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

Himanshu Thapliyal*, T. S. S. Varun*, Edgard Muñoz-Coreas*, Keith A. Britt† and Travis S. Humble†
*Department of Electrical and Computer Engineering
University of Kentucky, Lexington, KY
Email: hthapliyal@uky.edu
†Quantum Computing Institute
Oak Ridge National Laboratory, TN

***Abstract*—Quantum circuits for basic mathematical functions such as division are required to implement scientific computing algorithms on quantum computers. In this work, we propose two designs for quantum integer division. The designs are based on quantum Clifford+T gates and are optimized for T-count and T-depth. Quantum circuits that are based on Clifford+T gates can be made fault tolerant in nature but the T gate is very costly to implement. As a result, reducing T-count and T-depth have become important optimization goals. Existing quantum hardware is limited in terms of number of available qubits. Thus, ancillary qubits are a circuit overhead that needs to be kept to a minimum. We propose two quantum integer division circuits. The first quantum integer division circuit is based on the non-restoring division algorithm. The proposed non-restoring division circuit is optimized for total quantum hardware (T-count and T-depth) cost but requires $2 * n + 1$ ancillary qubits. We also propose a quantum integer division circuit based on the restoring division algorithm. The proposed restoring division circuit is optimized for total qubits. The design requires only n ancillary qubits but will need more quantum hardware than the non-restoring division circuit. Both proposed quantum circuits are based on (i) a new quantum conditional addition circuit, (ii) a new quantum adder-subtractor and (iii) a new quantum subtraction circuit. Further, both designs are compared and shown to be superior to existing work in terms of T-count and T-depth. The proposed quantum non-restoring integer division circuit has a 96% improvement in terms of T-count and a 93% improvement in terms of T-depth compared to existing work. The proposed quantum restoring integer division circuit has a 91% improvement in terms of T-count and a 86% improvement in terms of T-count compared to the existing work.**

I. Introduction and Background

Quantum circuits of arithmetic operations are needed to design quantum hardware for implementing quantum algorithms such as Shor's factoring algorithm, the discrete log problem, class number algorithm and triangle finding algorithm [1] [2]. Dividers are one of the major computational units in quantum arithmetic and have applications in circuit designs of quantum algorithms [3] [1].

Quantum circuits that are based on Clifford+T gates can be made fault tolerant in nature permitting reliable and scalable quantum computation [4] [5]. The Clifford+T gate family is illustrated in [6]. The T gate is very costly to implement compared to the Clifford gates making reducing T-count and T-depth important optimization goals [5] [7]. Existing quantum hardware is limited in terms of number of available qubits [8]. Thus, ancillary qubits are a circuit overhead that needs to be kept to a minimum.

In the existing literature, there are a handful of integer divider designs based on reversible gates targeting mostly reversible computing [9] [10] [11]. Among these designs we found only [12] to be suitable for quantum computing. The quantum integer division circuit in [12] implements the restoring division algorithm and uses the quantum Fourier transform to perform the division operation. However, the design in [12] is not optimized for T-depth and T-count. The quantum division circuit in [12] uses controlled phase shift gates. It is known that the controlled phase gates required by the design in [12] can only be approximated by Clifford+T gates [13]. The Clifford+T based approximations of the controlled phase gates have a high T gate cost [13]. Further, the T gate cost increases as the accuracy of the controlled phase gate approximation is improved [13]. Thus, implementing all the controlled phase gates required by the design in [12] with a high degree of accuracy will result in a design with high T-count and T-depth [13].

This paper presents two designs for quantum circuit integer division based on Clifford+T gates. The first quantum circuit is based on the non-restoring division algorithm and the second quantum circuit is based on the restoring division algorithm. Both proposed quantum integer division circuits are based on (i) a new quantum conditional ADD operation circuit, (ii) a new quantum adder-subtractor and (iii) a new quantum subtraction circuit. The proposed non-restoring division circuit is optimized for total quantum hardware (T-count and T-depth) cost. The trade off for reducing the quantum hardware of the design is the need to use more ancillary qubits. The non-restoring division circuit requires $2 * n + 1$ ancillary qubits. The proposed quantum restoring division is designed with the aim to minimize total qubits. We reduce the number of ancillary qubits to n but must use more quantum hardware than the proposed quantum non-restoring division circuit. Both the proposed restoring quantum integer division circuit and proposed non-restoring quantum integer division circuit are compared and shown to be superior to existing work in terms of T-depth and T-count.

978-1-5386-1356-6/17 $31.00 © 2017 IEEE
DOI 10.1109/iNIS.2017.34

IEEE computer society

This paper is organized as follows. Section II presents the design of the (i) new quantum conditional addition circuit, (ii) a new quantum adder-subtractor and (iii) a new quantum subtraction circuit used in the proposed quantum division circuits. In section III the design of the proposed quantum non-restoring integer division circuit is discussed. The design of the proposed quantum restoring integer division circuit is presented in section IV.

II. Design of Quantum Circuits Used In Proposed Integer Division Circuits

The quantum circuits that are required for developing the proposed non-restoring and restoring integer division circuits are: (i) controlled adder-subtractor, (ii) quantum subtractor and (iii) conditional ADD operation circuit. The quantum circuit designs of the quantum adder-subtractor, quantum subtractor and the conditional ADD operation circuit are discussed in the following sections.

A. Design of Quantum Subtractor

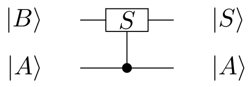

Fig. 1. Graphic symbol of quantum subtractor. S represents the quantum subtraction operation

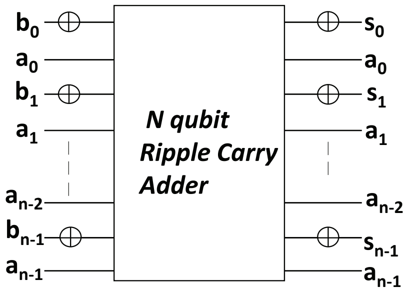

Fig. 2. Circuit design of N qubit quantum subtractor based on N qubit quantum ripple carry adder

Fig.1 shows the symbol of the quantum subtractor circuit. The subtractor circuit takes two n qubit inputs $|A\rangle$ and $|B\rangle$. The input a is regenerated at the output. The n-qubit output $|S\rangle$ has the result of the subtraction of b and a. Fig.2 shows the circuit design of N qubit subtractor based on N qubit quantum ripple carry adder. As shown in Fig.2, a quantum ripple carry adder is required to develop a quantum subtractor circuit. We use the quantum ripple carry adder proposed in [14] for developing the quantum subtractor circuit. To perform subtraction, we use the design approach presented in [15]. Thus, the input qubits $|B\rangle$ are complemented before being applied to the quantum ripple carry adder. Then, the ripple carry adder calculates $\overline{b}+a$. At the end of computation, the input qubits $|B\rangle$ are complemented again. As a result, the quantum subtractor calculates $(\overline{\overline{b}+a})$ which is equivalent to $b-a$ [15].

B. Design of Quantum Adder-Subtractor

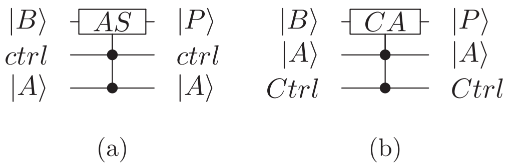

Fig. 3. Graphic symbols of (a) Adder-Subtractor (b) Conditional ADD operation circuit. AS represents add or subtract operation. CA represents conditional add operation

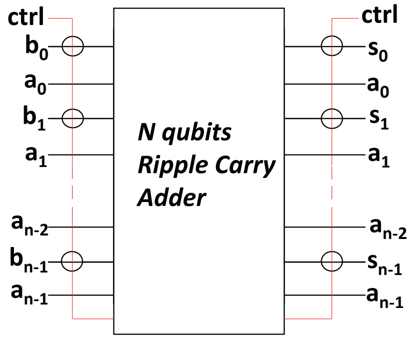

Fig. 4. Circuit design of N qubit quantum adder-subtractor based on N qubit quantum ripple carry adder

Fig. 3(a) shows the graphic symbol of the quantum controlled addition or subtraction circuit. The quantum adder-subtractor circuit operates as follows: (i) when the input labeled $ctrl$ is high (refer Fig. 3(a)), the circuit output is $|P\rangle = |B-A\rangle$, (ii) when the $ctrl$ input is low, the circuit output is $|P\rangle = |B+A\rangle$.

The complete working circuit of the quantum adder-subtractor circuit is shown in Fig. 4. The quantum adder-subtractor circuit is based on the design presented in [15] and uses the ripple carry adder in [14]. The quantum adder-subtractor calculates $(\overline{\overline{b}+a})$ when $ctrl$ is high. The expression $(\overline{\overline{b}+a})$ is equivalent to $b-a$.

C. Design of Quantum Conditional ADD Operation Circuit

Fig. 3(b) shows the graphic symbol of the quantum conditional ADD operation circuit. The quantum conditional ADD operation circuit operates as follows: (i) when the input labeled $ctrl$ is high (refer Fig. 3(b)), the circuit output is $|P\rangle = |B+A\rangle$, (ii) when the $ctrl$ input is low, the circuit output is $|P\rangle = |B\rangle$.

The complete working circuit of quantum conditional ADD operation circuit is shown in Fig.5 for 4 qubit operands. The quantum conditional ADD circuit uses a modified version of the ripple carry adder proposed in [14]. We were able

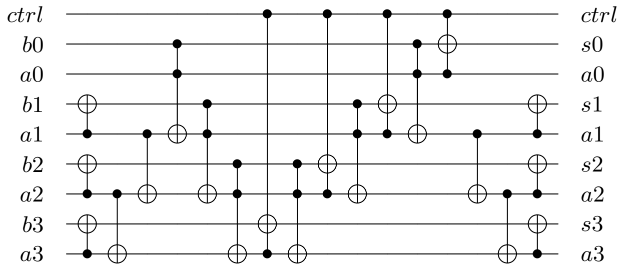

Fig. 5. Circuit design of quantum conditional ADD operation circuit

to remove the qubit that performs the carry out for the adder in [14] as we do not need the carry out qubit in the proposed integer dividers. The addition architecture in [14] uses Peres gates to perform the addition. The Peres gate can be decomposed into a Feynman and a Toffoli gate. By replacing the Feynman gate with a Toffoli gate, we can use the control line ($ctrl$) to determine whether the conditional ADD circuit will perform addition or no operation. Although, Fig.5 is just shown for 4 qubit operands, it can easily be extended to any operand size.

III. Design of Non-Restoring Quantum Integer Division Circuit

The quantum circuits that are required for developing the hardware implementation of the proposed non-restoring division algorithm are: (i) Leftshift operation circuit, (ii) controlled adder-subtractor, and (iii) conditional ADD operation circuit. We observed that we can eliminate the LeftShift operation circuit by combining $|R_{[0:n-2]}\rangle$ and $|Q_{[n-1]}\rangle$ to form an n qubit register there by saving the quantum resources.

The proposed non-restoring division algorithm for quantum circuits is shown in Table I. In Table I, the inputs to be given are: (a) $(|Q_{[0:n-1]}\rangle$, n qubit register in which the dividend is loaded; (b) $|D_{[0:n-1]}\rangle$, n qubit register in which the divisor is loaded; (c) $|R_{[0:n-1]}\rangle$, n qubit remainder register which is initiated to 0 at the start. At the end of computation, we get the quotient at $|Q_{[0:n-1]}\rangle$ and remainder at $|R_{[0:n-1]}\rangle$. The divisor is retained at the output. Also, $n+1$ garbage qubits are produced. The methodology to design our proposed quantum non-restoring integer division circuit is developed from the non-restoring division algorithm shown in Table I. The Steps of the methodology are presented below.

A. Design Methodology for Quantum Non-Restoring Integer Division Circuit

From Table I, we can see that the algorithm is divided into two phases. (i) Core Engine Phase and (ii) Supplementary Restoring Phase. The Core Engine Phase is iterated n times. Supplementary Restoring Phase takes place after the end of n iterations of the Core Engine Phase. The Supplementary Restoring Phase is repeated once. A quantum circuit is developed for each of these phases. The final circuit that performs the integer division using the non-restoring integer division algorithm is shown in Fig. 6. In Fig. 6, *I1* represents the first iteration of the Core Engine Phase, *I2* represents the second iteration and *In* represents the final iteration.

Algorithm 1: *Proposed quantum non-restoring division algorithm*

```
function $Non-Restore$  $(|Q_n\rangle, |R_n\rangle, |D_n\rangle)$
  for $i = 0$  $to$  $n-1$ do
      /* Start Core Engine Phase */
    if($|R_{[0:n-1]}\rangle > 0$)  then
      $(|Q_{[1:n-1]}\rangle, |R_{[0:n-1]}\rangle)$ = LEFTSHIFT $(|Q_{[0:n-1]}\rangle, |R_{[0:n-1]}\rangle)$;
      $|R_{[0:n-1]}\rangle = |R_{[0:n-1]}\rangle + |D_{[0:n-1]}\rangle$;
    else
      $(|Q_{[1:n-1]}\rangle, |R_{[0:n-1]}\rangle)$ = LEFTSHIFT $(|Q_{[0:n-1]}\rangle, |R_{[0:n-1]}\rangle)$;
      $|R_{[0:n-1]}\rangle = |R_{[0:n-1]}\rangle - |D_{[0:n-1]}\rangle$;
    end if;
    if($|R_{[0:n-1]}\rangle > 0$)  then
      $|Q_{[0]}\rangle = 1$;
    else
      $|Q_{[0]}\rangle = 0$;
     end if;
    /* End Core Engine Phase */
  end for;
 //after n iterations//
     /* Start Supplementary Restoring Phase */
    if($|R_{[0:n-1]}\rangle > 0$)  then
      $|R_{[0:n-1]}\rangle = |R_{[0:n-1]}\rangle$;
    else
      $|R_{[0:n-1]}\rangle = |R_{[0:n-1]}\rangle + |D_{[0:n-1]}\rangle$;
     end if;
    /* End Supplementary Restoring Phase */
 return $R$;
end function
```

TABLE I
PROPOSED QUANTUM NON-RESTORING DIVISION ALGORITHM

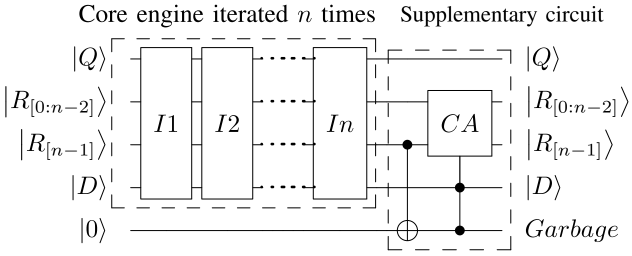

Fig. 6. Quantum non-restoring integer divider circuit design

1) Core Engine Phase: Fig. 7 represents the quantum circuit that does the operations that are marked under the Core Engine Phase in the algorithm in Table IV. We now elaborate on how the information moves in Fig. 7 .

- Step 1. $|D_{[0:n-1]}\rangle$ holds the divisor, $|R_{[0:n-1]}\rangle$ is initialised to zero, and $|Q_{[0:n-1]}\rangle$ holds the dividend.
- Step 2. We consider, $|Q_{[n-1]}\rangle$ and $|R_{[0:n-2]}\rangle$, as one combined register.
- Step 3. The combined register of Step 2 and $|D_{[0:n-1]}\rangle$ are applied as two n qubits inputs to the quantum adder-subtractor circuit. In Fig. 7, AS represents the adder-subtractor circuit. At the end of computation, register $|D_{[0:n-1]}\rangle$ emerges unchanged and the combined register

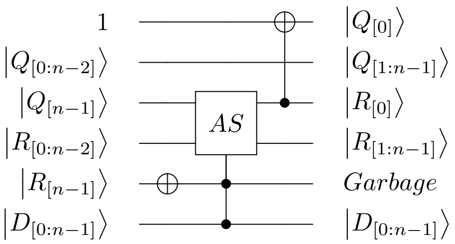

Fig. 7. Quantum non-restoring integer divider circuit design for first iteration(core engine)

now holds the sum or difference of the combined register and D.

- Step 4. Qubit $|R_{[n-1]}\rangle$ is complemented and applied as the $ctrl$ qubit to quantum adder-subtractor circuit.
- Step 5. The $ctrl$ qubit is left out as garbage.
- Step 6. An ancillary qubit set to 1 and qubit $|Q_{[n-1]}\rangle$ are applied to a CNOT gate. $|Q_{[n-1]}\rangle$ is the control qubit and 1 is the target qubit.

The Steps from 1 to 6 constitute the operations of the Core Engine Phase. From the algorithm in Table I, it can be seen that Steps 2 to 6 of the Core Engine Phase are iterated n times. So, the circuit in Fig. 7 that represents the Core Engine Phase is also iterated n times (see Fig. 6). The outputs of the first iteration as inputs to the second iteration and so on for all n iterations.

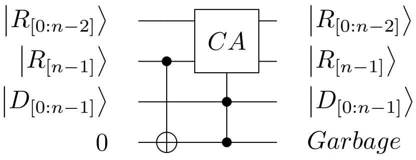

Fig. 8. Quantum circuit implementation of the Supplementary Restoring Phase(refer Table 1)

2) Supplementary Restoring Phase: After the end of n iterations of the Core Engine Phase, $|R_{[0:n-1]}\rangle$ might be negative at the end of n iterations. In that case, it has to be restored by adding the divisor. This restoration of the negative remainder is carried out by the Supplementary Restoring Phase quantum circuit shown in Fig. 8. The quantum circuit shown in Fig. 8 is the quantum implementation of the Supplementary Restoring Phase marked in the algorithm in Table I. We now elaborate on how the information moves in the supplementary circuit.

- Step 1. The qubit $|R_{[n-1]}\rangle$ and an ancillary qubit set to 0 are applied as inputs to a CNOT gate. $|R_{[n-1]}\rangle$ is the control qubit and the ancillary qubit is the target qubit. The target now holds the value of $|R_{[n-1]}\rangle$.
- Step 2. The ancillary qubit is used as $ctrl$ qubit to the conditional ADD operation quantum circuit.
- Step 3. Registers $|R_{[0:n-1]}\rangle$ and $|D_{[0:n-1]}\rangle$ are applied as inputs to conditional ADD operation quantum circuit. In Fig. 8, CA represents the conditional ADD operation circuit. $|D_{[0:n-1]}\rangle$ emerges unchanged and $|R_{[0:n-1]}\rangle$ will contain either the sum or emerge unchanged.
- Step 4. The control qubit $|R_{[0:n-1]}\rangle$ is left out as garbage.
- Step 5. After Step 4, we have the Quotient in $|Q_{[0:n-1]}\rangle$, and the remainder in $|R_{[0:n-1]}\rangle$. The divisor $|D_{[0:n-1]}\rangle$ is unchanged.

B. Cost Comparison With Existing Work

TABLE II
RESOURCE COUNT OF PROPOSED NON-RESTORING ALGORITHM DIVISION CIRCUIT

Designs	Adder-Subtractor	conditional ADD operation circuit	Non-Restoring Divider
T-count	$(14n-14)$	$(21n-14)$	$14n^2+21n-28$
T-depth	8	16	$8*n+7$
Ancilla qubits	0	0	$2*n+1$

TABLE III
COMPARISON OF RESOURCE COUNT BETWEEN PROPOSED AND EXISTING WORK

	1	Proposed	% impr. w.r.t. 1
T-count	$\approx 400n^2$	$14n^2+21n-28$	$\approx 96\%$
T-depth	$130*n$	$8*n+7$	$\approx 93\%$
Ancilla qubits	$2n$	$2*n+1$	-

1 is the work in [12]

The resources used in the design of the proposed quantum non-restoring integer division circuit is presented in Table II. As shown in Table II, the proposed design will require $2*n+1$ ancillary qubits. n ancillary qubits are used during initialization of remainder register and the remaining $n+1$ are transformed to garbage output. The T-count required by the design is given by summing the cost of adder-subtractor and conditional ADD operation quantum circuit at each stage. T-count of the proposed quantum non-restoring integer division circuit is $14n^2+21n-28$. The T-depth required by the design is given as $8*n+7$.

Comparison of resource costs between the proposed quantum non-restoring integer division circuit and the existing work is shown in Table III. To calculate the T-count and T-depth for [12] we use T-count and T-depth values from approximate phase gate implementations reported in [13]. The implementations with the poorest accuracy are used. This is because the T gate cost increases significantly as a function of accuracy. Table III shows that the proposed quantum circuit of integer division has an improvement ratio of 93% in terms of T-depth, and 96% in terms of T-count.

IV. DESIGN OF RESTORING QUANTUM INTEGER DIVISION CIRCUIT

The quantum circuits that are required for developing the hardware implementation of the proposed restoring division algorithm are (i) Leftshift operation circuit, (ii) n qubit quantum subtractor and (iii) Conditional ADD operation circuit. We observed that we can eliminate the LeftShift operation circuit by combining $|R_{[0:n-2]}\rangle$ and $(|Q_{[n-1]}\rangle$ to form an n qubit register which is actually equal to performing an left shift operation. By combining the qubits in this way, we do not have to use a separate left shift operation circuit.

The proposed restoring division algorithm is shown in Table IV. In Table IV, the inputs to be given are: (a) $(|Q_{[0:n-1]}\rangle$, n qubit register in which the dividend is loaded ; (b) $|D_{[0:n-1]}\rangle$, n qubit register in which the divisor is loaded; (c) $|R_{[0:n-1]}\rangle$, n qubit remainder register which is initiated to 0 at the start. The algorithm repeats n times. At the end of n iterations, we get the quotient at $(|Q_{[0:n-1]}\rangle$ and the remainder at $|R_{[0:n-1]}\rangle$. The divisor is retained at the output. The methodology to design our proposed quantum restoring integer division circuit is developed from the restoring division algorithm shown in Table IV. The Steps of the methodology are presented below.

Algorithm 1 : *Proposed Restoring division algorithm*

```
function $Restore$ $(|Q_n\rangle, |R_n\rangle, |D_n\rangle)$
  for $i = 0$ $to$ $n-1$ do
      $(|Q_{[1:n-1]}\rangle, |R_{[0:n-1]}\rangle)$ = LEFTSHIFT $(|Q_{[0:n-1]}\rangle, |R_{[0:n-1]}\rangle)$;
      $(|R-D_{[0:n-1]}\rangle = |R_{[0:n-1]}\rangle - |D_{[0:n-1]}\rangle$;
    if$(|R_{[0:n-1]}\rangle > 0)$ then
      $|Q_{[0]}\rangle = 1$
      $|R_{[0:n-1]}\rangle = |R - D_{[0:n-1]}\rangle$;
    else
      $|Q_{[0]}\rangle = 0$;
      $|R_{[0:n-1]}\rangle = |R - D_{[0:n-1]}\rangle + |D_{[0:n-1]}\rangle$;
    end if;
  end for;
//repeat for $n$ iterations//
return $R$;
end function
```

TABLE IV
PROPOSED RESTORING DIVISION ALGORITHM FOR QUANTUM CIRCUITS

A. Design Methodology for Quantum Restoring Integer Division Circuit

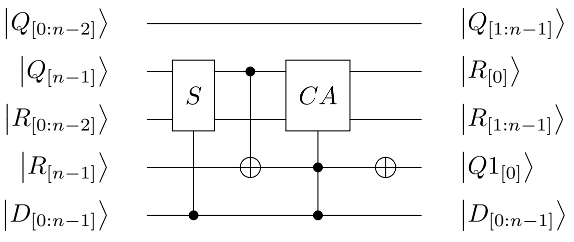

Fig. 9. Quantum restoring integer divider circuit design for a single iteration

Fig.9 shows the quantum circuit generated for the quantum restoring division circuit after 1 iteration of our design methodology. The Steps of the proposed methodology are repeated n times. Hence, the circuit in Fig. 9 is also iterated n times. This is done by using the outputs of the first iteration as inputs for the next iteration. Fig. 10 shows the complete quantum restoring division circuit where $I1$ represents the first iteration, $I2$ represents second iteration and In represents the final iteration. We now elaborate on how information moves through the circuit shown in Fig. 9.

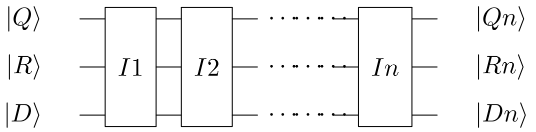

Fig. 10. Quantum restoring integer divider circuit design(for n iterations)

- Step 1. The $|D_{[0:n-1]}\rangle$ holds the divisor, $|R_{[0:n-1]}\rangle$ is initialised to zero, and $|Q_{[0:n-1]}\rangle$ holds the dividend.
- Step 2. We consider, $|Q_{[n-1]}\rangle$ and $|R_{[0:n-2]}\rangle$, as one combined register.
- Step 3. The combined register mentioned above in Step 2, and $|D_{[0:n-1]}\rangle$ are given as inputs to the quantum subtractor circuit. Register $|D_{[0:n-1]}\rangle$ emerges unchanged. The combined register now holds the result of subtraction of R and D registers. Let us call this result as $|R-D_{[0:n-1]}\rangle$.
- Step 4. Qubits $|R - D_{[n-1]}\rangle$ and $|R_{[n-1]}\rangle$ are supplied to a CNOT gate. $|R - D_{[n-1]}\rangle$ is the control qubit and the $|R_{[n-1]}\rangle$ is the target qubit. The target now holds the value of $|R - D_{[n-1]}\rangle$ because $|R_{[n-1]}\rangle$ is always zero throughout the computation.
- Step 5. Qubit $|R_{[n-1]}\rangle$ is the control qubit to the conditional ADD operation circuit.
- Step 6. Registers $|R - D_{[0:n-1]}\rangle$ and $|D_{[0:n-1]}\rangle$ are the two n qubit inputs to the conditional ADD operation circuit. Register $|D_{[0:n-1]}\rangle$ emerges unchanged. The combined register will contain either the sum or emerge unchanged..
- Step 7. $|R_{[n-1]}\rangle$ is complemented.

Steps 2 through 7 are repeated n times. At the end of n iterations, the Quotient will be in $|Q_{[0:n-1]}\rangle$, the remainder in $|R_{[0:n-1]}\rangle$ and the divisor emerges unchanged.

B. Cost Comparison With Existing Work

TABLE V
RESOURCE COUNT OF PROPOSED RESTORING DIVISION CIRCUIT

	Subtractor	conditional ADD operation circuit	Restoring Divider
T-count	$(14n - 14)$	$(21n - 14)$	$35n^2 - 28n$
T-depth	8	16	$18 * n$
Ancilla qubits	0	0	n

The resources used in the design of the proposed quantum restoring integer division circuit is presented in Table V. As shown in Table V, the proposed design will require n ancillary qubits during initialization of the remainder register. The T-count required by the design is given by summing the cost of

TABLE VI
COMPARISON OF RESOURCE COUNT BETWEEN PROPOSED AND EXISTING WORK

	1	Proposed	% impr. w.r.t. 1
T-count	$\approx 400n^2$	$35n^2 - 28n$	$\approx 91\%$
T-depth	$130 * n$	$18 * n$	86.15%
Ancilla qubits	$2n$	n	50%

1 is the work in [12]

subtractor and conditional ADD operation quantum circuit at each stage. T-count of the proposed quantum restoring integer division circuit is $35n^2 - 28n$. The T-depth required by the design is given as $18 * n$.

Comparison of resource estimation between proposed quantum circuit of integer division and the existing quantum circuit of integer division in [12] is shown in Table VI. To calculate the T-count and T-depth for [12] we use T-count and T-depth from approximate phase gate implementations reported in [13]. The implementations with the poorest accuracy were used. This is because the T-count increases significantly as a function of accuracy. Table VI showed that the proposed quantum circuit of integer division has an improvement ratio of 86.15% in terms of T-depth, and 91% in terms of T-count.

V. CONCLUSION

In this work, we have presented two designs for quantum circuit integer division based on Clifford+T gates. The first quantum circuit presented is based on the non-restoring division algorithm and the second quantum circuit presented is based on the restoring division algorithm. The design of subcomponents used in the proposed quantum integer division circuits such as the quantum conditional ADD operation circuit, quantum adder-subtractor and quantum subtraction circuit are also shown. The proposed quantum integer division circuits are shown to be superior to existing designs in terms of T-depth and T-count. We conclude that the proposed non-restoring division circuit can be integrated in a larger quantum data path system design where T-count and T-depth are of primary concern. We also conclude that the proposed restoring division circuit can be integrated in a larger quantum data path system design to implement quantum algorithms where qubits are limited and T-count and T-depth must be kept to a minimum.

Existing quantum circuit implementations do not include the additional qubit transformations that account for the available instruction set architecture, the hardware connectivity and layout constraints of a particular technology [16], [17]. For example, in trapped ion quantum computers (such as those presented in [18] and [19]) offer different methods to implement multi-qubit gates. These methods include piece-wise, nearest-neighbor interactions that address individual qubits as well as global interactions that apply coherent rotations uniformly to all available ions. The choice of which method to use depends on the layout of the device architecture and the relative complexity of the different instructions. Such constraints will significantly impact how quantum circuits are implemented in practice. The proposed quantum integer division circuit designs do not take into account technology constraints. However, the T-count and T-depth cost savings of our quantum integer division circuits are unaffected by these hardware considerations. To efficiently implement quantum algorithms, new designs need to be investigated for integer division that minimize the overhead imposed by technology constraints.

REFERENCES

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

[2] S. Beauregard, "Circuit for Shor's algorithm using 2n+3 qubits," *Quantum Information & Computation*, vol. 3, no. 2, pp. 175–185, Mar 2003.

[3] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

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

[5] X. Zhou, D. W. Leung, and I. L. Chuang, "Methodology for quantum logic gate construction," *Phys. Rev. A*, vol. 62, p. 052316, Oct 2000. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.62.052316

[6] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 6, pp. 818–830, 2013.

[7] S. J. Devitt, A. M. Stephens, W. J. Munro, and K. Nemoto, "Requirements for fault-tolerant factoring on an atom-optics quantum computer," *Nature Communications*, vol. 4, p. 2524, Oct. 2013.

[8] IBM, *Quantum Computing - IBM Q*, 2017, available at: https://www.research.ibm.com/ibm-q/.

[9] N. M. Nayeem, A. Hossain, M. Haque, L. Jamal, and H. M. H. Babu, "Novel reversible division hardware," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, Aug 2009, pp. 1134–1138.

[10] S. V. Dibbo, H. M. H. Babu, and L. Jamal, "An efficient design technique of a quantum divider circuit," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 2102–2105.

[11] F. Dastan and M. Haghparast, "A novel nanometric fault tolerant reversible divider," *International Journal of the Physical Sciences*, vol. 6, no. 24, pp. 5671–5681, October 2011.

[12] A. Khosropour, H. Aghababa, and B. Forouzandeh, "Quantum division circuit based on restoring division algorithm," in *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*. IEEE, 2011, pp. 1037–1040.

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

[14] H. Thapliyal and N. Ranganathan, "Design of efficient reversible logic-based binary and bcd adder circuits," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 9, no. 3, p. 17, 2013.

[15] H. Thapliyal, "Mapping of subtractor and adder-subtractor circuits on reversible quantum gates," in *Transactions on Computational Science XXVII*. Springer, 2016, pp. 10–34.

[16] K. A. Britt and T. S. Humble, "High-performance computing with quantum processing units," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, p. 39, 2017.

[17] K. A. Britt and T. S. Humble, "Instruction set architectures for quantum processing units," *arXiv preprint arXiv:1707.06202*, 2017.

[18] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, p. 201618020, 2017.

[19] E. A. Martinez, T. Monz, D. Nigg, P. Schindler, and R. Blatt, "Compiling quantum algorithms for architectures with multi-qubit gates," *New Journal of Physics*, vol. 18, no. 6, p. 063029, 2016.