



Optimized quantum Leading Zero Detector circuits

Francisco Orts^{a,*}, Gloria Ortega^a, Elías F. Combarro^b, Ignacio F. Rúa^c, Ester M. Garzón^a

^a*Informatics Department, University of Almería, Almería, Spain*

^b*Informatics Department, University of Oviedo, Oviedo, Spain*

^c*Mathematics Department, University of Oviedo, Oviedo, Spain*

Abstract

The algorithms that best demonstrate the potential of quantum computing are Shor's algorithm and Grover's algorithm. To this day, new evidence continues to emerge in the form of algorithms or ingenious applications that increase the field of application of this type of computing. However, given the limited number of qubits in current quantum computers, and also the noise problems they currently suffer from, implementing optimized circuits that allow us to take full advantage of the available resources, as well as detecting and correcting the errors caused by this noise, is a priority. In this work we present [several](#) Leading Zero Detector circuits for quantum computers and simulators, optimized in terms of noise tolerance and number of qubits. These circuits are a fundamental part in major circuits that perform operations as important and basic in computation as addition and division.

© 2011 Published by Elsevier Ltd.

Keywords:

Quantum circuit, Quantum Computing, Quantum arithmetic, Leading Zero Detector, LZD

1. Introduction

In quantum computing, small circuits that need few qubits and few operations are especially useful, even when they do not offer any quantum behavior. This is because they can be used as part of larger circuits that perform quantum operations of interest [1]. One advantage of the small size of these circuits is that today's quantum computers have a limited amount of resources, but these will be sufficient if the circuit has no major requirements. Although this amount of resources is increasing, a bad use of them will lead to a rapid consumption of resources [2]. On the other hand, quantum simulators are computationally expensive, so the lower the number of operations, the lower the requirements to implement a circuit in such simulators and therefore the lower the hardware requirements [3, 4]. A third fundamental reason to go for reduced circuit size is that current quantum computers are greatly affected by internal and external noise. Every qubit and every extra operation of a circuit increases the chances that noise will eventually affect the circuit's output [5].

Perhaps the clearest example of circuits designed to be components of other circuits is shown by Grover's algorithm [6]. This algorithm allows to find, in a set of disordered elements, one or more elements that meet a certain condition [7]. The interesting thing about Grover's algorithm is that it allows to solve this problem by performing fewer checks of the mentioned condition, but what interests us here is that this algorithm has two parts: one fixed, relative to increase the probability of occurrence of the desired result; and another totally dependent on the problem

*Corresponding author: Francisco Orts. Carretera de Sacramento, s/n, 04120 La Cañada de San Urbano, Almería, Spain. francisco.orts@ual.es

we are trying to solve. This second part, commonly called oracle, must check whether or not an element meets the condition of the problem [8]. Optimally implementing this oracle is fundamental, since in case of a bad implementation we have the risk of not only losing the advantage offered by Grover’s algorithm, but also of having a worse performance than a classical search due to the large number of operations added in the oracle [9].

From what is stated in the previous paragraph, we can understand the need for small and optimized circuits. However, reducing the number of operations and qubits is not enough; we still have the problem of noise. Nowadays it is common to use a specific gate called T gate, which allows the use of error detection codes and therefore mitigate the effects caused by noise. While T-gate offer this advantage, it also has its problems: it is much more expensive than other gates. In fact, the T gate exceeds the cost of the Clifford gates by as much as a factor of a hundred or even more [11]. That is why the goodness of quantum circuits is measured in terms of T gates, namely the number of T gates needed to implement the circuit (which would mark the cost of the circuit), and the number of T gates to be executed sequentially in the circuit (which would mark the speed or slowness of the circuit) [12, 13].

In this paper we focus on an operation called Leading Zero Detector (LZD). **In particular, three circuits are proposed to perform this operation with 8-digit numbers. To build such circuits, three circuits that work with 4-digit numbers are also proposed.** This operation is an essential part of the normalization process in classical floating point arithmetic units (we use the term classical to indicate non-quantum). Basically, it consists of performing a left shift until the first nonzero digit reaches the left-most position. At the same time, to complete the floating point operation the exponents are decremented according to the shift amount [14]. This operation, so fundamental in classical computing, is currently receiving a great deal of interest in quantum computing because it is involved in numerous larger operations and algorithms of interest [15, 16, 17, 10, 18, 19]. Among the applications that use an LZD circuit and are of interest in quantum computing, we can mention the arithmetic operations of addition and division.

In the case of addition, it is no exaggeration to say that it is possibly the operation whose optimization is currently of greatest interest in quantum computing [20]. This is mainly due to the fact that Shor’s famous algorithm [21] (the quantum algorithm with the best performance results) uses these circuits in its implementation. However, the importance of adders is not limited to this algorithm; there are a large number of quantum circuits that use them [22, 23, 24, 25, 26]. Fig. 1 shows an algorithm for the addition of two floating-point numbers presented by Nguyen et al [10]. This algorithm performs the addition between two 32-bit IEEE-754 single-precision floating-point numbers A and B . The IEEE-754 format involves a sign bit, 8 bits for the exponent, and the remaining 23 bits for the mantissa. A conditional swap is executed first to determine which number is the largest and which is the smallest, so that, at its output, X will be the largest, and Y the smallest. Before adding A and B (now labelled as X and Y) the number must be aligned. Therefore, if their exponents are not equal, the exponent of the smaller number increases until it reaches the exponent of the larger number, while shifting its mantissa to the right. Once the sum between the aligned numbers has been performed, the result is normalised and rounded. As an example, the result of adding $A = 01000001010001100110011001100110$ (12.4) and $B = 01000000110001100110011001100110$ (6.2) will produce the output $01000001100101001100110011001100$ (18.599998). In reality, at its output, the circuit will have 32 qubits containing $X + Y$ (the aforementioned result), another 32 qubits occupied by the digits of X , and several garbage outputs (the meaning of a garbage output is introduced later) resulting from normalisation and other intermediate operations.

The reader will notice that the previous algorithm involves, as mentioned above, a normalization process, which for the sake of clarity is shown in detail in Fig. 2. Normalising a number allows the integer part of the mantissa to be expressed as 1, and the fractional part as the remainder of the number. As an example (and using a non-IEEE-754 small number for clarity), the number 1101.01 (positive) would be normalised as 1.10101. Of course, the exponent will reflect this change (in the example, since the dot has been moved 3 positions, we would have an exponent of 2^3 , and a sign of 0 since we have mentioned that the value in the example is positive). During the process, several garbage outputs are generated, labelled in Fig. 2 with the letter G . As part of this normalization several operations are performed. Among such operations, there is an LZD circuit that performs the operation we are focusing on. Optimizing this small part will optimize the whole summation process.

The division operation is not as interesting as addition, but it is still involved in major circuits of scientific interest [27, 28, 29]. However, precisely because division circuits have not received as much interest as adders in quantum computing, current division circuits are not as optimized as adders. Therefore, a full or partial optimization of division circuits could be especially noteworthy in this family of circuits. As an example, Fig. 3 shows the schematic of a division circuit proposed in Gayathri et al. [16] in 2021. This is one of the most recent cir-

cuits in the literature at the time of writing these lines. Using the same values as in the addition example, i.e. $A = 01000001010001100110011001100110$ (12.4) and $B = 01000000110001100110011001100110$ (6.2), this circuit will return $A/B = 010000000000000000000000000000$ (2). The circuit output will be a combination of uncomputed qubits, some garbage outputs, and the result itself, as shown in Fig. 3. It is also clear from that figure that an LZD circuit is necessary for the implementation of the division. As in the case of addition, an optimization of this circuit will lead to an improvement of the whole assembly.

Having demonstrated the usefulness of LZD circuits in quantum computing and the benefits of optimizing them, the rest of the paper is organized as follows: In Section 2, we outline the metrics used in this work to allow us to measure and compare quantum circuits. These metrics, as we have discussed, are related to the number of T gates involved in the implementation of the circuits, as well as the number of required qubits. In Section 3 we present the methodology of our proposed LZD circuits. All three circuits perform the same operation, but the differences between their implementations allow us to choose the most suitable one depending on the metric being considered as the main one. Section 4 compares our proposed circuits with the best LZD circuits available in the state of the art. Finally, the last section presents the conclusions.

2. Quantum circuits

Since the priority of this work is to provide optimized circuits for their noise tolerance, it is necessary to define appropriate metrics for this purpose. We discussed in the previous section the fact that qubits in today's quantum computers are severely affected by internal and external noise. A first metric whose definition is trivial is the number of required qubits. The fewer the qubits, the lower the probability of error due to noise and the easier its detection and/or correction. We are talking about qubits that allow the target operation to be performed, but without including qubits dedicated to, precisely, error detection or correction functions.

We have also stated that, as a rule, fewer operations implies less exposure to noise. However, since there are an infinite number of possible operations, and a diverse number of possible ways to implement those operations, we cannot make a direct definition of the number of operations as a metric. In Mohammadi et al. [30] a set of metrics called quantum cost, delay, number of ancilla qubits, and number of garbage outputs are defined. These metrics are used in a large number of works [31, 32, 33, 34]. Quantum cost allows us to quantify the cost of an operation numerically. Knowing that the operations in quantum computing correspond to orthogonal matrices, the quantum cost defines as 1 the cost of any 2×2 or 4×4 matrix (that is, any operation that affects only one or two qubits). Consequently, all other operations will have as quantum cost the number of operations of cost 1 necessary for their computation.

Following Mohammadi et al. metrics, the delay is a metric oriented to define the speed of the circuit. The delay of any 2×2 or 4×4 matrix is defined as 1Δ . As before, the speed of a circuit is recursively defined as the sum of the delays of the operations that are part of the critical path of the circuit. The number of ancilla qubits indicates the number of qubits used to perform auxiliary operations. The number of garbage outputs specifies the number of qubits that, after circuit execution is complete, contain indeterminate values that are not part of the valid output of the circuit.

A first look at the metrics of Mohammadi et al. shows us that their number of ancilla qubits is closely related to our number of qubits. Since all circuits evaluated in this work must have the same number of inputs (always 8, not counting qubits dedicated to auxiliary operations), it is trivial to say that the number of inputs will always be at least 8 (predictably more, for reasons of maintaining reversibility). Similarly, if a circuit has fewer outputs than another, it will be because it has fewer ancilla inputs. Therefore, we consider that the use of the number of ancilla qubits instead of the number of total qubits allows us to compare circuits focusing only on the part that is suitable to be optimized.

The concepts of delay and quantum cost are useful to get an approximate idea of the size and speed of a circuit. At least, at a theoretical level. However, we indicated in the previous section that the quantum gate T has a much higher cost than the rest of the gates. The cost of the T-gate is such that it can make other gates insignificant. This is why the concepts of quantum cost and delay are not so useful when the T gate is part of the circuits. Instead, these two metrics are usually redefined as T-count and T-depth, which indicate the number of T gates in a circuit, and the number of T gates that are part of the critical path of that circuit, respectively. The concepts of T-count and T-depth are frequently used as reliable metrics of quantum circuits [12, 35, 16].

In relation to the number of garbage outputs, it is considered good practice to revert such garbage outputs to their original value. All the circuits evaluated in this paper are free of garbage outputs.

In the classical world only two gate are acting on 1 bit. In the quantum world, infinite gate are acting on 1 qubit. Extending this result, we quickly come to determine that we have infinite possibilities whatever the number of qubits we are dealing with. Trying to measure all these infinite gates in order to simplify the subsequent task of measuring and comparing our circuits would be unaffordable. Fortunately, only 5 quantum gates are used in the circuits studied in this work: the Pauli-X, the CNOT, the Toffoli gate, the temporary logical-AND, and the uncomputation of the temporary logical-AND gate.

- Pauli-X gate: this gate changes the probability amplitudes of the basis. Its implementation does not require any T gate, so its T-count and T-depth are 0.
- CNOT gate: although the CNOT gate has very interesting quantum implications, LZD circuits work only with the standard basis. Therefore, we can simplify the definition of this gate to say that it is similar to the previous one, but with a control qubit. This gate also does not use T-gates in its implementation.
- Toffoli gate: changes the probability amplitudes of the basis in the target qubit if and only if the two control qubits are active. The Toffoli gate has a T-count of 7, and a T-depth of 3 [36]. To uncompute its effects (to avoid garbage outputs), another Toffoli gate is necessary.
- Temporary logical-AND: this gate performs the same operation as the Toffoli gate, but with a T-count of 4 and a T-depth of 2 [35]. However, this gate requires the target qubit to be in the particular initial state $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle)$, so it cannot be applied over a qubit that is already being used.
- Uncomputation of the temporary logical-AND gate: this gate allows to uncompute a temporary logical-AND gate in order to avoid garbage outputs. This gate does not involve T gates.

Their symbols are shown in Fig. 4. These symbols will be used in the designs exposed in the next section to represent each gate. For the sake of clarity, we also show the implementation of the Toffoli gate and the temporary logical-AND gate in Fig. 5. The first three gates (Pauli-X, CNOT, and Toffoli gate) are widely mentioned and used in the literature on quantum computing. However, the temporary logical-AND gate is more recent and its use is focused on scientific works on quantum circuit optimisation. As the proper use of the temporary logical-AND gate is fundamental to the circuits proposed in this work, detailed information on the operation of this gate is given below.

2.1. Temporary logical-AND gate

The temporary logical-AND gate was introduced by Gidney in 2018 to reduce the T-count of the arithmetic sum [35]. Since its appearance, several scientific works have used it to reduce the T-count of their quantum circuits [2, 12].

The temporary logical-AND gate performs the same operation as the Toffoli gate, as shown in Table 1. The main differences of the temporary logical-AND gate with respect to the Toffoli gate are two. The first difference is that the temporary logical-AND gate has to act on a specifically prepared qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle)$. That is, while a Toffoli gate can be applied on a qubit already in use, to apply the temporary logical-AND gate it is always necessary to resort to a qubit already prepared in the mentioned state. The second difference consists in their implementations and the T-count associated with these implementations, as can be seen in Fig. 5. The temporary logical-AND gate has a T-count of 4, in contrast to the T-count of the Toffoli, which is 7.

It has been mentioned that it is good practice to reverse the garbage outputs of circuits. To uncompute a Toffoli gate, it is necessary another Toffoli gate. Using a second Toffoli gate to uncompute the operation of a first one will increase the T-count by 7. However, the temporary logical-AND gate can be reversed using a measure-and-fixup approach [35, 37]. This approach is shown, in circuit form, in Fig. 5. As can be seen in the figure, this circuit has a T-count of 0 as no T gates are involved in its implementation. Therefore, the temporary logical-AND gate not only improves the T-count of the Toffoli gate in the operation itself, but also in its uncomputation.

To understand how the temporary logical-AND gate works, it is necessary to go back to Jones' 2013 work [37]. In that paper, Jones presented an implementation of the Toffoli gate that used only 4 T gates, in contrast to the 7 used in the implementation proposed in [36] (shown in Fig. 5). To construct his Toffoli gate, Jones made use of a gate called

Table 1. Truth table of the temporary logical-AND gate. A, B and C are the inputs, and A', B' and C' the outputs. A and B are the control qubits, and C the target one (C must be prepared in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$). A' and B' will contain the same value as A and B , whereas C' will contain the result of the operation A AND B .

A	B	C	A'	B'	C'
$ 0\rangle$	$ 0\rangle$	$\frac{1}{\sqrt{2}}(0\rangle + e^{\frac{i\pi}{4}} 1\rangle)$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$\frac{1}{\sqrt{2}}(0\rangle + e^{\frac{i\pi}{4}} 1\rangle)$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$
$ 1\rangle$	$ 0\rangle$	$\frac{1}{\sqrt{2}}(0\rangle + e^{\frac{i\pi}{4}} 1\rangle)$	$ 1\rangle$	$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$ 1\rangle$	$\frac{1}{\sqrt{2}}(0\rangle + e^{\frac{i\pi}{4}} 1\rangle)$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$

iX presented in an earlier work [38]. The decomposition of this version of the Toffoli gate and the iX gate in basic gates [39, 40] are shown in Fig. 6(a) and Fig. 6(b), respectively. The iX gate performs the following operation:

$$iX|q_0\rangle|q_1\rangle|y\rangle = i^{q_0q_1}|q_0\rangle|q_1\rangle|y \oplus (q_0q_1)\rangle \quad (1)$$

being q_0 and q_1 control qubits, and y the target one. Jones' contribution consisted of correcting the phase of the iX gate using an S' gate. In particular, he uses the iX gate on an ancilla qubit:

$$iX|q_0\rangle|q_1\rangle|0\rangle = i^{q_0q_1}|q_0\rangle|q_1\rangle|(q_0q_1)\rangle \quad (2)$$

and then corrects the phase by applying the S' gate to the ancilla qubit. Once this value is copied to the target register, it is necessary to uncompute the state q_0q_1 . However, Jones proposed to use a Hadamard gate:

$$H|q_0q_1\rangle = \frac{1}{\sqrt{2}} \sum_{z \in \mathbb{Z}_2} (-1)^{q_0q_1z} |z\rangle \quad (3)$$

A measurement is then taken and the phase is corrected (which will be 1 if the measurement is 0, and $(-1)^{q_0q_1}$ if it is 1) with a controlled-Z gate. With this approach, it is not necessary to apply another Toffoli gate to avoid garbage outputs. From this point, Gidney proposed to delay the uncomputation of the product q_0q_1 , so he separated Jones' original circuit into two parts: the temporary logical-AND gate (Fig. 6(c)), and its uncomputation (Fig. 6(d)). This reduces the number of qubits needed to perform the operation by 1, with a T-count of 4 in the first circuit, and 0 in the second [41].

3. Proposed Quantum LZD circuits

For the design of our circuits, we started from the LZD circuit designed by Gayathri et al. in 2021 [16], which in turn is based on a previous work by Oklobdzija [14]. For the sake of clarity, we show in Fig. 7 the reproduction of the Gayathri et al. circuit for the 4-digit case. The circuit design (like ours) is focused on achieving an 8-digit operation, but the 8-digit circuit is composed of two equal 4-digit subcircuits as shown in Fig. 7, plus a Toffoli gate and two CNOT gates. We will detail these operations later. Focusing now on the 4-digit case, the input value is specified by the variables X_3, X_2, X_1, X_0 , with X_3 being the most significant digit and X_0 the least significant one. For its part, the variables Q_1, Q_0 compute the leading zero count, while the variable V will be set to 1 if and only if the input variable X_3, X_2, X_1, X_0 is composed only of zeros.

The output variables Q_1, Q_0 and V are implemented knowing that their values are governed by the following equations:

$$\overline{X_3}X_2 + \overline{X_3}\overline{X_1} = Q_0 \quad (4)$$

$$X_3 + X_2 = Q_1 \quad (5)$$

$$X_3 + X_2 + X_1 + X_0 = V \quad (6)$$

The 8-digit circuit is divided, as previously expressed, into two parts of 4 digits each, in order to evaluate the number of leading zeros. In this way, we will obtain the outputs V_i and Q_i corresponding to each of the two parts. In each of the subcircuits, V_i will be the logical sum as the complement of that part, while Q_1, Q_0 will represent the number of leading zeros. Finally, V will be obtained as the addition of V_1 and V_2 . To compute this last operation, the extra Toffoli gate and the two CNOT gates we mentioned early are used. The total circuit is composed of 4 Pauli-X gates, 28 CNOT gates, 19 Toffoli gates and 9 auxiliary qubits. It should be added that in the original paper they claim to have 2 more auxiliary qubits than those indicated here (that is, 11 ancilla qubits). This is because they use those last two qubits to copy the values of V_1 and V_2 to use these qubits to generate the output $VQ'_2Q'_1Q'_0$, used in another circuit. Since in this work we are focusing only on obtaining the Q_i and V values, it will not be necessary to consider these operations.

The circuit in Fig. 7 offers several optimization possibilities. Each of the 8 Toffoli gates (19, in the 8-digit version) increases the T-count by up to 7, and the T-depth by 3. Since the two 4-digit circuits can be run in parallel, it is only necessary to consider the T-depth of one of them in the 8-digit case. However, the T-count is still large. Replacing the Toffoli gates with temporary logical-AND gates will result in a saving of 3 in the T-count per gate replaced. Better yet, the Toffoli gates used to uncompute garbage outputs can be replaced by the uncomputation gate of the temporary logical-and gate, which has no cost in terms of T-count or T-depth. However, this drastic reduction of T gates (and thus T-count and T-depth) comes from dedicating an exclusive qubit to each of the temporary logical-AND gates used. On the other hand, there are more optimized implementations for the operations of equations 1, 2 and 3 than the used in Fig. 7. In fact, we can obtain the minimum cost of a (small) function with the help of a SAT solver [42]. In this way we obtain the circuit in Fig. 8, which we will call circuit 1 to differentiate it from the following circuits.

Circuit 1 improves in T-count and T-depth to the circuit of Gayathri et al. However, as we have already mentioned, the need of an extra qubit for each temporary logical-AND gate makes circuit 1 use a larger number of ancilla qubits. Nevertheless, circuit 1 still admits some improvements even though the SAT solver gave us the optimal implementation of the Q_0, Q_1 and V functions. These optimizations are achieved by considering the circuit as a whole, and not as three separate functions. In Fig. 8, the first temporary logical-AND gate implements X'_3X_2 as part of Q_0 . The circuit uses this value as a control in other gates. After this, the circuit uncomputes X'_3X_2 . However, later on, another temporary logical-AND gate is used to implement the value X_3X_2 in a new qubit as part of an auxiliary operation to compute Q_1 and V . By a simple logical operation we can convert the value X'_3X_2 into X_3X_2 . In this way, instead of uncomputing X'_3X_2 , we can reuse the qubit to contain X_3X_2 in it, saving us an extra qubit and a temporary logical-AND gate (in addition to the uncomputation gate of the first temporary logical-and gate). If in this new circuit we now apply a simplification to eliminate redundant operations, we will obtain the circuit shown in Fig. 9. We will label this circuit as circuit 2.

Circuit 2 further reduces the T-count and T-depth of circuit 1. It also reduces the number of ancilla qubits by two (one qubit per 4-digit subcircuit). However, it still requires more qubits than the original circuit of Gayathri et al. In an environment where the priority is to reduce the number of T-gates or at least to reduce some of the 2 metrics associated with those gates, circuit 2 will be the ideal choice. However, if the priority were to minimize the required number of qubits, circuit 2 can be modified so that it requires fewer qubits at the cost of sacrificing the gains achieved in T-count and T-depth. To do this, simply replace the second logical-AND temporary and its uncomputation gate with a Toffoli gate (each), so that its target qubit will be decomputed after its application and can be used, later, by another Toffoli gate as part of the computation process of V . In this way, and despite the increase of T gates, we reduce the number of ancilla qubits by 1 in each 4-digit subcircuit. We will call this circuit ‘‘Circuit 3’’. It is shown in Fig. 10.

As we have said from the beginning, two of these circuits are needed to build the 8-digit circuit. The complete 8-digit case is shown in Fig. 11. Circuit 2 is used in the figure to implement the 4-digit subcircuits, but any of the circuits studied in this section can be used. The only addition with respect to the Gayathri et al. proposal is that we will use a temporary logical-AND gate instead of a Toffoli gate to implement the value of V .

4. Analysis and Comparison

The proposed circuits consist of two 4-digit subcircuits, which are combined to form a larger 8-digit circuit by adding a temporary logical-AND gate and two CNOT gates. These three gates are common to all three circuits, assuming a T-count of 4 and a T-depth of 2 corresponding to the temporary logical-AND gate (as mentioned above, the CNOT gate has neither T-count nor T-depth). Otherwise, calculating the T-count and T-depth is trivial knowing

that we simply have to count the number of Toffoli and temporary logical-AND gates that compose them. This can be easily done by counting these gates from Fig. 8, 9, and 10, corresponding to circuits 1, 2 and 3, respectively. For the total T-count, we must count the T-count of each figure twice, since two 4-digit circuits are needed. However, the T-depth of the 4-digit circuit is similar to the 8-digit circuit (without adding the temporary logical-AND used to obtain V) since both circuits can be run in parallel. Starting to analyze the 4-digit modules, we obtain that circuit 1 (Fig. 8) has a T-count of 24 and a T-depth of 12. For circuit 2 (Fig. 9), the values obtained are a T-count of 20 and a T-depth of 10. Finally, measuring circuit 3 (Fig. 10) will give values of 33 and 15 of T-count and T-depth, respectively. Since we have to use two of these circuits, it is obtained that the values for each circuit are 48 T-count and 12 T-depth for the first circuit, 40 T-count and 10 T-depth for the second circuit, and 66 T-count and 15 T-depth for the third circuit. To these values must be added the operation necessary to obtain V, which gives an extra T-count of 4 and a T-depth of 2. Therefore, the final numbers for each of the circuits are as follows:

- **Circuit 1 (8 digits)**. T-count: 52, T-depth: 14, Number of ancilla inputs: 13.
- **Circuit 2 (8 digits)**. T-count: 44, T-depth: 12, Number of ancilla inputs: 11.
- **Circuit 3 (8 digits)**. T-count: 70, T-depth: 17, Number of ancilla inputs: 9.

As an example, the second complete 8-digit circuit is shown in Fig. 11. The 8-digit circuits 1 and 3 are obtained in the same way, but naturally replacing the 4-digit modules by the circuits in Figs. 8 and 10, respectively. A comparison of our three circuits along with the best LZD circuits available in the literature is shown in Table 2. The comparison, as described above, has been performed in terms of T-count, T-depth, and number of ancilla qubits. Regarding the T-count, the best option is our second proposal, with a value of 44. It is followed by our other two proposals, with values of 52 and 70. In the fourth position, with 76, is the circuit of Gayathri et al. The circuit of AnanthaLakshmi et al. would be next, with a T-count of 84, and in last place we find the circuit of Nguyen et al. with a value of 112. Thus, our best circuit in terms of T-count improves on the best available in the literature by 42.11%. On the other hand, our other two circuits improve by 31.58% (circuit 1) and 7.89% (circuit 3) the circuit of Gayathri et al. We can thus state that our three proposals are the best LZD circuits currently available in terms of T-count. In particular, our second proposal is the best option.

Table 2. Comparison of 8-digit LZD circuits in terms of T-count, T-depth, and ancilla qubits.

Circuit	T-count	T-depth	Ancilla qubits
Nguyen et al. [10]	112	54	38
AnanthaLakshmi et al. [15]	84	63	38
Gayathri et al. [16]	76	38	9
Proposed circuit 1	52	14	13
Proposed circuit 2	44	12	11
Proposed circuit 3	70	17	9

Concerning the T-depth, the best circuit in Table 2 is again our second proposal, with a value of only 12. It is followed by our proposals 1 and 3, with values of 14 and 17, respectively. The best state-of-the-art circuit in terms of T-depth is that of Gayathri et al. with a value of 38. In penultimate position is the circuit of Nguyen et al. with a T-depth of 54, followed by that of AnanthaLakshmi et al. with a T-depth of 63. Expressed as a percentage, our three proposals improve on the best available state-of-the-art circuit by 68.42%, 63.16%, and 55.26%, respectively. The results of our circuits in terms of T-depth are remarkable, as the worst of the three improves by more than 50% over what is available in the literature.

Finally, if we compare the circuits in Table 2 in terms of ancilla inputs, there is a tie between the circuit of Gayathri et al. and our circuit 3, both with 9 ancilla qubits. Next would be our proposals 2 and 1, with 11 and 13 ancilla qubits, respectively. Far behind these values are the rest of the circuits, with a value of 38. Thus, in terms of ancilla qubits, only one of our proposals comes close to matching (but not bettering) those currently available in the state of the art.

There is, therefore, no single circuit that is better on all three metrics. Our circuit 2 stands as the best circuit in terms of T-count and T-depth. However, it involves 2 extra qubits with respect to the best circuit in terms of ancilla qubit. In terms of ancilla inputs, the best circuits are that of Gayathri et al. and our proposal 3. However, our circuit

3 improves the T-count of the Gayathri et al. circuit by 7.89%, and its T-depth by 55.26%, so overall it would be a better choice. The reasonable choice would then be to choose between our circuits 2 and 3, knowing that:

- Circuit 2 improves the T-count of circuit 3 by 26, i.e., by 37.14%.
- Circuit 2 improves the T-depth of circuit 3 by 5, i.e. by 29.41%.
- Circuit 3 improves the number of ancilla inputs of circuit 2 by 2, i.e. by 18.18%.

Interpreting these numbers, we can only state that circuit 2 is better in terms of T gates, and circuit 3 is better in terms of ancilla inputs. Researchers interested in choosing an LZD circuit for use in their own circuits and algorithms will have to assess, based on their own resources, which option is more suitable for them between these two circuits.

5. Conclusions

In this paper we provide three LZD circuits for quantum computers and simulators. These three circuits improve on existing circuits in the literature in terms of noise tolerance. Their design is provided using only Clifford+T gates.

The three proposed circuits have been designed with the objective of reducing the T-count and T-depth with respect to the alternatives available in the state of the art. In order to achieve this goal, the most optimized way to perform each of the operations involved has been sought, and methodologies of proven success in classical computer structures have also been used. Likewise, the use of the temporary logical-AND gate in substitution of the Toffoli gate has been fundamental to achieve the mentioned reductions.

Since keeping the number of qubits as small as possible is important, both to avoid noise errors and to minimize resources (something fundamental in today’s quantum computers), we have also aimed to reduce the number of ancilla inputs of our circuits, obtaining equal or superior results with respect to the rest of the circuits.

Finally, we have carried out a comparison between our circuits and the best circuits available in the state of the art. As a result of this comparison, we can affirm that the circuits proposed in this work are the most suitable option to implement the LZD operation in current quantum computers.

Acknowledgments

This work was supported in part under grants PID2020-119082RB-C22, PID2021-123461NB-C22, PID2021-123278OB-I00 and MTM-2017-83506-C2-2-P funded by MCIN/AEI/ 10.13039/501100011033; by the Regional Ministry of Junta de Andalucía under the grants P20_00748, IC-DRUGS-P18-RT-1193, UAL2020-TIC-A2101, and UAL18-TIC-A020-B; by Gobierno del Principado de Asturias under grant AYUD/2021/50994, and by the European Regional Development Fund (ERDF).

Data Availability

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

- [1] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, J. Latorre, Data re-uploading for a universal quantum classifier, *Quantum* 4 (2020) 226.
- [2] F. Orts, G. Ortega, A. Cucura, E. Filatovas, E. Garzón, Optimal fault-tolerant quantum comparators for image binarization, *The Journal of Supercomputing* (2021) 1–12.
- [3] T. Jones, A. Brown, I. Bush, S. Benjamin, Quest and high performance simulation of quantum computers, *Scientific reports* 9 (1) (2019) 1–11.
- [4] D. Steiger, T. Häner, M. Troyer, Projectq: an open source software framework for quantum computing, *Quantum* 2 (2018) 49.
- [5] M. Nielsen, I. Chuang, *Quantum computation and quantum information* (2002).
- [6] L. Grover, A fast quantum mechanical algorithm for database search, in: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.
- [7] L. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Physical review letters* 79 (2) (1997) 325.
- [8] E. Combarro, J. Ranilla, I. Rúa, Experiments testing the commutativity of finite-dimensional algebras with a quantum adiabatic algorithm, *Computational and Mathematical Methods* 1 (1) (2019) e1009.

- [9] C. Bernhardt, *Quantum computing for everyone*, Mit Press, 2019.
- [10] T. Nguyen, R. Van Meter, A resource-efficient design for a reversible floating point adder in quantum computing, *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 11 (2) (2014) 1–18.
- [11] M. Amy, D. Maslov, M. Mosca, Polynomial-time t -depth optimization of clifford+ t circuits via matroid partitioning, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33 (10) (2014) 1476–1489.
- [12] H. Thapliyal, E. Muñoz-Coreas, V. Khalus, Quantum circuit designs of carry lookahead adder optimized for T-count, T-depth, and qubits, *Sustainable Computing: Informatics and Systems* 29 (2021) 100457.
- [13] F. Orts, G. Ortega, E. Garzón, Efficient reversible quantum design of sign-magnitude to two's complement converters, *Quantum Information & Computation* 20 (9-10) (2020) 747–765.
- [14] V. Oklobdzija, An algorithmic and novel design of a leading zero detector circuit: Comparison with logic synthesis, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2 (1) (1994) 124–128.
- [15] A. AnanthaLakshmi, G. Sudha, Design of an efficient reversible single precision floating point adder, *International Journal of Computational Intelligence Studies* 4 (1) (2015) 2–30.
- [16] S. Gayathri, R. Kumar, S. Dhanalakshmi, G. Dooly, D. Duraibabu, T-count optimized quantum circuit designs for single-precision floating-point division, *Electronics* 10 (6).
- [17] T. Nguyen, R. Van Meter, A space-efficient design for reversible floating point adder in quantum computing, *ACM Journal on Emerging Technologies in Computing Systems* 11 (2).
- [18] D. Nandan, J. Kanungo, A. Mahajan, Implementation of leading one detector based on reversible logic for logarithmic arithmetic, *International Journal of Computer Applications* 173 (8) (2017) 40–45.
- [19] H.-S. Li, P. Fan, H. Peng, S. Song, G.-L. Long, Multilevel 2-d quantum wavelet transforms, *IEEE Transactions on Cybernetics*.
- [20] F. Orts, G. Ortega, E. Combarro, E. Garzón, A review on reversible quantum adders, *Journal of Network and Computer Applications* (2020) 102810.
- [21] P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM review* 41 (2) (1999) 303–332.
- [22] R. Babbush, C. Gidney, D. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, H. Neven, Encoding electronic spectra in quantum circuits with linear t complexity, *Physical Review X* 8 (4) (2018) 041015.
- [23] C. Gidney, M. Ekerå, How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits, *Quantum* 5 (2021) 433.
- [24] D. Bernstein, T. Lange, C. Martindale, L. Panny, Quantum circuits for the csidh: optimizing quantum evaluation of isogenies, in: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2019, pp. 409–441.
- [25] Z. Sheng-Xing, L. Gui-Lu, L. Xiao-Shu, A remote quantum adding machine, *Chinese Physics Letters* 19 (11) (2002) 1579.
- [26] H. Li, P. Fan, H. Xia, H. Peng, G. Long, Efficient quantum arithmetic operation circuits for quantum image processing, *SCIENCE CHINA Physics, Mechanics & Astronomy* 63 (2020) 1–13.
- [27] R. Zhou, W. Hu, P. Fan, H. Ian, Quantum realization of the bilinear interpolation method for neqr, *Scientific reports* 7 (1) (2017) 1–17.
- [28] A. Wei, P. Naik, A. Harrow, J. Thaler, Quantum algorithms for jet clustering, *Physical Review D* 101 (9) (2020) 094015.
- [29] L. Gyongyosi, S. Imre, Quantum circuit design for objective function maximization in gate-model quantum computers, *Quantum Information Processing* 18 (7) (2019) 1–33.
- [30] M. Mohammadi, M. Eshghi, On figures of merit in reversible and quantum logic designs, *Quantum Information Processing* 8 (4) (2009) 297–318.
- [31] M. Noorallahzadeh, M. Mosleh, Parity-preserving reversible flip-flops with low quantum cost in nanoscale, *The Journal of Supercomputing* 76 (3) (2020) 2206–2238.
- [32] M. Noorallahzadeh, M. Mosleh, Efficient designs of reversible latches with low quantum cost, *IET Circuits, Devices & Systems* 13 (6) (2019) 806–815.
- [33] H. Gaur, A. Singh, U. Ghanekar, In-depth comparative analysis of reversible gates for designing logic circuits, *Procedia Computer Science* 125 (2018) 810–817.
- [34] F. Orts, G. Ortega, E. Garzón, An optimized quantum circuit for converting from sign-magnitude to two's complement, *Quantum Information Processing* 18 (11) (2019) 1–14.
- [35] C. Gidney, Halving the cost of quantum addition, *Quantum* 2 (2018) 74.
- [36] M. Amy, D. Maslov, M. Mosca, 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* 32 (6) (2013) 818–830.
- [37] C. Jones, Low-overhead constructions for the fault-tolerant toffoli gate, *Physical Review A* 87 (2) (2013) 022328.
- [38] P. Selinger, Quantum circuits of t -depth one, *Physical Review A* 87 (4) (2013) 042302.
- [39] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter, Elementary gates for quantum computation, *Physical review A* 52 (5) (1995) 3457.
- [40] Y. Liu, G. L. Long, Y. Sun, Analytic one-bit and cnot gate constructions of general n -qubit controlled gates, *International Journal of Quantum Information* 6 (03) (2008) 447–462.
- [41] M. Amy, N. Ross, The phase/state duality in reversible circuit design, *arXiv preprint arXiv:2105.13410*.
- [42] D. Große, R. Wille, G. Dueck, R. Drechsler, Exact synthesis of elementary quantum gate circuits for reversible functions with don't cares, in: *38th International Symposium on Multiple Valued Logic (ISMVL 2008)*, IEEE, 2008, pp. 214–219.

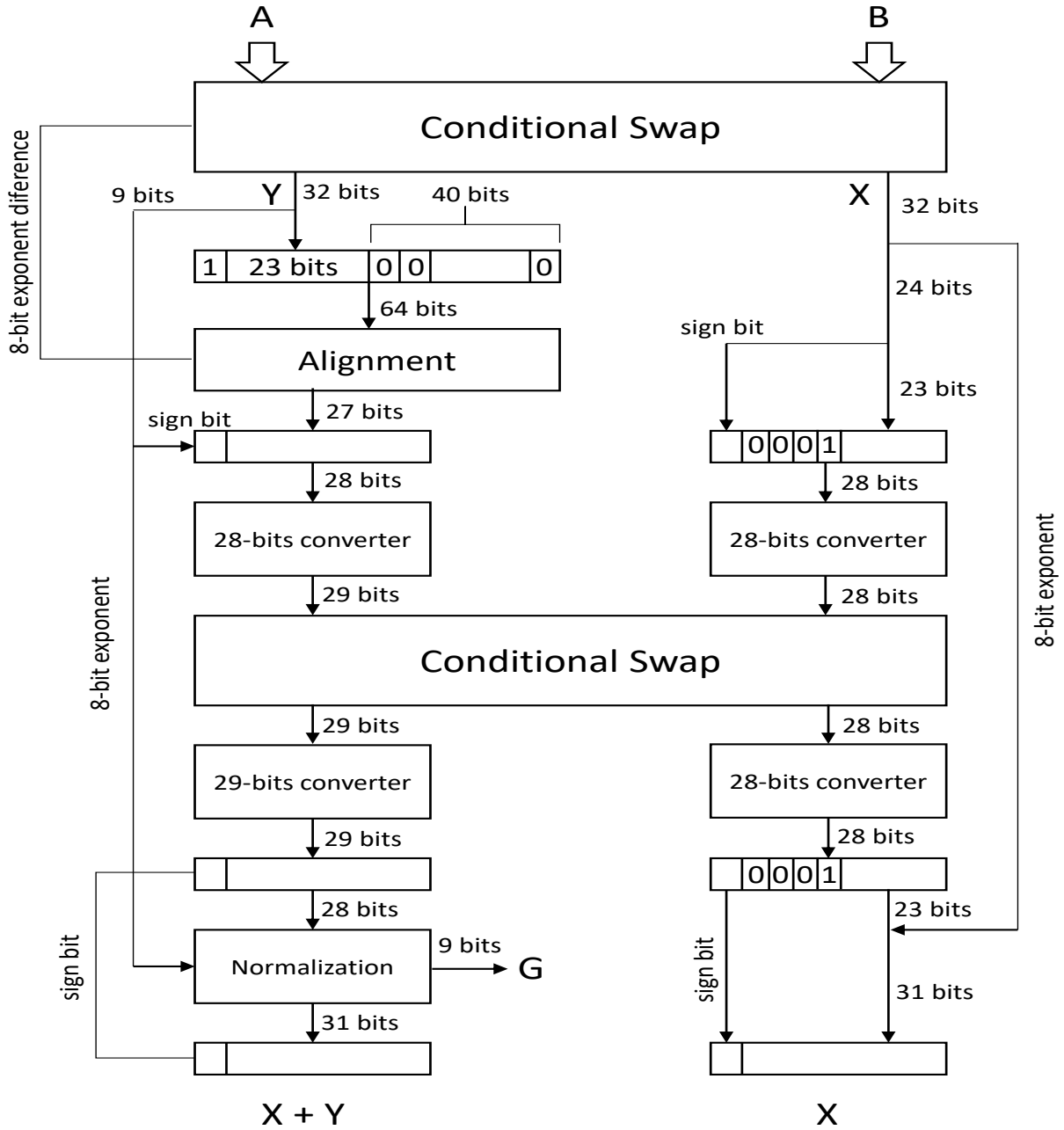


Figure 1. Algorithm for the addition of two floating-point numbers A and B in quantum computers, as it is presented in Nguyen et al [10]. The normalization part, which is the one we are interested in, is detailed in Fig. 2.

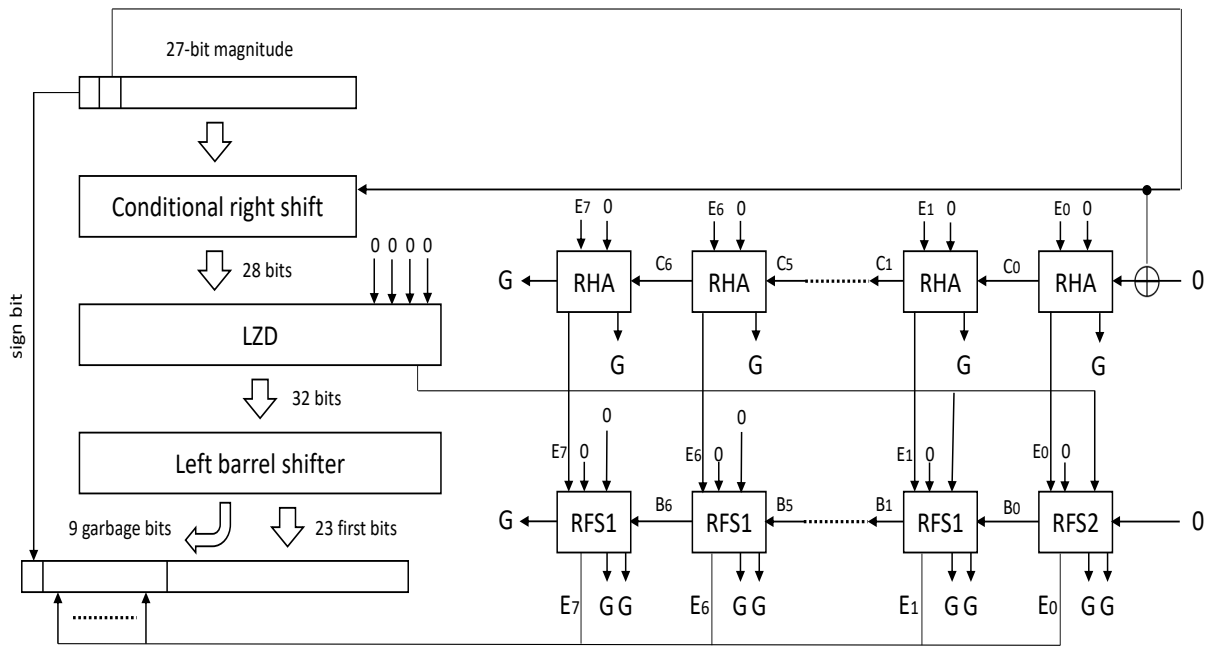


Figure 2. Normalization process performed in the addition of two floating-point numbers algorithm shown in Figure 1. This circuit involves a subcircuit dedicated to LZD operation, whose optimization is addressed in our work.

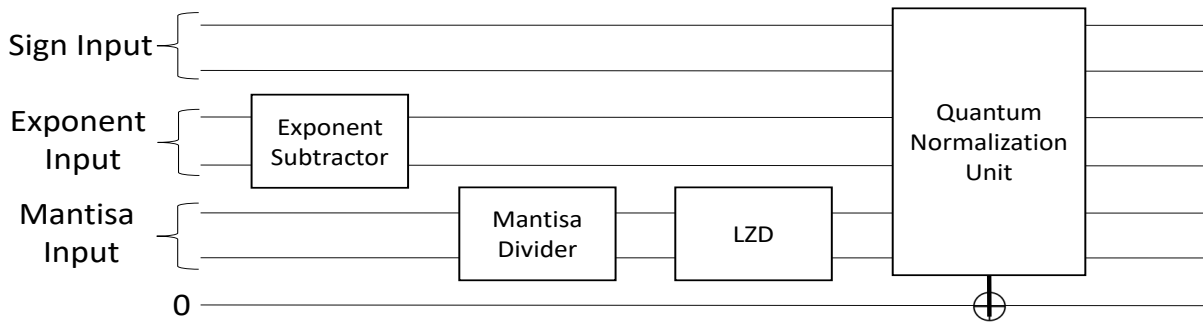


Figure 3. Quantum floating-point division circuit, as it is presented in Gayathri et al. [16]. It can be seen that a fundamental part of this operation is the LZD circuit.

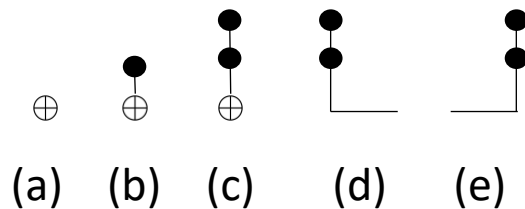


Figure 4. Symbol of the quantum gates used in this work. (a) Pauli-X gate, (b) Controlled Not (CNOT) gate, (c) Toffoli gate, (d) Temporary logical-AND gate, and (e) the uncomputation of the temporary logical-AND gate.

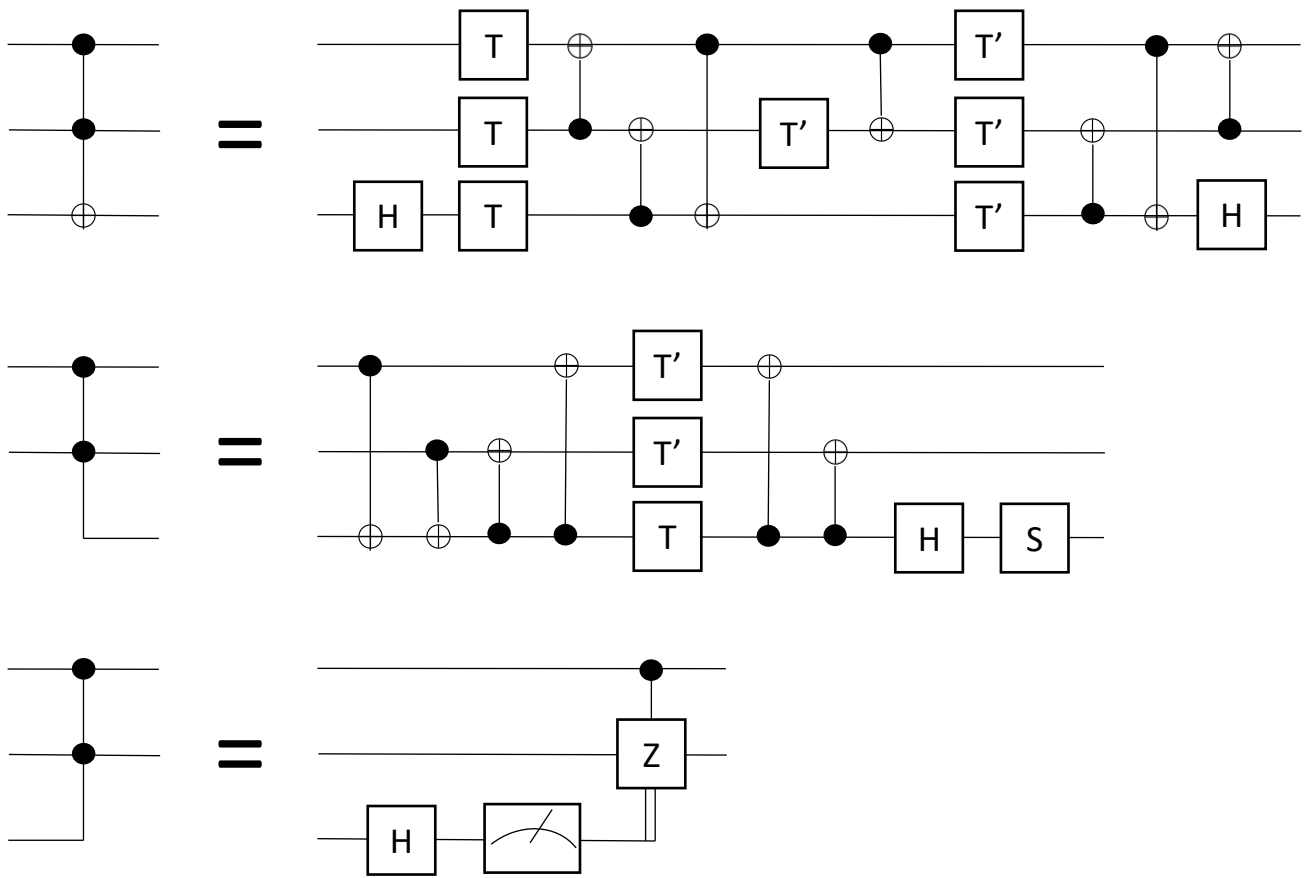


Figure 5. Implementation of the Toffoli gate, and the temporary logical-AND and its uncomputation.

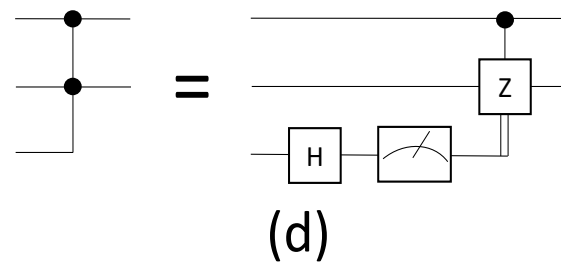
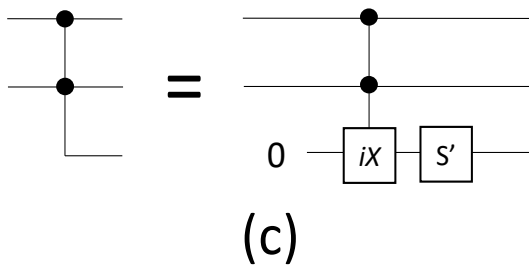
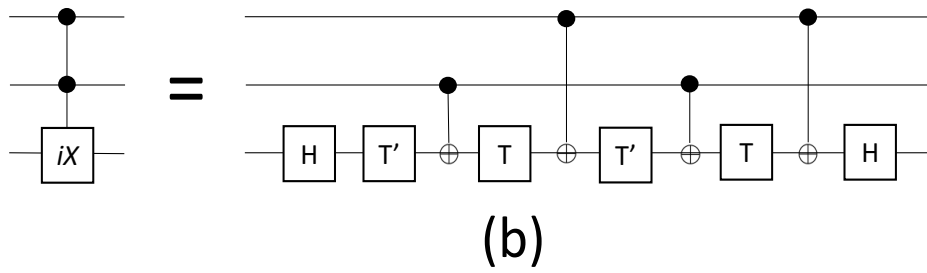
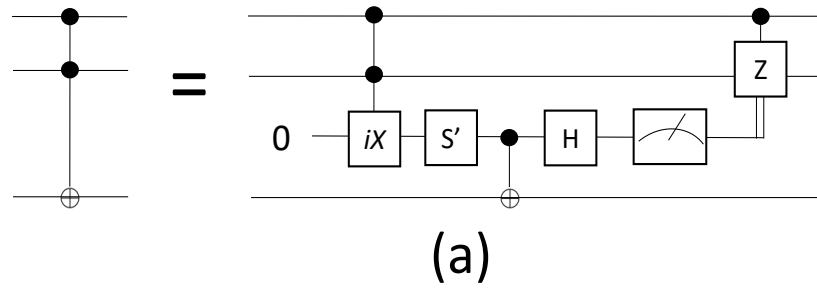


Figure 6. (a) Implementation of the Toffoli gate proposed by Jones [37]. (b) Composition of the iX gate [38]. (c and d) Construction of the temporary logical-AND operation, separating the Jones' circuit into two parts: the AND operation (c), and its uncomputation (d).

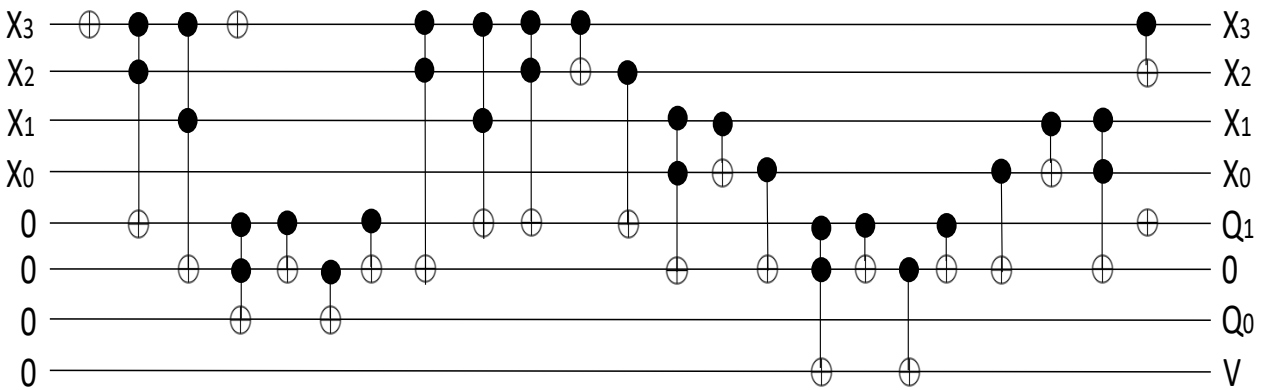


Figure 7. Quantum leading zero detector proposed in Gayathri et al. [16].

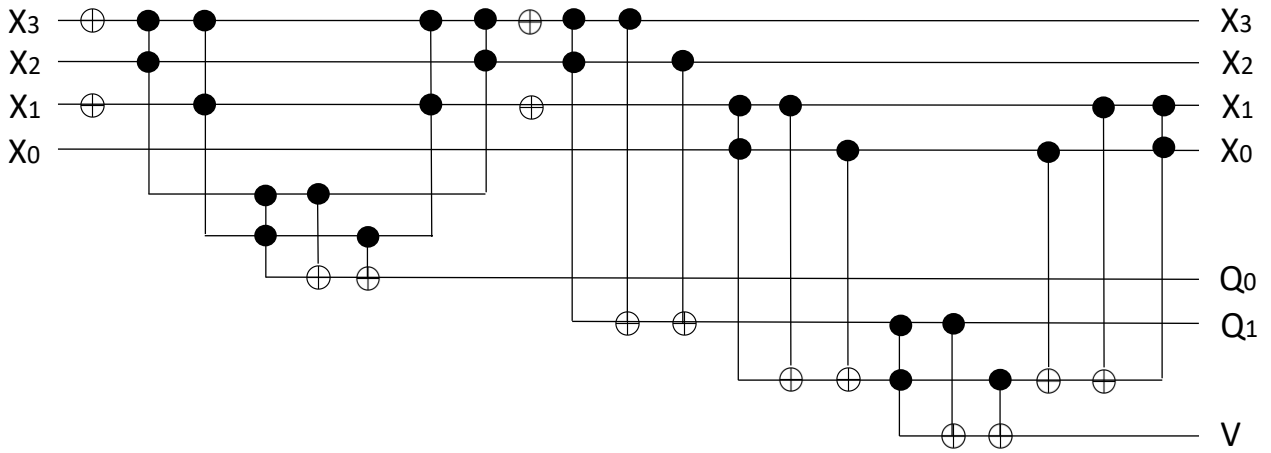


Figure 8. First proposed 4-qubit quantum leading zero detector.

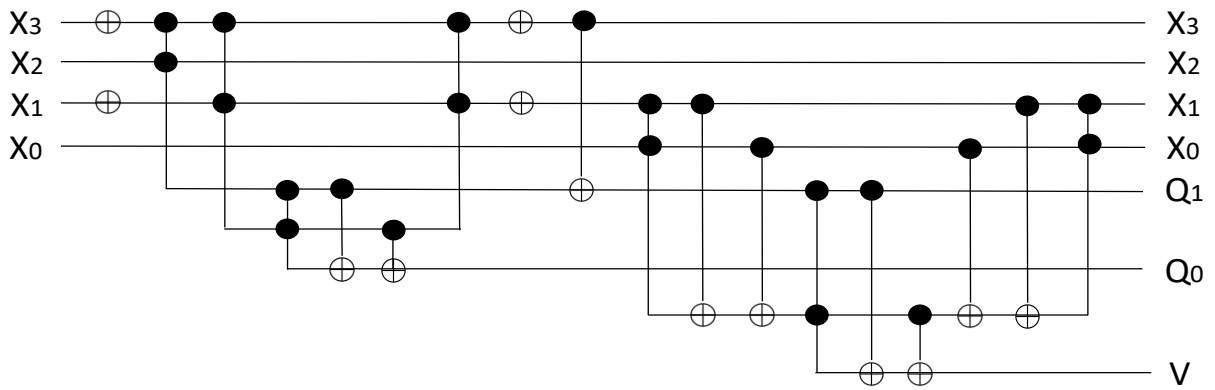


Figure 9. Second proposed 4-qubit quantum leading zero detector.

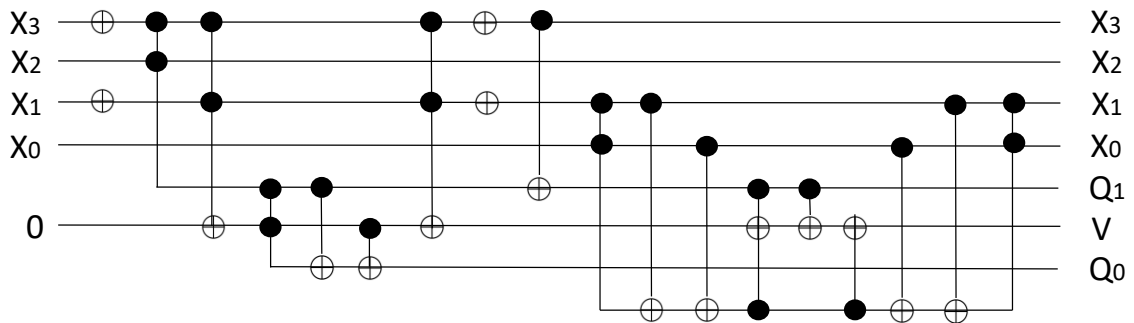


Figure 10. Third proposed 4-qubit quantum leading zero detector.

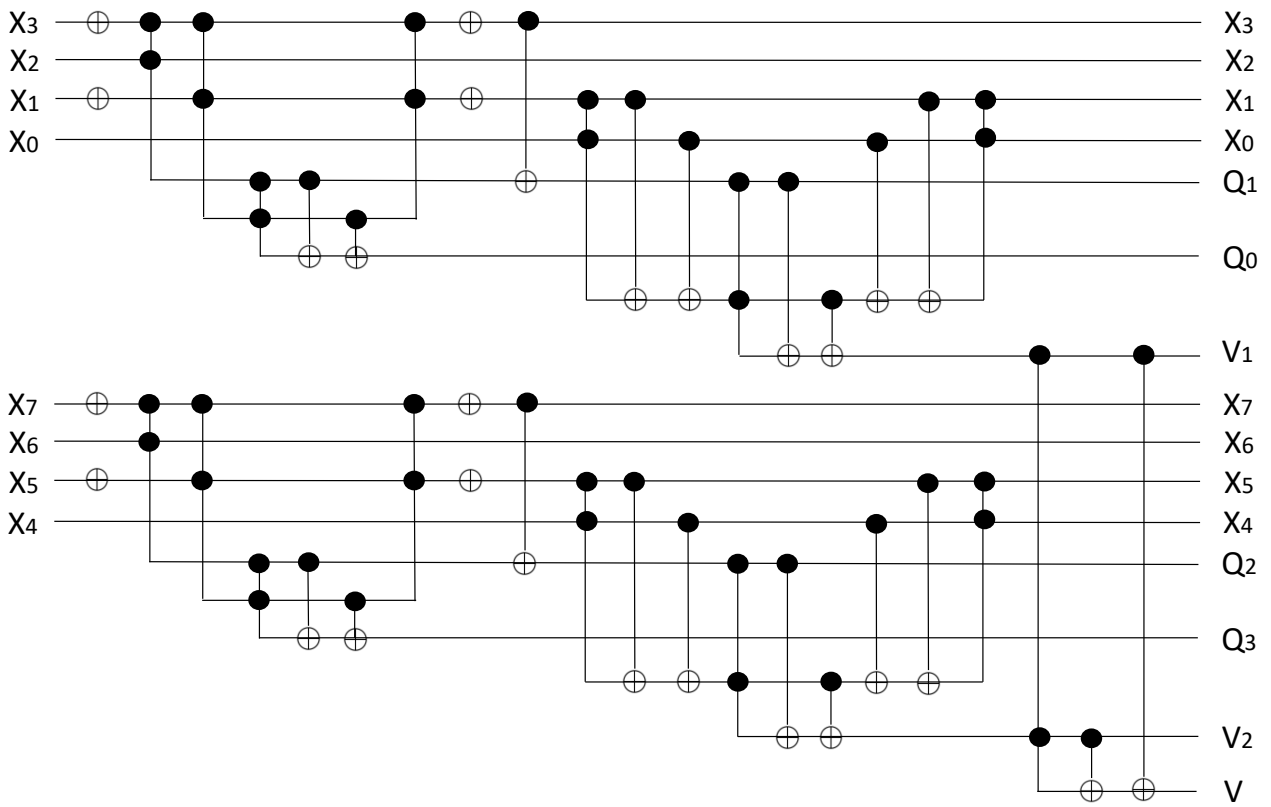


Figure 11. Second proposed 8-qubit quantum leading zero detector.