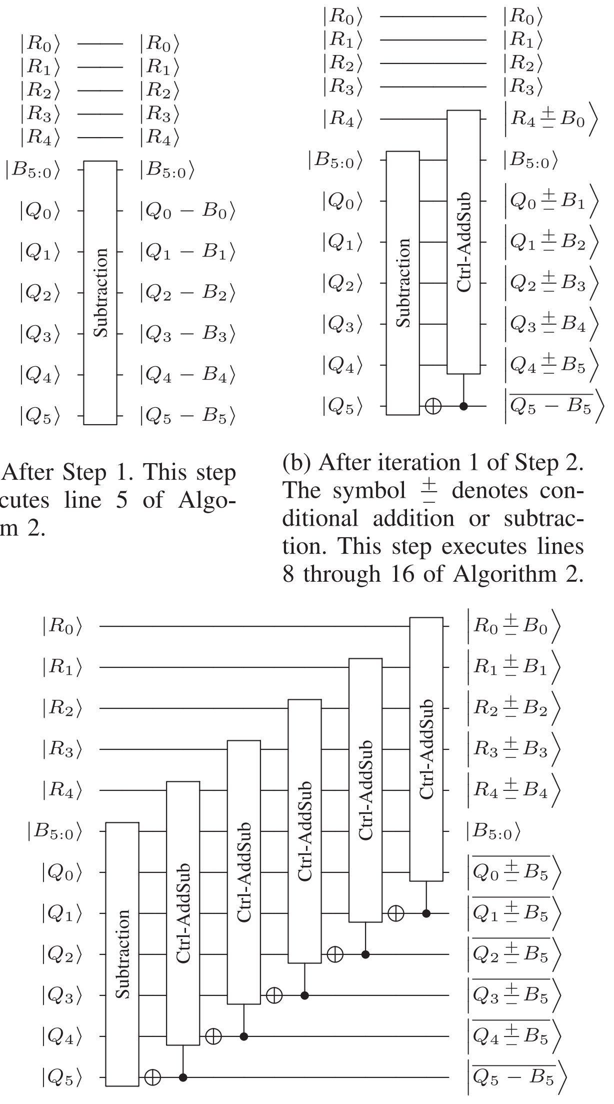
Algorithm 2: Non-Restoring Division Algorithm

```
Function Non-Restoring($a, b$)
Requirements: $a$ and $b$ are positive and 2's complement.
    //Takes 2 n bit values $a$ and $b$ as input.
    //Returns the quotient as an $n$ bit number $Q$ and
    //the remainder from the division as an $n-1$ bit
    //number $R$.

    $R = 0^{n-1}$; // Where $0^{n-1}$ are $n-1$ zeros.
    $Q = 0^{n-1}a_{n-1}$; // Where $0^{n-1}$ are $n-1$ zeros.
        // $Q$'s least significant bit has the value $a_{n-1}$
        // $a_{n-1}$ is the most significant bit of $a$.
    $Q = Q - b$

    For $i = 1$ to $n-1$
        $Q_{n-i} = \overline{Q_{n-i}}$
        $Y = Q_{n-1-i} \cdots Q_0 R_{n-2} \cdots R_{n-1-i}$
            // Where $Q_{n-1-i}$ is the most
            // significant bit of $Y$.
        If ($Q_{n-i} = 0$)
            $Y = Y + b$
        Else
            $Y = Y - b$
        End
    End

    If ($R < 0$)
        $R = R + b$
    End
    $Q_0 = \overline{Q_0}$
    Return $Q, R$
```

FIGURE 6. The non-restoring division algorithm.

This proposed quantum integer division circuit calculates division by implementing the non-restoring division algorithm. The non-restoring division algorithm is illustrated by Algorithm 2 shown in Figure 6. The non-restoring division algorithm is a well established technique to perform division [39], [40].

Consider the division of two n bit 2's complement positive binary numbers a and b. Let $|B\rangle$ be a n bit quantum register where that is initialized to the value b, let $|R\rangle$ be a $n-1$ quantum register where each register location is initialized with the value a_i for $0 \leq i \leq n-2$ and let $|Q\rangle$ be a n bit quantum register register location $|Q_0\rangle$ is initialized with the value a_{n-1} and the remaining $n-1$ locations in $|Q\rangle$ are initialized to 0. At the end of computation, the quantum register $|B\rangle$ will be restored to the value b while the quantum register $|R\rangle$ will have the



(a) After Step 1. This step executes line 5 of Algorithm 2.

(b) After iteration 1 of Step 2. The symbol ± denotes conditional addition or subtraction. This step executes lines 8 through 16 of Algorithm 2.

(c) After the final iteration of Step 2. The symbol ± denotes conditional addition or subtraction. This step executes lines 8 through 16 of Algorithm 2.

FIGURE 7. Circuit generation of the proposed quantum non-restoring division circuit: Steps 1-2. The register values after each step are shown. The symbol ± denotes conditional addition or subtraction.

remainder of the division of a by b. At the end of computation, the quantum register $|Q\rangle$ will have the quotient of the division of a by b.

The proposed methodology is generic in nature and can design a quantum non-restoring integer division circuit of any size. The steps of the proposed methodology are presented along with an illustrative example of the proposed quantum non-restoring integer division circuit for the division of two 6 bit numbers $a_0 \ldots a_5$ and $b_0 \ldots b_5$ shown in Figure 7 and Figure 8. The proposed methodology has three steps. A quantum circuit is generated for each step of the design. The steps of the proposed design methodology are as follows:

- Step 1: This step executes line 5 of Algorithm 2 in quantum hardware. Step 1 is shown for a 6 bit non-restoring divider in Figure 7(a). Apply the quantum registers $|Q\rangle$ and $|B\rangle$ to a quantum subtraction circuit such that, at the end of computation, the quantum register $|B\rangle$ is unchanged while quantum register $|Q\rangle$ now has the result of computation.
- Step 2: This step is repeated $n - 1$ times and has the following four sub-steps. This step executes lines 8 through 16 of Algorithm 2 in quantum hardware. Iteration 1 of Step 2 is shown for a 6 bit non-restoring divider in Figure 7(b). The final iteration of Step 2 is shown for a 6 bit non-restoring divider in Figure 7(c). We show the steps for iteration i where $1 \leq i \leq n - 1$.
  - Sub-step 1: This sub-step executes line 9 of Algorithm 2 in quantum hardware. Treat the locations $|R_{n-2}\rangle$ through $|R_{n-1-i}\rangle$ of quantum register $|R\rangle$ and locations $|Q_{n-1-i}\rangle$ through $|Q_0\rangle$ of register $|Q\rangle$ as a combined quantum register $|Y\rangle$. The values at locations $|R_{n-2}\rangle$ through $|R_{n-1-i}\rangle$ will occupy locations $|Y_{i-1}\rangle$ through $|Y_0\rangle$ and the values at locations $|Q_{n-}\ 1 - i\rangle$ to $|Q_0\rangle$ will occupy locations $|Y_{n-1}\rangle$ through $|Y_i\rangle$.
  - Sub-step 2: This sub-step executes line 8 of Algorithm 2 in quantum hardware and prepares location $|Q_{n-i}\rangle$ for use in subsequent sub-steps. At quantum register location $|Q_{n-i}\rangle$ apply a quantum NOT gate. Location $|Q_{n-i}\rangle$ now has the quotient bit q_{n-i} of the division of a by b.
  - Sub-step 3: Apply the quantum registers $|B\rangle$ and $|Y\rangle$ to a quantum *Ctrl-AddSub* circuit such that $|B\rangle$ is unchanged while $|Y\rangle$ will hold the result of computation.
  - Sub-step 4: Apply the quantum register location $|Q_{n-i}\rangle$ to the quantum *Ctrl-AddSub* circuit such that the operation of the circuit is conditioned on the value at location $|Q_{n-i}\rangle$. Location $|Q_{n-i}\rangle$ is unchanged. If $|Q_{n-i}\rangle = 0$, this step executes line 13 of Algorithm 2. If $|Q_{n-i}\rangle = 1$, this step executes line 15 of Algorithm 2.
- Step 3: This step executes lines 19 through 22 of Algorithm 2 in quantum hardware. Step 3 is shown for a 6 bit non-restoring divider in Figure 8. This step has the following three sub-steps:
  - Sub-step 1: Apply locations $|B_{n-2}\rangle$ through $|B_0\rangle$ of quantum register $|B\rangle$ and quantum register $|R\rangle$ to a quantum *Ctrl-AddNOP* circuit such that locations $|B_{n-2}\rangle$ through $|B_0\rangle$ are unchanged while quantum register $|R\rangle$ is transformed to the remainder.
  - Sub-step 2: Apply location $|Q_0\rangle$ of register $|Q\rangle$ to the quantum *Ctrl-AddNOP* circuit such that the operation of the circuit is conditioned on the value at register location $|Q_0\rangle$. Location $|Q_0\rangle$ is unchanged. If $|Q_0\rangle = 1$, Sub-step 1 and Sub-step 2 execute line 20. After this Sub-step, quantum register $|R\rangle$ will contain the remainder of the division of a by b.
  - Sub-step 3: At quantum register location $|Q_0\rangle$, apply a quantum NOT gate. Step 3 executes line 23

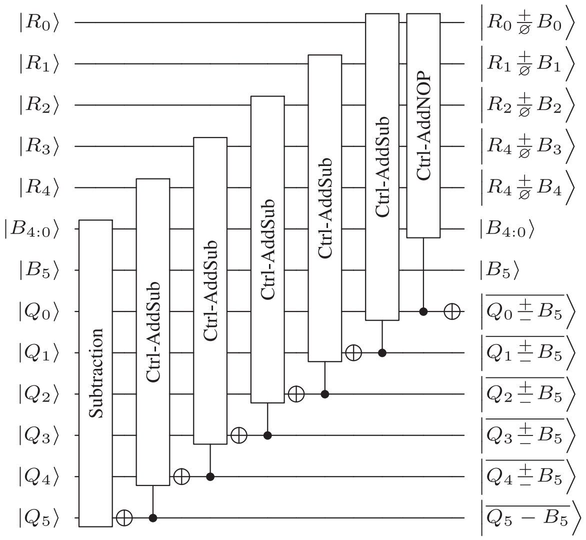

$|R_0\rangle$
$|R_1\rangle$
$|R_2\rangle$
$|R_3\rangle$
$|R_4\rangle$
$|B_{4:0}\rangle$
$|B_5\rangle$
$|Q_0\rangle$
$|Q_1\rangle$
$|Q_2\rangle$
$|Q_3\rangle$
$|Q_4\rangle$
$|Q_5\rangle$
Subtraction
Ctrl-AddSub
Ctrl-AddSub
Ctrl-AddSub
Ctrl-AddSub
Ctrl-AddSub
Ctrl-AddNOP
$\left|R_0 \frac{+}{\varnothing} B_0\right\rangle$
$\left|R_1 \frac{+}{\varnothing} B_1\right\rangle$
$\left|R_2 \frac{+}{\varnothing} B_2\right\rangle$
$\left|R_4 \frac{+}{\varnothing} B_3\right\rangle$
$\left|R_4 \frac{+}{\varnothing} B_4\right\rangle$
$|B_{4:0}\rangle$
$|B_5\rangle$
$\left|\overline{Q_0 \frac{+}{-} B_5}\right\rangle$
$\left|\overline{Q_1 \frac{+}{-} B_5}\right\rangle$
$\left|\overline{Q_2 \frac{+}{-} B_5}\right\rangle$
$\left|\overline{Q_3 \frac{+}{-} B_5}\right\rangle$
$\left|\overline{Q_4 \frac{+}{-} B_5}\right\rangle$
$\left|\overline{Q_5 - B_5}\right\rangle$

FIGURE 8. Circuit generation of proposed quantum non-restoring division circuit after Step 3. The symbol $\frac{+}{-}$ denotes conditional addition or subtraction and the symbol $\frac{+}{\varnothing}$ denotes conditional addition. This step executes lines 19 through 22 of Algorithm 2.

of Algorithm 2. Location $|Q_0\rangle$ now has the quotient bit q_0 of the division of a by b. After this step, quantum register $|Q\rangle$ will contain the quotient of the division of a by b.

VI. COST ANALYSIS OF THE PROPOSED NON-RESTORING DIVISION CIRCUIT

A. T-COUNT ANALYSIS

The T-count of the proposed quantum integer division circuit is illustrated shortly for each step of the proposed design methodology:

- The T-count for Step 1 is $14 \cdot n - 14$. We use a quantum subtraction circuit of T-count $14 \cdot n - 14$ in this step.
- Step 2 is repeated $n - 1$ times. The T-count for each iteration of Step 2 is $14 \cdot n - 14$. We use a quantum *Ctrl-AddSub* circuit of T-count $14 \cdot n - 14$ in this step.
- The T-count for Step 3 is $21 \cdot n - 21$. We use a quantum *Ctrl-AddNOP* circuit of size $n - 1$ in this step. We use a quantum *Ctrl-AddNOP* circuit of T-count $21 \cdot n - 14$ in this step.

We determine the total T-count by summing the T-count for each step in the design as shown below:

$$14 \cdot n - 14 + (14 \cdot n - 14) \cdot (n - 1) + 21 \cdot n - 21. \tag{4}$$

This expression can be simplified to the following:

$$14 \cdot n^2 + 7 \cdot n - 21. \tag{5}$$

B. T-DEPTH COST

Our proposed design is based on T-depth efficient designs of quantum subtraction circuits, quantum *Ctrl-AddSub* circuits and quantum *Ctrl-AddNOP* circuits. We determined that garbageless and T gate optimized quantum *Ctrl-AddSub* circuits in the literature such as the design in [34] have a T-depth that is constant and independent of the circuit size n. Thus, these *Ctrl-AddSub* circuits have T-depth of order $\mathcal{O}(1)$. As explained in Section IV-B of Section IV, we use a quantum subtraction circuit with T-depth of order $\mathcal{O}(1)$ and a quantum *Ctrl-AddNOP* with a T-depth of order $\mathcal{O}(n)$.

To calculate T-depth, we must consider the T-depth seen by each quantum register $|B\rangle$, $|Q\rangle$ and $|R\rangle$ during each step. The quantum register that sees the most T gate layers will determine the T-depth of the quantum division circuit. Figure 8 shows that register $|B\rangle$ along with a location in $|Q\rangle$ see the most functional block layers. By considering how each register is applied to the functional blocks it encounters, we could determine that quantum register $|B\rangle$ sees the most T gate layers. The T-depth of the proposed quantum integer division circuit is illustrated shortly for each step of the proposed design methodology.

- Step 1 has a constant T-depth of 10. This T-depth is seen by locations $|B_1\rangle$ through $|B_{n-2}\rangle$ of quantum register $|B\rangle$.
- Step 2 is repeated $n - 1$ times. Each iteration of Step 2 has a constant T-depth of 10. This T-depth is seen by locations $|B_1\rangle$ through $|B_{n-2}\rangle$ of quantum register $|B\rangle$.
- In Step 3, quantum register $|B\rangle$ sees a constant T-depth of 13 at locations $|B_1\rangle$ through $|B_{n-2}\rangle$.

We determine total T-depth for the proposed quantum non-restoring integer division circuit by summing the T-depth for each step of the methodology as shown below:

$$10 + 10 \cdot (n - 1) + 13. \tag{6}$$

This expression can be simplified to the following:

$$10 \cdot n + 13. \tag{7}$$

C. COST COMPARISON

Comparison of the proposed design with the existing quantum non-restoring division circuits by Jamal *et al.* [23] are presented in Tables 3 and 4. We also compare against the recently proposed alternative design methodology in Dibbo *et al.* [24] as shown in Tables 3 and 4. We use Bennett's garbage removal scheme to remove the garbage outputs from the designs in Jamal *et al.* and Dibbo *et al.* [41].

1) COST COMPARISON IN TERMS OF NUMBER OF TOFFOLI GATES

Table 3 shows that our proposed design and the designs by Jamal *et al.* each have a Toffoli gate count of order $\mathcal{O}(n^2)$. Table 3 also shows that the design by Dibbo *et al.* has a Toffoli gate count of order $\mathcal{O}(n^3)$. To perform the comparison we determined the number of Toffoli gates used in each design. To determine the number of Toffoli gates operating as CCNOT gates and 2-control AND gates we examined the inputs and outputs to each Toffoli gate for each design.

2) COST COMPARISON IN TERMS OF T-COUNT

Table 4 illustrates that the T-count cost of the proposed design and the designs by Jamal *et al.* are $\mathcal{O}(n^2)$. The design by Dibbo

TABLE 3. Comparison of resource count between proposed and existing work in terms of toffoli gates.

	Jamal *et al.* [23]†	Jamal *et al.* [23]†	Dibbo *et al.* [24]†	Proposed
Total Toffoli gates	$4 \cdot n^2$	$6 \cdot n^2 + 4 \cdot n$	$\approx n^3$	$2 \cdot n^2 + n + 1$
2-control AND*	$2 \cdot n^2$	$4 \cdot n^2 + 4 \cdot n$	$\approx n^2$	0
CCNOT**	$2 \cdot n^2$	$2 \cdot n^2$	$\approx n^3 - n^2$	$2 \cdot n^2 + n + 1$

† *The designs by Jamal et al. [23] and Dibbo et al. are modified to remove garbage output.*
* *2-control AND is a Toffoli gate that calculates $A \cdot B \oplus 0$.*
** *CCNOT is a Toffoli gate that calculates $A \cdot B \oplus C$.*

TABLE 4. Comparison of resource count between proposed and existing work in terms of T gates.

	Jamal *et al.* [23]†	Jamal *et al.* [23]†	Dibbo *et al.* [24]†	Proposed
T count	$28 \cdot n^2$	$42 \cdot n^2 + 28 \cdot n$	$\approx 9 \cdot n^3$	$14 \cdot n^2 + 7 \cdot n - 21$
T-depth	NA	NA	NA	$10 \cdot n + 13$
Qubits	$2 \cdot n^2 + 5 \cdot n - 1$	$3 \cdot n^2 + 14 \cdot n$	$\approx \frac{1}{2}n^3 + 4 \cdot n$	$3 \cdot n - 1$

† *The designs by Jamal et al. [23] and Dibbo et al. are modified to remove garbage output.*
Table entries are marked NA where a closed-form expression is not available for the designs by Jamal et al. and Dibbo et al.

et al. has a T-count cost of order $\mathcal{O}(n^3)$. We calculated that our proposed design methodology achieves improvement ratios ranging from 48.05 to 49.95 percent, 66.68 to 70.54 percent and 59.90 to 99.70 percent compared to the designs by Jamal *et al.* and the design by Dibbo *et al.* in terms of T-count. To perform the comparison we implemented each design with Clifford+T gates. To realize quantum gates such as the Toffoli gate, we use the Clifford+T implementations presented in [19].

3) COST COMPARISON IN TERMS OF QUBITS

Table 4 shows that our proposed design has a qubit cost of order $\mathcal{O}(n)$ while the qubit cost for the designs by Jamal *et al.* are of order $\mathcal{O}(n^2)$. Table 4 also illustrates that the design by Dibbo *et al.* has a qubit cost of order $\mathcal{O}(n^3)$. We determined that our proposed design methodology achieves improvement ratios ranging from 78.43 to 99.71 percent, 89.42 to 99.81 percent and 77.08 to 99.99 percent compared to the designs by Jamal *et al.* and the design by Dibbo *et al.* We determined the qubit cost for each design by summing the qubits required for the quotient, remainder, garbage outputs, and primary inputs.

4) COST COMPARISON IN TERMS OF T-DEPTH

The T-depth cost of the proposed design is $\mathcal{O}(n)$. A closed-form expression is not available for the designs by Jamal *et al.* and the design by Dibbo *et al.* for the T-depth. To perform the comparison we implemented each design with Clifford+T gates.

VII. APPLICATION OF THE PROPOSED QUANTUM INTEGER DIVISION CIRCUITS IN IMAGE PROCESSING

Quantum algorithms have been proposed for image orientation, image pattern recognition and image template matching problems [8], [42], [43]. Quantum circuit implementations are required to implement these promising image processing quantum algorithms. Thus, researchers have proposed quantum image representations such as the Novel Enhanced Quantum Representation (NEQR) and proposed quantum circuits for image operations such as bilinear interpolation [13], [44].

In this Section, we consider the impact of our proposed quantum integer division circuits on the total resource costs of the quantum bilinear interpolation circuits recently presented in [13]. Circuits for the scale up and scale down operation are presented. Scaling down an image by an integer value n results in reducing the original y and x positions of each pixel by 2^n. Scaling up an image by an integer value n results in increasing the original y and x positions of each pixel by 2^n. Illustrative examples of the quantum bilinear interpolation circuits for the scale up and scale down operations can be found in [13]. The quantum bilinear interpolation circuits presented in [13] are based on (i) a quantum subtractor, (ii) a quantum addition circuit, (iii) a quantum multiplication circuit, and (iv) a quantum division circuit. We evaluated the reduction in the resources required to implement quantum bilinear interpolation circuits with the use of the proposed quantum integer division circuits. In Table 5, a comparison of the quantum bilinear interpolation circuits implemented with our proposed quantum integer division circuits and the original implementation in [13] is presented.

Table 5 shows that each implementation of the quantum bilinear interpolation circuits by Zhou *et al.* [13] have T gate counts of order $\mathcal{O}(n^2)$. With our proposed restoring division circuit, the quantum bilinear interpolation circuits in Zhou *et al.* achieves up to ≈ 83.91 percent reduction in T gates compared to the original implementation. When our proposed non-restoring division circuit is used to implement the quantum bilinear interpolation circuits shown in Zhou *et al.*, the circuits achieve up to a ≈ 88.74 percent reduction in T gates compared to the original implementation. To perform the comparison, we implemented each design with Clifford +T gates. To realize quantum gates such as the Toffoli gate, we use the Clifford+T implementations presented in [19]. To implement the controlled phase gates in the design by Khosropour *et al.* with Clifford+T gates we must use the Clifford+T

TABLE 5. Comparison of quantum bilinear interpolation circuit implementations in terms of T gates *.

Design	T Gate Count
With Divider used in Zhou *et al.* [13]	$\approx 870 \cdot n^2 + 203 \cdot n - 168 + 112 \sum_{i=1}^{log_2(n)} \frac{n}{2^i} \cdot (n + i - 2^{i-1} - 1)$
With Our Restoring Divider	$\approx 140 \cdot n^2 + 147 \cdot n - 168 + 112 \sum_{i=1}^{log_2(n)} \frac{n}{2^i} \cdot (n + i - 2^{i-1} - 1)$
With Our Non-Restoring Divider	$\approx 98 \cdot n^2 + 217 \cdot n - 144 + 112 \sum_{i=1}^{log_2(n)} \frac{n}{2^i} \cdot (n + i - 2^{i-1} - 1)$

* *The quantum bilinear interpolation circuits for the scale up and scale down operations in [13] have the same resource costs.*

approximations presented in works such as [30]. To have a fair comparison, we select the approximations for the controlled phase gates with the poorest accuracy in our calculations because they are shown in [30] to have the lowest T-count. Thus, we compare the proposed work against the best case T-count of the quantum bilinear interpolation circuits in [13].

VIII. CONCLUSION

In this work, we have proposed two designs for quantum circuit integer division based on Clifford+T gates. The first quantum integer division circuit proposed is based on the restoring division algorithm and the second is based on the non-restoring division algorithm. We also show the design of components used in our proposed quantum integer division circuits such as the quantum subtraction circuit, quantum *Ctrl-AddSub* circuit and quantum *Ctrl-AddNOP* circuit. The proposed quantum restoring division circuit is shown to be superior to existing quantum restoring division designs in terms of T-depth, T-count and qubits. Likewise, the proposed quantum non-restoring division circuit is shown to be superior to existing quantum non-restoring division designs in terms of T-count and qubits. The proposed quantum restoring division circuit and proposed non-restoring division circuit are verified for functional correctness by mathematical proof and Verilog simulation. In conclusion, we presented the quantum circuit realizations of two common division algorithms (restoring and non-restoring division). We compared the proposed quantum restoring division circuit and proposed non-restoring division circuit. Despite the fact that the restoring division circuit produces the quotient and remainder simultaneously at the end of computation, the proposed non-restoring division circuit has the advantage in terms of qubit cost, T-depth and T-count. Thus, by thorough analysis of the quantum circuit designs, we conclude that our proposed non-restoring division circuit will be most useful in applications where T-count, T-depth and total qubits are of primary concern. Finally, we evaluated the reduction in the resources required to implement quantum bilinear interpolation circuits with the use of the proposed quantum integer division circuits. As a future work, it will be interesting to explore if the design techniques such as multivalued logic ([45]) or encoding methods ([46]) could result in additional resource savings in our proposed division circuits.

ACKNOWLEDGMENTS

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

Travis Humble acknowledges support from the Department of Energy, Office of Science Early Career Research Program.

REFERENCES

[1] P. Selinger *et al.*, *The Quipper System*, 2016. [Online]. Available: http://www.mathstat.dal.ca/ selinger/quipper/doc/

[2] S. Hallgren, “Polynomial-time quantum algorithms for pell’s equation and the principal ideal problem,” *J. ACM*, vol. 54, no. 1, pp. 4:1–4:19, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1206035.1206039

[3] W. van Dam and S. Hallgren, “Efficient quantum algorithms for shifted quadratic character problems,” *eprint arXiv:quant-ph/0011067*, Nov. 2000. [Online]. Available: https://arxiv.org/abs/quant-ph/0011067

[4] W. van Dam, S. Hallgren, and L. Ip, “Quantum algorithms for some hidden shift problems,” *SIAM J. Comput.*, vol. 36, no. 3, pp. 763–778, 2006. [Online]. Available: https://doi.org/10.1137/S009753970343141X

[5] S. Guodong, S. Shenghui, and X. Maozhi, “Quantum algorithm for polynomial root finding problem,” in *Proc. 10th Int. Conf. Comput. Intell. Secur.*, Nov. 2014, pp. 469–473.

[6] S. Hallgren, “Polynomial-time quantum algorithms for pell’s equation and the principal ideal problem,” *J. ACM*, vol. 54, no. 1, pp. 4:1–4:19, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1206035.1206039

[7] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.*, vol. 103, Oct. 2009, Art. no. 150502. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502

[8] G. Beach, C. Lomont, and C. Cohen, “Quantum image processing (quip),” in *Proc. 32nd Appl. Imagery Pattern Recognit. Workshop*, 2003, pp. 39–44.

[9] W. van Dam and I. E. Shparlinski, “Classical and quantum algorithms for exponential congruences,” in *Theory of Quantum Computation, Communication, and Cryptography*, Y. Kawano and M. Mosca, Eds. Berlin, Germany: Springer, 2008, pp. 1–10.

[10] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172

[11] M. Soeken, M. Roetteler, N. Wiebe, and G. D. Micheli, “Design automation and design space exploration for quantum computers,” in *Proc. Des. Autom. Test Eur. Conf. Exhibition*, Mar. 2017, pp. 470–475.

[12] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, “Quantum Algorithms and Circuits for Scientific Computing,” *Quantum Inf. Comput.*, vol. 16, pp. 197–236, 2016.

[13] R.-G. Zhou, W. Hu, P. Fan, and H. Ian, “Quantum realization of the bilinear interpolation method for NEQR,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–37, 2017.

[14] D. Wecker *et al.*, *Language-Integrated Quantum Operations: LIQUi*$|\rangle$, 2016. [Online]. Available: https://www.microsoft.com/en-us/research/project/language-integrated-quantum-operations-liqui/

[15] E. Fredkin and T. Toffoli, “Conservative logic,” *Int. J. Theoretical Phys.*, vol. 21, no. 3, pp. 219–253, 1982. [Online]. Available: http://dx.doi.org/10.1007/BF01857727

[16] X. Zhou, D. W. Leung, and I. L. Chuang, “Methodology for quantum logic gate construction,” *Phys. Rev. A*, vol. 62, Oct. 2000, Art. no. 052316. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.62.052316

[17] M. Amy, D. Maslov, and M. Mosca, “Polynomial-time t-depth optimization of clifford+t circuits via matroid partitioning,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 33, no. 10, pp. 1476–1489, Oct. 2014.

[18] A. Paler, I. Polian, K. Nemoto, and S. J. Devitt, “Fault-tolerant, high-level quantum circuits: form, compilation and description,” *Quantum Sci. Technol.*, vol. 2, no. 2, 2017, Art. no. 025003. [Online]. Available: http://stacks.iop.org/2058-9565/2/i=2/a=025003

[19] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, “A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, Jun. 2013.

[20] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, “An algorithm for the t-count,” *Quantum Inf. Comput.*, vol. 14, no. 15–16, pp. 1261–1276, 2014. [Online]. Available: http://www.rintonpress.com/xxqic14/qic-14-1516/1261-1276.pdf

[21] P. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, “A new universal and fault-tolerant quantum basis,” *Inf. Process. Lett.*, vol. 75, no. 3, pp. 101–107, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020019000000843

[22] A. Khosropour, H. Aghababa, and B. Forouzandeh, “Quantum division circuit based on restoring division algorithm,” in *Proc. 8th Int. Conf. Inf. Technol.: New Generations*, 2011, pp. 1037–1040.

[23] L. Jamal and H. M. H. Babu, “Efficient approaches to design a reversible floating point divider,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 3004–3007.

[24] S. V. Dibbo, H. M. H. Babu, and L. Jamal, “An efficient design technique of a quantum divider circuit,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2016, pp. 2102–2105.

[25] I. Gassoumi, L. Touil, and B. Ouni, “Division circuit using reversible logic gates,” in *Proc. Int. Conf. Adv. Syst. Elect. Technol.*, Mar. 2018, pp. 60–65.

[26] N. M. Nayeem, A. Hossain, M. Haque, L. Jamal, and H. M. H. Babu, “Novel reversible division hardware,” in *Proc. 52nd IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2009, pp. 1134–1138.

[27] F. Dastan and M. Haghparast, “A novel nanometric fault tolerant reversible divider,” *Int. J. Phys. Sci.*, vol. 6, no. 24, pp. 5671–5681, Oct. 2011.

[28] H. M. H. Babu and M. S. Mia, “Design of a compact reversible fault tolerant division circuit,” *Microelectronics J.*, vol. 51, no. Supplement C, pp. 15–29, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026269216000161

[29] A. Bolhassani and M. Haghparast, “Optimised reversible divider circuit,” *Int. J. Innov. Comput. Appl.*, vol. 7, no. 1, pp. 13–33, Mar. 2016. [Online]. Available: http://dx.doi.org/10.1504/IJICA.2016.075465

[30] V. Kliuchnikov, D. Maslov, and M. Mosca, “Fast and efficient exact synthesis of single-qubit unitaries generated by clifford and t gates,” *Quantum Inf. Comput.*, vol. 13, no. 7–8, pp. 607–630, Jul. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2535649.2535653

[31] H. Thapliyal, T. S. S. Varun, and E. Muñoz-Coreas, “Quantum circuit design of integer division optimizing ancillary qubits and t-count,” in *Proc. 16th Asian Quantum Inf. Sci. Conf.*, Aug. 2016, pp. 197–199.

[32] H. Thapliyal, T. S. S. Varun, E. Muñoz-Coreas, K. A. Britt, and T. S. Humble, “Quantum circuit designs of integer division optimizing t-count and t-depth,” in *Proc. IEEE Int. Symp. Nanoelectronic Inf. Syst.*, Dec. 2017, pp. 123–128.

[33] IBM, *Quantum Computing - IBM Q*, 2017. [Online]. Available: https://www.research.ibm.com/ibm-q/

[34] H. Thapliyal, “Mapping of subtractor and adder-subtractor circuits on reversible quantum gates,” in *Transactions on Computational Science XXVII*. New York, NY, USA: Springer, 2016, pp. 10–34.

[35] H. Thapliyal and N. Ranganathan, “Design of efficient reversible logic-based binary and bcd adder circuits,” *ACM J. Emerging Technol. Comput. Syst.*, vol. 9, no. 3, 2013, Art. no. 17.

[36] E. Muñoz-Coreas and H. Thapliyal, “Quantum circuit design of a T-count optimized integer multiplier,” *IEEE Trans. Comput.*, vol. 68, no. 5, pp. 729–739, May. 2019, [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8543237&isnumber=8683959.

[37] S. S. Zhou, T. Loke, J. A. Izaac, and J. B. Wang, “Quantum fourier transform in computational basis,” *Quantum Inf. Process.*, vol. 16, no. 3, Feb. 2017, Art. no. 82. [Online]. Available: https://doi.org/10.1007/s11128-017-1515-0

[38] B. Luong, *The Quantum Fourier Transform*. Boston, MA, USA: Birkhäuser, 2009, pp. 131–139. [Online]. Available: https://doi.org/10.1007/978-0-8176-4916-6_8

[39] M. D. Ercegovac, *Division and Square Root : Digit-Recurrence Algorithms and Implementations*. Boston, MA, USA: Kluwer, 1994.

[40] N. R. Scott (Norman Ross), *Computer Number Systems and Arithmetic*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[41] C. H. Bennett, “Logical reversibility of computation,” *IBM J. Res. Develop.*, vol. 17, no. 6, pp. 525–532, Nov. 1973. [Online]. Available: http://dx.doi.org/10.1147/rd.176.0525

[42] S. Caraiman and V. Manta, “ Image processing using quantum computing,” in *Proc. 16th Int. Conf. Syst. Theory Control Comput*, Oct. 2012, pp. 1–6.

[43] R. Schützhold, “Pattern recognition on a quantum computer,” *Phys. Rev. A*, vol. 67, Jun. 2003, Art. no. 062311. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.67.062311

[44] Y. Zhang, K. Lu, Y. Gao, and M. Wang, “Neqr: A novel enhanced quantum representation of digital images,” *Quantum Inf. Process.*, vol. 12, no. 8, pp. 2833–2860, Aug. 2013. [Online]. Available: https://doi.org/10.1007/s11128-013-0567-z

[45] M. Khan, H. Thapliyal, and E. Muñoz Coreas, “ Automatic synthesis of quaternary quantum circuits,” *J. Supercomputing*, vol. 73, no. 5, pp. 1733–1759, 2017. [Online]. Available: http://search.proquest.com/docview/1894985387/?pq-origsite=primo

[46] A. Zulehner and R. Wille, “Exploiting coding techniques for logic synthesis of reversible circuits,” in *Proc. Asia South Pacific Des. Autom. Conf.*, 2018, pp. 670–675.

HIMANSHU THAPLIYAL received the PhD degree in computer science and engineering from the University of South Florida, Tampa, in 2011. He has published more than 100 journal/conference articles and received Best Paper awards at 2012 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) and 2017 Cyber and Information Security Research Conference (CISR). He is the recipient of the NSF CAREER award. His research interests include circuit design of quantum computing and emerging technologies, and hardware assisted cybersecurity. He is an assistant professor and Endowed Robley D. Evans faculty fellow with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington.

EDGARD MUÑOZ-COREAS received the BS degree in electrical engineering from the University of Michigan, Ann Arbor, in 2010 and the MS degree in electrical engineering from the University of Kentucky, Lexington, in 2015. Currently, he is working towards the PhD degree in electrical engineering at the University of Kentucky. His research interests include circuit design of quantum computing and emerging technologies.

T. S. S. VARUN received the BTech degrees in electrical and communication engineering from JNTU Anantpur, India in 2015, and the MS degree in electrical engineering from the University of Kentucky, Lexington, Kentucky, in 2017. His research interests include circuit design of quantum computing and emerging technologies.

TRAVIS S. HUMBLE received the doctorate degree in theoretical chemistry from the University of Oregon, in 2005. He then joined Oak Ridge National Laboratory as an intelligence community postdoctoral research fellow before becoming a member of the research staff. He is the director of the Quantum Computing Institute, Oak Ridge National Laboratory. He oversees research and development of quantum computing technologies and their use in solving problems for scientific discovery and energy security. He was recently awarded the prestigious Department of Energy Early Career Award to research how quantum computing can be used to support high-performance computing applications.