The Journal of Systems and Software 215 (2024) 112091

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Quantum circuit optimization of an integer divider☆

Francisco Orts *, Remigijus Paulavičius, Ernestas Filatovas

Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4, LT-08412 Vilnius, Lithuania

ARTICLE INFO

Keywords:
Quantum divider
Quantum circuit
Clifford+T gates
Quantum circuit optimization

ABSTRACT

Efficient arithmetic operations in quantum circuits play a vital role in the implementation of quantum algorithms. Quantum circuits constructed exclusively using gates of the Clifford+T group are compatible with error detection and correction codes available in the quantum literature. However, the T gate, a member of this group, has a higher cost compared to other gates, making it crucial to minimize its usage to reduce circuit expenses. While the T gate cannot be entirely avoided since the Clifford group is not a universal set of gates, circuit optimization can effectively reduce the number of T gates required for implementation. In this work, we present a novel divider circuit for quantum computing that focuses on reducing the number of T gates while maintaining a reasonable number of qubits for this type of operation. To achieve this, we introduce variants of minor circuits, including a comparator and two types of subtractors. These circuits are based on published literature but undergo modifications to optimize their resource utilization for performing the division operation. The obtained results demonstrate that the proposed divider circuit outperforms other currently published divider circuits in terms of T gate usage, highlighting its efficiency and potential practicality in quantum algorithms.

1. Introduction

Quantum computing is a computing paradigm that offers efficient ways to solve some problems that cannot be efficiently solved by classical computing (Grover, 1996; Shor, 1999). Instead of being bit-based, the so-called quantum computers use the quantum bit (qubit) as their unit of information (Bernhardt, 2019). These qubits possess properties inherited from quantum mechanics, such as superposition or entanglement. Such properties make it possible to find ingenious new ways to solve problems (Nielsen and Chuang, 2011). However, to build quantum algorithms, it is usually necessary to implement quantum circuits that compute classical functions (Romero and Aspuru-Guzik, 2021). For instance, arithmetic operations are often used in algorithms that offer quantum advantages (Thomsen et al., 2010; Li et al., 2020a). To implement Shor's algorithm, the most powerful of the quantum algorithms (Nielsen and Chuang, 2011), it is common to use adder circuits for certain operations (Orts et al., 2020). Similarly, division, a fundamental arithmetic operation crucial for several proven quantum algorithms, also relies on circuits (Van Dam and Hallgren, 2000; Van Dam et al., 2006; Hallgren, 2007; Wei et al., 2020; Houssein et al., 2021; Giani and Eldredge, 2021). Therefore, an efficient divider circuit will be of great interest in quantum computing even if it does not offer any quantum advantage since it can be integrated into such algorithms that offer such advantages (Pérez-Salinas et al., 2020).

A quantum circuit shares a conceptual similarity with a classical circuit (Humble et al., 2019). Like classical circuits, it takes an input and undergoes transformations through gates to produce an output. However, the similarities end there. Quantum circuits differ in several key aspects: they must be reversible, maintain the same number of inputs and outputs, disallow fan-in or fan-out, and cannot be cloned into another qubit (Nielsen and Chuang, 2011). Implementing a classical function by means of a quantum circuit is, therefore, not a trivial task, as it requires careful consideration of these aspects (Bernhardt, 2019). To add a further degree of difficulty, current quantum devices, commonly referred to as Noisy Intermediate-Scale Quantum (NISQ) devices, have limited resources and are extremely sensitive to internal and external noise (Preskill, 2018). It is essential to design quantum circuits that optimize the use of the few available resources; otherwise, they will be quickly consumed (Endo et al., 2021). It is imperative to devise strategies to mitigate the effects of noise; otherwise, the results obtained by the circuits will contain high error rates (Endo et al., 2018).

A common approach to mitigate the effects of noise is by employing quantum gates from the Clifford+T group (Bravyi and Gosset, 2016). Circuits built using only gates belonging to the universal Clifford+T group are of interest due to their compatibility with error detection and correction codes (Amy et al., 2013). However, they bring an added challenge due to the presence of the T gate (Barenco et al.,

☆ Editor: Raffaela Mirandola.

* Corresponding author.
E-mail addresses: francisco.gomez@mif.vu.lt (F. Orts), remigijus.paulavicius@mif.vu.lt (R. Paulavičius), ernestas.filatovas@mif.vu.lt (E. Filatovas).

https://doi.org/10.1016/j.jss.2024.112091
Received 1 August 2023; Received in revised form 29 February 2024; Accepted 7 May 2024
Available online 10 May 2024

1995). The T gate is essential to convert the Clifford group into the universal Clifford+T group (Nielsen and Chuang, 2011). Nevertheless, the fault-tolerant version of the T gate carries a greater cost compared to the other gates within the Clifford group (Gosset et al., 2013; Paler et al., 2017; Litinski, 2019), resulting in a significant increase in the overall cost of the circuits. Moreover, classical devices struggle to simulate this operation efficiently; therefore, a high number of T gates can imply considerable computational burdens on many simulators (Kissinger et al., 2022). There is interest in the scientific community in designing circuits that allow the number of T gates (commonly denoted as T-count) to be reduced (Bocharov et al., 2015; Heyfron and Campbell, 2018; Muñoz-Coreas and Thapliyal, 2018; Amy and Mosca, 2019; Kissinger and van de Wetering, 2019; Orts et al., 2023b). Associated with this T-count, a second metric called T-depth represents the maximum number of T gates that a circuit has to compute sequentially (Selinger, 2013). So, the T-depth allows an estimate of the depth of the circuit to be made. Of course, given the scarcity of resources in quantum devices, it is equally important to keep the required number of qubits in a circuit small (Pérez-Salinas et al., 2020).

In this paper, a divider circuit for quantum computing is proposed. This circuit performs the integer division between two numbers D and Q, with the requirement that Q must be less than or equal to D. The objective of the circuit is to reduce both the T-count and T-depth with respect to the divider circuits currently available in the quantum literature. It is common in state-of-the-art circuits to achieve improvements in T-count at the cost of sacrificing qubits (Gidney, 2018). However, in this work, we have tried to keep the number of qubits in the same range as the rest of the dividers. On the other hand, to achieve the implementation of the divider, it has been necessary to use a comparator circuit and two (different) subtractor circuits. Efficient circuits already published in the literature have been used, although several adaptations have been made to reduce their metrics. We do not want to claim the merits of such circuits but only to point out that, with slight improvements, they are even more efficient for integration in the divider circuit.

The most important contributions of this paper are the following:

- Introducing metrics to assess circuit's performance.
- Proposing improvements (in terms of such metrics) in several comparator and conditional subtractor circuits.
- Proposing an efficient divider circuit exclusively using Clifford+T gates, enabling the utilization of existing error detection and correction codes.
- Conducting a comprehensive comparative analysis with state-of-the-art dividers, clearly demonstrating the superior performance of the proposed circuit.

The remainder of the paper is structured as follows. In Section 2, the necessary concepts are introduced to understand this work. Section 3 presents the proposed circuit, including a detailed description of the subcircuits forming the circuit and the construction of the divider itself. Section 4 discusses the divider and its components while also conducting a comparative analysis of the proposed circuit against the most efficient dividers currently available. Finally, Section 5 presents the concluding remarks, summarizing the key findings of the paper.

2. Background

This section provides the background needed to comprehend the work presented in this paper. Initially, we introduce the quantum gates from the Clifford+T group, which are utilized in this study to develop a novel divider circuit. Subsequently, we detail the metrics used to evaluate the circuit's performance. Finally, the need to optimize this operation in quantum computing is underscored, along with the scope outlined by this work.

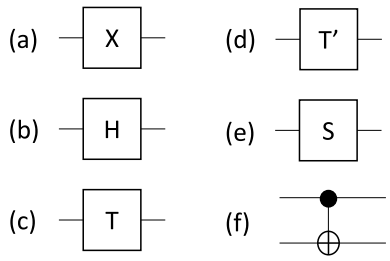

Fig. 1. Symbols of the Clifford+T gates used in this work. (a) Pauli-X gate, (b) Hadamard gate, (c) T gate, (d) hermitian of the T gate, (e) S gate, and (f) CNOT gate.

2.1. Quantum circuits and gates

Quantum circuit design is the most common (but not the only) way to program a current quantum device (Combarro and Gonzalez-Castillo, 2023). Such quantum circuits are made up of gates that make modifications to the state of the qubits. There are infinite quantum gates, but the Clifford+T group allows one to approximate the remaining infinite quantum gates (Bernhardt, 2019). Moreover, in this work, not all the gates of the Clifford+T group are used, but only a subset of them:

- Pauli-X gate: given a qubit in state $\alpha|0\rangle + \beta|1\rangle$, the Pauli-X gate inverts the value of the base amplitudes. That is, it produces the result $\beta|0\rangle + \alpha|1\rangle$ (Pauli, 1988).
- Hadamard gate: this gate allows a qubit to be placed in superposition (Nielsen and Chuang, 2011). For instance, given a qubit in state $|0\rangle$, the result of applying this gate on it will be state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$.
- T gate: this has already been noted for its high cost, and it is the aim of this work to reduce the number of these gates in the divider circuit. The gate introduces a phase of $\frac{\pi}{4}$ into the qubit state (Barenco et al., 1995). It is also interesting to mention that this gate is not efficiently simulatable on a classical computer (Nielsen and Chuang, 2011).
- Hermitian of the T gate: This other gate introduces a phase of $-\frac{\pi}{4}$ into the qubit state (Barenco et al., 1995). The hermitian of the T gate shares all the problems of the T gate. Therefore, for T-count and T-depth counting purposes, it is counted as if it were a T gate.
- S gate: the S gate introduces a phase of $\frac{\pi}{2}$ (Barenco et al., 1995). It should be noted that this gate does not share the high cost of the two previous gates.
- CNOT gate: this is the Clifford+T gate responsible for enabling entanglement between two qubits, a fundamental requirement for a universal set of quantum gates. It is a controlled version of the Pauli-X gate: it will have a similar effect to the Pauli-X gate on a B qubit, but only if another A qubit is in the $|1\rangle$ state. This operation is usually represented by $A \oplus B$.

The symbol for each gate is shown in Fig. 1. There are three more gates used in this work: the Toffoli gate (Toffoli, 1980), the temporary logical-AND gate (Gidney, 2018), and the uncomputation gate of the temporary logical-AND operation (Gidney, 2018). Such gates do not belong by definition to the Clifford+T group but can be implemented using only gates from the Clifford+T group, as shown in Fig. 2 (where their symbols are also shown). For the Toffoli gate, the implementation proposed by Amy et al. (2013) has been chosen because it is the most efficient in terms of T-count and T-depth.

The Toffoli gate and the temporary logical-AND gate operate over three qubits. They perform the operation $AB \oplus C$. This operation is similar to the CNOT but with a second control qubit. In fact, in some sources, the Toffoli gate is labeled as CCNOT gate (Zahedinejad et al., 2015). However, there are important differences between the two gates.

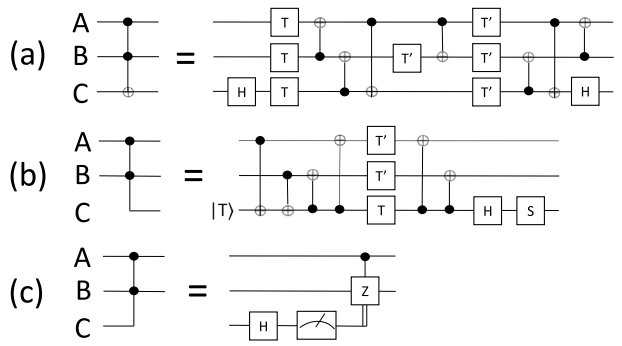

Fig. 2. Symbol and Clifford+T implementation of the (a) Toffoli gate (Amy et al., 2013), (b) temporary logical-AND gate (Gidney, 2018), and (c) uncomputation of the temporary logical-AND gate (Gidney, 2018). The qubit marked as T in the temporary logical-AND gate must be prepared in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$. This state can be achieved using a Hadamard gate and a T gate.

The Toffoli gate has a T-count of 7, and a T-depth of 3, while the temporary logical-AND gate has a T-count of 4, and a T-depth of 2. On the other hand, the Toffoli gate is capable of using an existing qubit as a target qubit. For instance, it is capable of performing the operation $|1\rangle|1\rangle \oplus |1\rangle$ (which, by completeness, we will say, causes the target qubit to change its state to $|0\rangle$). However, the temporary logical-AND gate is unable to perform this operation (at least directly), as it is focused on performing the AND operation between the control qubits and storing this operation in an auxiliary qubit. It was said before that both the Toffoli gate and the temporary logical-AND gate perform the operation $AB \oplus C$. In reality, the temporary logical-AND gate will only work if C is an auxiliary qubit reserved for this operation and must be prepared in the $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$ state. For the sake of clarity, it is worth mentioning explicitly that the preparation of this state requires a Hadamard gate and a T gate acting consecutively on a qubit in state $|0\rangle$. The aforementioned T gate required for the preparation of this state is already included in the metrics mentioned for the temporary logical-AND gate (T-count: 4, T-depth: 2).

As explained in the next subsection, when auxiliary operations are performed, it is necessary to revert them to free resources. The way to reverse a Toffoli operation is to apply another Toffoli operation of the same type. It is trivial to show that this operation will double the values of T-count and T-depth. However, the uncomputation gate of the temporary logical-AND operation can revert to the temporary logical-AND with a T-count and T-depth of 0. Therefore, in situations where operations must be reversed, the temporary logical-AND gate and uncomputation are more efficient than the Toffoli gate. At least in terms of T-count or T-depth, as their usage always involves employing an auxiliary qubit. However, as explained above, it is not always possible to replace a Toffoli gate with a temporary logical-AND gate.

2.2. Metrics

The following metrics have been employed in this study to assess the performance of both the proposed circuit and the remaining circuits:

- T-count: the number of T gates (including the hermitian of the T gate) contained in a circuit. The reasons for a reduction in the T-count have already been explained.
- T-depth: this metric indicates the maximum number of T gates that the circuit must execute sequentially. It can be interpreted as the depth of the critical path of the circuit in terms of T gates.
- Number of ancilla qubits: indicates the number of qubits that have a constant initial value (Mohammadi and Eshghi, 2009). In other words, it specifies the number of auxiliary qubits in the circuit. Any qubit that does not belong to this category is because it is intended to contain the input of the circuit. For example, in the case of a divider circuit that computes D/Q, all qubits that do not initially contain D or Q are considered ancilla qubits.
- Number of garbage outputs: a garbage output is any qubit that, after completion of circuit execution, contains a value that is neither part of the theoretical output of the circuit nor its initial value (Mohammadi and Eshghi, 2009). The problem with qubits that contain garbage output is that their values are unknown and cannot be entangled with qubits from other circuits, thus wasting resources. The most common way to clean up the garbage outputs in a circuit is to reverse the circuit using Bennett's scheme (Bennett, 1973), although there are other ingenious ways to achieve this, such as the aforementioned uncomputation gate of the temporary logical-AND operation. However, strategies such as the latter may have consequences such as, for example, preventing a sub-circuit from being used in specific algorithms that require full reversibility (López et al., 2023).

2.3. Applicability and scope of the proposed circuitry

The necessity for quantum circuits that compute division is justified by the existence of quantum algorithms that require this operation. These algorithms belong to areas as diverse as objective function maximization (Gyongyosi and Imre, 2019), quantum image processing (Zhou and Wan, 2021), and transcendental functions (Wang et al., 2020). Given that division plays a fundamental role in these algorithms, the optimization of division circuits directly contributes to enhancing their efficiency. Reducing the cost of the division will mean that, for instance, the image processing proposed by Zhou and Wan (2021) will be less expensive. This rationale applies equally to other related works, further emphasizing the necessity for efficient divider circuits. Thus, it underscores not only the demand for divider circuits but also the imperative for them to operate with high efficiency.

Despite the significance of division and its associated algorithms, it is essential to consider the current limitations regarding their implementation. Present quantum computers contain at most a few hundred qubits (Khalid et al., 2023), a capacity insufficient for executing algorithms at moderate or large scales. Consequently, hybrid approaches such as the Variational Quantum Eigensolver or the Quantum Approximate Optimization Algorithms are more suitable than the use of quantum arithmetic circuits (Zhang et al., 2023). However, these alternative circuits require their specific configurations, which will depend on the size of the inputs at any given time, the inputs themselves, and even the current calibration of the quantum device. Therefore, once the NISQ era has passed and a sufficient number of qubits are available, the more generalized and dependency-free algorithmic circuits for each situation will become predominant. This is also the case for simulators where noise is not considered or with sufficient computational capacity to simulate the qubits necessary for the corresponding error tolerance.

3. Proposed divider circuit

This section presents the proposed divider circuit. Firstly, it introduces the division algorithm that serves as the basis for the circuit design. Next, it describes the subcircuits employed in the formation of the divider circuit. Finally, a detailed explanation of the construction of the proposed circuit is provided.

3.1. Division algorithm

The proposed circuit is based on the methodology proposed by Yuan et al. (2022). This methodology can be summarized using Algorithm 1. The idea presented in this algorithm is a customized version of the so-called "long division" algorithm, which is widely used in digital electronics (Hennessy and Patterson, 2011). This algorithm consists of comparing the divisor Q with a part of the dividend D of the same size,

always starting with the most significant digits of the dividend. If the divisor is less than or equal to that part, a subtraction is performed. Otherwise, the selected fragment of the dividend is simply increased by one digit, the most significant one of those not yet considered. This process is repeated until all the digits of D have been traversed, which occurs in $N - M + 1$ iterations, being N and M the number of digits of D and Q, respectively.

Algorithm 1 Division algorithm proposed by Yuan et al. (2022)

Require: Dividend $D = D_{N-1} \dots D_0$, and divisor $Q = Q_{M-1} \dots Q_0$, with $M \leq N$.
Ensure: $S = S_{N-M} \dots S_0$
  for $i = 0$ to $N - M$ **do**
    if $D_{N-1-i:N-M-i} \geq Q_{M-1:0}$ **then**
      $S_{N-M+1-k} = \overline{S_{N-M+1-k}}$
      $D = [D_{N-1-i:N-M-i} - Q_{M-1:0}] \dots D_0$
      $S_{N-M-k} = D_{N-1-i}$
    else
      $S_{N-M+1-k} = S_{N-M+1-k}$
      $S_{N-M-k} = D_{N-1-i}$
      $D = [D_{N-1-i:N-M-i-1} - Q_{M-1:0}] \dots D_0$
    end if
  end for
  return $S_{N-M+1-k}$

Carrying out an iteration of Algorithm 1 requires the comparison $D_{N-1-i:N-M-i} - Q_{M-1:0}$, as well as the subtractions $D = [D_{N-1-i:N-M-i} - Q_{M-1:0}] \dots D_0$ and $D = [D_{N-1-i:N-M-i-1} - Q_{M-1:0}] \dots D_0$, in addition to other minor operations. Therefore, a comparator circuit and a subtractor circuit are necessary. The comparison $D_{N-1-i:N-M-i} \geq Q_{M-1:0}$ can be performed with one of the many comparator media available in the literature (Xia et al., 2018, 2019; Li et al., 2020b; Xia et al., 2020; Orts et al., 2021, 2023a). However, there are some nuances in the two subtractions. Firstly, both subtractions do not have to be performed in the same iteration. Instead, one or the other is performed based on the result of the comparison, making them conditional subtractions based on a previous outcome. Secondly, the first subtraction is performed between numbers of equal lengths, while the second subtraction is performed between numbers of different lengths. The same circuit can be used for the second subtraction, but it would be inefficient, as it would not represent the smaller number effectively, leading to resource wastage.

3.2. Comparator

The comparator of Xia et al. (2019) is one of the most optimized comparators in the quantum literature. It allows the comparison of two numbers of size N with only 2 ancilla qubits. An example of this circuit, for the case $N = 2$, is shown in Fig. 3(a). The comparator of Li et al. (2020b) improves on that of Xia et al. (2019) by requiring only one ancilla qubit, although it has a higher T-count and T-depth. In this work, we have chosen the Xia et al. circuit precisely because of its better T-count and T-depth values. In addition, we have included minor changes in the Xia et al. circuit that allow us to reduce its T-count and T-depth further. In particular, we have replaced the Toffolis gates of the Xia et al. circuit with temporary logical-AND gates and the uncomputation gate of the temporary logical-AND gate. Of the $2N - 1$ Toffoli gates involved in the Xia et al. comparator, the first N gates are replaced by temporary logical-AND gates so that for each replaced gate, the T-count and T-depth are reduced by 4 and 1, respectively. The remaining $N-1$ Toffoli gates, intended to reverse the above operations, are replaced by uncomputation gates of the temporary logical-AND gate, achieving (for each gate replaced) a reduction in the T-count and T-depth of 7 and 3, respectively. An example of this new version of the circuit of Xia et al. is shown in Fig. 3(b).

Replacing each Toffoli gate with a temporary logical-AND gate does involve adding an ancilla qubit. However, the first Toffoli gate of the original circuit, as well as the last, operates on an auxiliary qubit. Thus, replacing these gates with a temporary logical-AND gate does not imply increasing the number of ancilla qubits since these same qubits can be used. In other words, for the case $N = 2$ (Fig. 3(b)), replacing the Toffoli gates with temporary logical-AND gates does not increase the number of qubits, while the T-count and T-depth are significantly reduced (13 and 5, respectively). For $N \geq 3$, $N - 2$ ancilla qubits will be needed.

To implement the comparator, for any number of digits N, the following steps must be followed:

1. Encode A and B into $2N$ qubits. Prepare N ancilla qubits in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$.
2. From $i = 0$ to $N - 1$, apply a CNOT gate such that it performs the operation $b_i \oplus a_i$.
3. Apply a temporary logical-AND gate that has as control qubits the qubits that initially contained a_0 and b_0 and whose target qubit is the first ancilla qubit. Apply a CNOT gate to this last ancilla qubit, b_1 being the control qubit.
4. From $i = N - 1$ to $i = 1$, apply a Pauli-X to the qubit that initially contained a_i. Apply a temporary logical-AND gate whose control qubits are the target qubit of the last temporary logical-AND gate applied in the circuit and the qubit that initially contained a_i. Its target qubit is an unused ancilla qubit. Apply another Pauli-X in the same place to revert to the initial one. Finally, apply a CNOT gate with b_i as the control qubit and the ancilla qubit of the previous temporary-logical-AND as the target one.
5. The result will be in the last ancilla qubit.
6. Apply the circuit inverse to reverse the garbage output. To revert to the temporary logical-AND gate, apply the uncomputation gate of the temporary logical-AND gate.

3.3. Conditional equal-bit subtractor

Although there are various subtractors in the quantum literature (Thapliyal and Ranganathan, 2011; Orts et al., 2019), this operation is usually performed using adders since such circuits allow the operation to be performed using fewer resources (Orts et al., 2020). Given two numbers A and B, subtraction using an adder can be carried out as $A - B = \overline{\overline{A} + B}$ (Hennessy and Patterson, 2011). The same is true for conditional subtraction: it can be performed using a conditional adder. The most efficient conditional adder currently published in terms of T-count and ancilla qubits is the one published by Muñoz-Coreas and Thapliyal (2018).

In the conditional adder of Muñoz-Coreas and Thapliyal, a qubit marked ctr acts as a control qubit of the entire circuit, so that if this qubit has a value of $|1\rangle$, the operation $A - B$ will be performed. Otherwise, no operation will be performed, and the input qubits will keep their initial value. To act as a subtractor, A must be negated at the beginning of the circuit (only if $ctr = |1\rangle$). The output ($S = s_1 s_0$) must also be negated at the end, and -again- A to avoid garbage outputs. On the other hand, in no case will $A - B$ need more digits than A or B since before this circuit it has been verified that B is not larger than A (Algorithm 1). Therefore, the computation of the carry output performed in the original circuit can be omitted. Making these slight modifications to the circuit of Muñoz-Coreas and Thapliyal results in the circuit shown in Fig. 4.

Muñoz-Coreas and Thapliyal's conditional adder, acting as a conditional subtractor, can be constructed for any number of digits N by following these steps:

1. Encode A and B into $2N$ qubits. Another qubit ctr will contain the value used as a control.

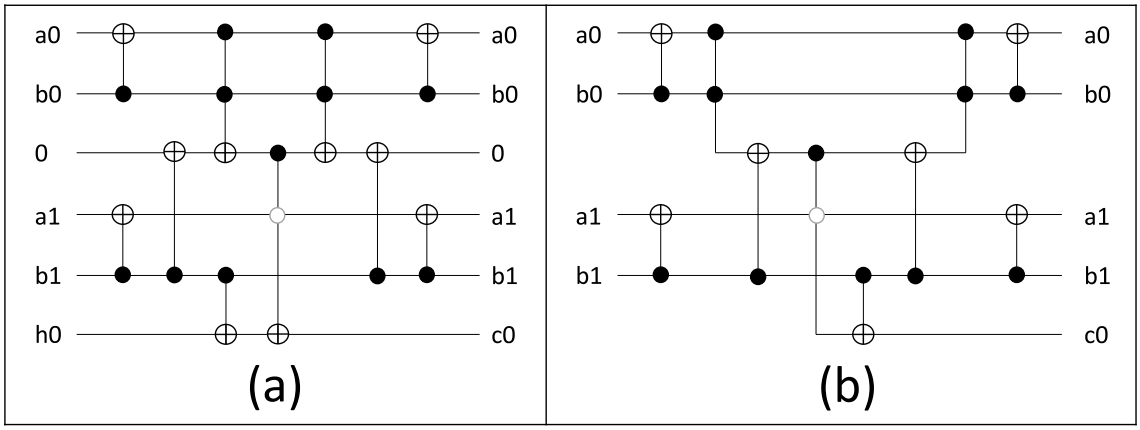

Fig. 3. (a) Comparator proposed by Xia et al. (2019), for the case $N = 2$. The circuit compares two numbers $A = a_1a_0$ and $B = b_1b_0$ using two ancilla qubits (initialized as $|0\rangle$). The qubit c_0 will return $|0\rangle$ if $A \geq B$, or $|1\rangle$ otherwise. (b) Proposed modifications to the Xia et al. comparator to reduce the T-count without increasing the number of ancilla qubits.

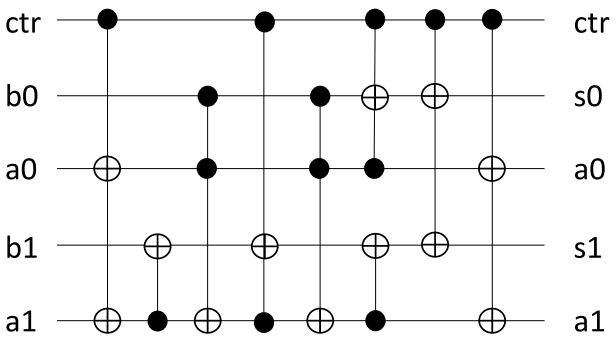

Fig. 4. Conditional adder proposed by Muñoz-Coreas and Thapliyal (2018). The circuit has been adapted to carry out the operation $a - b = \overline{\overline{a} + b}$ by using several CNOT gates to negate a at the beginning and s (the result) at the end. To avoid garbage outputs, a is negated again at the end to reverse it. Moreover, the generation of the original carry output has been removed, as it is not necessary for its integration in the divider.

2. Reverse (or not) all digits of A using CNOT gates controlled by ctr.
3. From $i = N - 1$ to $i = 1$, apply a CNOT to compute $\overline{a_i} \oplus b_i$.
4. From $i = 0$ to $i = N-2$, apply a Toffoli gate whose control qubits are the qubits that originally contained a_i and b_i, and whose target qubit is the qubit that originally contained a_{i+1}.
5. From $i = N - 1$ to $i = 1$, compute a Toffoli gate over the qubit that originally contained b_i, and whose control qubits are ctr and the qubit that originally contained a_1. Then, apply a Toffoli gate whose control qubits are the qubits that originally contained a_{i-1} and b_{i-1}, and whose target qubit is the qubit that originally contained a_i.
6. Repeat the gates applied in Step 3, but in reverse order.
7. Reverse all digits of the solution (qubits that originally contained the digits of B) using CNOT gates controlled by ctr. Such qubits will be $S = A - B$ if $ctr = |1\rangle$, or $S = B$ otherwise.
8. Repeat Step 2 to uncompute such qubits.

3.4. Conditional unequal-bit subtractor

The first subtraction in Algorithm 1 is performed between numbers of equal length. However, the second one is performed between numbers of different lengths. Being A and B the numbers to be subtracted and A the longest number, the operation $A - B$ can be performed using the conditional subtractor described in the previous subsection, simply by equalizing the number of digits of A and B and setting the most significant digits of B to 0 until it reaches the length of A. However, it is possible to make adaptations in the circuit of the previous Subsection to suit this specific case of subtraction with numbers of different lengths so that the T-count and T-depth can be reduced.

To build this second version of the conditional subtractor, it is started in a way similar to the previous subtractor (see Section 3.3) until Step 5 is reached. Any gate or qubit that affects b_{N-1} is omitted as that value will not be encoded. In the first iteration of Step 5 ($i = N-1$), the first Toffoli gate is replaced by a temporary logical-AND gate using the same control qubits but having as target qubit an auxiliary qubit. The CNOT gate of Step 6 acting on b_{N-1} is also omitted, but the CNOT gate indicated in Step 7 that should be applied over b_{N-1} is applied on the auxiliary qubit of the temporary logical-AND gate. An example of the resulting circuit for the case $N = 3$ is shown in Fig. 5.

3.5. Divider circuit

Once circuits have been chosen to perform the comparison and the two subtractions shown in Algorithm 1, the implementation of the divider circuit can begin. In Algorithm 1, the same iteration $N-M+1$ is repeated $N - M + 1$ times, so the goal is to achieve a circuit that allows this iteration to be performed and to repeat it (with the appropriate values in each case) the required number of times. For each iteration i, with $i = 0$ to $N - M$, one must:

- Compare $D_{N-1-i:N-M-i}$ with $Q_{M-1:0}$.
- Compute $D = [D_{N-1-i:N-M-i} - Q_{M-1:0}] \dots D_0$ if the condition $D_{N-1-i:N-M-i} \geq Q_{M-1:0}$ is fulfilled.
- Compute $D = [D_{N-1-i:N-M-i-1} - Q_{M-1:0}] \dots D_0$ otherwise.

In more detail, the following steps must be followed to compute the division D/Q, with N and M the number of digits of D and Q, respectively:

- For $i = 0$ to $N - M$
  1. Prepare a comparator circuit that performs the comparison between $D_{N-1-i:N-M-i}$ and $Q_{M-1:0}$.
  2. Store the inverse result of the comparison in an ancilla qubit and reverse the qubit that contained the result.
  3. Prepare a conditional equal-bit subtractor controlled by the inverted result of the comparator (previously saved in Step 2) that performs the subtraction $D = [D_{N-1-i:N-M-i} - Q_{M-1:0}] \dots D_0$.
  4. Invert the result of the comparator again (reverse it). Apply a temporary logical-AND gate with this value and D_{N-1-i}

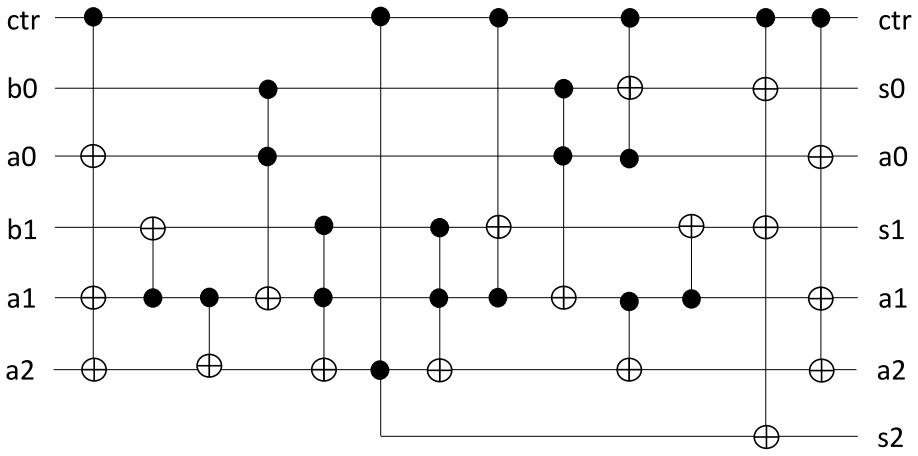

Fig. 5. Proposed modification for the conditional adder of Muñoz-Coreas and Thapliyal (2018). This circuit allows for the subtracting of numbers of different lengths. For the needs of the divider, this circuit performs $a - b$ subtraction, always requiring a to be the longest. *ctr* controls if the subtraction is carried out or not. Again, the result of the subtraction will be stored in the qubits marked s.

as control qubits. Prepare a conditional unequal-bit subtractor to perform the subtraction $D = [D_{N-1-i:N-M-i-1} - Q_{M-1:0}] \dots D_0$, which will be controlled by the target qubit of the previous temporary logical-AND gate.

5. Uncompute the temporary logical-AND gate and the Pauli-X gate applied in the previous step.

Upon completing an iteration, the original value of Q is no longer encoded in any of the qubits. Consequently, it becomes necessary to reintroduce it into the circuit. The practice of reintroducing values throughout a circuit is not uncommon and has been employed in multiple works within the quantum literature (Pérez-Salinas et al., 2020; Orts et al., 2023a). Following the second subtractor, the qubits containing S (the output of the subtractor) must be reset (Shende et al., 2005). Subsequently, Q is re-uploaded using the same approach as at the beginning of the circuit – Pauli-X gates are applied when the corresponding digit is 1, and no gate is applied when the corresponding digit is 0. This process ensures the proper re-encoding of the value Q back into the circuit, facilitating subsequent operations.

An example of the first iteration of the divider circuit is shown in Fig. 6. This example corresponds to a division between two numbers D and Q with 6 and 2 digits, respectively. The qubits marked as *aux* are ancilla qubits. It can be seen how the initial comparison is made between the two digits of Q and the two most significant digits of D. Depending on the outcome of this comparison, the conditional equal-bit subtractor (labeled "Subtractor 1") or the conditional unequal-bit subtractor (labeled "Subtractor 2") is applied. Though not explicitly shown in the figure, the qubits labeled r_i must be reset, and in two of these qubits, Q must be re-uploaded, as explained in the previous paragraph.

4. Analysis and comparison

In this section, we thoroughly examine the proposed divider circuit and its individual components. Next, we conduct a comprehensive comparative analysis, contrasting the proposed circuit with the most efficient state-of-the-art dividers, clearly demonstrating the superiority of our circuit in terms of performance.

As can be seen in Fig. 6, each iteration of the proposed circuit contains a comparator, a controlled subtractor for numbers of equal length, and a controlled subtractor for numbers of different lengths. Therefore, to know the metrics of the iteration, the metrics of these operations are needed. Apart from these three operations, it also needs two temporary logical-AND gates. The rest of the gates that can be seen in Fig. 6 (Pauli-X, CNOT, uncomputation of the temporary logical-AND gate) have a T-count of 0.

The comparator proposed for use in the circuit is the modified version of the comparator proposed by Xia et al. (2019) This modified version (Fig. 3(b)) requires N temporary logical-AND gates. The rest of the gates involved have no cost in terms of T gates. Therefore, it can be easily obtained that the comparator has a T-count of $4N$, as well as a T-depth of $2N$. The circuit needs N ancilla qubits, of which all but the last one containing the result are restored. There are no garbage outputs in the comparator.

As a controlled subtractor for numbers of identical length, the conditional adder proposed by Muñoz-Coreas and Thapliyal (2018) is used (Fig. 4). This adder involves $3N - 2$ Toffoli gates. The remaining gates are CNOT gates, whose T-count is 0. Therefore, the T-count of the circuit is $21N - 14$, and its T-depth is $9N - 6$. The circuit has no ancilla inputs and no garbage outputs.

The modification of the conditional adder of Muñoz-Coreas and Thapliyal proposed in Section 3 is used as a controlled subtractor for numbers with different lengths. This circuit is built with $3N - 2 - K$ Toffoli gates and K temporary logical-AND gates, K being the difference between the length of the two numbers to be subtracted (and N the length of the larger number). Since in the divider circuit, this difference will be 1 (since we always take a part of the dividend that has one more digit than the divisor), it can be assumed that $K = 1$. Therefore, the T-count will be $21N - 17$, and the T-depth will be $9N - 7$. The circuit has no garbage output. It needs one ancilla qubit for the temporary logical-AND gate, but it also implies one less qubit saved from the a_{N-1} encoding. Therefore, the number of qubits remains $2N+1$ as in the original circuit of Muñoz-Coreas and Thapliyal.

For the sake of clarity, the metrics obtained for these components are shown in Table 1. Therefore, the metrics of an iteration of the divider circuit will be defined by the sum of all these values. However, it is necessary to consider them as a single circuit. First, qubits are shared between components. For instance, qubits reversed by the comparator can be used by the scheme. Second, the divider iteration works with two numbers D and Q, of sizes N and M, respectively. The size of the iteration is defined by M, the length of the divisor. Thus, this size of M digits will be used for the comparator and for the first subtractor, while for the second one $M + 1$ will be considered because the next least significant digit of D is added.

For two numbers D and Q of sizes N and M, respectively, the metrics of an iteration will be:

- **T-count**: $4M$ (comparator) $+21M - 14$ (first subtractor) $+21M + 4$ (second subtractor) $+8 = 46M - 2$.
- **T-depth**: $2M$ (comparator) $+9M - 6$ (first subtractor) $+9M + 2$ (second subtractor) $+4 = 20M$.

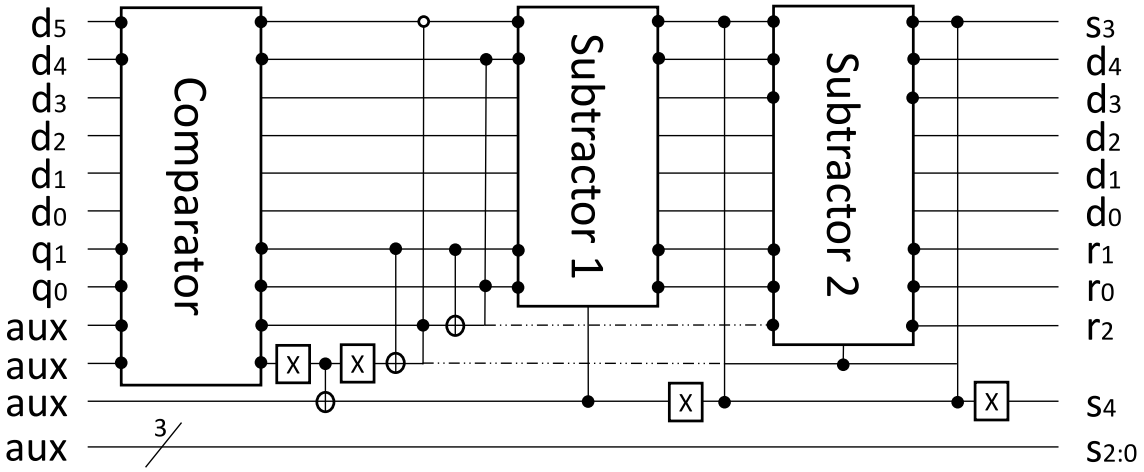

Fig. 6. First iteration of the proposed circuit, for an example of the division between two numbers $D = d_5 \dots d_0$ and $Q = q_1 q_0$. It can be seen that it consists of a comparator (labeled "Comparator"), a controlled subtractor for numbers of equal length ("Subtractor 1"), and a controlled subtractor circuit for numbers of different lengths ("Subtractor 2"). The inputs labeled "aux" are auxiliary qubits. The outputs marked s_i will contain the quotient, while those marked r_i will contain the result of the two subtractions and are, therefore, garbage outputs.

Table 1
T-count, T-depth, and the number of ancilla qubits of each of the operations involved in an iteration of the divider circuit, as shown in Fig. 6. The "Scheme" operation refers to the costs that are part of the scheme itself. None of the operations produce garbage outputs.

Operation	T-count	T-depth	Ancilla qubits
Comparator	$4N$	$2N$	N
Subtractor 1	$21N - 14$	$9N - 6$	0
Subtractor 2	$21N - 17$	$9N - 7$	1
Scheme	8	4	1

- **Ancilla inputs**: The N ancilla inputs of the comparator. These qubits are reversed at the end of the comparison so that all other operations can use them. Moreover, at the end of the iteration, these qubits, except for 2, will be free so that they can be used in the following iterations.
- **Garbage outputs**: 0. The garbage outputs (r in Fig. 6) are reversed at the end of the iteration, as explained in Section 3.

These values correspond to a single iteration. To compute the complete division, $N - M + 1$ iterations will be necessary. Hence, the total T-count of the divider circuit will be $46MN - 46M^2 + 48M - 2N - 2$. Similarly, the T-depth of the complete circuit will be $20MN - 20M^2 + 20M$. Each iteration initially requires N ancilla qubits, but all except 1 are restored. Thus, the circuit necessitates $2N - M + 1$ ancilla qubits. As previously explained in Section 3, the divider is free of garbage output.

Table 2 shows a comparison of the proposed circuit with the most important dividers published in the literature in terms of T-count and T-depth. Notably, the values of the circuit of Yuan et al. (2022) and the proposed circuit depend on both N and M, while the metrics of the other circuits rely solely on N. A straightforward comparison between the proposed circuit and the circuit of Yuan et al. reveals that the proposed circuit achieves lower values of T-count and T-depth, making it a more efficient choice. On the other hand, comparing the values of the proposed circuit with those of Thapliyal et al. (2019) (an improvement on Khosropour et al., 2011) is more intricate. Fig. 7 shows the T-count and T-depth values of the Thapliyal et al. and proposed circuits for N values between 5 and 100. For the proposed circuit, the maximum and minimum values of T-count and T-depth are shown for each value of N, labeled 'Best case (proposed circuit)' and 'Worst case (proposed circuit),' respectively. Depending on the M value, the T-count and T-depth vary within this range of maximum and minimum values. The figure clearly demonstrates that even in the worst case, the proposed circuit outperforms the circuits of Thapliyal et al.

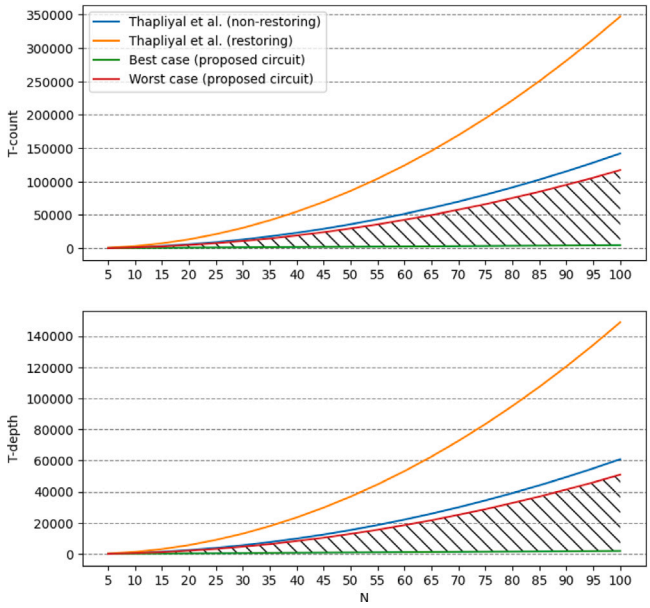

Fig. 7. Comparison, in terms of T-count (top) and T-depth (bottom), and for various digit sizes N, between the dividers of Thapliyal et al. and the proposed circuit. Since the proposed circuit also depends on M, for each case of N the best case and the worst case are shown. The range between these two values comprises all possible values that the proposed circuit can take. The graph has been generated with the Matplotlib library (Hunter, 2007).

On the other hand, Table 3 shows the number of ancilla qubits and garbage outputs. Also, for clarity, it shows which of them are fully reversible and which are not. In terms of the number of ancilla qubits, the most efficient circuit is that of Thapliyal et al. (restoring), requiring N ancilla qubits. In comparison, the proposed circuit necessitates $2N - M + 1$ ancilla qubits. Notably, as the value of M increases, the required number of ancilla qubits decreases. In the worst case, $2N$ ancilla qubits are needed, and in the best case $N + 1$. Regarding the number of garbage outputs, three circuits, including the proposed one, achieve a value of 0, indicating an absence of garbage outputs. Finally, it is important to note that both the proposed circuit and the circuit of Yuan et al. involve reset operations. As a result, they may not be fully compatible with certain quantum algorithms, such as Grover's, which

Table 2
Comparison, in terms of the number of T-count and T-depth, between state-of-the-art and proposed circuits.

Circuit	T-count	T-depth
Khosropour et al. (2011)	$\approx 400N^2$	$\approx 170N^2$
Thapliyal et al. (2019) (non-restoring)	$14N^2 + 21N - 28$	$6N^2 + 9N - 4$
Thapliyal et al. (2019) (restoring)	$35N^2 - 28N$	$15N^2 - 12N$
Yuan et al. (2022)	$126MN - 126M^2 + 133M - 7N - 7$	$54MN - 54M^2 + 57M - 3N - 3$
Proposed	$46MN - 46M^2 + 48M - 2N - 2$	$20MN - 20M^2 + 20M$

Table 3
Comparison of ancilla qubits, number of garbage outputs, and reversibility, between state-of-the-art and proposed circuits.

Circuit	Ancilla qubits	Garbage outputs	Fully reversible
Khosropour et al. (2011)	$2N$	$\geq N + 1$	Yes
Thapliyal et al. (2019) (non-restoring)	$2N + 1$	$N + 1$	Yes
Thapliyal et al. (2019) (restoring)	N	0	Yes
Yuan et al. (2022)	$3N - 2M + 2$	0	No
Proposed	$2N - M + 1$	0	No

require continuous recovery of the initial values of the circuit assigned as an oracle.

5. Conclusions

In this work, a circuit for computing the integer division in quantum computing has been presented. The circuit is based on the possibility of adapting the size of successive subtractions involved in the algorithm known as "long division" in such a way that the required number of iterations is reduced relative to other published circuits, with the consequent saving of resources.

In order to build the divider circuit, it was necessary to build a comparator, a conditional subtractor for numbers of equal lengths, and a conditional subtractor for numbers of different lengths. For these computations, we utilized the best circuits currently available in the literature, customizing and optimizing them for our specific division implementation. As a result, we achieved improvements in terms of T-count and T-depth in these subcircuits.

Furthermore, we performed a thorough comparison between the proposed circuit and those available in the state-of-the-art, demonstrating the benefits and limitations of the divider circuit. The proposed circuit outperformed existing circuits in the literature, reducing T-count and T-depth, while keeping the number of ancilla qubits within the typical range for such circuits. Additionally, the circuit is free of garbage output, ensuring that after completion, it will only occupy the number of qubits strictly necessary to store the result.

In conclusion, our work contributes a novel and resource-efficient approach to quantum integer division, showcasing the potential for practical implementation of this operation in quantum computing.

CRediT authorship contribution statement

Francisco Orts: Investigation, Software, Validation, Writing – original draft, Writing – review & editing. **Remigijus Paulavičius:** Funding acquisition, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Ernestas Filatovas:** Formal analysis, Investigation, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Francisco Orts reports financial support was provided by Research Council of Lithuania.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research has received funding from the Research Council of Lithuania under the Program "University Excellence Initiatives" of the Ministry of Education, Science and Sports of the Republic of Lithuania (Measure No. 12-001-01-01-01 "Improving the Research and Study Environment"). Project No.: S-A-UEI-23-11.

References

Amy, M., Maslov, D., Mosca, M., Roetteler, M., 2013. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 32 (6), 818–830.

Amy, M., Mosca, M., 2019. T-count optimization and Reed–Muller codes. IEEE Trans. Inform. Theory 65 (8), 4771–4784.

Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H., 1995. Elementary gates for quantum computation. Phys. Rev. A 52 (5), 3457.

Bennett, C.H., 1973. Logical reversibility of computation. IBM J. Res. Dev. 17 (6), 525–532.

Bernhardt, C., 2019. Quantum Computing for Everyone. MIT Press.

Bocharov, A., Roetteler, M., Svore, K.M., 2015. Efficient synthesis of universal repeat-until-success quantum circuits. Phys. Rev. Lett. 114 (8), 080502.

Bravyi, S., Gosset, D., 2016. Improved classical simulation of quantum circuits dominated by Clifford gates. Phys. Rev. Lett. 116 (25), 250501.

Combarro, E.A., Gonzalez-Castillo, S., 2023. A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-On Primer to Quantum Computing: from Qubits to Quantum Machine Learning and Beyond. Vol. 1, Packt Publishing.

Endo, S., Benjamin, S.C., Li, Y., 2018. Practical quantum error mitigation for near-future applications. Phys. Rev. X 8 (3), 031027.

Endo, S., Cai, Z., Benjamin, S.C., Yuan, X., 2021. Hybrid quantum-classical algorithms and quantum error mitigation. J. Phys. Soc. Japan 90 (3), 032001.

Giani, A., Eldredge, Z., 2021. Quantum computing opportunities in renewable energy. SN Comput. Sci. 2 (5), 393.

Gidney, C., 2018. Halving the cost of quantum addition. Quantum 2, 74.

Gosset, D., Kliuchnikov, V., Mosca, M., Russo, V., 2013. An algorithm for the T-count. arXiv preprint arXiv:1308.4134.

Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212–219.

Gyongyosi, L., Imre, S., 2019. Quantum circuit design for objective function maximization in gate-model quantum computers. Quantum Inf. Process. 18 (7), 1–33.

Hallgren, S., 2007. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. J. ACM 54 (1), 1–19.

Hennessy, J.L., Patterson, D.A., 2011. Computer Architecture: a Quantitative Approach. Elsevier.

Heyfron, L.E., Campbell, E.T., 2018. An efficient quantum compiler that reduces T count. Quant. Sci. Technol. 4 (1), 015004.

Houssein, E.H., Mahdy, M.A., Eldin, M.G., Shebl, D., Mohamed, W.M., Abdel-Aty, M., 2021. Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm. J. Adv. Res. 29, 147–157.

Humble, T.S., Thapliyal, H., Munoz-Coreas, E., Mohiyaddin, F.A., Bennink, R.S., 2019. Quantum computing circuits and devices. IEEE Des. Test 36 (3), 69–94.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 9 (3), 90–95.

Khalid, U., ur Rehman, J., Paing, S.N., Jung, H., Duong, T.Q., Shin, H., 2023. Quantum network engineering in the NISQ age: Principles, missions, and challenges. IEEE Netw..

Khosropour, A., Aghababa, H., Forouzandeh, B., 2011. Quantum division circuit based on restoring division algorithm. In: 2011 Eighth International Conference on Information Technology: New Generations. IEEE, pp. 1037–1040.

Kissinger, A., van de Wetering, J., 2019. Reducing T-count with the ZX-calculus. arXiv preprint arXiv:1903.10477.

Kissinger, A., van de Wetering, J., Vilmart, R., 2022. Classical simulation of quantum circuits with partial and graphical stabiliser decompositions. arXiv preprint arXiv:2202.09202.

Li, H.-S., Fan, P., Xia, H., Peng, H., Long, G.-L., 2020a. Efficient quantum arithmetic operation circuits for quantum image processing. Sci. China Phys. Mech. Astron. 63, 1–13.

Li, H.-S., Fan, P., Xia, H.-Y., Peng, H., Long, G.-L., 2020b. Efficient quantum arithmetic operation circuits for quantum image processing. Sci. China Phys. Mech. Astron. 63, 1–13.

Litinski, D., 2019. Magic state distillation: Not as costly as you think. Quantum 3, 205.

López, L.O., Orts, F., Ortega, G., González-Ruiz, V., Garzón, E.M., 2023. Fault-tolerant quantum algorithm for dual-threshold image segmentation. J. Supercomput. 1–14.

Mohammadi, M., Eshghi, M., 2009. On figures of merit in reversible and quantum logic designs. Quantum Inf. Process. 8, 297–318.

Muñoz-Coreas, E., Thapliyal, H., 2018. Quantum circuit design of a T-count optimized integer multiplier. IEEE Trans. Comput. 68 (5), 729–739.

Nielsen, M.A., Chuang, I.L., 2011. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press.

Orts, F., Filatovas, E., Garzón, E.M., Ortega, G., 2023a. A quantum circuit to generate random numbers within a specific interval. EPJ Quant. Technol. 10 (1), 17.

Orts, F., Ortega, G., Combarro, E.F., Garzón, E.M., 2020. A review on reversible quantum adders. J. Netw. Comput. Appl. 170, 102810.

Orts, F., Ortega, G., Cucura, A., Filatovas, E., Garzón, E., 2021. Optimal fault-tolerant quantum comparators for image binarization. J. Supercomput. 77 (8), 8433–8444.

Orts, F., Ortega, G., Garzón, E.M., 2019. A faster half subtractor circuit using reversible quantum gates. Balt. J. Mod. Comput. 7 (1), 99–111.

Orts, F., Paulavičius, R., Filatovas, E., 2023b. Improving the implementation of quantum blockchain based on hypergraphs. Quantum Inf. Process. 22 (9), 330.

Paler, A., Polian, I., Nemoto, K., Devitt, S.J., 2017. Fault-tolerant, high-level quantum circuits: form, compilation and description. Quant. Sci. Technol. 2 (2), 025003.

Pauli, W., 1988. Zur Quantenmechanik des Magnetischen Elektrons. Springer.

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., Latorre, J.I., 2020. Data re-uploading for a universal quantum classifier. Quantum 4, 226.

Preskill, J., 2018. Quantum computing in the NISQ era and beyond. Quantum 2, 79.

Romero, J., Aspuru-Guzik, A., 2021. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. Adv. Quant. Technol. 4 (1), 2000003.

Selinger, P., 2013. Quantum circuits of T-depth one. Phys. Rev. A 87 (4), 042302.

Shende, V.V., Bullock, S.S., Markov, I.L., 2005. Synthesis of quantum logic circuits. In: Proceedings of the 2005 Asia and South Pacific Design Automation Conference. pp. 272–275.

Shor, P.W., 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. 41 (2), 303–332.

Thapliyal, H., Munoz-Coreas, E., Varun, T., Humble, T.S., 2019. Quantum circuit designs of integer division optimizing T-count and T-depth. IEEE Trans. Emerg. Top. Comput. 9 (2), 1045–1056.

Thapliyal, H., Ranganathan, N., 2011. A new design of the reversible subtractor circuit. In: 2011 11th IEEE International Conference on Nanotechnology. IEEE, pp. 1430–1435.

Thomsen, M.K., Glück, R., Axelsen, H.B., 2010. Reversible arithmetic logic unit for quantum arithmetic. J. Phys. A 43 (38), 382002.

Toffoli, T., 1980. Reversible computing. In: Automata, Languages and Programming: Seventh Colloquium Noordwijkerhout, the Netherlands July 14–18, 1980 7. Springer, pp. 632–644.

Van Dam, W., Hallgren, S., 2000. Efficient quantum algorithms for shifted quadratic character problems. arXiv preprint quant-ph/0011067.

Van Dam, W., Hallgren, S., Ip, L., 2006. Quantum algorithms for some hidden shift problems. SIAM J. Comput. 36 (3), 763–778.

Wang, S., Wang, Z., Li, W., Fan, L., Cui, G., Wei, Z., Gu, Y., 2020. Quantum circuits design for evaluating transcendental functions based on a function-value binary expansion method. Quantum Inf. Process. 19 (10), 1–31.

Wei, A.Y., Naik, P., Harrow, A.W., Thaler, J., 2020. Quantum algorithms for jet clustering. Phys. Rev. D 101 (9), 094015.

Xia, H.-Y., Li, H., Zhang, H., Liang, Y., Xin, J., 2018. An efficient design of reversible multi-bit quantum comparator via only a single ancillary bit. Internat. J. Theoret. Phys. 57 (12), 3727–3744.

Xia, H., Li, H., Zhang, H., Liang, Y., Xin, J., 2019. Novel multi-bit quantum comparators and their application in image binarization. Quantum Inf. Process. 18, 1–17.

Xia, H.-Y., Zhang, H., Song, S.-X., Li, H., Zhou, Y.-J., Chen, X., 2020. Design and simulation of quantum image binarization using quantum comparator. Modern Phys. Lett. A 35 (09), 2050049.

Yuan, S., Gao, S., Wen, C., Wang, Y., Qu, H., Wang, Y., 2022. A novel fault-tolerant quantum divider and its simulation. Quantum Inf. Process. 21 (5), 182.

Zahedinejad, E., Ghosh, J., Sanders, B.C., 2015. High-fidelity single-shot toffoli gate via quantum control. Phys. Rev. Lett. 114 (20), 200502.

Zhang, S.-X., Allcock, J., Wan, Z.-Q., Liu, S., Sun, J., Yu, H., Yang, X.-H., Qiu, J., Ye, Z., Chen, Y.-Q., et al., 2023. Tensorcircuit: a quantum software framework for the nisq era. Quantum 7, 912.

Zhou, R., Wan, C., 2021. Quantum image scaling based on bilinear interpolation with decimals scaling ratio. Internat. J. Theoret. Phys. 60 (6), 2115–2144.

Francisco Orts is a senior researcher at the Vilnius University, Lithuania. Additionally, he is a Ph.D. Professor at the International University of La Rioja, and a collaborating professor at the Universitat Oberta de Catalunya, both universities in Spain. He also collaborates actively with the Supercomputing-Algorithms group at the University of Almeria, Spain. He has worked as a Computer Engineer in construction, stock market and IT services companies, with more than 15 years of experience in the sector.

Remigijus Paulavičius received a Ph.D. degree in computer science from Vytautas Magnus University, Kaunas, Lithuania, in 2010. He was a Postdoctoral Researcher at Vilnius University, Vilnius, Lithuania, and a Research Associate at Imperial College London, London, UK. He is currently a Professor at Vilnius University. His research interests include global optimization, optimization software, parallel and quantum computing, and distributed ledger technologies.

Ernestas Filatovas received a Ph.D. degree in Informatics Engineering from Vilnius University, Lithuania, in 2012. He is currently a senior researcher and co-founder of the Blockchain Technologies Group at the Institute of Data Science and Digital Technologies, Vilnius University. His main research interests include blockchain and distributed ledger technologies, parallel and quantum computing, global and multiobjective optimization, and machine learning.