Journal of Network and Computer Applications 170 (2020) 102810

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

A review on reversible quantum adders

F. Orts[a,*], G. Ortega[a], E.F. Combarro[b], E.M. Garzón[a]

[a] Group of Supercomputation-Algorithms, Dpt. of Informatics, Univ. of Almería, ceiA3, 04120, Almería, Spain
[b] Computer Science Department, Univ. of Oviedo, 33007, Oviedo, Spain

ARTICLE INFO

Keywords:
Reversible adder
Quantum computing
Reversible circuit
Adder

2010 MSC:
00-01
99-00

ABSTRACT

Reversible adders are essential circuits in quantum computing systems. They are a fundamental part of the algorithms implemented for such systems, where Shor's celebrated factoring algorithm is one of the most prominent examples in which reversible arithmetic is needed. There is a wide variety of works in the existing literature which tackle the design of an adder for quantum systems, and today there is still a great interest in the creation of new designs and the perfection of the existing ones. Similar to how it happens in classical digital systems, there are different methodologies to approach the addition using reversible circuits. Some methodologies focus on minimizing the necessary resources, others on optimizing computing time, etc. In this work we analyze the reversible adders in the state-of-the-art for quantum computing, classifying them according to their type, and finally, comparing each other using referenced and validated metrics that allow highlighting the strengths and weaknesses of each adder.

1. Introduction

Reversible computation was first considered in the pioneering works of Landauer (1961), Lecerf (1963) and Bennett (1973) in the context of the energetic cost of computational operations. These authors unveiled deep connections between the thermodynamics of computation (in particular, the minimum amount of heat that a physical computing machine needs to dissipate per instruction) and the logical irreversibility of some operations. Surprisingly enough, it was discovered that the only computational task that implies an energy consumption is information erasure. Thus, in principle, computations may be physically executed without using energy as long as all operations are kept reversible and no information is lost in the process.

These profound results motivated further studies of, among others, Fredkin and Toffoli (Toffoli, 1980; Fredkin and Toffoli, 1982), who showed how any function computed by a logical circuit can be also computed by a reversible circuit. The key element is the existence of reversible gates that are universal in the sense that they can be used to simulate any other possible logic gate (reversible or not). With them, any circuit can be transformed in a reversible one with only a linear increase in the number of wires and gates (Fredkin and Toffoli, 1982). This opens the possibility of using reversible circuits in order to decrease the energetic consumption of computations, a topic that has gained interest in recent years (Cohen et al., 2016; Anamika, 2018; Chaves et al., 2018; Sahu et al., 2019).

Interest in reversible computation in general, and in reversible gates in particular, also comes from an intimate connection with quantum computing (Nielsen and Chuang, 2011). Quantum computing is a computational paradigm that exploits the physical properties of subatomic particles in order to achieve speedups in solving computational problems (Zhou et al., 2018; Zhang et al., 2018). Far from being solely a theoretical model, several quantum computer prototypes have been constructed in recent years (Linke et al., 2017; Michielse et al., 2017; Neill et al., 2018). In fact, Google has recently reported solving, with a quantum computer, a problem that would be unfeasible to solve with only classical resources, thus achieving the so-called quantum supremacy (Arute et al., 2019).

The main model of quantum computing is that of quantum circuits, in which logical gates are replaced with quantum ones (Nielsen and Chuang, 2011). These gates must obey the laws of quantum physics and, as a consequence, they are always reversible. Reversibility is therefore no longer the interesting energy saving option that we have described: it is now a fundamental requirement of quantum computing. This quantum paradigm requires us to reversibly implement even the most trivial algorithms that exist in classical computing. In fact, classical reversible gates such as the Toffoli gate play a very important role in quantum computing since they can be used, in combination with a few others, to approximate any possible quantum circuit (Shi, 2003). But in general,

* Corresponding author.
E-mail address: francisco.orts@ual.es (F. Orts).

https://doi.org/10.1016/j.jnca.2020.102810
Received 7 November 2019; Received in revised form 11 July 2020; Accepted 14 August 2020
Available online 28 August 2020

this conversion to a reversible methodology is not trivial, and implies an increase in the necessary resources compared to the classic counterpart (with the consequent effort to optimize their use), and even the search for more efficient alternative approaches from the point of view of reversibility (Vartiainen et al., 2004; Fowler et al., 2004).

Circuits for performing the addition are especially relevant for several quantum algorithms that achieve a speedup over the best known classical methods. Chief among them are Shor's algorithms, which can famously factor numbers and compute discrete logarithms in polynomial time (Shor, 1994), with momentous implications for classical cryptographic protocols such as the RSA cryptosystem (Rivest et al., 1978) or Diffie-Hellman key exchange (Diffie and Hellman, 1976). For instance, the most computationally intensive part of the algorithm for integer factorization is the modular exponentiation circuit. The most usual approach to compute the modular exponentiation is to use modular multiplier circuits, which are constructed using adders (Pavlidis and Gizopoulos, 2014). Although classically tractable, the design of the arithmetic part of the method usually requires considerable ingenuity in order to minimize the number of gates used and reduce the operational error, especially because all the operations must be conducted in a reversible way, making actual implementations highly non-trivial, as we have mentioned previously.

If checking that reversible adders are used in the most computationally critical part of the probably most important quantum algorithm were not enough to highlight the importance of such circuits, there are more examples. In addition to Shor's algorithms, quantum methods for achieving a quadratic speedup over classical algorithms in search and detection tasks have been proposed, most notably Grover's algorithm (Grover, 1996) and quantum walks (Venegas-Andraca, 2012). Although, in general, these methods do not involve arithmetical operations, they use a quantum oracle that is problem-depending and that may, in some cases, benefit from optimized reversible circuits for addition, as for instance when algebraic structures are involved (Combarro et al., 2019a; Combarro et al., 2019b).

Then, to build a circuit that implements any of these algorithms, is it necessary to design a reversible adder? Are there alternatives already implemented? In the literature, there is a wide variety of reversible circuits for basic arithmetic operations such as addition or multiplication. However, it is not always easy to analyze or compare them, because, on the one hand, the reported figures of merit are not consistent from one author to another and, on the other, not all parameters that are of potential interest are always clearly acknowledged. How can we know if a circuit is the right one for us if we do not have all of its information? How do we know that there is no better one?

For these reasons, together with the above mentioned connections of reversible circuits to computation energy reduction and to quantum computing, we think that a thorough, exhaustive, clear and impartial review of the existing reversible circuits for binary addition is needed. A review that seeks and establishes suitable metrics to accurately and verifiably measure a reversible circuit. A review that finds and analyzes the state-of-the-art adders based on these metrics, conveniently and visually offering all this information to anyone interested in using a reversible adder. A review that, in summary, is a reliable database of reversible adders. In this work, we aim to provide such a review, with special emphasis on being consistent on the parameters under which the circuits are evaluated and on highlighting their particular merits and flaws. We report the analysis of more than 40 references on reversible adders, clearly classifying them according to their different types and studying all their relevant parameters (including delay, quantum cost and the presence of garbage outputs). We also summarize all the pertinent information in several tables that interested researchers can quickly refer to in order to select the adder that is more suitable for their needs and provide original figures that exemplify some of the most prominent reversible adders for some values of their inputs.

The rest of the paper is organized as follows. In Section 2, we introduce and explain the different metrics that will be used to compare all the reversible adders studied in this work (Section 2.1), explaining how we have used them to carry out the review (Section 2.2). Since the adders are analyzed and compared based on their methodology, the types of adders and their main characteristics are presented in Section 3. Section 4 reviews the reversible adders that have been proposed in the literature, paying attention first to half-adders (Subsection 4.1), then to full adders (Subsection 4.2) and, finally, to carry propagate adders (Subsection 4.3, which includes ripple-carry adders and carry-lookahead adders). The comparison of all these adders is carried out in Section 5, where we also provide summary tables for quick reference of our findings in each category. Finally, in Section 6, we raise some conclusions of our study.

2. Metrics

2.1. Choice and justification of metrics

In the classical, non-reversible setting, measuring the complexity of a digital circuit is usually straightforward. A set of universal gates (for instance, AND, OR and NOT or just NAND) is fixed and the circuit complexity can be computed as the number of gates plus the number of bits that are needed to implement it together with a measure of its depth (which captures how many gates can be executed in parallel). When dealing with reversible circuits, in addition to considering the number of gates and the depth of the circuit, it is also important to take into account other aspects, such as the presence of garbage outputs. Also, as previously mentioned, one of the most important applications of reversible circuits comes from its use in quantum computing, something that affects the gates that can used to decompose the circuits. For these reasons, in this section we clearly define the parameters that will be used throughout the paper in order to study the complexity of reversible adders.

There are a large number of adder circuits available for quantum computing, as will be seen in this review. They all have a common goal: to make the addition of two numbers as efficient as possible. However, the concept of efficiency often changes among the authors of these circuits. And most importantly, each author frequently measures his circuit using the metrics he considers appropriate or even metrics defined by himself. Comparing adders becomes a tedious task because each circuit has been evaluated differently and therefore their metrics cannot be directly compared. We want to illustrate this problem with a specific example. Li et al. presented in Mohammadi et al. (2020) an adder which involves 28 quantum gates to perform and addition between two 5-digit binary numbers. On the other hand, Gidney presented an adder that needs 29 gates to perform the same operation (Gidney, 2018). However, none of them mentioned this information in their results. Li et al. evaluated their circuit in terms of the *quantum cost* and *delay*, while Gidney measured his circuit in terms of the *T-count*. This example reveals the difficulty in comparing the different quantum adders and the need to carry out a comparative study according to a wide and recognized set of characteristic parameters associated with quantum circuits.

The objective is to measure and to compare the existing adders using a common methodology that allows a direct comparison between them, also avoiding differences in the nomenclature. For instance, the quantum cost of a circuit is defined in several works as the number of gates which composes a circuit. According to this, a circuit which consists of 2 Toffoli gates has the same quantum cost than other circuit which consists of 2 CNOT gate s. Taking into account that a Toffoli gate is composed of 2 CNOT gates and other 3 gates (Nielsen and Chuang, 2011), this definition of quantum cost is imprecise. Moreover, an entire personalized circuit built with 5 Toffoli gates could be defined as a *novel reversible gate*, being its quantum cost 1. Comparing this new gate with a circuit which has 2 Toffoli gates would show that the first one has a quantum cost of 1 and the second one a quantum cost of 2.

This review is focused on the digital and logic levels of the adders. Therefore, exact and verifiable metrics at these levels are desirable. For these reasons, the metrics defined in Mohammadi et al. (2009) are followed. Four parameters are defined in Mohammadi et al. (2009) to evaluate reversible circuits:

- Quantum Cost (*QC*): the quantum cost of a circuit or a $X \times X$ gate is defined as the number of the 1×1 and 2×2 gates which composes it. The quantum cost of 1×1 and 2×2 gates is 1. This is a sensible metric, since we are mainly interested in the possibility of using arithmetical reversible circuits in quantum computing and most quantum computers use only 1×1 and 2×2 gates as primitives.
- Delay (*D*): the delay of a circuit defines its speed. A higher delay implies that a circuit is slower. $\triangle$ is the unit of delay defined in Mohammadi et al. (2009). 1×1 and 2×2 gates have a delay of $1\triangle$. The delay of a circuit or a $X \times X$ gate is defined by the number of 1×1 or 2×2 which must be computed sequentially. Therefore, if 2 or more gates can be computed in parallel, the delay will be determined by the delay of the slowest gate. To facilitate the evaluation of the delay, several schematic diagrams of this work graphically analyze the steps to complete the corresponding specific operations.
- Number of auxiliary Inputs (*I*): inputs which are set to a constant value (usually 0 or 1) and are used to do auxiliary operations.
- Garbage Outputs (*GO*): outputs which cannot be used at the end of the circuit since they have useless values. Garbage outputs must be reversibly removed (uncomputed) or these outputs may not be used later, which would result in a waste of resources. An output which is uncomputed to its original (and known) value is not considered as a garbage output. Uncomputing garbage outputs is especially important if the circuits are to be used in quantum computations, for garbage outputs can prevent the interference that quantum algorithms need to work properly.

According to Mohammadi et al. (2009), the quantum cost and delay of the basic gates used by the adders studied in this work are shown in Table 1, and their symbols in Fig. 1. Several gates are only used in specific adders, and they are analyzed along such adders.

The final idea is to show as much useful information as possible about a circuit (using the same metrics across all circuits to enable comparisons), understanding that there is no single better parameter. For example, on a machine with few resources, the general interest might be to reduce the number of qubits and the quantum cost; while in a machine that has more resources the interest could be to reduce the delay. However, we recognize that these metrics are not perfect and that it needs to be supplemented in some aspects. First, and as described below, there are various methodologies for performing addition. That is why we have considered it appropriate to classify and compare the adders according to their methodology instead of making a single comparison. The types of adders are explained in the next section. Second, there is a growing interest in implementing adders that allow the use of error detection and correction codes. These adders suffer from an increase in their metrics, but they have the advantage that they allow such error handling. In the review, we considered it convenient to indicate which adders have this capacity. It is important to mention that in these cases, the implementation of the gates in Table 1 may be different, increasing the quantum cost and the delay due to, usually, the incorporation of *T* gates. In relation to this matter, we also want to remark the work done in Gidney (2018), which is focused on improving the number of *T* gates needed to build *N*-bit adders.

Table 1
Gates and their quantum cost and delay.

Gate	QC	D
Pauli-*X*	1	1
V	1	1
V^+	1	1
Feynman/*CNOT*	1	1
Controlled-*V*	1	1
Controlled-V^+	1	1
Peres (Hung et al., 2006)	4	4
Toffoli (Nielsen and Chuang, 2011)	5	5

There is a third point to consider. As it has been mentioned, this work is focused on the logic level of the adders. However, it must be remarked that behind this level there are several physical realizations of these reversible gates and circuits like quantum computation, optic computation, quantum-dot cellular automata or ultra low power *VLSI* design (Thapliyal and Ranganathan, 2010). Each of these technologies has its own rules and limitations, which are out of the scope of this paper. For instance, we use the version of the Toffoli gate described in Nielsen and Chuang (2011) (except for several adders focused on error detection) since it optimizes the quantum cost and delay. Nevertheless, in linear optics, it is more important to optimize the number of controlled-unitary gates since the CNOT gate can only be probabilistically implemented (Orts et al., 2019). In these terms, versions of the Toffoli gate like the presented in Lanyon et al. (2009) and Lemr et al. (2015) are better options than the one described in Nielsen and Chuang (2011) since they are focused in reducing the number of controlled gates.

2.2. Review methodology

In this review, we have tried to analyze all the adders published at the time of writing these lines. However, it could be possible that we did not notice the existence of some adders due to the enormous amount of related works, which sometimes include the design of adders as part of a larger circuit without indicating it externally. Therefore, the existence of such adders goes unnoticed by anyone who does not read the article in depth. On the other hand, we have tried to make a thorough review in terms of the metrics described in the previous subsection. We have not limited ourselves to gather the information described in each work, but we have 1) implemented and tested the corresponding adder, and 2) measured the circuit using the proposed metrics.

For the implementation and testing of each adder, we have used the ProjectQ simulator, an open-source software framework for quantum computing (Steiger et al., 2018). The circuits have been implemented in Python under this framework and subjected to software tests to verify their correct operation. On the other hand, the measurement in terms of the metrics of Mohammadi et al. (2009) has also been done in Python over the circuits taking into account the following:

- The quantum cost can be easily measured setting a weight for each circuit and multiplying the number of gates of each type by its weight.
- The delay can be measured by dividing the circuit into levels in which no qubit acts twice. The delay of each level is given by the gate with the greatest weight.
- To count the number of ancilla inputs is trivial.
- The number of garbage outputs is measured by labeling the qubits which are not used to contain the result, and checking if they have been reverted symmetrically.

Some circuits offer designs adaptable to variable data size. In these cases, the circuit has been implemented in a way that dynamically adapts to the size of the input data. Thus it is possible to obtain the corresponding equation to each metric since the part to repeat of each circuit to increase it by each digit is perfectly defined.

Finally, we have made a comparison with the information obtained, gathering this information in tables to facilitate both its understanding and its use. In order not to make the comparison unnecessarily long, some of the analyzed adders have not been included. The main reason for discarding is the presence of garbage outputs in circuits that do not improve in any metric to those that do not present garbage outputs.

2.2.1. About the implementation of functions

Several types of adders are presented in the next section. The first

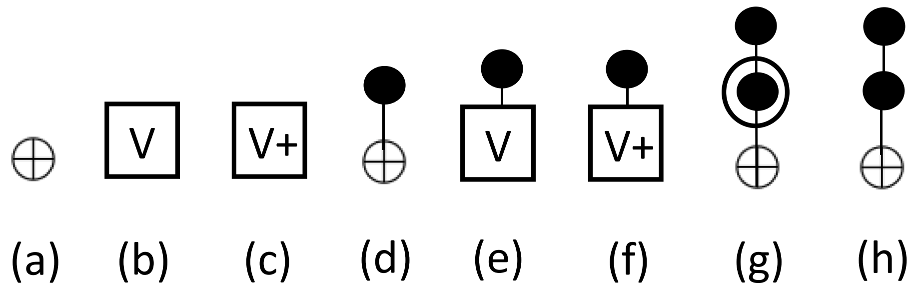


Fig. 1. Gates symbols: (a) Pauli-X, (b) V, (c) V^+, (d) Feynman/*CNOT*, (e) Controlled-V, (f) Controlled-V^+, (g) Peres and (h) Toffoli.

two, called *half adder* and *full adder*, are functions that implement a truth table of 2 and 3 inputs, respectively (see Tables 2 and 3). However, the implementation of these small functions has no merit: since we have the optimal implementations for the gates described in Table 1, it is possible to determine the optimal design of these functions in terms of the metrics described in this work using a SAT solver (Große et al., 2008). However, for the completeness of the review, we have included the analysis corresponding to the half and full adders.

2.2.2. *About error detection and correction codes*

We have mentioned that when analyzing an adder, we indicate whether or not it is fault-tolerant. However, an equivalent but fault-tolerant circuit can be obtained from any adder presented in this review by following these steps:

1. To apply the "Initial expansion algorithm" presented in Miller et al. (2014) to map the adder into a Clifford + T Circuit.
2. To remove redundant gates if necessary.
3. To minimize and parallelize the T gates according to the method described in He et al. (2019).

This review focuses on finding, analyzing, and comparing the work done by authors, so we do not make this adaptation in the circuits. Therefore, when in this review it is indicated that a circuit is fault-tolerant, it is because the authors of the adder have oriented its methodology to optimize it in these terms. In other words, authors present a circuit already prepared for fault-tolerance.

3. Types of adders

Addition is one of the basic operations in digital systems (Harris and Harris, 2015). Despite its apparent simplicity, there are a wide variety of ways to implement an adder. Since the review analyzes and catalogs the adders according to their type, it is important to make clear what each of the different types of adders consists of.

We have followed the terminology and classification order of adders described in Harris and Harris (2015) for classical adders:

- The *half adder* is the simplest case of an adder. This kind of circuit has two inputs: two digits A and B. Its objective is the computation of $A + B$. Notice that the result of the half adder needs two digits as the case $1 + 1$ returns 10. Therefore, the half adder has two outputs: S, which contains the least significant digit of the addition, and C_{out}, which contains the most significant digit (usually called carry out).

Table 2
Truth table of the half adder.

A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 3
Truth table of the full adder.

C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

  Table 2 shows the truth table of the half adder. As consequence, it can be established that $C_{out} = AB$ and $S = A \oplus B$.
- A *full adder* is similar to a half adder, but accepting the carry in, C_{in}, as an input. Therefore, a full adder has 3 inputs (A, B and C_{in}) and 2 outputs S and C_{out}. According to its truth table (Table 3), it can be deduced that $S = A \oplus B \oplus C_{in}$ and $C_{out} = AB + AC_{in} + BC_{in}$.
- *Carry propagate adders* are able to sums two N-bit numbers A and B (usually with a carry in C_{in}). Their output consists on a N-bit number S, the result of the addition, and the carry out of that operation, C_{out}. The name *carry propagate adder* is used because the C_{out} of every pair of bits A_i and B_i is propagated into the next pair A_{i+1} and B_{i+1} (Harris and Harris, 2015). There are two kinds of carry propagate adders: ripple-carry adders and carry-lookahead adders.
  - A N-bit *ripple-carry adder* is built chaining N full adders, just connecting the C_{out} output of every full adder with the C_{in} input of the next full adder. This is shown in Fig. 2.
  - Carry-lookahead adders divide the addition into blocks to accelerate the computation of the carry out.

4. Analysis of adders

4.1. *Half adder*

On the one hand, a half adder can be built using a Toffoli gate to compute $C_{out} = AB$ followed by a CNOT gate to compute $S = A \oplus B$ (Nielsen and Chuang, 2011). This circuit has a quantum cost of 6, a delay of $6\triangle$, an auxiliary qubit and no garbage outputs, as it is shown in Fig. 3. This design has been widely used to implement schemes in different experimental systems (Chatterjee and Roy, 2015; Barbosa, 2006; Srivastava et al., 2017; Wu and Cain, 2014; Dridi et al., 2015; Eloie et al., 2018).

On the other hand, the Peres gate (Peres, 1985) can also act as a half adder (Akbar et al., 2011; Batish et al., 2018). The version of the Peres gate presented in Hung et al. (2006) achieves the best quantum cost among the 3×3 reversible gates (Thapliyal and Ranganathan, 2010). This version consists of 1 CNOT gate, 1 Controlled-V gate and 2 Controlled-V^+ gates. The Peres gate has 3 inputs A, B and C, and produces 3 outputs $P = A$, $Q = A \oplus B$ and $R = AB \oplus C$. Setting $C = 0$, the outputs are $P = A$, $Q = A \oplus B$ and $R = AB$, which are the outputs of a half adder. Fig. 4 shows this use of the Peres gate. It has a quantum cost of 4, a delay

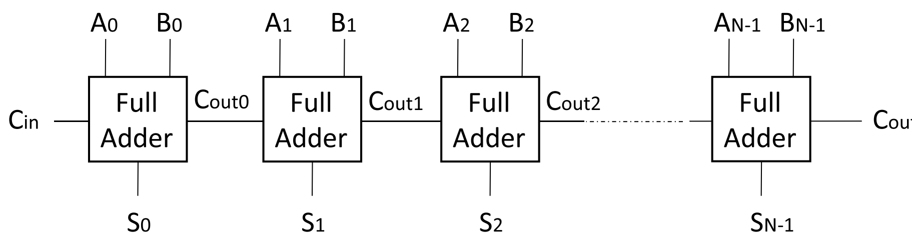


Fig. 2. *N* ripple-carry adder.

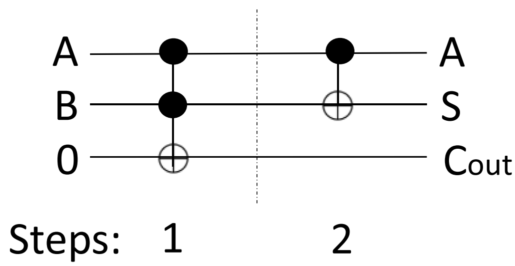


Fig. 3. Quantum implementation of the half adder proposed in Nielsen and Chuang (2011).

of 4△, an auxiliary qubit and no garbage outputs.

In Yamashita et al. (2008), a quantum circuit for half adder is mentioned in Fig. 3 of Section 2.2 as an example of semi-classical quantum circuit. The circuit is reproduced in Fig. 5. It has a quantum cost of 5 and a delay of 5△. To illustrate how it works, its truth table is shown in Table 4. In a similar way than the Peres gate, this circuit works as a half-adder if C is set to 0 and the variables P and Q denote C_{out} and S, respectively. Considering this case, the circuit has not garbage output and an auxiliary qubit.

There are several reversible half adder/subtractors (that is, circuits which compute half addition and subtraction at once) in the literature. These circuits have a quantum cost higher than regular half adders since they also perform the subtraction. A fault tolerant[1] full adder/subtractor using reversible gates was presented in Kaur and Dhaliwal (2012). The circuit of Kaur and Dhaliwal (2012) consists of two Feynman double gates (quantum cost 2 (Parhami, 2006)) and two Fredkin gates (quantum cost 5), with a total quantum cost of 14, the same delay, 3 auxiliary inputs, 1 selection qubit and 3 garbage outputs. Sarma and Jain (2018) presented a novel reversible half adder and subtractor circuit. This

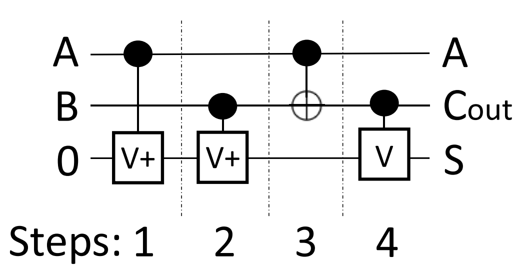


Fig. 4. Peres gate acting as a half adder, using the quantum implementation proposed in Hung et al. (2006).

[1] A fault tolerant circuit protects the information while it dynamically undergoes computation. This kind of circuit is specially useful since the error probability per gate is guaranteed to be lower than a given constant threshold. Of course, they need extra quantum cost to achieve this result (Nielsen and Chuang, 2011). Although interesting, the study of the techniques used to achieve this remarkable result is beyond the scope of this review.

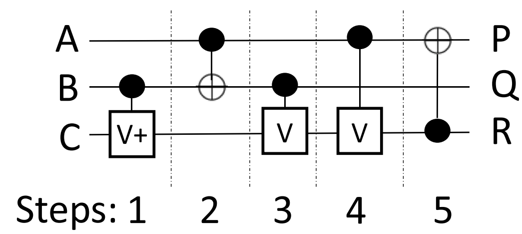


Fig. 5. Quantum implementation of the half adder presented in Yamashita et al. (2008).

Table 4
Truth table of the half adder of Fig. 5.

A	*B*	*C*	*P*	*Q*	*R*
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	0	0	1

circuit has a quantum cost of 5, the same delay, 1 auxiliary qubit and no garbage outputs. It is shown in Fig. 6. Authors named this circuit *RSG* gate, and it can also be used to build a full adder/subtractor circuit. That functionality will be analyzed in the next section. In the same year, 2018, Balaji et al. (2018) presented a fault tolerant half adder/-subtractor, which is similar in term s of quantum cost to Kaur and Dhaliwal (2012) (it also consists of two Fredkin gates and two Feynman double gates). The circuit of Balaji et al. (2018) improves the number of garbage outputs and auxiliary inputs, from 5 to 3 and from 4 to 2 respectively. However, it has fan-out. Fan-out is not allowed in reversible logic design (Nielsen and Chuang, 2011). Both circuit s are shown in Fig. 7 and Fig. 8.

4.2. Full adder

A simple way to build a full adder is to use three half-adders. A first half adder is used to compute $S_1 = A \oplus B$ and $C_{out1} = AB$. Then, a second half adder computes $S = S_1 \oplus C_{in}$ and $C_{out2} = S_1 C_{in}$. Finally, a third half adder computes $C_{out} = C_{out2} \oplus C_{out1}$ and an unused value (garbage)

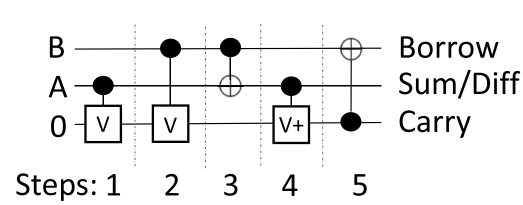


Fig. 6. Quantum implementation of the reversible half adder and subtractor circuit presented in Sarma and Jain (2018).

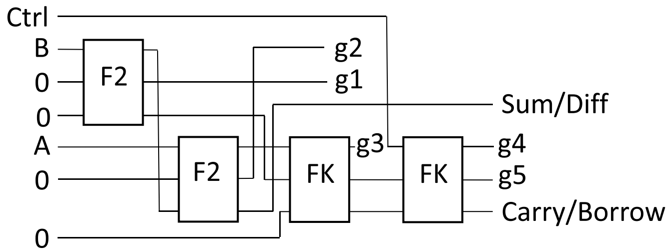


Fig. 7. Circuit of reversible fault tolerant half Adder/subtractor proposed in Kaur and Dhaliwal (2012). *F2* represents a Feynman double gate, and *FK* a Fredkin gate.

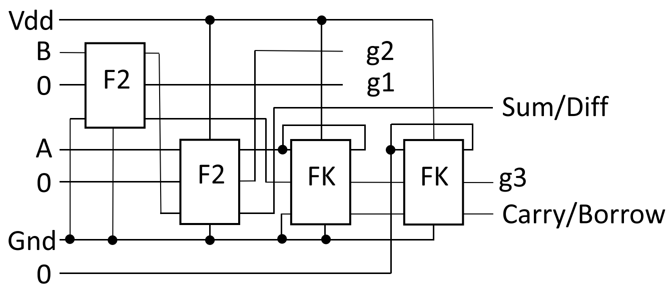


Fig. 8. Circuit of reversible fault tolerant half Adder/subtractor proposed in Balaji et al. (2018). *F2* represents a Feynman double gate, and *FK* a Fredkin gate.

$C_{out2}C_{out1}$. This circuit is shown in Fig. 9, and can be built using any of the half adders described in the previous subsection. However, this design can be improved using 2 Peres gates (Bhagyalakshmi and Venkatesha, 2010). A first Peres gate computes $Q = A \oplus B$ and $R = AB$ (second and third outputs respectively), and a second one accepts Q, C_{in} and R as inputs to compute $S = Q \oplus C_{in}$ and $C_{out} = QC_{in} \oplus R$ (second and third outputs respectively). This circuit can be seen in Fig. 10. It has a quantum cost of 8, a delay of 8△, 1 auxiliary qubit and 1 garbage output. These metrics have been calculated considering the version of the Peres gate presented by (Hung et al., 2006).

Khlopotine et al. (2002) proposed a full adder which uses the Fredkin gate (the Fredkin gate has a quantum cost of 5). This circuit consists of 5 Fredkin gates, so it has a quantum cost of 25. This version was improved in Bruce et al. (2002), reducing the necessary number of Fredkin gates into 4. In 2004, Cuccaro et al. (2004) proposed a new ripple-carry adder. It is based in 2 components (gates): a gate called Majority (*MAJ*), and another called UnMajority (*UMA*). These two gates can be combined to act as a full adder. The *MAJ* gate has three inputs, C_{in}, B and A, and three outputs, $U = C_{in} \oplus A$, $V = B \oplus A$ and C_{out}. Once the *MAJ* gate has been applied, C_{out} must be used or saved since the computation of the *UMA* gate will reverse this value into A. When the use of C_{out} is finished, the *UMA* gate is computed. It has three inputs (U, V and C_{out}) and three outputs: C_{in} and A (those values are reversed to avoid garbage outputs) and the sum S. The complete circuit to compute this process is shown in Fig. 11. It has a quantum cost of 14, the same delay, 0 auxiliary qubits and no garbage outputs (the complete ripple-carry adder of Cuccaro et al. (2004) will be analyzed in a later section). The design of Cuccaro et al. (2004) was improved in later works (Takahashi and Kunihiro, 2005; Takahashi and Kunihiro, 2008; Trisetyarso and Van Meter, 2010; Meter et al., 2008). In 2016, Wang et al. (2016) proposed a new design which keeps C_{out}. This circuit is shown in Fig. 12. It has a quantum cost of 10, a delay of 8△, 1 auxiliary qubit and no garbage outputs.

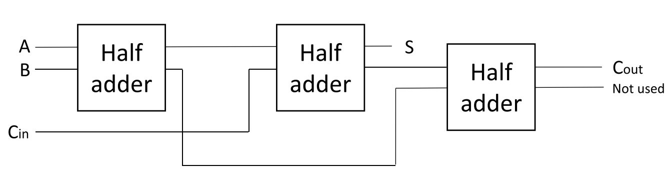


Fig. 9. Quantum implementation of a full adder using half adders.

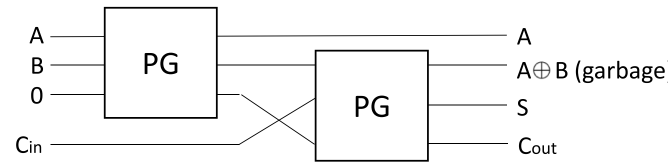


Fig. 10. Quantum implementation of a full adder using two Peres gates.

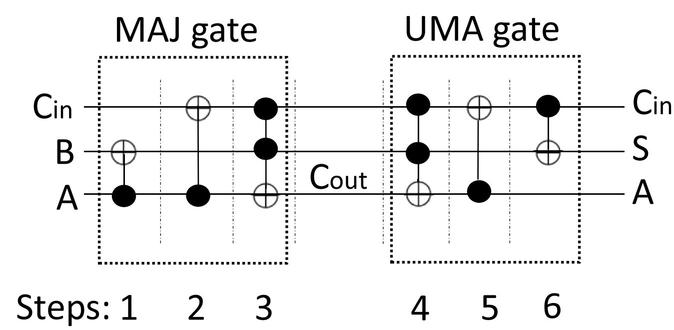


Fig. 11. Full adder proposed in Cuccaro et al. (2004). It consists of two gates called MAJ and UMA. C_{out} must be used before applying the UMA gate.

Maslov et al. (2008) designed a full adder which consists of 1 Controlled-V^+ gate, 3 Controlled-V gates and 2 CNOT gates. As it is shown in Fig. 13, this adder has a quantum cost of 6, a delay of 4△, 1 auxiliary qubit and 1 garbage output. In Nagamani et al. (2014), it was presented a full adder with a quantum cost of 12, delay 12△ and keeping 1 auxiliary qubit but avoiding garbage outputs. A circuit proposed in Thapliyal (2016) improves them. This circuit has the same quantum cost, delay and number of auxiliary qubits than (Maslov et al., 2008), but with no garbage outputs. It consists of 3 Controlled-V^+ gates, 1 Controlled-V gate and 2 CNOT gates. The circuit of Thapliyal (2016) is shown in Fig. 14. Also in 2016, Singh and Rai (2016) proposed two alternative designs of full adder based on reversible gates, but none of them improves the adder of Thapliyal (2016). The best of the adders of Singh and Rai (2016) has a quantum cost of 8, a delay of 8△, 1 auxiliary input and 1 garbage output.

Several fault tolerant full adders have been proposed. As it was mentioned, a fault tolerant circuit has a higher quantum cost because of parity preservation (Valinataj et al., 2016). For instance, Mitra and Chowdhury (2012) proposed a fault tolerant full adder with a quantum cost of 11, the same delay, 2 auxiliary inputs and 3 garbage outputs.

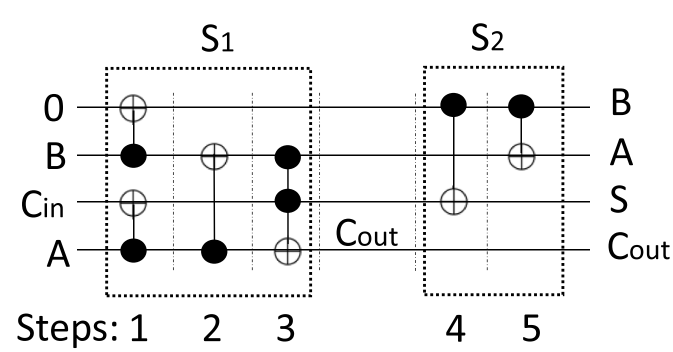


Fig. 12. Full adder proposed in Wang et al. (2016). The first sub-circuit $S1$ computes C_{out} and the second one computes S starting from the outputs of $S1$ without erasing C_{out}.

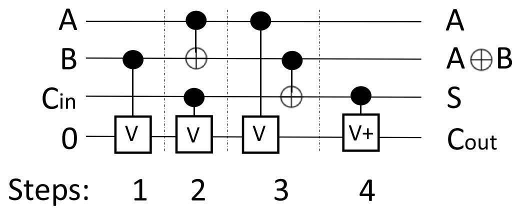


Fig. 13. Full adder proposed in Maslov et al. (2008).

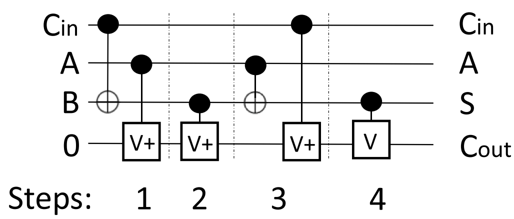


Fig. 14. Full adder proposed in Thapliyal (2016).

Previously to Mitra and Chowdhury (2012), several fault tolerant full adders were proposed: (Islam et al., 2009a) with a quantum cost of 14 and 3 garbage outputs, Bruce et al. (2002) which has been already analyzed in this section, and Haghparast and Navi (2008) with a quantum cost of 18 and 6 garbage outputs. Other fault tolerant adders are (Islam et al., 2009b) with a quantum cost of 14 and 3 garbage outputs, Dastan and Haghparast (2011) with a quantum cost of 14 and 3 garbage outputs, and Zhou et al. (2014) with a quantum cost of 8, delay 7 and 2 garbage outputs (Fig. 15). The circuit of Valinataj et al. (2016) has a quantum cost of 10 and 3 garbage outputs, but it offers interesting benefits against (Zhou et al., 2014) in terms of the transistor count or the total logical calculation (the number of *XOR*, *AND*, and *NOT* operations).

Similar to what happened with half adders, there are several reversible full adder/subtractors in the literature. Again, these circuits have a quantum cost higher than normal full adders since they also perform the subtraction. Rangaraju et al. (2010) proposed three designs. The best one consists of 2 CNOT gates and 2 Peres gate. It has a quantum cost of 10, the same delay, 1 auxiliary qubit and 3 garbage outputs. It also needs an extra selection qubit in order to select the operation to be computed (addition or subtraction). The half adder/subtractor of Kaur and Dhaliwal (2012) can be used to build a fault tolerant full adder/subtractor. 2 of these half adder/subtractors and 1 Feynman double gate (quantum cost 2 (Parhami, 2006)) are needed, with a total quantum cost of 30, the same delay, 9 auxiliary inputs, 1 selection qubit and 11 garbage outputs. A similar circuit was proposed in Saligram and Rakshith (2013), which consists of 4 Feynman double gates and 2 Fredkin gates, reducing the quantum cost to 18, the auxiliary inputs to 5 and the number of garbage outputs to 6. Moreover, Kumar et al. (2017) improved this design, using 3 Feynman double gates and only 1 Fredkin gate (total quantum cost of 11). It has 4 garbage outputs and 4 auxiliary inputs. The full adder of Thapliyal (2016) (Fig. 14) can be converted into a full adder/subtractor adding a selection qubit and a CNOT gate, having a quantum cost of 8 and a delay of 5△. On the other hand, the *RSG* gate presented in Sarma and Jain (2018), whose use as half adder/subtractor has been studied in the previous section, can be used to built a full adder/subtractor. It is able to compute both operations in parallel, without a selection qubit. This circuit has a quantum cost of 15, a delay of 10△, 1 auxiliary input and no garbage outputs. If we only consider the adder path of the circuit, the quantum cost is reduced to 10. Also in 2018, Balaji et al. (2018) proposed a fault tolerant full adder/subtractor using its half adder/subtractor described in the previous section. The complete circuit requires 2 of those half adder/subtractor s (quantum cost of 14 each one) and 1 Feynman double gate. Its final quantum cost is 30, with 5 auxiliary inputs and 3 garbage outputs.

In Batish et al. (2018), a comparative analysis for performance evaluation of reversible full adders is carried out. As a part of the analysis, they considered several methods to implement full adders using reversible gates:

- Full adder using *PCTG* gates: the *PCTG* gate consists of 1 Fredkin gate and 1 Feynman double gate. The full adder is built using 2 of these gates and 2 Feynman double gate s. As it is shown in Fig. 16, it has a quantum cost of 18, the same delay, 5 auxiliary inputs and 6 garbage outputs.
- Full adder using *BKG* gates: this gate was defined in Bhuvana and VS (2016). In that work, it is said that the *BKG* has a quantum cost of 1 since it is only 1 gate. However, according to the metrics of Mohammadi et al. (2009), a 4×4 gate cannot have a quantum cost of 1. The internal design of this gate is not described. It is detailed that it has four inputs A, B, C and D, and four outputs $P = A, Q = A\overline{D} \oplus C, R = (A\overline{D} \oplus C) \oplus B$ and $S = (A\overline{D} \oplus C)B \oplus AC \oplus A\overline{D}$. Setting $D = 0$, it acts as a full adder with 1 garbage output.
- Full adder using *DKG* gates: this circuit is defined in Krishnaveni et al. (2012). Similar to *BKG* gate, the internal design of this gate is not described. It has four inputs A', B', C' and D', and four outputs $P = B', Q = \overline{A'}C' + A'\overline{D'}, R = (A' \oplus B')(C' \oplus D') \oplus C'D'$ and $S = B' \oplus C' \oplus D'$. If the inputs were set to $A' = 0$, $B' = A$, $C' = B$ and $D' = C_{in}$, the outputs would be $P = A$, $Q = B$, $R = A(B \oplus C) \oplus BC = C_{out}$ and $S = A \oplus B \oplus C = Sum$. According to Mohammadi et al. (2009), it has no garbage outputs.
- Full adder using Peres gates: this can be seen in Fig. 10.
- Full adder using Peres and CNOT gates: this idea was introduced in Rohini and Rajashekar (2016). It s quantum cost is higher than the version of Fig. 10, and it has more garbage outputs.
- Full adder using *IG* gates: the *IG* gate was presented in Islam et al. (2009c). It has 4 inputs A, B, C and D, and four outputs $P = A$, $Q = A \oplus B$, $R = AB \oplus C$, and $S = BD \oplus \overline{B}(A \oplus D)$. Two *IG* gates connected in cascade can act as a full adder, as shown in Fig. 17. It has 3 garbage outputs. Once again, the internal design is not covered.

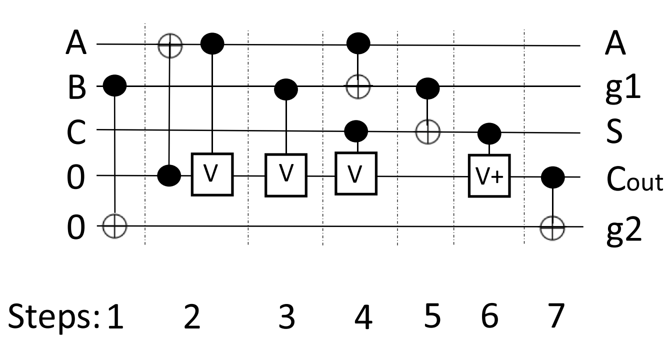


Fig. 15. Fault tolerant full adder proposed in Zhou et al. (2014).

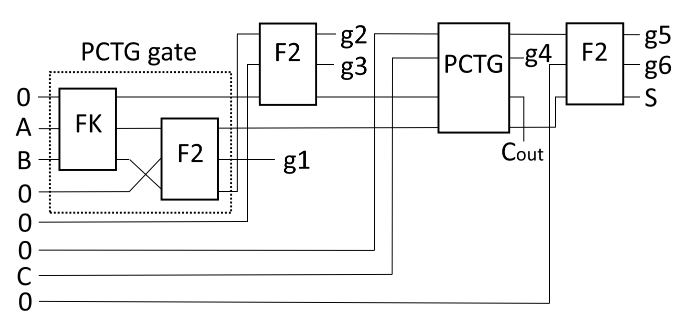


Fig. 16. Full adder using *PCTG* gates. A *PCTG* gate consists of 1 Fredkin gate (*FK*) and 1 Feynman double gate (*F2*).

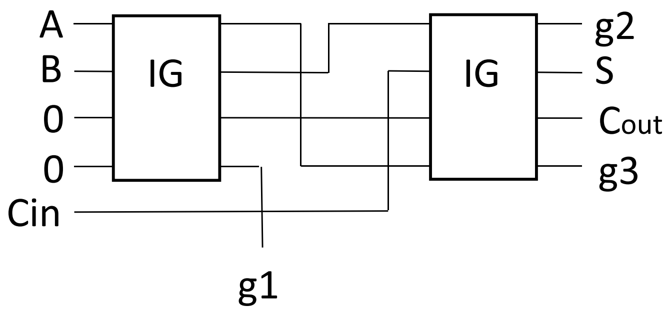


Fig. 17. Full adder using *IG* gates.

- Full Adder using Feynman and Fredkin gates: this full adder is the version proposed in Singh and Rai (2016), which has already been studied in this section.

4.3. Carry propagate adder

4.3.1. Ripple-carry adder

Since a N-digit ripple-carry adder is composed of N full adders, its QC, D, I and GO are given by the following equations:

$$\begin{aligned} QC_{ripple} &= N \cdot QC_{fulladder} \\ D_{ripple} &= N \cdot D_{fulladder} \\ I_{ripple} &= N \cdot I_{fulladder} \\ GO_{ripple} &= N \cdot GO_{fulladder} \end{aligned}$$

If the ripple-carry adder did not have carry in ($C_{in} = 0$), the least significant full adder could be replaced by a half adder. In this case, the equations are:

$$\begin{aligned} QC_{ripple} &= (N-1) \cdot QC_{fulladder} + QC_{halfadder} \\ D_{ripple} &= (N-1) \cdot D_{fulladder} + D_{halfadder} \\ I_{ripple} &= (N-1) \cdot I_{fulladder} + I_{halfadder} \\ GO_{ripple} &= (N-1) \cdot GO_{fulladder} + GO_{halfadder} \end{aligned}$$

It is possible to use any of the full adders of subsection 4.2 to build a ripple-carry adder. For the case $C_{in} = 0$, the least significant full adder can be replaced by any of the half adders of subsection 4.1. Ripple-carry adders are the best of the carry propagate adders in terms of quantum cost. However, due to their linear nature, they have a delay higher than others adders since the carry signals must propagate though every pair of bits A_i and B_i (Harris and Harris, 2015). In terms of quantum computers, which currently have few resources, this kind of adders are the best option as they minimize the quantum cost.

In addition to the ripple-carry adders that can be formed by combining the described full (and half) adders, it is worth noting some special cases. For instance, in Cuccaro et al. (2004) a ripple-carry adder is built using their full adder (Fig. 11). Then, this ripple-carry adder is optimized swapping and avoiding unnecessary gates. Assuming $C_{in} = 0$, the circuit can be even optimized further. For the general case, the circuit needs $2N-1$ Toffoli gates, $5N-3$ *CNOT* gates, and $2N-4$ Pauli-X gates, with a total quantum cost of $(2N-1) \times 5 + (5N-3) \times 1 + (2N-4) \times 1 = 17N - 12$. The delay is $10N\triangle$ as it has $2N-1$ Toffoli time-slices and 5 *CNOT* time-slices ($(2N-1) \times 5 + 5 \times 1$). It has 1 auxiliary input and no garbage outputs. As an example, the optimized circuit for the case $N = 6$ is shown in Fig. 18. Other proposals in the literature which presented a ripple-carry adder without C_{in} are (Takahashi and Kunihiro, 2005) (Quantum cost: $26N-29$, delay: $24N-27\triangle$, number of auxiliary inputs: 0, number of garbage outputs: 0), (Takahashi et al., 2010) (QC: $15N-9$, D: $13N-7\triangle$, AI: 0, GO: 0), and Thapliyal and Ranganathan (2013) (QC: $13N-8$, D: $11N-4\triangle$, AI: 0, GO: 0). The circuit of Thapliyal and Ranganathan (2013) is shown in Fig. 19 for the case $N = 4$.

On the other hand, there are several proposals (Cuccaro et al., 2004; Vedral et al., 1996; Skoneczny et al., 2008) which consider the input C_{in}. Thapliyal and Ranganathan (2011) presented an adder which optimizes the reduction of the computation in the ripple-carry process thanks to the use of a new gate called *TR*. This gate was defined in Thapliyal and Ranganathan (2009), but in Thapliyal and Ranganathan (2011) is optimized, using only 1 *CNOT* gate, 2 Controlled-V gates and 1 Controlled-V^+ gate. It has a quantum cost of 4 and a delay of $4\triangle$. The resulting adder has a quantum cost of $15N-6$, a delay of $9N+5\triangle$, and neither auxiliary inputs nor garbage outputs (Fig. 20). The ripple-carry adder of Nagamani et al. (2014) improves the quantum cost and delay of Thapliyal and Ranganathan (2011) ($12N$ and $10N\triangle$ respectively) at the cost of using $4N$ auxiliary inputs. The optical reversible ripple-carry adder with C_{in} proposed in Kotiyal (2016) is remarkable. It is not better than (Thapliyal and Ranganathan, 2011) in terms of the metrics of Mohammadi et al. (2009), but it improves it in terms of optical cost. Optical cost is one of the most important metric parameters in optical computing.

Moving away from the goal of reducing the quantum cost, Gidney presented an alternative gate to the Toffoli gate focused on reducing the cost of T gates (Gidney, 2018). This new gate has a higher quantum cost than the Toffoli gate if we consider the version proposed in Nielsen and Chuang (2011). However, it improves upon Toffoli gate implementations focused on fault-tolerance. As an example of the advantages of this gate (called *temporary logical-AND*), the author proposes an implementation of a fault-tolerant adder. This circuit has a quantum cost of

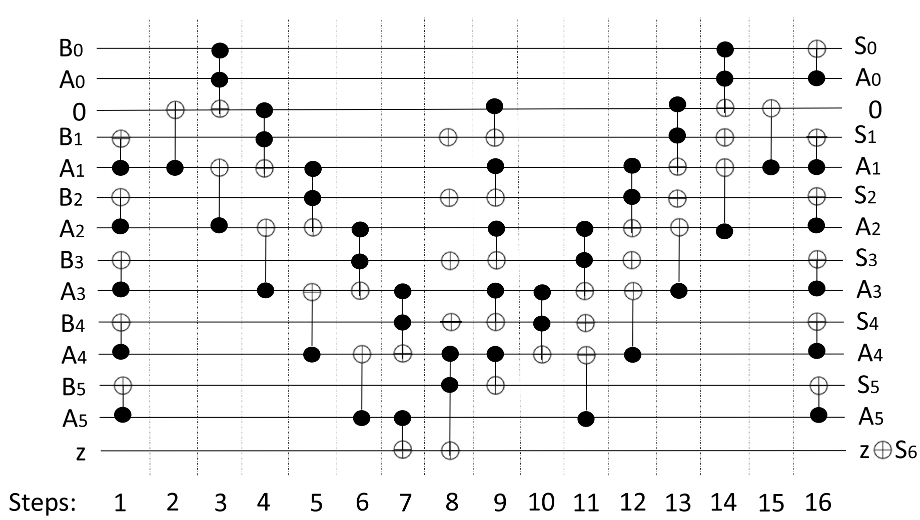


Fig. 18. Ripple-carry adder for $N = 6$ (assuming $C_{in} = 0$) proposed in Cuccaro et al. (2004).

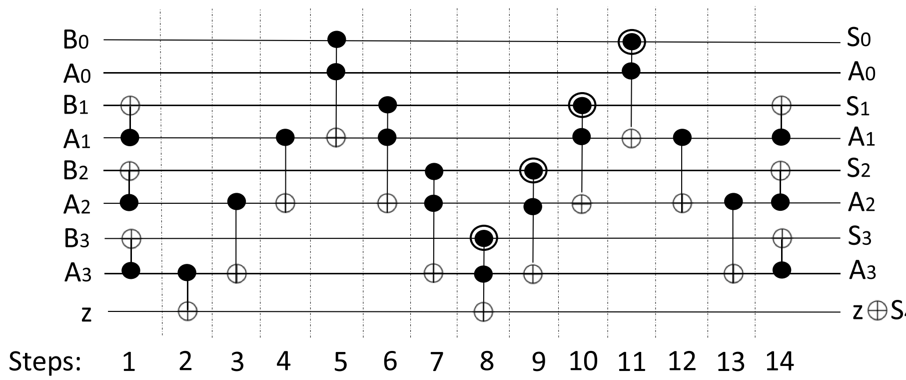


Fig. 19. Ripple-carry adder without carry in for $N = 4$ proposed in Thapliyal and Ranganathan (2013).

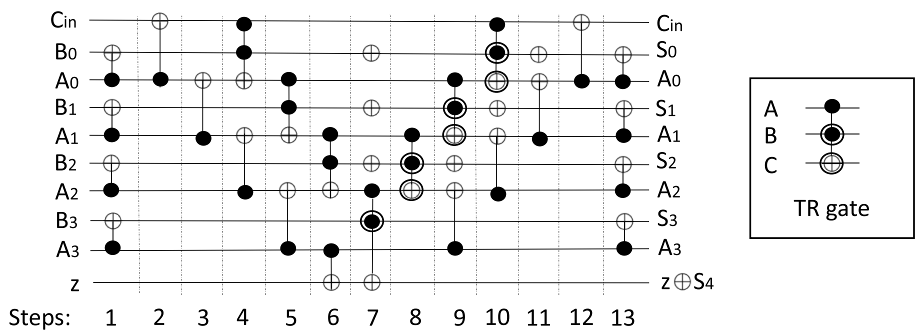


Fig. 20. Ripple-carry adder with carry in for $N = 4$ proposed in Thapliyal and Ranganathan (2011).

$18N - 2$, a delay of $15N - 5\triangle$, and requires N ancillary entries and 0 garbage outputs. Although these numbers are worse than previous circuits, that is due to their fault tolerance.

At the time of writing this article, the last adder in this category corresponds to the one published by Li et al. (Mohammadi et al., 2020). This adder combines Peres and *TR* gates to achieve the addition with a quantum cost of $13N - 10$, a delay of $10N - 4$, only 1 ancilla input and no garbage outputs. An example of this adder is shown in Fig. 21. The authors of this adder also describes its implementation in terms of T gates, obtaining an equivalent but error-oriented circuit. This second version has a quantum cost of $35N - 25$, a delay of $16N - 3$, and the same number of ancilla inputs and garbage outputs (1 and 0, respectively).

4.3.2. Carry-lookahead adder

This kind of adders employs two special signals to compute the carry out: generate signal (G) and propagate signal (P) (Harris and Harris, 2015):

- The carry out C_{out} of a pair of bits A_i and B_i is always 1 if both values are 1. This is called *generation* of a carry. Following this idea, G_i, the generate signal for the i-th pair, can be computed as $G_i = A_iB_i$.
- If a carry out C_{out} is produced when there is a carry in C_{in}, it is said that the carry is *propagated*. P_i, the propagate signal, can be computed as $P_i = A_i + B_i$.

Considering both signals, the carry out can be computed as:

$$C_i = A_iB_i + (A_i + B_i)C_{i-1} = G_i + P_iC_{i-1}$$

These adders are faster than the previous 1 s. However, they have a higher quantum cost since they need more operations to anticipate the computation of the *Gi* and *Pi* signals (Harris and Harris, 2015).

In 2004, Draper et al. proposed a logarithmic-depth reversible carry-lookahead adder which improves the delay of the previous linear -depth

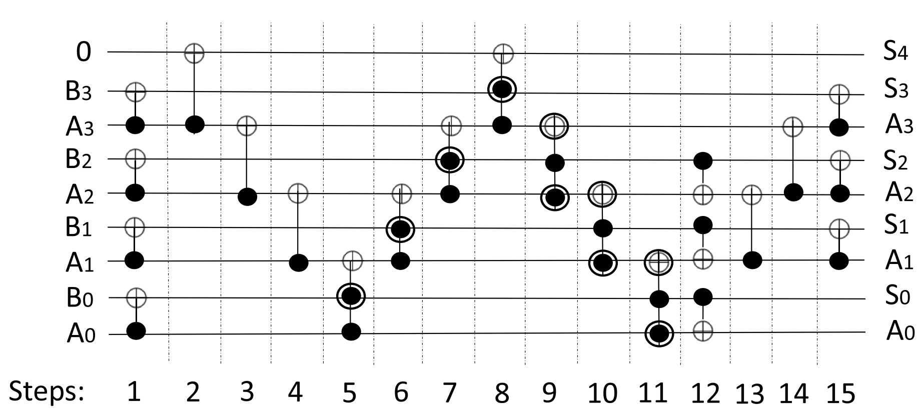


Fig. 21. Ripple-carry adder with carry in for $N = 4$ proposed in Mohammadi et al. (2020).

adders (Draper et al., 2004). It has a quantum cost of $28N - 15W(N) - 15\log(N) - 6$ (where $W(N)$ represents the number of ones in the binary expansion of N), a delay of $logN + logN/3 + 7$, $5N/4$ auxiliary inputs and no garbage outputs. Thapliyal et al. optimized the methodology to compute a carry-lookahead addition (without C_{in}), improving the quantum cost and delay of the previous adder (Thapliyal et al., 2013). It has a quantum cost of $26N - 15W(N) - 15\log(N-4)$ and a delay of $logN + logN/3 + 2$. The optimization is possible by computing G_i and P_i in parallel, and also replacing several *CNOT* and Toffoli gates by Peres gates. The circuit involves the use of several ancilla inputs, Zg_i and Zp_i, to compute G_i and P_i respectively. An example of this circuit is shown in Fig. 22. It works as follow s:

- Step 1: This step computes G_{i+1} and P_{i+1}, where $0 \leq i \leq N-1$. Zg_0 is transformed into $Zg_0 \oplus = A_0B_0$ with a Toffoli gate. For the case $i > 0$, $B_i \oplus = A_i$ and $Zg_i \oplus = A_iB_i$ using Peres gates.
- Step 2: This step computes G_{i+2} and P_{i+2}, where $0 \leq i \leq N-2$ for G and $2 \leq i \leq N-2$ for P. Using Toffoli gates, compute $Zp_i \oplus = B_iB_{i+1}$ for $i = 2$ to $N-2$ and $Zg_{i+1} \oplus = Zg_iB_{i+1}$ for $i = 0$ to $N-2$.
- Step 3: This step computes G_{i+3}, G_{i+4} and P_{i+4}, where $i = N/2$ and 0 for G and $i = N/2$ for P. For $i = N/2$ and $i = 0$, compute $Zp_{i+3} \oplus = Zg_{i+1}Zp_{i+1}$. When $i = N/2$, compute $Zp_{i+1} \oplus = Zp_iZp_{i+2}$.
- Step 4: This step computes G_{i+1} for $i = N$ and $i = N-2$. Compute $Zg_{i-1} \oplus = Zg_{i-3}Zp_{i-2}$ for $i = N/2$ and $Zg_{i-1} \oplus = Zg_{i-5}Zp_{i-3}$ for $i = N$. Also, this step uncomputes the values of Zp computed in step 3.
- Step 5: For $i = 2$ to $i = N-2$, transform Zg_i into $Zg_i \oplus = Zg_{i-1}B_i$, and Zp_{i+1} into $Zp_{i+1} \oplus = Zp_{i-1}Zp_{i+1}$ for $i = N/2$.
- Step 6: Uncompute Zp_i to avoid garbage outputs, and transform Zg_i into $Zg_i \oplus = B_{i+1}$ to compute S_{i+1}.
- Step 7: Compute S_0 and uncompute B_i.

Two reversible carry-lookahead adders were presented in Rahmati et al. (2017). They are shown (for the case $N = 4$) in Figs. 23 and 24. The first one has a quantum cost of $(2 \times N \times 4) + (N \times 1)$, better than the

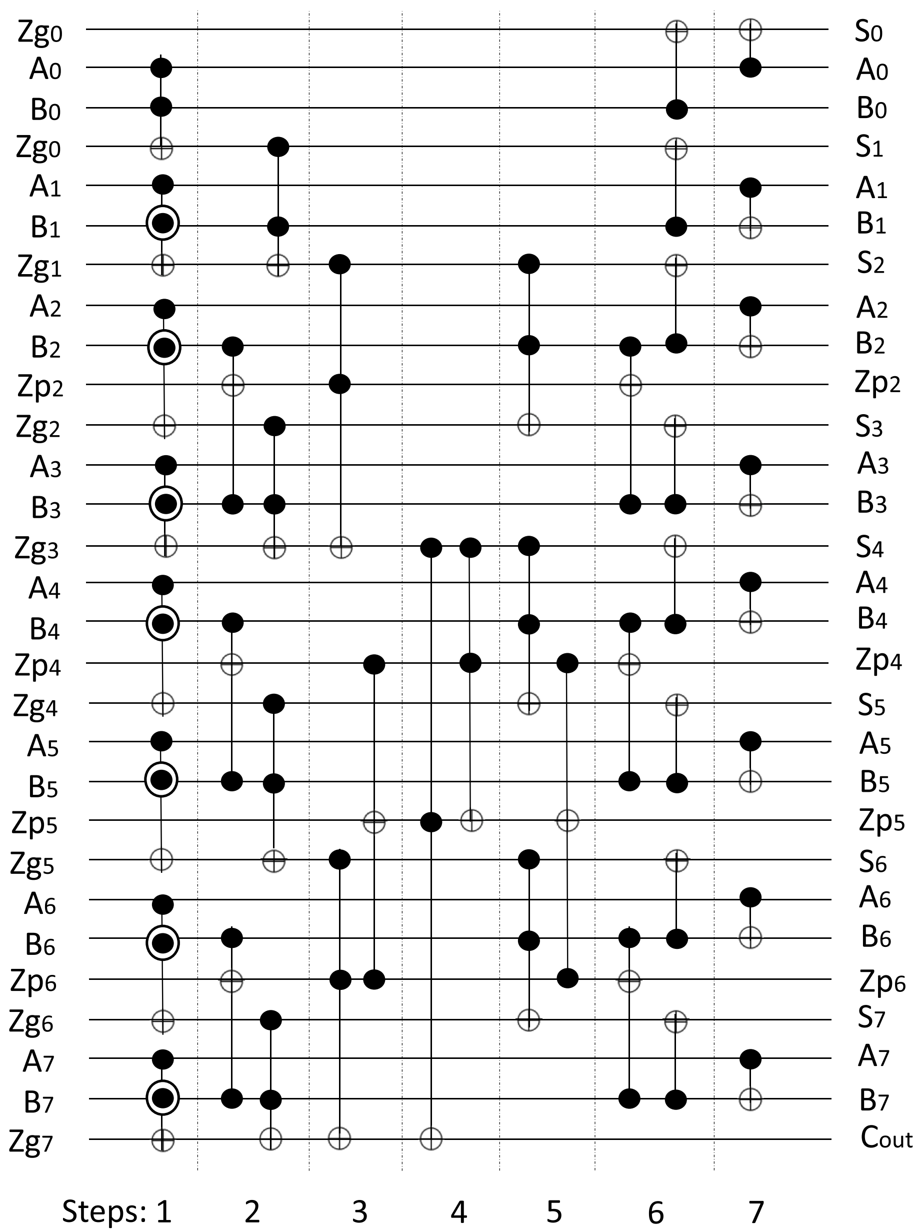


Fig. 22. Example of the carry-lookahead adder proposed in Thapliyal et al. (2013) for the case $N = 8$. It is built using Toffoli, *CNOT* and Peres gates. A_i and B_i are the numbers to be added, C_{out} are the carry out and S_i are the digits of the sum. Zg_i and Zp_i are auxiliary inputs used to compute G_i and P_i respectively.

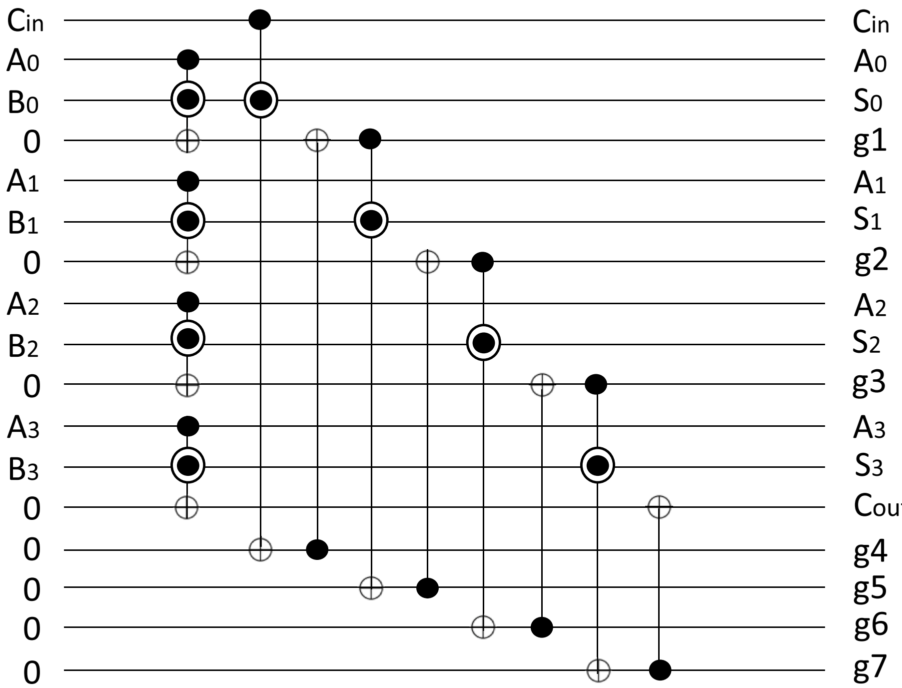


Fig. 23. Example of the first design of a carry-lookahead adder proposed in Rahmati et al. (2017) for the case $N = 4$. It is built using *CNOT* and Peres gates. A_i and B_i are the numbers to be added, C_{in} and C_{out} are the carry in and the carry out respectively, S_i are the digits of the sum, and g_i are garbage outputs.

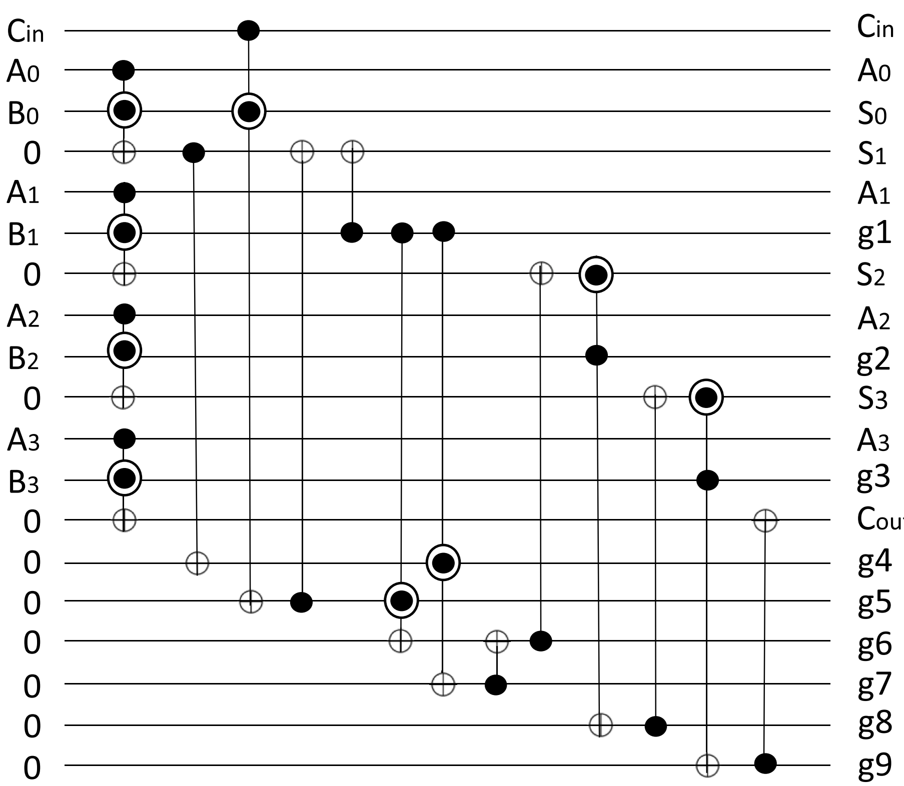


Fig. 24. Example of the second design of a carry-lookahead adder proposed in Rahmati et al. (2017) for the case $N = 4$. It is built using *CNOT* and Peres gates. A_i and B_i are the numbers to be added, C_{in} and C_{out} are the carry in and the carry out respectively, S_i are the digits of the sum, and g_i are garbage outputs.

presented in Thapliyal et al. (2013). However, this circuit presents $3 \times N$ garbage outputs, whereas the circuit of Thapliyal et al. (2013) has no garbage outputs. Following Bennett's garbage removal scheme (Bennett, 1973), it would be necessary to add $N + 1$ extra qubits to save S_i and C_{out}, four extra *CNOT* gates to copy S_i and C_{out} to those qubits, and to apply the reverse of the circuit. Therefore, uncomputing the garbage outputs would mean a final quantum cost of $2 \times ((2 \times N \times 4) + (N \times 1)) + N + 1$, which is higher than the quantum cost of Thapliyal et al. (2013). The second adder presented in Rahmati et al. (2017) also presents garbage outputs, so the same procedure can be applied to it. Its final quantum cost and delay do not improve that of Thapliyal et al. (2013).

A reversible adder presented in Lisa and Babu (2015) improves the quantum cost and delay of Thapliyal et al. (2013) using a novel technique for generating carry output. Nevertheless, it also has garbage outputs (Fig. 25). This adder was built using a new 4×4 gate called *RPA* (Reversible Partial Adder) gate, which has a quantum cost of 5 and delay $5\triangle$. It also uses several 4×4 Fredkin gates (the 3×3 Fredkin gate has a quantum cost of 5 and the same delay). The quantum cost is $11N$, but uncomputing the garbage outputs with (Bennett, 1973) would increase this number to $2 \times 11N + (N + 1)$. For instance, for the case $N = 4$, this circuit would have a quantum cost of 93, whereas the adders of Rahmati et al. (2017) and Thapliyal et al. (2013) have 61 and 55 respectively.

Focusing on fault-tolerance, Thapliyal et al. presented in Thapliyal et al. (2020) four adders based on their own design of Thapliyal et al. (2013) but optimizing the number of T gates at the cost of increasing the rest of the metrics. To achieve this optimization they use the Gidney's *temporary logical-AND* gate. Through the four circuits they explore several combinations of temporary logical-AND and Toffoli gates to find the best possibilities in terms of T-count and number of ancilla inputs. At best, $2N - W(N) - \log(N) + 1$ ancilla inputs are required (as opposed to $5N/5$ of the original circuit). Their quantum cost and delay are much higher than the (Thapliyal et al., 2013) circuit. It is the cost to pay for fault tolerance.

5. Comparative analysis

In this section, the analyzed adders are compared. Since comparing adders of different types makes no sense, the comparison is carried out between adders of the same kind. Therefore, four comparison are presented: half adders, full adders, ripple-carry adders and carry-lookahead adders. Nevertheless, at the end of the section an overview of all the results is made to consider the comparison as a whole.

5.1. Half adders

Table 5 shows the quantum cost, delay, number of auxiliary inputs and number of auxiliary outputs of the most representative half adders. The final column indicates if the adder can be used as a subtractor. In terms of quantum cost, the Peres gate proposed in Hung et al. (2006) (Fig. 4) achieves the best value. The proposals of Sarma and Jain (2018) and Yamashita et al. (2008) have a quantum cost of 5, one more than (Hung et al., 2006), but (Sarma and Jain, 2018) can also act as a subtractor, so the extra value is justified in this case. Nielsen and Chuang (2011) has a quantum cost of 6, and Kaur and Dhaliwal (2012) has the highest quantum cost, 14. The quantum cost of Kaur and Dhaliwal (2012) is justified since this adder was the first reversible adder/-subtractor. None of the half adders can compute any operation in parallel, so their delay is equal to their quantum cost. In terms of auxiliary inputs, all of them have 1 input, except (Kaur and Dhaliwal, 2012) which has 3. Only (Kaur and Dhaliwal, 2012) presents garbage outputs.

Table 5
Comparative evaluation of half adders.

Adder	Quantum cost	Delay $\triangle$	Ancilla inputs	Garbage outputs	Adder/ subtractor
Kaur and Dhaliwal (2012)	14	14	3	3	Yes
Nielsen and Chuang (2011)	6	6	1	0	
Yamashita et al. (2008)	5	5	1	0	
Sarma and Jain (2018)	5	5	1	0	Yes
Hung et al. (2006)	4	4	1	0	

5.2. Full adders

Table 6 focus the comparison on the most relevant full adders. A new columns ha been added to this comparison in order to identify which adders are fault tolerant. The most optimized adder in terms of quantum cost, delay and garbage output is Thapliyal (2016) [a], with 6, $4\triangle$ and 0 respectively. The adder o f (Maslov et al., 2008) also presents the same quantum cost, delay and number of auxiliary inputs, but it has 1 garbage output. The only full adder which has no auxiliary inputs is Cuccaro et al. (2004), but it has a quantum cost and a delay higher than the average (14 and $14\triangle$ respectively). Bhagyalakshmi and Venkatesha (2010), Maslov et al. (2008), Singh and Rai (2016), Mitra and Chowdhury (2012), Zhou et al. (2014), Rangaraju et al. (2010), Saligram and Rakshith (2013) and Kumar et al. (2017) present garbage outputs, so their quantum cost and delay would be higher if they were uncomputed (Bennett, 1973). The best adder/subtractor is the proposed in Thapliyal (2016) [b] as it has no garbage outputs, and it has the best quantum cost and delay among the adder/subtractor s, 8 and $5\triangle$ respectively. It is followed by (Rangaraju et al., 2010) with a quantum cost of 10 and delay $10\triangle$, but with 3 garbage outputs. Therefore, the adder of Sarma and Jain (2018), which has a quantum cost of 15, would be a better option

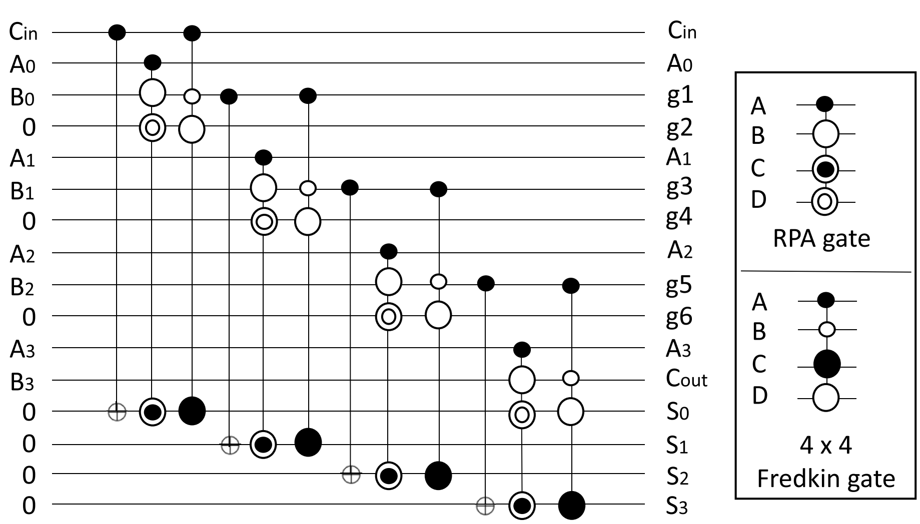


Fig. 25. Carry-lookahead adder proposed in Lisa and Babu (2015), for the case $N = 4$. It is built using 4×4 *RPA* and Fredkin gates. A_i and B_i are the numbers to be added, C_{in} and C_{out} are the carry in and the carry out respectively, S_i is the sum, and g_i are garbage outputs.

Table 6
Comparative evaluation of full adders.

Adder	Quantum cost	Delay △	Ancilla inputs	Garbage outputs	Adder/subtractor	Fault tolerant
Saligram and Rakshith (2013)	18	18	5	6	Yes	
Yamashita et al. (2008)	15	15	3	0		
Sarma and Jain (2018)	15	10	1	0	Yes	
Cuccaro et al. (2004)	14	14	0	0		
Hung et al. (2006)	12	12	3	0		
Nagamani et al. (2014)	12	12	1	0		
Mitra and Chowdhury (2012)	11	11	2	3		Yes
Kumar et al. (2017)	11	11	4	4	Yes	
Wang et al. (2016)	10	8	1	0		
Rangaraju et al. (2010)	10	10	1	3	Yes	
Singh and Rai (2016)	8	8	1	1		
Bhagyalakshmi and Venkatesha (2010)	8	8	1	1		
Thapliyal (2016) [b]	8	5	2	0	Yes	
Zhou et al. (2014)	8	7	2	2		Yes
Maslov et al. (2008)	6	4	1	1		
Thapliyal (2016) [a]	6	4	1	0		

than (Rangaraju et al., 2010) as it has no garbage outputs. Finally, focusing on the fault tolerant adders, Zhou et al. (2014) and Mitra and Chowdhury (2012) are the most optimized options. Zhou et al. (2014) presents lower values of quantum cost and delay, and both have the same number of auxiliary inputs. Moreover, Zhou et al. (2014) has less garbage outputs.

5.3. Ripple-carry adders

The comparison between ripple-carry adders is shown in Table 7. A N-bits ripple-carry adder could be built chaining N full adders of Table 6. The better the selected full adder, the better the ripple-carry adder. However, even choosing (Thapliyal, 2016)[a] (the best full adder) results in a ripple-carry adder which is worse than, for instance, the improved ripple-carry adder presented in Thapliyal and Ranganathan (2013). For that reason, the resulting ripple-carry adders are not included in Table 7. In this table, the new column C_{in} indicates if the adder support s C_{in} or not.

In terms of auxiliary inputs, only Cuccaro et al. (2004) and Nagamani et al. (2014) have them. The adder of Cuccaro et al. (2004) has only 1, whereas the adder of Nagamani et al. (2014) employs $4N$. In terms of quantum cost, Nagamani et al. (2014) achieves the best value -even considering that it supports C_{in}- thanks to the use of the mentioned extra ancilla inputs. The non fault-tolerant adder of Mohammadi et al. (2020) improves the quantum cost of Nagamani et al. (2014), but only when $N < 10$. Thapliyal and Ranganathan (2013) is the third best adder in these terms. However, both Mohammadi et al. (2020) and Thapliyal and Ranganathan (2013) do not support C_{in}. In terms of delay, the circuit of Thapliyal and Ranganathan (2011) gets the best value, $9N + 5\triangle$, followed by the non fault-tolerant adder of Mohammadi et al. (2020), which has $10N - 4\triangle$. Finally, we can highlight that no circuit has garbage outputs.

The adder presented by Gidney (2018) is optimized for fault tolerance, and that is why it has higher values of quantum cost and delay. It even sacrifices multiple ancillary inputs to reduce the number of T gates. The fault-tolerant adder proposed in Mohammadi et al. (2020) does not improve the quantum cost nor the delay with respect to the previous one, but it represents a substantial improvement in terms of ancilla inputs.

Table 8
Comparative evaluations of carry-lookahead adders. $W(N)$ is the number of ones in the binary expansion of N. The quantum cost and delay of the circuits of Thapliyal et al. (2020) cannot be precisely indicated because certain transformations in the quantum state of the ancilla qubits need to be taken into account for use with temporary logical-AND gates. The study of how these transformations can influence the logarithmic propagation of the adders is not trivial.

Adder	Quantum cost	Delay △	Ancilla inputs	Fault-tolerance
Thapliyal et al. (2020)[a]	$>40N$	$O(logN)$	$4N - 2W(N) - 2\log(N)$	Yes
Thapliyal et al. (2020)[b]	$>40N$	$O(logN)$	$2N - W(N) - \log(N) + 1$	Yes
Thapliyal et al. (2020)[c]	$>40N$	$O(logN)$	$4N - 2W(N) - 2\log(N)$	Yes
Thapliyal et al. (2020)[d]	$>40N$	$O(logN)$	$2N - W(N) - \log(N) + 1$	Yes
Draper et al. (2004)	$28N - 15W(N) - 15\log(N) - 6$	$logN + logN/3 + 7$	$5N/4$	
Thapliyal et al. (2013)	$26N - 15W(N) - 15\log(N - 4)$	$logN + logN/3 + 2$	$5N/4$	

Table 7
Comparative evaluation of ripple-carry adders.

Adder	Quantum cost	Delay △	Ancilla inputs	Garbage outputs	C_{in}	Fault tolerant
Li et al. (2) (Mohammadi et al., 2020)	$35N - 25$	$16N - 3$	0	0		Yes
Takahashi and Kunihiro (2005)	$26N - 29$	$24N - 27$	0	0		
Gidney (2018)	$18N - 2$	$15N - 5$	N	0		Yes
Cuccaro et al. (2004)	$17N - 12$	$10N$	1	0		
Takahashi et al. (2010) (2)	$15N - 9$	$13N - 7$	0	0		
Thapliyal and Ranganathan (2011)	$15N - 6$	$9N + 5$	0	0	Yes	
Thapliyal and Ranganathan (2013) (2)	$13N - 8$	$11N - 4$	0	0		
Li et al. (1) (Mohammadi et al., 2020)	$13N - 10$	$10N - 4$	0	0		
Nagamani et al. (2014)	$12N$	$10N$	$4N$	0	Yes	

5.4. Carry-lookahead adders

Finally, the carry-lookahead adders are compared in Table 8. As it has been mentioned in the subsection of the carry-lookahead adders, there are several adders whose garbage outputs have not been uncomputed. They are the two adders of Rahmati et al. (2017) and Lisa and Babu (2015). It is not useful to compare such circuits with those which have uncomputed their garbage outputs since uncomputing these outputs following (Bennett, 1973) would increase the quantum cost, delay and auxiliary inputs (Orts et al., 2019). On the other hand, a 4 × 4 Fredkin gate is used in the case of Lisa and Babu (2015). The quantum cost and delay of this gate is not addressed, so it is not possible to determine the quantum cost and delay of this circuit with precision. Therefore, they have not been included in the table (but they have been analyzed in the subsection of the carry-lookahead adders for the sake of clarity). Considering the remaining non fault-tolerant adders, Draper et al. (2004) and Thapliyal et al. (2013), it can be concluded that Thapliyal et al. (2013) presents the best quantum cost and delay. Both of them have $5N/4$ auxiliary inputs and 0 garbage outputs. Considering now the four adders presented in Thapliyal et al. (2020), they do not improve any of the metrics from Mohammadi et al. (2009) to the previous adders. However, they are optimized in terms of the T gate, being the only adders focused on fault-tolerance in this category. In Table 8, the quantum cost and delay of these four adders are not shown accurately: this is because certain transformations in the quantum state of their ancilla qubits need to be taken into account for use with temporary logical-AND gates. The influence of these transformations in the quantum cost and delay of the adders is not trivial, and their effect is not included in the analysis done in citethapliyal2020tcount as it is focused on the optimization of T gates and necessary qubits.

5.5. General discussion

From the tables of the comparative evaluations, it can be concluded something that it is already known in classical circuits: ripple-carry adders have a lower cost, and carry-lookahead adders are the fastest. On the one hand, Mohammadi et al. (2020) and Thapliyal and Ranganathan (2013) for the case without C_{in} and Nagamani et al. (2014) and Thapliyal and Ranganathan (2011) for the case with C_{in} are the most optimized ripple-carry adders nowadays in terms of quantum cost, delay, auxiliary inputs and garbage outputs. On the other hand, Thapliyal et al. (2013) is the most optimized carry-lookahead adder. On the other hand, in the most recent works there is a growing interest in the optimization of circuits in terms of T gates. (Mohammadi et al., 2020; Gidney, 2018) (ripple-carry adders), and the four adders of Thapliyal et al. (2020) (carry-lookahead adders), are the best exponents in this new stage of optimization.

6. Conclusions

In this work, a revision on the state-of-the-art reversible adders has been carried out. First, appropriate metrics have been considered for the measurement and comparison of quantum circuits. Second, the adders have been classified in one of the four possible types: half adders, full adders, ripple-carry adders, and carry-lookahead adders, explaining their particular calculation methods and structures. Third, a complete analysis of each existing reversible adder has been done in terms of those metrics. Finally, a comparison between the analyzed adders has been done using the metrics and the category to determine which adders are the most beneficial in terms of quantum cost, delay, number of auxiliary inputs and/or number of garbage outputs, and taking into account the peculiarities of the category in question (if any) and fault tolerance.

The analysis has been carried out with two essential goals in mind: first, to collect and compare, using a set of standard metrics, all the reversible adders existing in the literature; second, to focus on the possibility of applying these adders in quantum circuits. To this extent, our emphasis has been on analyzing quantum costs and delays, as well as the presence of garbage outputs (which prevent useful interference from arising in quantum algorithms) and in presenting our findings in a clear and concise way, with tables that summarize the main properties of all the most important adders and that can be used as a quick reference by the interested researchers. In addition, we also provide figures that exemplify some of the most remarkable adders for some values on the number of inputs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

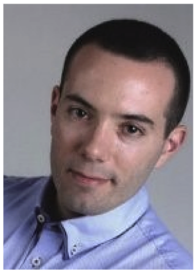
Acknowledgments

This work has been partially supported by the Spanish Ministry of Science throughout Project RTI2018-095993-B-I00, by the Regional Ministry of the Principado de Asturias under grant FC-GRUPIN-IDI/2018/000226 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

Akbar, E.P.A., et al., 2011. Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology. Microelectron. J. 42 (8), 973–981.
Anamika, R. Bhardwaj, 2018. Reversible logic gates and its performances. In: 2018 2nd International Conference on Inventive Systems and Control. ICISC, pp. 226–231.
Arute, F., et al., 2019. Quantum supremacy using a programmable superconducting processor. Nature 574 (7779), 505–510.
Balaji, B., et al., 2018. Full adder/subtractor using reversible logic. Int. J. Pure Appl. Math. 120 (6), 437–446.
Barbosa, G.A., 2006. Quantum half-adder. Phys. Rev. A 73, 052321.
Batish, K., et al., 2018. Comparative analysis for performance evaluation of full adders using reversible logic gates. In: 2018 International Conference on Intelligent Circuits and Systems (ICICS). IEEE, pp. 126–132.
Bennett, C.H., 1973. Logical reversibility of computation. IBM J. Res. Dev. 17 (6), 525–532.
Bhagyalakshmi, H., Venkatesha, M., 2010. An improved design of a multiplier using reversible logic gates. Int. J. Eng. Sci. Technol. 2 (8), 3838–3845.
Bhuvana, B., VS, K.B., 2016. Design of reversible adders using a novel reversible bkg gate. In: 2016 Online International Conference on Green Engineering and Technologies (IC-GET). IEEE, pp. 1–6.
Bruce, J., et al., 2002. Efficient adder circuits based on a conservative reversible logic gate. In: Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI, IEEE, pp. 83–88, 2002.
Chatterjee, D., Roy, A., 2015. A transmon-based quantum half-adder scheme. Prog. Theor. Exp. Phys. (9).
Chaves, J., et al., 2018. Energy efficient QCA circuits design: simulating and analyzing partially reversible pipelines. J. Comput. Electron. 17 (1), 479–489.
Cohen, E., et al., 2016. All-optical design for inherently energy-conserving reversible gates and circuits. Nat. Commun. 7, 11424.
Combarro, E.F., et al., 2019a. A quantum algorithm for the commutativity of finite dimensional algebras. IEEE Access 7, 45554–45562.
Combarro, E.F., et al., 2019b. Quantum walks for the determination of commutativity of finite dimensional algebras. J. Comput. Appl. Math. 354, 496–506.
Cuccaro, S.A., Moulton, D.P., et al., 2004. A New Quantum Ripple-Carry Addition Circuit. arXiv:quant-ph/0410184.
Dastan, F., Haghparast, M., 2011. A novel nanometric fault tolerant reversible divider. Int. J. Phys. Sci. 6 (24), 5671–5681.
Diffie, W., Hellman, M., 1976. New directions in cryptography. IEEE Trans. Inf. Theor. 22 (6), 644–654.
Draper, T.G., et al., 2004. A Logarithmic-Depth Quantum Carry-Lookahead Adder. arXiv: quant-ph/0406142.
Dridi, G., et al., 2015. The mathematics of a quantum Hamiltonian computing half adder boolean logic gate. Nanotechnology 26 (34), 344003.
Eloie, L., et al., 2018. Quantum arithmetics via computation with minimized external control: the half-adder. Phys. Rev. A 97, 062321.
Fowler, A.G., et al., 2004. Implementation of Shor's algorithm on a linear nearest neighbour qubit array. Quant. Inf. Comput. 4 (4), 237–251.
Fredkin, E., Toffoli, T., 1982. Conservative logic. Int. J. Theor. Phys. 21 (34), 219–253.
Gidney, C., 2018. Halving the cost of quantum addition. Quantum 2 (74), 10–22331.
Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96. ACM, New York, NY, USA, pp. 212–219.

Groe, D., et al., 2008. 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, pp. 214–219.
Haghparast, M., Navi, K., 2008. Design of a novel fault tolerant reversible full adder for nanotechnology based systems. World Appl. Sci. J. 3 (1), 114–118.
Harris, S., Harris, D., 2015. Digital Design and Computer Architecture, arm edition. Morgan Kaufmann.
He, X., et al., 2019. The mapping and optimization method of quantum circuits for Clifford+T gate. J. Appl. Math. Phys. 7 (11), 2796–2810.
Hung, W.N., et al., 2006. Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 25 (9), 1652–1663.
Islam, M., et al., 2009a. Efficient approaches for designing fault tolerant reversible carry look-ahead and carry-skip adders. MASAUM J. Basic Appl. Sci. 1 (3), 354–360.
Islam, M.S., et al., 2009b. Fault tolerant reversible logic synthesis: carry look-ahead and carry-skip adders. In: 2009 International Conference on Advances in Computational Tools for Engineering Applications. IEEE, pp. 396–401.
Islam, M.S., et al., 2009c. Synthesis of fault tolerant reversible logic circuits. In: 2009 IEEE Circuits and Systems International Conference on Testing and Diagnosis. IEEE, pp. 1–4.
Kaur, P., Dhaliwal, B.S., 2012. Design of fault tolerant full adder/subtractor using reversible gates. In: 2012 International Conference on Computer Communication and Informatics. IEEE, pp. 1–5.
Khlopotine, A.B., et al., 2002. Reversible logic synthesis by iterative compositions. In: Iwls, pp. 261–266.
Kotiyal, S., 2016. Design Exploration and Application of Reversible Circuits in Emerging Technologies. Ph.D. thesis. Computer science and engineering department. University of South Florida.
Krishnaveni, D., et al., 2012. Design of an efficient reversible 8x8 Wallace tree multiplier. World Appl. Sci. J. 20 (8), 1159–1165.
Kumar, P.K., et al., 2017. Optimal design of reversible parity preserving new full adder/ full subtractor. In: 2017 11th International Conference on Intelligent Systems and Control (ISCO). IEEE, pp. 368–373.
Landauer, R., 1961. Irreversibility and heat generation in the computing process. IBM J. Res. Dev. 5 (3), 183–191.
Lanyon, B.P., et al., 2009. Simplifying quantum logic using higher-dimensional hilbert spaces. Nat. Phys. 5 (2), 134.
Lecerf, Y., 1963. Machines de Turing rversibles. Comptes Rendus Hebd. Sances Acad. Sci. 257, 2597–2600.
Lemr, K., et al., 2015. Experimental implementation of optimal linear-optical controlled-unitary gates. Phys. Rev. Lett. 114 (15), 153602.
Linke, N.M., et al., 2017. Experimental comparison of two quantum computing architectures. Proc. Natl. Acad. Sci. 114 (13), 3305–3310.
Lisa, N.J., Babu, H.M.H., 2015. Design of a compact reversible carry look-ahead adder using dynamic programming. In: 2015 28th International Conference on VLSI Design. IEEE, pp. 238–243.
Maslov, D., Negrevergne, C., et al., 2008. Quantum circuit simplification and level compaction. IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 27 (3), 436–444.
Meter, R.V., et al., 2008. Arithmetic on a distributed-memory quantum multicomputer. ACM J. Emerg. Technol. Comput. Syst. (JETC) 3 (4), 2.
Michielse, K., et al., 2017. Benchmarking gate-based quantum computers. Comput. Phys. Commun. 220, 44–55.
Miller, D.M., et al., 2014. Mapping NCV circuits to optimized Clifford+T circuits. In: International Conference on Reversible Computation. Springer, pp. 163–175.
Mitra, S.K., Chowdhury, A.R., 2012. Minimum cost fault tolerant adder circuits in reversible logic synthesis. In: 2012 25th International Conference on VLSI Design. IEEE, pp. 334–339.
Mohammadi, M., et al., 2009. On figures of merit in reversible and quantum logic designs. Quant. Inf. Process. 8 (4), 297–318.
Mohammadi, M., et al., 2020. Efficient quantum arithmetic operation circuits for quantum image processing. Sci. China Phys. Mech. Astron. 63, 1–13.
Nagamani, A., et al., 2014. Design of optimized reversible binary adder/subtractor and BCD adder. In: 2014 International Conference on Contemporary Computing and Informatics (IC3I). IEEE, pp. 774–779.
Neill, C., et al., 2018. A blueprint for demonstrating quantum supremacy with superconducting qubits. Science 360 (6385), 195–199.
Nielsen, M.A., Chuang, I.L., 2011. Quantum Computation and Quantum Information, 10th anniversary edition.
Orts, F., et al., 2019. An optimized quantum circuit for converting from signmagnitude to two's complement. Quant. Inf. Process. 18 (11), 332.
Parhami, B., 2006. Fault-tolerant reversible circuits. In: 2006 Fortieth Asilomar Conference on Signals, Systems and Computers. IEEE, pp. 1726–1729.
Pavlidis, A., Gizopoulos, D., 2014. Fast quantum modular exponentiation architecture for Shor's factorization algorithm. Quant. Inf. Comput. 14 (7), 649–682.
Peres, A., 1985. Reversible logic and quantum computers. Phys. Rev. A 32 (6), 3266.
Rahmati, M., et al., 2017. Novel designs of a carry/borrow look-ahead adder/subtractor using reversible gates. J. Comput. Electron. 16 (3), 856–866.
Rangaraju, H., et al., 2010. Low power reversible parallel binary adder/subtractor. Int. J. VLSI Des. Commun. Syst. 1 (3), 23–34.
Rivest, R.L., et al., 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21 (2), 120–126.
Rohini, H., Rajashekar, S., 2016. Design of reversible logic based basic combinational circuits. Commun. Appl. Electron. 5 (9), 38–43.
Sahu, L., Kumar, U., Singh, L., 2019. Implementation of improved energy-efficient FIR filter using reversible logic. In: Data Science and Big Data Analytics. Lecture Notes on Data Engineering and Communications Technologies, vol. 16. Springer, Singapore, pp. 229–238.
Saligram, R., Rakshith, T., 2013. Design of low logical cost adders using novel parity conserving toffoli gate. In: 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA). IEEE, pp. 1–6.
Sarma, R., Jain, R., 2018. Quantum gate implementation of a novel reversible half adder and subtractor circuit. In: 2018 International Conference on Intelligent Circuits and Systems (ICICS). IEEE, pp. 72–76.
Shi, Y., 2003. Both Toffoli and controlled-not need little help to do universal quantum computing. Quant. Inf. Comput. 3 (1), 84–92.
Shor, P.W., 1994. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, SFCS '94. IEEE Computer Society, Washington, DC, USA, pp. 124–134.
Singh, V.P., Rai, M., 2016. Verilog design of full adder based on reversible gates. In: 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall). IEEE, pp. 1–5.
Skoneczny, M., et al., 2008. Reversible fourier transform chip. In: 2008 15th International Conference on Mixed Design of Integrated Circuits and Systems. IEEE, pp. 281–286.
Srivastava, S., et al., 2017. Quantum half-adder boolean logic gate with a nano-graphene molecule and graphene nano-electrodes. Chem. Phys. Lett. 667, 301–306.
Steiger, D.S., et al., 2018. ProjectQ: an open source software framework for quantum computing. Quantum 2 (49).
Takahashi, Y., Kunihiro, N., 2005. A linear-size quantum circuit for addition with no ancillary qubits. Quant. Inf. Comput. 5 (6), 440–448.
Takahashi, Y., Kunihiro, N., 2008. A fast quantum circuit for addition with few qubits. Quant. Inf. Comput. 8 (6), 636–649.
Takahashi, Y., et al., 2010. Quantum addition circuits and unbounded fan-out. Quant. Inf. Comput. 10 (9), 872–890.
Thapliyal, H., 2016. Mapping of subtractor and adder-subtractor circuits on reversible quantum gates. In: Transactions on Computational Science XXVII. Springer, pp. 10–34.
Thapliyal, H., Ranganathan, N., 2009. Design of efficient reversible binary subtractors based on a new reversible gate. In: 2009 IEEE Computer Society Annual Symposium on VLSI. IEEE, pp. 229–234.
Thapliyal, H., Ranganathan, N., 2010. Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. ACM J. Emerg. Technol. Comput. Syst. (JETC) 6 (4), 14.
Thapliyal, H., Ranganathan, N., 2011. A new reversible design of bcd adder. In: 2011 Design, Automation & Test in Europe. IEEE, pp. 1–4.
Thapliyal, H., Ranganathan, N., 2013. Design of efficient reversible logic-based binary and bcd adder circuits. ACM J. Emerg. Technol. Comput. Syst. (JETC) 9 (3), 17.
Thapliyal, H., et al., 2020. T-count and Qubit Optimized Quantum Circuit Designs of Carry Lookahead Adder. arXiv:2004.01826.
Thapliyal, H., et al., 2013. Progress in reversible processor design: a novel methodology for reversible carry look-ahead adder. In: Transactions on Computational Science XVII. Springer, pp. 73–97.
Toffoli, T., 1980. Reversible computing. In: Proceedings of the 7th Colloquium on Automata, Languages and Programming. Springer-Verlag, Berlin, Heidelberg, pp. 632–644.
Trisetyarso, A., Van Meter, R., 2010. Circuit design for a measurement-based quantum carry-lookahead adder. Int. J. Quant. Inf. 8, 843–867, 05.
Valinataj, M., et al., 2016. Novel low-cost and fault-tolerant reversible logic adders. Comput. Electr. Eng. 53, 56–72.
Vartiainen, J.J., et al., 2004. Implementing Shor's algorithm on Josephson charge qubits. Phys. Rev. A 70, 012319.
Vedral, V., et al., 1996. Quantum networks for elementary arithmetic operations. Phys. Rev. A 54 (1), 147.
Venegas-Andraca, S.E., 2012. Quantum walks: a comprehensive review. Quant. Inf. Process. 11 (5), 1015–1106.
Wang, F., et al., 2016. Improved quantum ripple-carry addition circuit. Sci. China Inf. Sci. 59 (4), 042406.
Wu, C., Cain, C., 2014. A non-qubit quantum adder as one-dimensional cellular automaton. Phys. E Low-dimens. Syst. Nanostruct. 59, 243–247.
Yamashita, S., et al., 2008. DDMF: an efficient decision diagram structure for design verification of quantum circuits under a practical restriction. IEICE Trans. Fund. Electron. Commun. Comput. Sci. 91 (12), 3793–3802.
Zhang, D., et al., 2018. Novel optimized link state routing protocol based on quantum genetic strategy for mobile learning. J. Netw. Comput. Appl. 122, 37–49.
Zhou, R.G., et al., 2014. Novel designs for fault tolerant reversible binary coded decimal adders. Int. J. Electron. 101 (10), 1336–1356.
Zhou, L., et al., 2018. Quantum technique for access control in cloud computing ii: encryption and key distribution. J. Netw. Comput. Appl. 103, 178–184.



F. Orts is a predoctoral researcher at the Informatics Department at the University of Almería, Spain. He studied the Master in Computer Engineering at the University of Almería. He is currently doing his PhD thanks to the Spanish FPI program. His publications and more information about him can be found in %http://hpca.ual.es/forts/. His research interests are Multi-Dimensional Scaling, Quantum Computing and High Performance Computing.

G. Ortega received the Ph.D. degree from the University of Almería (Spain) in 2014. From 2009, she has been working as a member of the TIC-146 Supercomputing-Algorithms research group. She is a PhD Assistant Professor at the Informatics Department at the University of Almería. Her current research work is focused on High Performance Computing and Optimization. Some of her research interest include the study of strategies for load balancing the workload on heterogeneous systems, the parallelization of optimization problems and image processing.

E.F. Combarro received the B.S. degree in mathematics (1997), the M.S. degree in computer science (2002), and the PhD In mathematics (2001) from the University of Oviedo (Spain), where he currently is an Associate Professor at the Computer Science Department. He has authored more than 40 research papers in topics such as computability theory, the theory of fuzzy measures, the computational classification of semifields and text categorization. His current research focus mainly in quantum computing and its applications to algebraic and optimization problems.

E.M. Garzón received her B.Sc. degree in Physics in 1985 from the University of Granada (Spain) and her PhD degree in Computer Engineering in 2000 from the University of Almería (Spain). From 1989 to 1993 she was an Assistant Professor at University of Granada. She is a Full Professor of the Department of Informatics at the University of Almería. Currently, she is head of the Supercomputing-Algorithms research group. Her research interest lies in the field of High-Performance Computing for irregular problems related to Matrix Computation, Image Processing and Global Optimization on heterogeneous HPC platforms and Quantum Computers.