Quantum Information and Computation, Vol. 0, No. 0 (2003) 000–000

EFFICIENT REVERSIBLE QUANTUM DESIGN OF SIGN-MAGNITUDE TO TWO'S COMPLEMENT CONVERTERS

F. ORTS, G. ORTEGA, E.M. GARZÓN
University of Almería, Department of Informatics
Almería, 04120, Spain [a]

Received (received date)
Revised (revised date)

Despite the great interest that the scientific community has in quantum computing, the scarcity and high cost of resources prevent to advance in this field. Specifically, qubits are very expensive to build, causing the few available quantum computers are tremendously limited in their number of qubits and delaying their progress. This work presents new reversible circuits that optimize the necessary resources for the conversion of a sign binary number into two's complement of N digits. The benefits of our work are two: on the one hand, the proposed two's complement converters are fault tolerant circuits and also are more efficient in terms of resources (essentially, quantum cost, number of qubits, and T-count) than the described in the literature. On the other hand, valuable information about available converters and, what is more, quantum adders, is summarized in tables for interested researchers. The converters have been measured using robust metrics and have been compared with the state-of-the-art circuits. The code to build them in a real quantum computer is given.

Keywords: Quantum Computing, Quantum circuits, Reversible circuit, Two's complement, Sign-magnitude representation to two's complement converter
Communicated by: to be filled by the Editorial

1 Introduction

Quantum computing uses the principles of quantum mechanics to define a new paradigm of computing that allows reaching goals that cannot be achieved by those already known algorithms or classical computing. There is a great interest in the scientific community for quantum computing [1]. Specifically, efforts must be made to determine what kind of problems can be solved more efficiently by quantum computing than by traditional computation. In any case, there are problems that can be solved using quantum computing more efficiently. Grover's and Shor's algorithms are the best examples of such problems [2, 3].

The most remarkable difference between both paradigms, quantum computing and classical computing, is the basic unit of each one. In the case of classical computing, the bit is the fundamental unit. A bit has only two possibilities, 0 and 1. Despite the simplicity of the bit, current computers can perform any type of operation if they have enough number of bits. On the other hand, quantum computers define their own unit, the quantum bit (qubit). These qubits can also be in states 0 or 1 and, what is more, because of the properties of quantum mechanics they can be simultaneously in both states. This property is called superposition.

[a]Corresponding author: F. Orts. e-mail: francisco.orts@ual.es.

Thanks to the superposition, a qubit can easily and completely imitate the behavior of, for example, an atom. For this reason among others, quantum computing is interesting in many research fields such as chemistry [4], image processing [5] or mathematics [6].

Nowadays, quantum computing is focused on making possible the implementation of the aforementioned Shor's algorithm. This algorithm allows a number N to be decomposed into factors. It is faster than any similar algorithm in classical computation [3]. The problem of computing Shor's algorithm does not come from the algorithm itself, but from the current quantum computers since they have only a few qubits [1]. However, companies like Google, IBM or Intel are currently working on getting larger quantum computers, that is, with more qubits. Meanwhile, to accelerate the achievements of the computation of Shor's algorithm, current works are focusing on optimizing the necessary resources for its computation.

Following the line of optimizing resources, it is especially important to optimize basic arithmetic operations. There are current works that present optimized versions to perform addition [7, 8], subtraction [9, 10, 11], multiplication [12, 13], and division [14, 15] in quantum computers. Very relevant is the addition, which has a fundamental role in Shor's algorithm. However, the optimization of the involved operations is not the only possibility to achieve the effective computation of Shor's algorithm and the advance of quantum computing in general. For example, another possibility is to take an appropriate numeric representation and to provide tools for its use. In classical computing, the use of the two's complement representation is common. Among other improvements, the two's complement allows the addition of negative numbers to be computed more efficiently [16]. Working in two's complement can be beneficial for the optimal use of the available resources in quantum computing, and for the acceleration of several operations [17, 18, 19, 20, 21, 22, 23, 24, 25].

Quantum circuits have several parameters such as the number of required qubits or the quantum cost (the number of used logical gates). This work proposes sign-magnitude to two's complement converters which are focused on optimizing resources to make their implementation as simple and inexpensive as possible. Circuits that involve a small amount of resources are really appreciated in quantum computation. They are relevant even when they do not achieve quantum advantages since they may be a useful part in bigger circuits and algorithms [26]. To achieve this, a study of the state-of-the-art circuits that allow the conversion of sign-magnitude to two's complement has been carried out. This study has analyzed the benefits of each circuit to be able to implement new ones reducing their quantum cost, the number of involved qubits, the number of T gates, and other interesting parameters in an effort to minimize the use of resources. A solid metric has been defined to evaluate these parameters in both the proposed converters and the state-of-the-art circuits.

This paper is organized as follows. Section 2 presents background information on quantum gates, defines the metrics in terms of quantum computing, explains the two's complement methodology and finally explores the state-of-the-art circuits which can compute a conversion from sign-magnitude to two's complement and their methodology. In Section 3 the designs of the proposed sign-magnitude to two's complement converters are presented and discussed in detail. Finally, the conclusions are exposed in Section 4.

The circuits have been tested in the real quantum computer *ibmq_ourense*. Our implemented circuits are freely available through the following website: https://github.com/2forts/QuantumConversor. The code is ready to be used in the IBM Quantum Experience

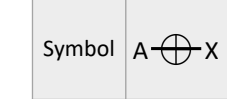
Name	Pauli-X	V	V+	CNOT	Controlled-V	Controlled-V+
Symbol						
Matrix Form	$\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}$	$\begin{vmatrix} 1+i & 1-i \\ 1-i & 1+i \end{vmatrix}$	$\begin{vmatrix} 1-i & 1+i \\ 1+i & 1-i \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1+i & 1-i \\ 0 & 0 & 1-i & 1+i \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1-i & 1+i \\ 0 & 0 & 1+i & 1-i \end{vmatrix}$

Fig. 1: Information of the basic reversible gates used in quantum circuits. (a) Pauli-X gate, (b) V gate, (c) V+ gate, (d) CNOT gate, (e) controlled-V gate, and (f) controlled-V+ gate.

platform, which includes quantum simulators and real quantum computers [27]. Due to the topology of each platform, several switch operations could be necessary. Luckily, the IBM Quantum Experience platform is able to perform them automatically.

2 Background

2.1 Measures in quantum circuits

This work has considered the metrics described in [28] to measure quantum circuits. Such metrics define five essential factors:

- Quantum cost: The quantum cost of a circuit defines the number of basic gates integrated into the circuit. This parameter is the most important when optimization of resources is mandatory [11].
- Delay: The delay of a circuit is its speed. A circuit is faster than other if its delay is lower.
- Normal inputs: inputs whose value is given by the user.
- Ancilla inputs: extra qubits used to compute auxiliary operations. Qubits remain a scarce resource, so the number of required qubits should be reduced whenever possible.
- Garbage outputs: outputs that are not part of the solution and whose value are unknown. For instance, an ancilla input whose value has not been uncomputed. All the outputs which are not part of the solution must be restored to its initial values or they will not be available to be entangled with other inputs of circuits since this operation will generate anomalous results [1].

In [28], authors set an important rule for measuring: the quantum cost of 1×1 and 2×2 gates is 1. Progressively, the quantum cost of an $N \times N$ gate consists of its number of 1×1 and 2×2 gates. The quantum cost of a circuit can be easily calculated adding the quantum cost of all its gates. Moreover, [28] defines the delay of 1×1 and 2×2 gates (that is, gates

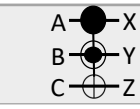
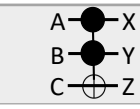
Name	Peres	Toffoli
Symbol		
Matrix Form	$\begin{vmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0 \end{vmatrix}$	$\begin{vmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{vmatrix}$

Fig. 2: Information of the basic reversible gates used in quantum circuits. (a) Peres gate and (b) Toffoli gate.

with quantum cost 1) as $1\triangle$. In the same way, the delay of an $N \times N$ gate is its depth when it is built using 1×1 and 2×2 gates. In this work, all the measures of circuits have been carried out using these terms so that the comparisons were coherent. Therefore, the gates shown in Fig. 1 have a quantum cost of 1 and a delay of $1\triangle$. The Peres gate has a quantum cost of 4 and a delay of $4\triangle$, and the Toffoli gate a quantum cost and delay of 5 and $5\triangle$ respectively. Both gates are shown in Fig. 2.

Apart from the mentioned factors, because of quantum computers are extremely vulnerable to noise errors, other factors can be considered to include fault tolerance and palliate noise problems [29]. The T gates are used to make possible the use of error-correcting codes to ensure fault-tolerance in quantum circuits. But their drawback is that they are more expensive than the rest in terms of space and time cost due to their increased tolerance to noise errors [29, 30]. Therefore, there are two important factors to minimize the cost when using such gates: the T-count, that is, the number of T gates; and the T-depth, that is, the number of T gates which must be computed sequentially. Two or more T gates computed in parallel are in the same level and they only increase the T-depth in 1 unit. Therefore, the T-count and the T-depth of the circuits are also considered in this work.

2.2 Two's complement

Two's complement is a binary numeric representation. Its main advantage is that the representation of an integer is easily manipulated by the hardware. For instance, negating a number can be done inverting all its digits and also adding one to the whole number. Negation in sign-magnitude representation involves more operations because it is necessary to handle the sign of the number as a special digit. Other examples are the addition and subtraction. Adding negative numbers is identical to adding positive numbers. Therefore, it is not necessary additional logic to handle the negative case.

The representation of an N-digit number in two's complement consists of its complement with respect to 2^N. The numeric range of notation, for the N-digit case, is $-2^{N-1} \leq X \leq (-2^{N-1} - 1)$ [31]. As an example, the case $N = 4$ is shown in Table 1. Converting a number A from signed binary to two's complement is as follows. If $A >= 0$, no conversion is necessary as both representations are equal, that is, A has the same representation in signed binary as in two's complement. However, in the case of $A < 0$, the conversion is necessary. Such a

conversion can be computed as the inversion of all the digits of A and then $\overline{\mathrm{A}} + 1$.

Table 1: Decimal, Sign-magnitude and Two's complement representation of binary numbers with $N = 4$.

Sign-magnitude	**Two's complement**	**Decimal**
0111	0111	7
0110	0110	6
0101	0101	5
0100	0100	4
0011	0011	3
0010	0010	2
0001	0001	1
0000	0000	+0
1000	–	−0
1001	1111	−1
1010	1110	−2
1011	1101	−3
1100	1100	−4
1101	1011	−5
1110	1010	−6
1111	1001	−7
–	1000	−8

The conversion from sign-magnitude to two's complement can be done in two ways;

- Designing a specific circuit which computes the conversion described in Table 1.
- Using an addition circuit.

Instead of using a specific converter, the conversion of a number A can be computed inverting its digits and then computing $\overline{\mathrm{A}} + 1$. Therefore, the conversion can be done with N Pauli-X gates and an existing addition circuit. In the next subsection, the state-of-the-art converters and adders are reviewed.

2.3 Related work

Only a few works can be found in the literature related to the specific implementation of sign-magnitude to two's complement converters without using adders. And worse, none of these works is based on fault-tolerant Clifford + T gates. [33] presented the mathematical design (but not a circuit design) of a gate called $SSMT$ which computes the conversion of a 4-digit number. [34] proposed a gate called TCG. This gate is only valid for the 4-digit case like the previous converter but, in this case, the logic design was presented. TCG was optimized in [32]. This design has a quantum cost of 25 for the case $N = 4$, being the best converter in terms of quantum cost. Although these converters have a good quantum cost, they are not able to compute numbers with more than 4 digits. They do not have carry input nor carry output, so several converters cannot be joined to compute numbers with more than 4 digits. In [35], a novel converter valid for any N-digit number was presented. It is based on a reversible out-of-place carry look-ahead adder presented in [8], adapted to compute the

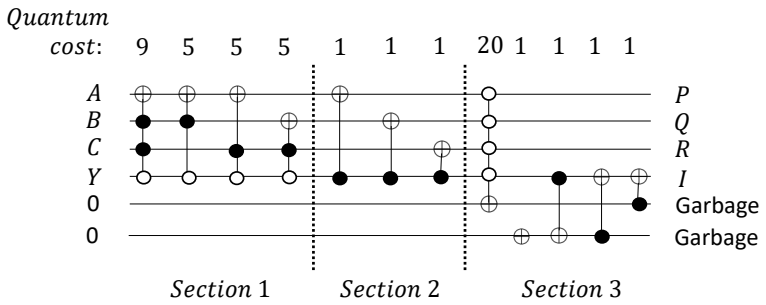


Fig. 3: Ripple-carry sign-magnitude to two's complement converter. It is based on the circuit presented in [32], adding it carry-in (Y) and carry-out (I). It accepts a number or a piece of the number of 3 digits (A, B, C), giving its conversion to two's complement (P, Q, R). It has two auxiliary inputs and two garbage outputs. Moreover, in the Figure can be observed the quantum cost of every step.

conversion specifically. Its delay improves the delay of the rest of converters and adders that are able to compute the conversion.

The sign-magnitude to two's complement converter of [32] has the lowest quantum cost, 35. However, as it has been mentioned, it is only valid for the case $N = 4$ as it does not support carry output nor carry input of any type. We have modified this gate with the objective to make it available for the general N-digit case, converting it in a ripple-carry converter. This circuit is shown in Fig. 3. Nevertheless, the cost of adding this extra feature is expensive because three qubits of the original gate need to be dedicated to the carry: on the one hand, it is necessary to add one new qubit to accept and compute the carry-in (Y in Fig. 3). We have replaced an original input qubit of the circuit of [32] by Y, so the circuit can compute a number (or a piece of the number) of $N = 3$. If $Y = 0$, the circuit acts like in [32] (Fig. 3, Section 1), but if $Y = 1$, the values of the inputs A, B and C are inverted (Fig. 3, Section 2). On the other hand, there are extra costs to generate the carry out (Fig. 3, Section 3). Two auxiliary qubits are needed. The first one is the target qubit of a multiple Toffoli gate (MTG, quantum cost of 20 [36, 37]), which is set to 1 for the case which generates carry-out, it means $\overline{A}\,\overline{B}\,\overline{C}\,\overline{Y}$. The second one is used to change the value of Y to I when necessary. The total quantum cost is 51, and the delay is also 51$\triangle$. The quantum cost of each gate of the circuit is shown in Fig. 3. However, the two garbage outputs have not been uncomputed, so the quantum cost and delay are assumed to be bigger than 51. For instance, following the Bennett's garbage removal scheme [38], it would be necessary four extra qubits to save the results, four extra CNOT gates to translate the result into these qubits, and also to apply the logical reverse of the entire circuit (except for the part which involves the new four qubits).

As it has been mentioned, the conversion of a number A can be done computing $\overline{\mathrm{A}} + 1$. For this, it is necessary to invert the digits of A using Pauli-X gates (one gate per digit) and to use an adder circuit to compute $\overline{A} + 1$. This is the most usual way to compute the two's complement since there are lots of quantum addition circuits in the state-of-the-art [7, 8, 11, 39, 40, 41, 42, 43, 44, 45], contrary to what happens with specific converters. Moreover, there are several fault tolerant adders [46, 47]. We have summarized the information of the most relevant adders in Table 2, but without including fault tolerant adders in this

first comparison since they have a higher quantum cost due to their tolerance to noise errors. Adders with garbage outputs have not been included, for the reasons explained in subsection 2.1 about garbage outputs. Table 2 shows that the two most efficient addition circuits in terms of delay are [7, 8], with a delay of O(log N) (they are the only carry-lookahead adders in the table). However, they are expensive in terms of quantum cost. Ripple-carry adders improve them in terms of quantum cost. In general, ripple-carry adders are more optimized in terms of quantum cost, and carry look-ahead adder are better than them in terms of delay [48]. On the other hand, the ripple-carry adder which can be built using the full adder presented in [11] is the adder with the best quantum cost. They are adders with better quantum cost or delay, but they have garbage outputs [49, 50].

Table 2: Comparison of adders for N-digit numbers. The comparison only includes the ripple-carry and carry-lookahead adders which do not have garbage outputs and without the use of the fault-tolerant T gates.

Circuit	**Quantum cost**	**Delay** $\triangle$	**Ancilla inputs**	**Garbage outputs**
[7]	$28N - 15W(N) - 15log(N) - 6$	$logN + logN/3 + 7$	$5N/5$	0
[8]	$26N - 15W(N) - 15log(N - 4)$	$logN + logN/3 + 2$	$5N/4$	0
[41]	$26N - 29$	$24N - 27$	0	0
[39]	$17N - 12$	$10N$	1	0
[51]	$15N$	$15N$	$3N$	0
[52]	$15N$	$10N$	N	0
[43]	$15N - 9$	$13N - 7$	0	0
[53]	$15N - 6$	$9N + 5$	0	0
[54]	$13N - 8$	$11N - 4$	0	0
[55]	$12N$	$12N$	$3N$	0
[56]	$12N$	$12N$	N	0
[56]	$12N$	$10N$	$4N$	0
[44]	$10N$	$8N$	N	0
[11]	$6N$	$4N$	N	0

The proposed circuits in this work are focused on optimizing the necessary resources of the current converters and adders acting as converters. Their details are presented in the next section.

3 Design of the new sign-magnitude to two's complement converters

We have developed three converters focused on minimizing the metrics of [28]. We propose three models:

- A first one focused on quantum simulators, which do not have noise problems. Its priorities are optimizing the quantum cost and the number of qubits.
- A second one focused on real quantum computes. Its priorities are achieving a balance between quantum cost, T-count and number of necessary qubits.
- A third one also focused on real quantum computes, but prioritizing T-count and number of necessary qubits.

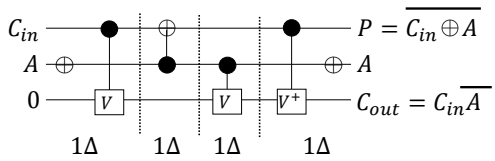


Fig. 4: Proposed sign-magnitude to two's complement converter. C_{in} is the input carry, A is the input, P is the output and C_{out} is the output carry. It has 1 auxiliary input and no garbage outputs. Its quantum cost is 6 and it has a delay of 4△.

3.1 First design: optimal quantum cost converter

We have developed a first sign-magnitude to two's complement converter focusing on the reduction of the quantum cost and the ancilla inputs. Neither the T-count nor the T-depth are taken into account since this design is focused on its use in simulators. This converter has two inputs: the digit to be converted (A) and the carry input (C_{in}). In the same way, it has two outputs: the converted digit (P) and the carry out (C_{out}). The idea behind this converter is that a sign-magnitude to two's complement conversion consists of negating a number and adding 1. That is, $\overline{\mathrm{A}} + 1$. Focusing on this specific addition, the conversion can be done digit by digit, taking into account that there is a carry which must be propagated. Since a digit only has two possibilities (0 or 1), its conversion depends on whether there is carry-in or not. Likewise, the conversion of such digit may generate carry-out or not. As there are only four possibilities (considering A and C_{in}), this can be expressed in a simple truth table as it is shown in Table 3.

Table 3: Truth table of the proposed converter.

C_{in}	A	P	C_{out}
0	0	1	0
0	1	0	0
1	0	0	1
1	1	1	0

C_{in} is the input carry, A is the input, P is the output and C_{out} is the output carry.

The proposed converter to compute the function of Table 3 has been built using Controlled-V, Controlled-V^+, $CNOT$ and Pauli-X gates. The circuit is shown in Fig. 4. It has 1 auxiliary input and no garbage outputs. It has been implemented using the properties of the Controlled-V and Controlled-V^+ gates described in Subsection 2.1. As the circuit has 1 V^+ gate and 2 V gates, the quantum cost of these gates is 3. It also uses a $CNOT$ gate, which has a quantum cost of 1, and two Pauli-X gates which also have a quantum cost of 1 each one. In conclusion, the circuit has a quantum cost of 6 and a delay of 4△. From Table 3 can be deduced that $P = \overline{C_{in}}\ \overline{A} + C_{in}A$ and $C_{out} = C_{in}\overline{A}$. On the one hand, $C_{in}\overline{A}$ can be computed using the properties of the Controlled-V^+ and Controlled-V gates. On the other hand, $\overline{C_{in}}\ \overline{A} + C_{in}A$ can be easily computed using a $CNOT$ gate. Although the $CNOT$ gate calculates the result inverted, the desired result can be achieved inverting A at the beginning with a Pauli-X gate. A is reverted at the end to avoid a garbage output.

For a better understanding of the circuit, all possible combinations and the obtained

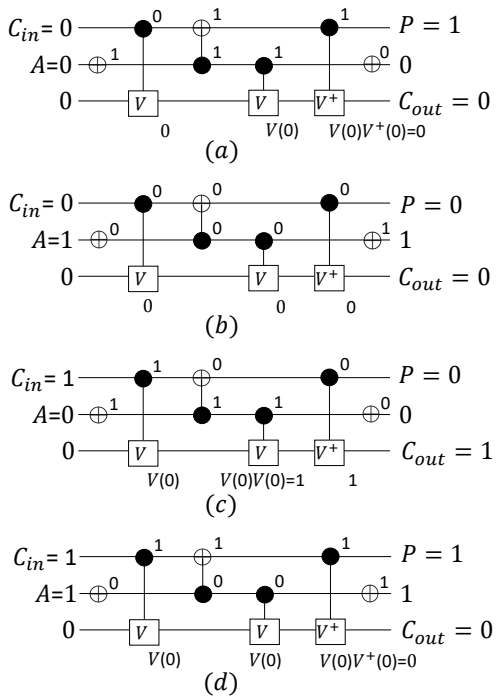


Fig. 5: For the sake of clarity, all the possible states of the proposed circuit are checked in detail, verifying that the obtained results are equal to those of Table 3. In (a), the circuit is tested for the case $C_{in} = 0, A = 0$. In (b), the circuit is tested for the case $C_{in} = 0, A = 1$. (c) shows the results of $C_{in} = 1, A = 0$. In (d), it is verified that the circuit also responds adequately for $C_{in} = 1, A = 1$.

results are shown in Fig. 5. They coincide with the results of Table 3, verifying that the circuit works properly:

- Fig. 5(a) shows the case $C_{in} = 0$, $A = 0$. The second Controlled-V gate and the Controlled-V^+ are activated. $V(A) \times V^+(A) = V^+(A) \times V(A) = A$, so C_{out} maintains the initial value (0). The $CNOT$ gate is activated, so P is inverted to 1.
- Fig. 5(b) shows the case $C_{in} = 0$, $A = 1$. Neither gates Controlled-V nor Controlled-V^+ are activated, so C_{out} maintains the initial value (0). The $CNOT$ gate is not activated, so P maintains its value.
- Fig. 5(c) shows the case $C_{in} = 1$, $A = 0$. The two Controlled-V gates are activated, so C_{out} inverts the initial value to 1. The $CNOT$ gate is activated, so P is inverted to 0.
- Fig. 5(d) shows the case $C_{in} = 1$, $A = 1$. The first Controlled-V gate and the Controlled-V^+ are activated. As $V(A) \times V^+(A) = V^+(A) \times V(A) = A$, C_{out} maintains the initial value (0). The $CNOT$ gate is not activated, so P maintains its value.

The real utility of the converter is its scalability, that is, the possibility of connecting it in cascade with other converters of the same type, to convert numbers of any size. A

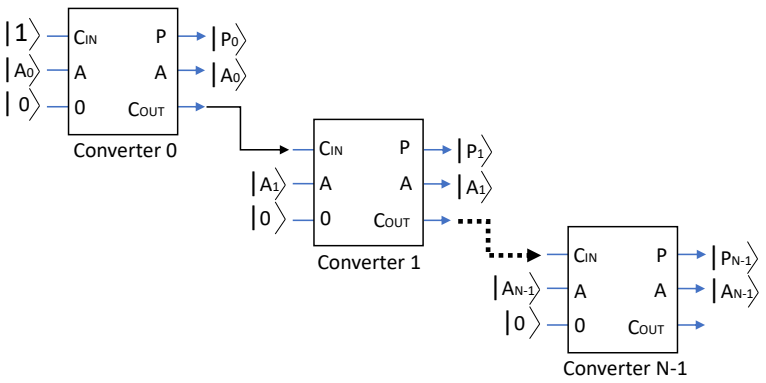


Fig. 6: Ripple-carry sign-magnitude to two's complement converters computing the conversion of an N-digit number.

ripple-carry sign-magnitude to two's complement converter can be constructed with several converters in cascade, with the carry output from each converter connected to the carry input of the next one in the chain. In the general case, N converters can compute the conversion of an N-digit number (i.e. A), as it is shown in Fig. 6. In the figure, Converter 0 computes the least significant digit of A, it means A_0, and Converter $N-1$ computes the most significant digit of A (A_{N-1}). It is important to always start the process setting the initial carry input to 1. As an example, we explain how to convert the number -5 (1101 in sign-magnitude representation). The most significant digit is 1, so the value represented is negative. The number to be converted is 1101. Therefore, $C_{in0} = 1$ and $A = 101$ ($A_2 = 1, A_1 = 0, A_0 = 1$). Given that the number to be converted has 3 digits, the ripple-carry converter will consist of 3 converters:

- The inputs of the Converter 0 are $C_{in0} = 1$ and $A_0 = 1$. According to Fig. 5(d), the results of this case are $P_0 = 1$ and $C_{out0} = 0$.
- The inputs of the Converter 1 are $C_{in1} = 0$ and $A_1 = 0$. According to Fig. 5(a), the results of this case are $P_1 = 1$ and $C_{out1} = 0$.
- The inputs of the Converter 2 are $C_{in2} = 0$ and $A_2 = 1$. According to Fig. 5(b), the results of this case are $P_2 = 0$ and $C_{out2} = 0$.

The result is given by P, that is, (1)011.

3.2 Second design: balance between quantum cost and T-count

A second circuit can be obtained from the circuit described in the previous subsection replacing its non-Clifford + T gates and optimizing the use of the T gates. It has the same number of qubits than the previous proposed converter and also no garbage outputs. The first step consists on achieving a circuit with only Clifford + T gates. In [58], a heuristic called "Initial expansion algorithm" is presented. This heuristic, which is very easy to be applied, is focused on transform a circuit into another circuit which only has Clifford + T gates. We have chosen the definition of the Controlled-V and V^+ in terms of Clifford + T gates proposed in [57] to

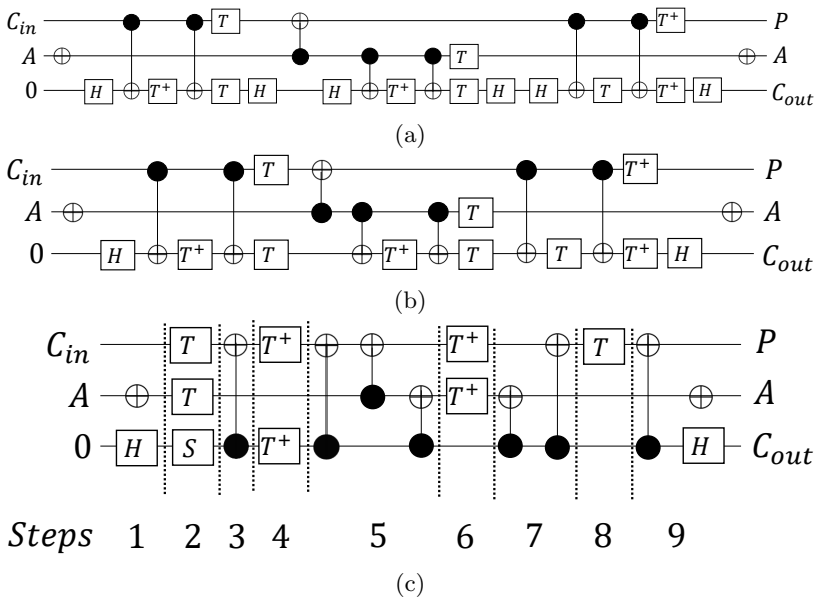


Fig. 7: Second proposed converter. (a) Resulting circuit of expanding the Controlled-V and V^+ gates in the circuit of Fig. 4 as shown in [57] using the "Initial expansion algorithm" proposed in [58]. (b) Removing some redundant Hadamard gates from the previous circuit. (c) Minimization and parallelization of the T gates according to the method described in [59].

expand these gates. The result is shown in Fig. 7a. Moreover, several Hadamard gates can be removed since they are applied consecutive, reducing the quantum cost of the circuit (Fig. 7b). Until now, we have limited ourselves to convert the original design to another one that uses only Clifford-T gates.

Finally, it is mandatory to optimize the T-count and the T-depth. The circuit can be optimized in these terms using a heuristic for the parallelization of the T gates [59]. Two T gates can be sequentially computed after several transforms, so they can be replaced by a Phase gate, reducing the T-count in one gate. At the end of the heuristic, the final circuit has a T-count of 8 and a T-depth of 4. The final circuit is shown in Fig. 7c. The quantum cost and delay of this new proposed circuit is bigger than the best ones presented in Table 5. However, to compare these circuits with the proposed one in this section in terms of quantum cost and delay is not coherent since the proposed circuit is the only one with fault tolerant capabilities.

3.3 Third design: optimal T-count and number of qubits

As it has been mentioned in Section 1, ripple-carry adders involve lesser resources than carry-lookahead adders. We can adapt the methodology of a ripple-carry adder, which computes $A + B$, to perform the operation $\overline{\mathrm{A}} + 1$. On the one hand, the resulting converter simplifies the operations of the original adder since $B_0 = 1$ and $B_i = 0$ for $i > 0$. For instance, The first carry out depends only on A_0: if A_0, then the carry out will be 0, and 1 otherwise as $A_0 + 1 = 10$ in this last case. The computation of the remaining carries also depends only of

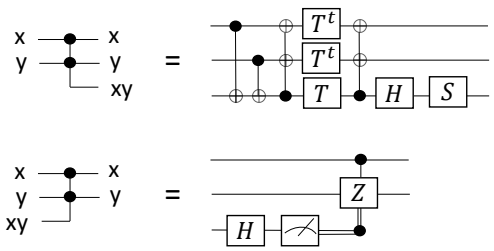


Fig. 8: Temporary logical-AND gate and its uncomputation gate [46]. The temporary logical-AND gate has a T-count of 4 and a T-count of 2. The uncomputation gate does not involves T gates.

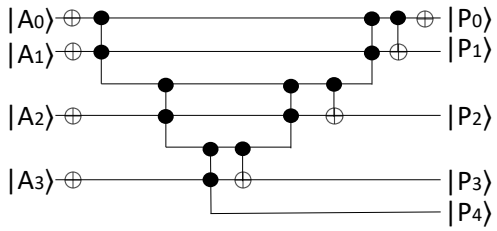


Fig. 9: Example of the third proposed converter, for the 4-digits case.

the digits of A. Therefore, the only difficulty is keeping a constant product of all the digits of A, which can be expensive if Toffoli gates are employed. On the other hand, all the qubits used to contain B can be removed, resulting in a significant reduction of necessary resources.

In [46], a new construction called temporary logical-AND gate, which performs the AND operation of two qubits into an ancilla qubit, is presented. This gate is similar to the Toffoli gate, but its T-count and T-depth are 4 and 2, respectively (whereas the T-count and T-depth of the Toffoli gate are 7 and 3, respectively. Moreover, its uncomputation does not need any T gate (the uncomputation of a Toffoli gate should be done using another Toffoli gate, so the same T-count and T-depth are required). We reproduce the temporary logical-AND gate and its uncomputation gate in Fig. 8. Using the reduced T-count and T-depth of this gate, we can achieve an optimized converter. An example of this converter for the 4-digit case is shown in Fig. 9. The circuit can be obtained for any N-digit number A following these steps:

1. For $i = 0$ to $i = N - 1$, apply a Pauli-X gate at every digit (qubit) A_i.
2. Apply a temporary logical-AND gate at A_0, A_1 over an ancilla qubit. We call this ancilla qubit as C_1, since its contains the output carry.
3. For $i = 2$ to $i = N - 1$, apply a temporary logical-AND gate at C_{i-1}, A_i over an ancilla qubit C_i.
4. For $i = N - 1$ to $i = 2$, apply a CNOT gate at C_{i-1} over A_i. Then, uncompute C_{i-1} using the uncomputation gate.
5. Apply a CNOT gate at A_0 over A_1, and finally a Pauli-X gate over A_0.

3.4 Cost analysis

The first proposed circuit in this work consists of 2 Controlled-V gates, 1 Controlled-V^+ gate, 1 $CNOT$ gate and 2 Pauli-X gates (quantum cost: 6). In the general case and assuming that N converters are necessary to compute the conversion of an N-digit number, the quantum cost is $6N$. In terms of delay and following the same metrics of [28], it has a delay of $4N\triangle$ since a gate with the quantum cost of 1 has a delay of $1\triangle$, and the Pauli-X gates can be computed in parallel with other gates, as it is shown in Fig. 4. Every converter has an auxiliary input, so the general case needs N auxiliary qubits. There are not garbage outputs in the proposed circuit. To calculate the T-count and T-depth of the Controlled-V gates we have considered the values given in [57]. Then, we conclude that the T-count and T-depth of the circuit are $9N$ and $6N$, respectively.

In this work, we have obtained the circuit directly from the truth table. However, it is also possible to use a SAT solver to quickly search the minimum quantum cost of a function with small number of intputs [60]. To assure the quality of our circuit, we have checked it using such solvers. It can be concluded that the minimum cost of implementing the function given by the proposed circuit (considering the uncomputation of the garbage outputs) is 6, so we can affirm that the proposed circuit is optimal in that term. On the other hand, the number of garbage outputs cannot be improved since it is 0. The number of ancilla inputs (1) cannot be reduced either due to the maintenance of reversibility.

Focusing now on the second circuit, it has a T-count of 8 and a T-depth of 4, which can be obtained directly in Fig. 7c using the same metrics than in the previous circuit. The number of necessary qubits is $3N$ (the same number of the first circuit). It also has no garbage outputs. It optimizes the T-count and T-depth of the first circuit into $8N$ and $4N$, respectively.

Finally, focusing on the third circuit, only steps 2 and 3 require T gates as they involve temporary logical-AND gates. According to the T-count and the T-depth of the temporary logical-AND gate, it can be concluded that this circuit requires $4N-4$ T gates. These gates are computed sequentially, so the T-depth will be $2N-1$. The circuit requires N normal inputs (the digits of the number) and $N-1$ ancilla inputs, that is, a total of $2N-1$ qubits. For the rest of the metrics we have proceeded as in the previous cases, obtaining a quantum cost of $15-13$, a delay of $12N-10$, and 0 garbage outputs.

For the sake of clarity, we show the numbers of our three proposals in Table 4.

Table 4: Comparison of our three proposed converters for N-digit numbers, in terms of the metrics proposed in [28], but also in terms of T-count and T-depth.

Circuit	**Quantum Cost**	**Delay**	**Ancilla Inputs**	**Total Qubits**	**Garbage Outputs**	**T-Count**	**T-Depth**
First Proposed Circuit	$6N$	$4N$	N	$3N$	0	$9N$	$6N$
Second Proposed Circuit	$19N$	$13N$	N	$3N$	0	$8N$	$4N$
Third Proposed Circuit	$15N-13$	$12N-10$	$N-1$	$2N-1$	0	$4N-4$	$2N-2$

Table 5: Comparison of converters for N-digit numbers.

Circuit	**Quantum cost**	**Delay** $\triangle$	**Normal inputs**	**Ancilla inputs**	**Garbage outputs**
[7]	$28N - 15w(N) - 15log(N) - 6$	$logN + logN/3 + 9$	$2N$	$5N/4$	0
[8]	$26N - 15w(N - 15log(N - 4)$	$logN + logN/3 + 4$	$2N$	$5N/4$	0
[35]	$21N - 15w(N) - 15log(N - 4)$	$logN + logN/3 + 1$	N	$5N/4 + 1$	0
[41]	$28N - 29$	$26N - 27$	$2N$	0	0
[39]	$19N - 12$	$10N + 2$	$2N$	1	0
[51]	$17N$	$15N + 2$	$3N$	$3N$	0
[52]	$17N$	$10N + 2$	$3N$	N	0
[43]	$17N - 9$	$13N - 5$	$2N$	0	0
[53]	$17N - 6$	$9N + 7$	$2N$	0	0
[54]	$15N - 8$	$11N - 2$	$2N$	0	0
[55]	$14N$	$12N + 2$	$2N$	$3N$	0
[56]	$14N$	$12N + 2$	$2N$	N	0
[56]	$14N$	$10N + 2$	$2N$	$4N$	0
[44]	$12N$	$8N + 2$	$3N$	N	0
[11]	$8N$	$4N + 2$	$3N$	N	0
[32] extended	$51(((N - 1)/3) + 1)$	$51(((N - 1)/3) + 1)$	$4((N - 1)/3 + 1)$	N	$2N$
Proposed Circuit 1	$6N$	$4N$	$2N$	N	0

The adders include two extra levels of delay (A level for the initial inversion and another to restore and avoid garbage outputs) and $2N$ extra quantum cost to act as a converters.

3.5 Cost comparison

3.5.1 Non Fault tolerant circuits

A comparison between several state-of-the-art converters and our first proposed circuit, in terms of quantum cost, delay, number of inputs, ancilla inputs and garbage outputs, is shown in Table 5. The converters presented in [32], [34] and [36] have not been included in the comparison since they are only valid for the $N = 3$ or $N = 4$ cases. We have also not included fault tolerant adders since these circuits will be compared to our fault tolerant proposals. Table 5 include all the adders described in Subsection 2.2, but acting as a converter in this case (this has extra quantum cost and delay derived to compute the inversion of all the digits of the number to be converted and to restore the number to avoid garbage outputs). [35] is a specifically designed circuit to compute the sign-magnitude to two's complement conversion. Notice that the circuits of Figs. 3 and 4 will hereinafter be referred to as [32] extended and proposed circuit, respectively.

Focusing on delay on Table 5, circuits [7], [8] and [35] are $O(logN)$, and the remaining circuits are $O(N)$. So, the fastest circuit of the comparison is [35] with a delay of $logN + logN/3 + 1\triangle$. Therefore, [35] is the best choice when speed is the priority. In terms of normal inputs (qubits which are not auxiliary), [11, 44, 55, 51] have $3N$ inputs (which are the number A to be converted), the 1 to be added to A (is 1 in the first digit and 0 in the remaining digits) and the carries. [32] extended has $4((N - 1)/3 + 1)$ inputs since it computes three digits (and the carry-in) at once. [7], [8] and the proposed circuit have $2N$ inputs. In the rest of the circuits except [35] and the proposed one, the inputs are A and also the 1 to be added. In the proposed circuit, the inputs are the digit and the carry-in. Finally, [35] optimizes the number of inputs since it was specifically designed to compute the conversion, so it only needs the number to be converted.

In terms of ancilla inputs, [7] and [8] have $5N/4$, and [35] has $5N/4 + 1$. These circuits prioritize the delay, using more qubits to achieve the highest speed. The most optimized circuits of the table have N ancilla qubits (they prioritize the reduction of the number of

necessary qubits). In terms of garbage outputs, only [32] extended contain uncomputed outputs. It has been included only to demonstrate that an optimization starting from an existing specific converter was not possible (or, at least, not trivial). As it has been mentioned, the cost of adding carry-in and carry-out to the existing converters is expensive.

The quantum cost is the most important factor when the optimization of resources is the priority, especially considering the current scarcity of resources in quantum computing [1]. In terms of quantum cost, the proposed circuit is the best choice. It improves the quantum cost in a 33% with respect to [45] (which is not included in the table since it has garbage outputs) and [11], which are the circuits currently available with the best quantum cost. [44] has a quantum cost of $12N$, so the proposed circuit improves the quantum cost in a 50%. Circuits [7, 8, 35] have a higher quantum cost than the others since they decrease their delays at the expense of the quantum cost. Finally, [32] extended has an expensive quantum cost $(51(((N-1)/3)+1))$, as it has been mentioned. Focused on quantum cost, the proposed circuit overcomes the other options, therefore is the best option when the optimization of resources is mandatory.

Focusing on our proposal, we have demonstrated that our converter is the best circuit in terms of quantum cost. Moreover, Table 4 have shown that it outperforms the rest of the circuits in terms of delay, with the exception of the first three converters. However, these three circuits have a very expensive quantum cost, being non-viable in current quantum computers and simulators. In the same way, only the circuit of [35] improves our proposal in terms of normal inputs, but again at the cost of using three times more quantum gates than our proposal. In terms of ancilla inputs, several circuits outperform ours, but once again at the cost of using (at least) two times more gates. Our focus was to achieve a converter optimized in terms of quantum cost, but without neglecting the relevance of the rest of the metrics. Table 4 have shown that our proposal is the most balanced converter in all metrics.

3.5.2 Fault tolerant circuits

This comparison is focused on the T-count, T-depth and the number of required qubits to perform the operation. We mentioned that there is no fault tolerant converters in the literature. Fortunately, there are several fault tolerant adders. On the one hand, the adder presented in [46] is the best adder of the state-of-the-art in terms of T-count. On the other hand, [47] proposes four new adders. Since none of the adders included in [47] outperform the adder of [46] in terms of T-count, we have selected the best of the four adders in terms of T-depth. Therefore, Table 6 shows a comparison between our converters and the best adders in terms of T-count, T-depth and number of qubits. Our third circuit outperforms the rest of the circuits in terms of T-count, with $4N-4$. The second best circuit in T-count is the circuit of [46], followed by our second proposal. The worst circuit in terms of T-count is [47], with a value of $16N-8w(N)-8log(N)-4$ (where $w(n)=n-\sum_{i=1}^{\infty}(\frac{n}{2^y})$).

The circuit of [47] is the best option in terms of T-depth, with $4log(N)+2log(\frac{2N}{3})+3$. The second best circuit is our third proposal, with $2N-2$, followed by [46] with a T-depth of $2N$, and by our second proposal with $4N$. Finally, in terms of number of qubits, the best option is again our third circuit, with only $2N-1$. The second best circuit is [46] with $3N-1$, and the third is our second proposal with $3N$. The worst is terms of necessary qubits is [47], which we have already seen that it sacrifices T-count and qubits to improve its T-depth.

Table 6: Comparison of converters for N-digit numbers, in terms of T-count, T-depth and number of necessary qubits.

Circuit	**T-count**	**T-depth**	**Number of qubits**
Second Proposed Circuit	$8N$	$4N$	$3N$
[46]	$4N$	$2N$	$3N-1$
[47]	$16N-8w(N)-8log(N)-4$	$4log(N)+2log(\frac{2N}{3})+3$	$6N-2w(N)-2log(N)$
Third Proposed Circuit	$4N-4$	$2N-2$	$2N-1$

To finish, we can conclude with the following classification:

- The best quantum cost: the first proposed circuit, with an improvement of a 33% with respect to the best current converters.
- The best delay: [35].
- Less number of inputs: [35] and the third proposed circuit.
- No garbage outputs: all the included except [32] extended.
- The best in terms of T-count: the third proposed circuit.
- The best in terms of T-depth: [47].
- Less number of qubits: The third proposed circuit.

4 Conclusion

In this work, we have presented the design of several scalable reversible sign-magnitude to two's complement converters. The first one optimizes the quantum cost, being also the most balanced in the rest of the metrics defined in [28]. Obtained results have shown that the quantum cost of our first proposed circuit improves a 33% with respect to the state-of-the-art circuits based on reversible gates and quantum adders. We have also presented a version of this circuit (the second proposal), achieving a balance between quantum cost and T-count. Additionally, we have designed an specific version (third proposal) that optimizes ancillary qubits and T-count. This last version outperforms the best fault tolerant circuits in the state-of-the-art. An advantage of our proposals is that they do not contain any garbage output, therefore they could be entangled with any other reversible circuit which needed to operate with two's complement. Moreover, we have demonstrated that adding carry-in and carry-out to the current best converter (in terms of quantum cost), which is limited to the case $N = 4$, is not a competitive option. A comparison, using a solid metric, between the proposed circuits and the best converters has been carried out in terms of quantum cost, delay, number of inputs, auxiliary qubits, garbage outputs, T-count, T-depth and number of qubits.

Additionally, we have analyzed the state-of-the-art converters and adders, giving valuable information summarized in tables which will be very useful for interested researchers in order to select the correct adder to their own applications.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science throughout Project RTI2018-095993-B-I00 and by the European Regional Development Fund (ERDF). F. Orts is supported by an FPI Fellowship (attached to Project TIN2015-66680-C2-1-R) from the Spanish Ministry of Education.

References

1. Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information 10th edition, 2017.
2. Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.
3. Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.
4. Hans Primas. *Chemistry, quantum mechanics and reductionism: perspectives in theoretical chemistry*, volume 24. Springer Science & Business Media, 2013.
5. Salvador Elías Venegas-Andraca. Introductory words: Special issue on quantum image processing published by quantum information processing. *Quantum Information Processing*, 14(5):1535–1537, 2015.
6. Elías F Combarro, José Ranilla, and Ignacio F Rúa. A quantum algorithm for the commutativity of finite dimensional algebras. *IEEE Access*, 7:45554–45562, 2019.
7. Thomas G Draper, Samuel A Kutin, Eric M Rains, and Krysta M Svore. A logarithmic-depth quantum carry look-ahead adder. *Quantum Information & Computation*, 6(4):351–369, 2006.
8. Himanshu Thapliyal, HV Jayashree, AN Nagamani, and Hamid R Arabnia. Progress in reversible processor design: a novel methodology for reversible carry look-ahead adder. In *Transactions on Computational Science XVII*, pages 73–97. Springer, 2013.
9. KVRM Murali, Neeraj Sinha, TS Mahesh, Malcolm H Levitt, KV Ramanathan, and Anil Kumar. Quantum-information processing by nuclear magnetic resonance: Experimental implementation of half-adder and subtractor operations using an oriented spin-7/2 system. *Physical Review A*, 66(2):022313, 2002.
10. F Orts, G Ortega, and Ester M Garzón. A faster half subtractor circuit using reversible quantum gates. *Baltic Journal of Modern Computing*, 7(1):99–111, 2019.
11. Himanshu Thapliyal. Mapping of subtractor and adder-subtractor circuits on reversible quantum gates. In *Transactions on Computational Science XXVII*, pages 10–34. Springer, 2016.
12. Heumpil Cho and Earl E Swartzlander Jr. Adder and multiplier design in quantum-dot cellular automata. *IEEE Transactions on Computers*, 58(6):721–727, 2009.
13. Majid Haghparast, Somayyeh Jafarali Jassbi, Keivan Navi, and Omid Hashemipour. Design of a novel reversible multiplier circuit using hng gate in nanotechnology. In *World Appl. Sci. J.* Citeseer, 2008.
14. Alireza Khosropour, Hossein Aghababa, and Behjat Forouzandeh. Quantum division circuit based on restoring division algorithm. In *2011 Eighth International Conference on Information Technology: New Generations*, pages 1037–1040. IEEE, 2011.
15. F Orts, G Ortega, and Ester M Garzón. A quantum circuit for solving divisions using grover's search algorithm. In *Proc. 18th Int. Conf. Comput. Math. Method. Sci. Eng*, 2018.
16. Sarah Harris and David Harris. *Digital design and computer architecture: ARM edition*. Morgan Kaufmann, 2015.
17. Michael Kirkedal Thomsen, Robert Glück, and Holger Bock Axelsen. Reversible arithmetic logic unit for quantum arithmetic. *Journal of Physics A: Mathematical and Theoretical*, 43(38):382002, 2010.
18. Andreas Klappenecker and Martin Rotteler. Discrete cosine transforms on quantum computers. In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, pages 464–468. IEEE, 2001.

19. Mihai Udrescu, Lucian Prodan, and Mircea Vlăduţiu. Implementing quantum genetic algorithms: a solution based on grover's algorithm. In *Proceedings of the 3rd Conference on Computing Frontiers*, pages 71–82. ACM, 2006.
20. Fei Yan, Abdullah M Iliyasu, Yiming Guo, and Huamin Yang. Flexible representation and manipulation of audio signals on quantum computers. *Theoretical Computer Science*, 752:71–85, 2018.
21. S Manjula Gandhi, J Devishree, and S Sathish Mohan. A new reversible smg gate and its application for designing two's complement adder/subtractor with overflow detection logic for quantum computer-based systems. In *Computational Intelligence, Cyber Security and Computational Models*, pages 259–266. Springer, 2014.
22. Michael Nachtigal, Himanshu Thapliyal, and Nagarajan Ranganathan. Design of a reversible floating-point adder architecture. In *2011 11th IEEE International Conference on Nanotechnology*, pages 451–456. IEEE, 2011.
23. Ajinkya Borle and Samuel J Lomonaco. Analyzing the quantum annealing approach for solving linear least squares problems. In *International Workshop on Algorithms and Computation*, pages 289–301. Springer, 2019.
24. Maii T Emam and Layle AA Elsayed. Reversible full adder/subtractor. In *2010 XIth international workshop on symbolic and numerical methods, modeling and applications to circuit design (SM2ACD)*, pages 1–4. IEEE, 2010.
25. Jenil Jain and Rahul Agrawal. Design and development of efficient reversible floating point arithmetic unit. In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 811–815. IEEE, 2015.
26. Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.
27. Davide Castelvecchi. Ibm's quantum cloud computer goes commercial. *Nature News*, 543(7644):159, 2017.
28. Majid Mohammadi and Mohammad Eshghi. On figures of merit in reversible and quantum logic designs. *Quantum Information Processing*, 8(4):297–318, 2009.
29. Edgard Muñoz-Coreas and Himanshu Thapliyal. T-count optimized design of quantum integer multiplication. *arXiv preprint arXiv:1706.05113*, 2017.
30. Fang Zhang and Jianxin Chen. Optimizing t gates in clifford+ t circuit as pi/4 rotations around paulis. *arXiv preprint arXiv:1903.12456*, 2019.
31. Israel Koren. *Computer arithmetic algorithms*. AK Peters/CRC Press, 2001.
32. Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, Chitrita Chaudhuri, and Atal Chaudhuri. A reversible approach to two's complement addition using a novel reversible tcg gate and its 4 dot 2 electron qca architecture. *Microsystem Technologies*, pages 1–11, 2018.
33. Vandana Shukla, OP Singh, GR Mishra, and RK Tiwari. Design of a 4-bit 2's complement reversible circuit for arithmetic logic unit applications. In *The International Conference on Communication, Computing and Information Technology (ICCCMIT), Special Issue of International Journal of Computer Applications*, pages 1–5, 2012.
34. Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, and Atal Chaudhuri. A novel reversible two's complement gate (tcg) and its quantum mapping. In *Devices for Integrated Circuit (DevIC), 2017*, pages 252–256. IEEE, 2017.
35. F. Orts, G. Ortega, and E. M. Garzón. An optimized quantum circuit for converting from sign–magnitude to two's complement. *Quantum Information Processing*, 18(11):332, Sep 2019.
36. D Michael Miller, Robert Wille, and Zahra Sasanian. Elementary quantum gate realizations for multiple-control toffoli gates. In *2011 41st IEEE International Symposium on Multiple-Valued Logic*, pages 288–293. IEEE, 2011.
37. Kamalika Datta, Indranil Sengupta, and Hafizur Rahaman. A post-synthesis optimization technique for reversible circuits exploiting negative control lines. *IEEE Transactions on Computers*, 64(4):1208–1214, 2014.
38. Charles H Bennett. Logical reversibility of computation. *IBM journal of Research and Develop-*

ment, 17(6):525–532, 1973.
39. Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit. *arXiv preprint quant-ph/0410184*, 2004.
40. Igor L Markov and Mehdi Saeedi. Constant-optimized quantum circuits for modular multiplication and exponentiation. *Quantum Information & Computation*, 12(5–6):361–394, 2012.
41. Yasuhiro Takahashi and Noboru Kunihiro. A linear-size quantum circuit for addition with no ancillary qubits. *Quantum Information & Computation*, 5(6):440–448, 2005.
42. Yasuhiro Takahashi and Noboru Kunihiro. A fast quantum circuit for addition with few qubits. *Quantum Information & Computation*, 8(6):636–649, 2008.
43. Yasuhiro Takahashi, Seiichiro Tani, and Noboru Kunihiro. Quantum addition circuits and unbounded fan-out. *Quantum Information & Computation*, 10(9):872–890, 2010.
44. Feng Wang, Mingxing Luo, Huiran Li, Zhiguo Qu, and Xiaojun Wang. Improved quantum ripple-carry addition circuit. *Science China Information Sciences*, 59(4):042406, 2016.
45. Dmitri Maslov, Gerhard W Dueck, D Michael Miller, and Camille Negrevergne. Quantum circuit simplification and level compaction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(3):436–444, 2008.
46. Craig Gidney. Halving the cost of quantum addition. *Quantum*, 2:74, 2018.
47. Himanshu Thapliyal, Edgard Muñoz-Coreas, and Vladislav Khalus. T-count and qubit optimized quantum circuit designs of carry lookahead adder. *arXiv preprint arXiv:2004.01826*, 2020.
48. Md Saiful Islam, Muhammad Mahbubur Rahman, Zerina Begum, and Mohd Zulfiquar Hafiz. Fault tolerant reversible logic synthesis: Carry look-ahead and carry-skip adders. In *Advances in Computational Tools for Engineering Applications, 2009. ACTEA'09. International Conference on*, pages 396–401. IEEE, 2009.
49. Nusrat Jahan Lisa and Hafiz Md Hasan Babu. Design of a compact reversible carry look-ahead adder using dynamic programming. In *2015 28th International Conference on VLSI Design*, pages 238–243. IEEE, 2015.
50. Maryam Rahmati, Monireh Houshmand, and Masoud Houshmand Kaffashian. Novel designs of a carry/borrow look-ahead adder/subtractor using reversible gates. *Journal of Computational Electronics*, 16(3):856–866, 2017.
51. Shigeru Yamashita, Shin Minato, and D Michael Miller. DDMF: An efficient decision diagram structure for design verification of quantum circuits under a practical restriction. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91(12):3793–3802, 2008.
52. Rajkumar Sarma and Ritika Jain. Quantum gate implementation of a novel reversible half adder and subtractor circuit. In *2018 International Conference on Intelligent Circuits and Systems (ICICS)*, pages 72–76. IEEE, 2018.
53. Himanshu Thapliyal and Nagarajan Ranganathan. A new reversible design of bcd adder. In *2011 Design, Automation & Test in Europe*, pages 1–4. IEEE, 2011.
54. Himanshu Thapliyal and Nagarajan Ranganathan. Design of efficient reversible logic-based binary and bcd adder circuits. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 9(3):17, 2013.
55. William NN Hung, Xiaoyu Song, Guowu Yang, Jin Yang, and Marek Perkowski. Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(9):1652–1663, 2006.
56. AN Nagamani, S Ashwin, and Vinod Kumar Agrawal. Design of optimized reversible binary adder/subtractor and BCD adder. In *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pages 774–779. IEEE, 2014.
57. Mathias Soeken, D Michael Miller, and Rolf Drechsler. Quantum circuits employing roots of the pauli matrices. *Physical Review A*, 88(4):042322, 2013.
58. D Michael Miller, Mathias Soeken, and Rolf Drechsler. Mapping ncv circuits to optimized clifford+t circuits. In *International Conference on Reversible Computation*, pages 163–175. Springer,

2014.
59. Xianya He, Zhijin Guan, and Fei Ding. The mapping and optimization method of quantum circuits for clifford+ t gate. *Journal of Applied Mathematics and Physics*, 7(11):2796–2810, 2019.
60. Daniel Große, Robert Wille, Gerhard W Dueck, and Rolf 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)*, pages 214–219. IEEE, 2008.