

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*

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.

Keywords: reversible adder, quantum computing, reversible circuit, adder

2010 MSC: 00-01, 99-00

1. Introduction

Reversible computation was first considered in the pioneering works of Landauer [1], Lecerf [2] and Bennet [3] in the context of the energetic cost of com-

*Corresponding author

Email address: `francisco.orts@ual.es` (F. Orts)

putational operations. These authors unveiled deep connections between the
5 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 discov-
ered that the only computational task that implies a energy consumption is
information erasure. Thus, in principle, computations may be physically exe-
10 cuted 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 [4, 5], 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
15 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 [5]. This opens the possibility of using reversible circuits in
order to decrease the energetic consumption of computations, a topic that has
20 gained interest in recent years [6, 7, 8, 9].

Interest in reversible computation in general, and in reversible gates in par-
ticular, also comes from an intimate connection with quantum computing [10].
Quantum computing is a computational paradigm that exploits the physical
properties of subatomic particles in order to achieve speedups in solving com-
putational problems [11, 12]. Far from being solely a theoretical model, several
25 quantum computer prototypes have been constructed in recent years [13, 14, 15].
In fact, Google has recently reported solving, with a quantum computer, a prob-
lem that would be unfeasible to solve with only classical resources, thus achieving
the so-called quantum supremacy [16].

30 The main model of quantum computing is that of quantum circuits, in which
logical gates are replaced with quantum ones [10]. These gates must obey the
laws of quantum physics and, as a consequence, they are always reversible. **Re-**
versibility is therefore no longer the interesting energy saving option that we
have described: it is now a fundamental requirement of quantum computing.

35 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 [17]. But in general, 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 [18, 19].

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 [20], with momentous implications for classical cryptographic protocols such as the RSA cryptosystem [21] or Diffie-Hellman key exchange [22]. 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 [23]. 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 [24] and quantum walks [25]. Although, in general, these methods do not involve arithmetical operations, they use a quantum oracle that is problem-dependent and that may, in some cases, benefit from optimized

reversible circuits for addition, as for instance when algebraic structures are involved [26, 27].

Then, to build a circuit that implements any of these algorithms, is it necessary to design a reversible adder? Are there alternatives already implemented?

70 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.

75 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.

95 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 be 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
130 example. Li et al. presented in [28] an adder which involves 28 quantum gates to perform an addition between two 5-digit binary numbers. On the other hand, Gidney presented an adder that needs 29 gates to perform the same operation [29]. 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
135 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
140 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 gates. Taking into account
145 that a Toffoli gate is composed of 2 CNOT gates and other 3 gates [10], 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
150 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 [30] are followed. Four parameters are defined in [30] to evaluate reversible circuits:

- 155 • 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. Δ is the unit of delay defined in [30]. 1×1 and 2×2 gates have a delay of 1Δ . 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 [30], 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), un-

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 [31]	4	4
Toffoli [10]	5	5

Table 1: Gates and their quantum cost and delay.

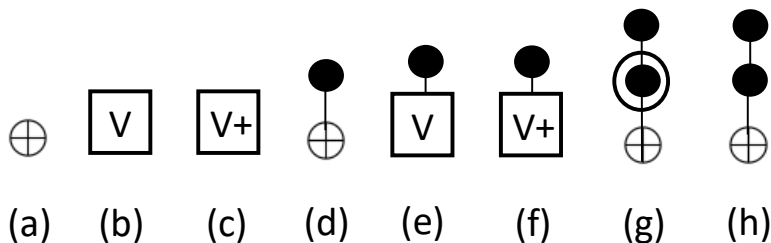


Figure 1: Gates symbols: (a) Pauli- X , (b) V , (c) V^+ , (d)Feynman/ $CNOT$, (e)Controlled- V , (f)Controlled- V^+ , (g)Peres and (h)Toffoli.

185 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
190 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 correc-

195 tion 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
200 incorporation of T gates. In relation to this matter, we also want to remark the
work done in [29], which is focused on improving the number of T gates needed
to build N -bit adders.

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
205 behind this level there are several physical realizations of these reversible gates
and circuits like quantum computation, optic computation, quantum-dot cellu-
lar automata or ultra low power *VLSI* design [32]. 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 [10] (except for
210 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 proba-
bilistically implemented [33]. In these terms, versions of the Toffoli gate like the
presented in [34, 35] are better options than the one described in [10] since they
215 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,
220 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

225 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 [36]. The
circuits have been implemented in Python under this framework and subjected
230 to software tests to verify their correct operation. On the other hand, the
measurement in terms of the metrics of [30] 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.
- 235 • 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
240 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
245 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, gather-
ing 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
250 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 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 [37]. 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 [38] 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 [39].

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 [40]. 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 [40] 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 shows the truth table of the half adder. As consequence, it can be established that $C_{out} = AB$ and $S = A \oplus B$.

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

Table 2: Truth table of the half adder.

- 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 sum 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} [40]. There are two kinds of carry propagate adders: ripple-carry adders and carry-lookahead adders.

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 3: Truth table of the full adder.

- 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.

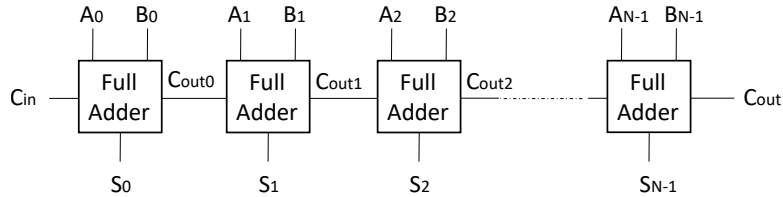


Figure 2: N ripple-carry adder.

305

- 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 $S = A \oplus B$ [10]. This circuit
 310 has a quantum cost of 6, a delay of 6Δ , 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 [41, 42, 43, 44, 45, 46]

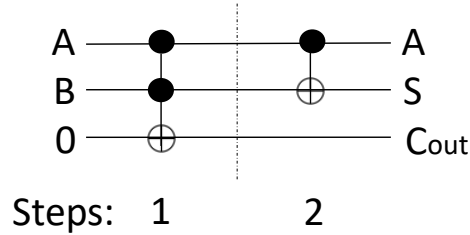


Figure 3: Quantum implementation of the half adder proposed in [10].

On the other hand, the Peres gate [47] can also act as a half adder [48, 49].
 315 The version of the Peres gate presented in [31] achieves the best quantum cost among the 3×3 reversible gates [32]. 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
 320 of a half adder. Fig. 4 shows this use of the Peres gate. It has a quantum cost of 4, a delay of 4Δ , an auxiliary qubit and no garbage outputs.

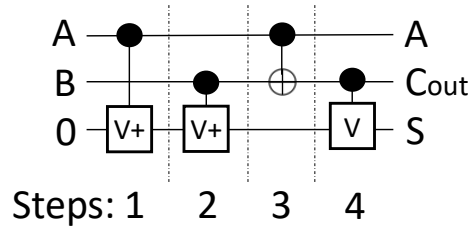


Figure 4: Peres gate acting as a half adder, using the quantum implementation proposed in [31].

In [50], 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 5Δ . To illustrate how
 325 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 a auxiliary qubit.

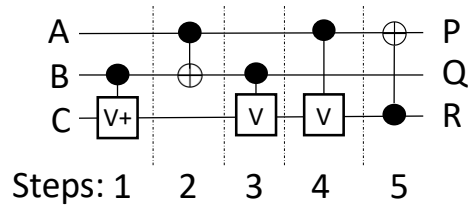


Figure 5: Quantum implementation of the half adder presented in [50].

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

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

330 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 tolerant¹ full adder/subtractor using reversible gates

¹A 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 [10]. Although interesting, the study of the techniques used to achieve this remarkable result is beyond the scope of this review.

was presented in [51]. The circuit of [51] consists of two Feynman double gates (quantum cost 2 [52]) 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. [53] presented a novel reversible half adder and subtractor circuit. This 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, [54] presented a fault tolerant half adder/subtractor, which is similar in terms of quantum cost to [51] (it also consists of two Fredkin gates and two Feynman double gates). The circuit of [54] 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 [10]. Both circuits are shown in Fig. 7 and Fig. 8.

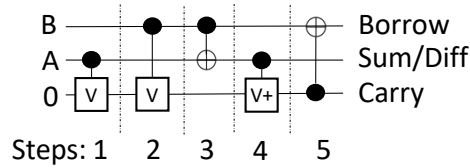


Figure 6: Quantum implementation of the reversible half adder and subtractor circuit presented in [53].

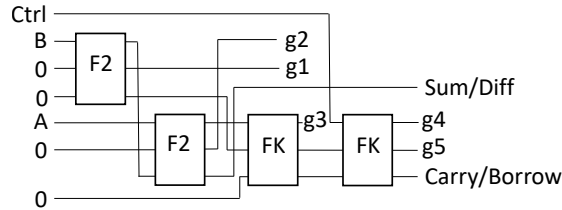


Figure 7: Circuit of reversible fault tolerant half Adder/subtractor proposed in [51]. *F2* represents a Feynman double gate, and *FK* a Fredkin gate.

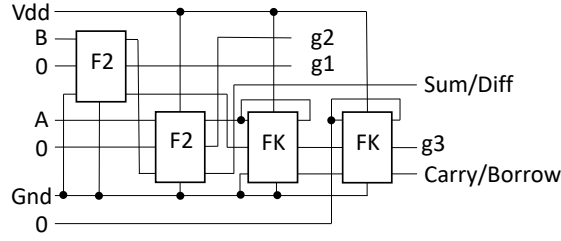


Figure 8: Circuit of reversible fault tolerant half Adder/subtractor proposed in [54]. $F2$ represents a Feynman double gate, and FK a Fredkin gate.

4.2. Full Addder

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) $C_{out2} C_{out1}$. 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 [55]. 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 8Δ , 1 auxiliary qubit and 1 garbage output. These metrics have been calculated considering the version of the Peres gate presented by [31].

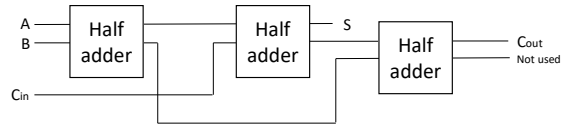


Figure 9: Quantum implementation of a full adder using half adders.

[56] 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 [57], reducing the necessary number of Fredkin gates into 4. In 2004, [58] proposed a new ripple-carry adder.

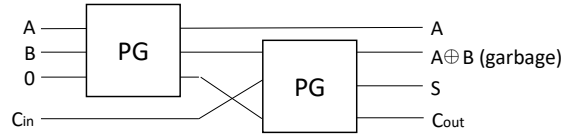


Figure 10: Quantum implementation of a full adder using two Peres gates.

It is based in 2 components (gates): a gate called Majority (*MAJ*), and another
 365 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
 370 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 [58] will be analyzed in a later section). The design of [58] was
 375 improved in later works [59, 60, 61, 62]. In 2016, [63] 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 8Δ , 1 auxiliary qubit and no garbage outputs.

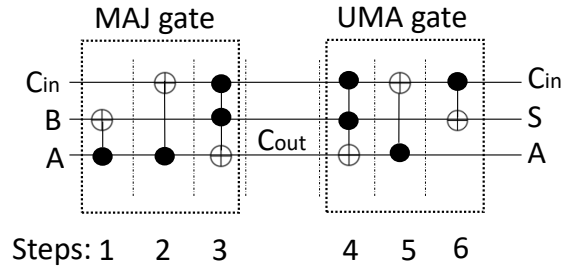


Figure 11: Full adder proposed in [58]. It consists of two gates called MAJ and UMA. C_{out} must be used before applying the UMA gate.

[64] 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
 380 cost of 6, a delay of 4Δ , 1 auxiliary qubit and 1 garbage output. In [65], it was

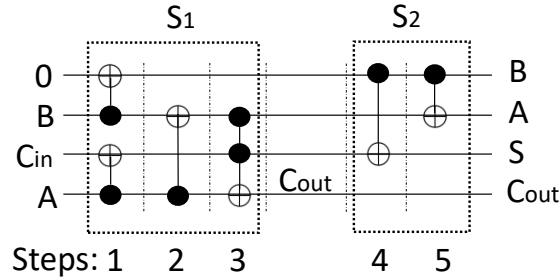


Figure 12: Full adder proposed in [63]. The first sub-circuit $S1$ computes C_{out} and the second one computes S starting from the outputs of $S1$ without erasing C_{out} .

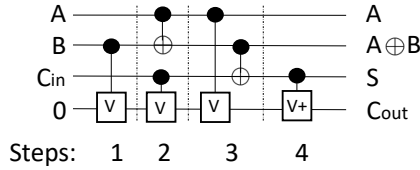


Figure 13: Full adder proposed in [64].

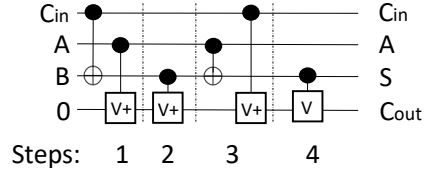


Figure 14: Full adder proposed in [66].

presented a full adder with a quantum cost of 12, delay 12Δ and keeping 1 auxiliary qubit but avoiding garbage outputs. A circuit proposed in [66] improves them. This circuit has the same quantum cost, delay and number of auxiliary qubits than [64], but with no garbage outputs. It consists of 3 Controlled- V^+ gates, 1 Controlled- V gate and 2 CNOT gates. The circuit of [66] is shown in Fig. 14. Also in 2016, [67] proposed two alternative designs of full adder based on reversible gates, but none of them improves the adder of [66]. The best of the adders of [67] has a quantum cost of 8, a delay of 8Δ , 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 [68]. For instance, [69] proposed a fault tolerant full adder with a quantum cost of 11, the same delay, 2 auxiliary inputs and 3 garbage outputs. Previously to [69], several fault tolerant full adders were proposed: [70] with a quantum cost of 14 and 3 garbage outputs, [57] which has been already analysed in this section, and [71] with a quantum cost of 18 and 6 garbage outputs. Other fault

tolerant adders are [72] with a quantum cost of 14 and 3 garbage outputs, [73] with a quantum cost of 14 and 3 garbage outputs, and [74] with a quantum cost of 8, delay 7 and 2 garbage outputs (Fig. 15). The circuit of [68] has a quantum cost of 10 and 3 garbage outputs, but it offers interesting benefits against [74] in terms of the transistor count or the total logical calculation (the number of *XOR*, *AND*, and *NOT* operations).

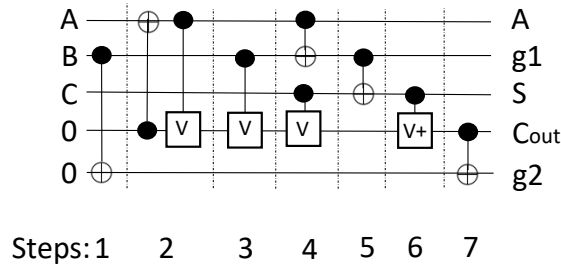


Figure 15: Fault tolerant full adder proposed in [74].

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. [75] 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 [51] 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 [52]) 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 [76], 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, [77] 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 [66] (Fig. 14) can be converted into a full adder/subtractor adding a selection qubit and a CNOT gate, having a quantum

420 cost of 8 and a delay of 5Δ . On the other hand, the *RSG* gate presented in [53], 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 10Δ , 1 auxiliary input and no garbage outputs. If we only consider the
425 adder path of the circuit, the quantum cost is reduced to 10. Also in 2018, [54] 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/subtractors (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.

430 In [49], 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
435 and 2 Feynman double gates. 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 [78]. 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 [30], a 4×4 gate cannot have a
440 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 $P = A, Q = A\bar{D} \oplus C, R = (A\bar{D} \oplus C) \oplus B$ and $S = (A\bar{D} \oplus C)B \oplus AC \oplus A\bar{D}$. Setting $D = 0$, it acts as a full adder with 1 garbage output.
- Full adder using *DKG* gates: this circuit is defined in [79]. Similar to
445 *BKG* gate, the internal design of this gate is not described. It has four inputs A', B', C' and D' , and four outputs $P = B', Q = \bar{A}'C' + A'\bar{D}', R = (A' \oplus B')(C' \oplus D') \oplus C'D'$ 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 $S = A \oplus B \oplus C = Sum$.

450

According to [30], 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 [80]. Its quantum cost is higher than the version of Fig. 10, and it has more garbage outputs.
- 455 • Full adder using IG gates: the IG gate was presented in [81]. 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.
- 460 • Full Adder using Feynman and Fredkin gates: this full adder is the version proposed in [67], which has already been studied in this section.

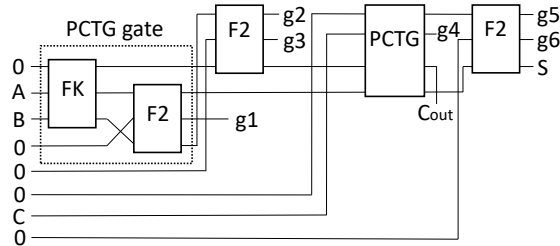


Figure 16: Full adder using $PCTG$ gates. A $PCTG$ gate consists of 1 Fredkin gate (FK) and 1 Feynman double gate ($F2$).

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 ,

465

I and GO are given by the following equations:

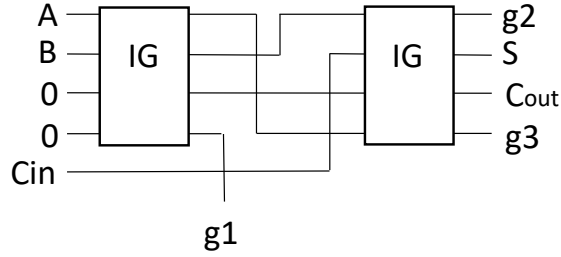


Figure 17: Full adder using IG gates.

$$QC_{ripple} = N \cdot QC_{fulladder}$$

$$D_{ripple} = N \cdot D_{fulladder}$$

$$I_{ripple} = N \cdot I_{fulladder}$$

$$GO_{ripple} = N \cdot GO_{fulladder}$$

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:

$$QC_{ripple} = (N - 1) \cdot QC_{fulladder} + QC_{halfadder}$$

$$D_{ripple} = (N - 1) \cdot D_{fulladder} + D_{halfadder}$$

$$I_{ripple} = (N - 1) \cdot I_{fulladder} + I_{halfadder}$$

$$GO_{ripple} = (N - 1) \cdot GO_{fulladder} + GO_{halfadder}$$

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
 470 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 through every pair of bits A_i and B_i [40]. In terms of quantum computers, which currently have few resources, this kind of adders are the best
 475 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 [58] 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 $10N\Delta$ 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 [59] (Quantum cost: $26N - 29$, delay: $24N - 27\Delta$, number of auxiliary inputs: 0, number of garbage outputs: 0), [82] (QC: $15N - 9$, D: $13N - 7\Delta$, AI: 0, GO: 0), and [83] (QC: $13N - 8$, D: $11N - 4\Delta$, AI: 0, GO: 0). The circuit of [83] is shown in Fig. 19 for the case $N = 4$.

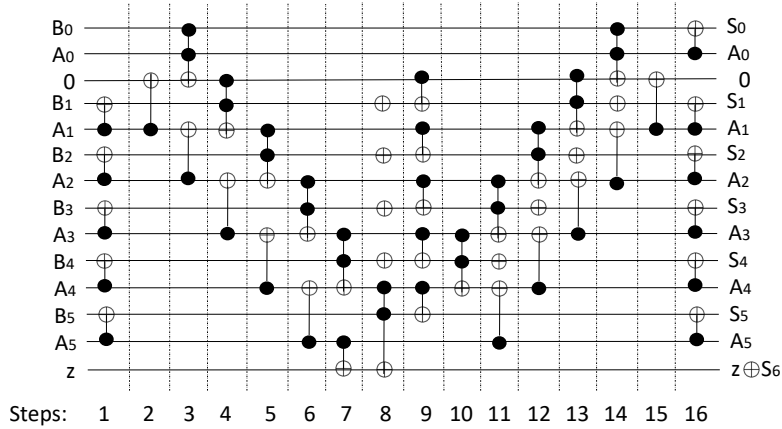


Figure 18: Ripple-carry adder for $N = 6$ (assuming $C_{in} = 0$) proposed in [58].

On the other hand, there are several proposals [58, 84, 85] which consider the input C_{in} . [86] 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 [87], but in [86] is optimized, using only 1 $CNOT$

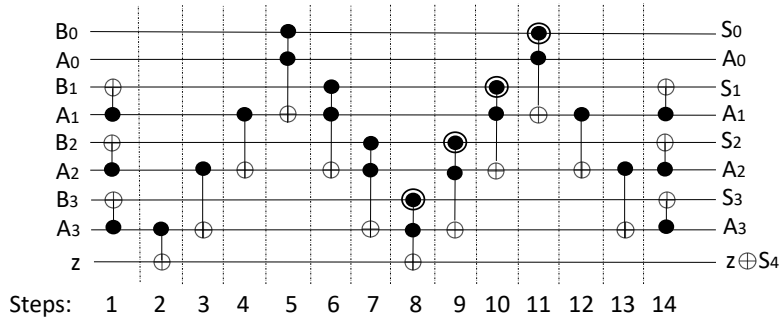


Figure 19: Ripple-carry adder without carry in for $N = 4$ proposed in [83].

495 gate, 2 Controlled- V gates and 1 Controlled- V^+ gate. It has a quantum cost of 4 and a delay of 4Δ . The resulting adder has a quantum cost of $15N - 6$, a delay of $9N + 5\Delta$, and neither auxiliary inputs nor garbage outputs (Fig. 20). The ripple-carry adder of [65] improves the quantum cost and delay of [86] ($12N$ and $10N\Delta$ respectively) at the cost of using $4N$ auxiliary inputs. The optical
 500 reversible ripple-carry adder with C_{in} proposed in [88] is remarkable. It is not better than [86] in terms of the metrics of [30], but it improves it in terms of optical cost. Optical cost is one of the most important metric parameters in optical computing.

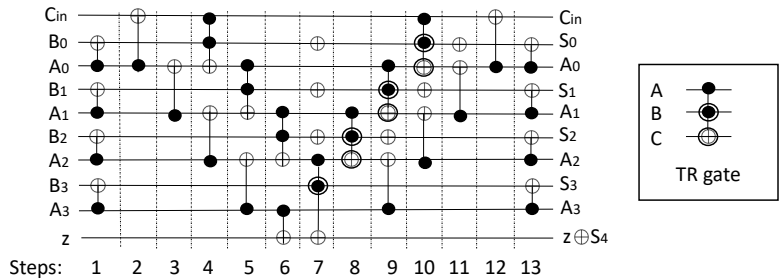


Figure 20: Ripple-carry adder with carry in for $N = 4$ proposed in [86].

Moving away from the goal of reducing the quantum cost, Gidney presented
 505 an alternative gate to the Toffoli gate focused on reducing the cost of T gates [29]. This new gate has a higher quantum cost than the Toffoli gate if we consider the version proposed in [10]. 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 $18N - 2$, a delay of $15N - 5\Delta$, 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. [28]. 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).

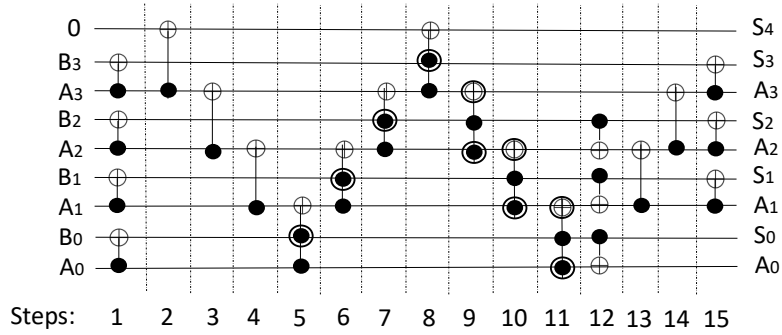


Figure 21: Ripple-carry adder with carry in for $N = 4$ proposed in [28].

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) [40]:

- 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 $G_i = A_i B_i$.

- 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:

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

These adders are faster than the previous ones. However, they have a higher quantum cost since they need more operations to anticipate the computation of the G_i and P_i signals [40].

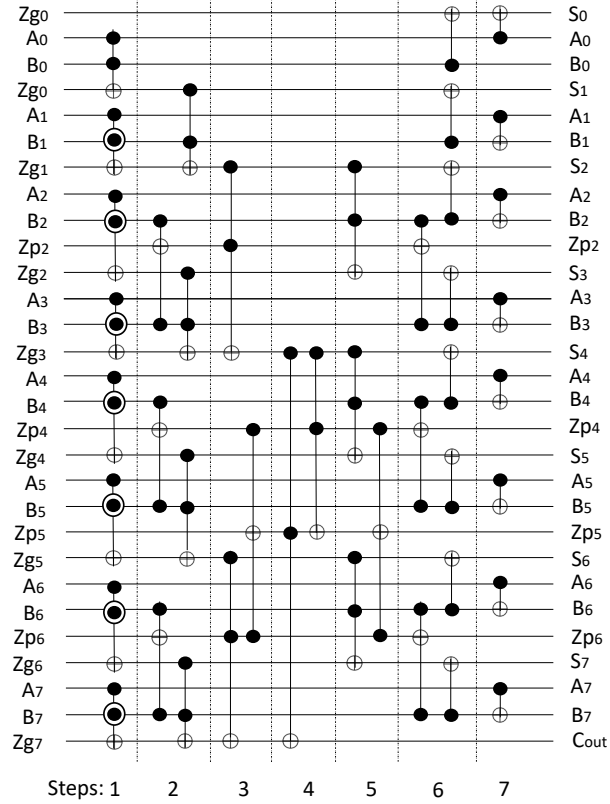


Figure 22: Example of the carry-lookahead adder proposed in [89] for the case $N = 8$. It is built using Toffoli, $CNOT$ and Peres gates. A_i and B_i are the numbers to be added, C_{out} are the carry out and S_i are the digits of the sum. Zg_i and Zp_i are auxiliary inputs used to compute G_i and P_i respectively.

In 2004, Draper et al. proposed a logarithmic-depth reversible carry-lookahead
535 adder which improves the delay of the previous linear-depth adders [90]. 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 $\log N + \log N/3 + 7$,
 $5N/4$ auxiliary inputs and no garbage outputs. Thapliyal et al. optimized the
methodology to compute a carry-lookahead addition (without C_{in}), improving
540 the quantum cost and delay of the previous adder [89]. It has a quantum cost
of $26N - 15W(N) - 15\log(N - 4)$ and a delay of $\log N + \log N/3 + 2$. 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
545 example of this circuit is shown in Fig. 22. It works as follows:

- 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
550 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
555 $Zg_{i-1} \oplus = Zg_{i-3}Zp_{i-2}$ 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
560 $Zg_i \oplus = B_{i+1}$ to compute S_{i+1} .

- Step 7: Compute S_0 and uncompute B_i .

Two reversible carry-lookahead adders were presented in [91]. 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 presented in [89]. However, this circuit presents $3 \times N$ garbage outputs, whereas the circuit of [89] has no garbage outputs. Following Bennett's garbage removal scheme [3], 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 [89]. The second adder presented in [91] also presents garbage outputs, so the same procedure can be applied to it. Its final quantum cost and delay do not improve that of [89].

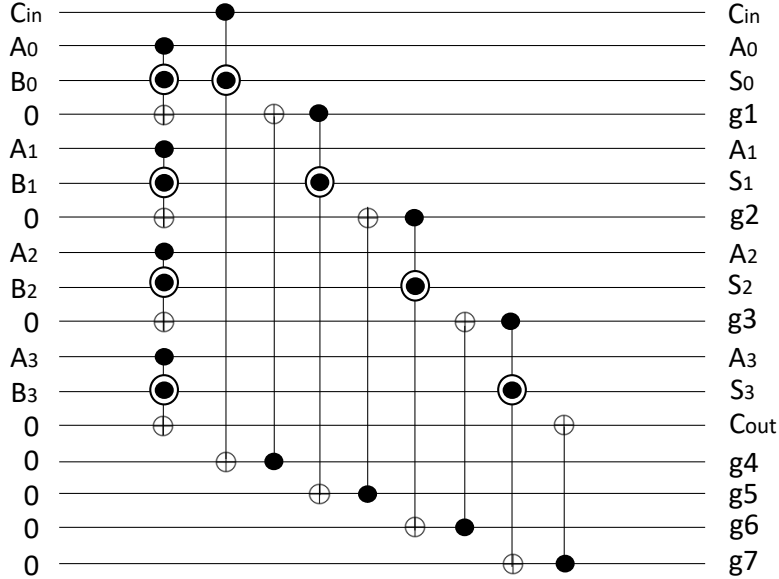


Figure 23: Example of the first design of a carry-lookahead adder proposed in [91] for the case $N = 4$. It is built using $CNOT$ and Peres gates. A_i and B_i are the numbers to be added, C_{in} and C_{out} are the carry in and the carry out respectively, S_i are the digits of the sum, and g_i are garbage outputs.

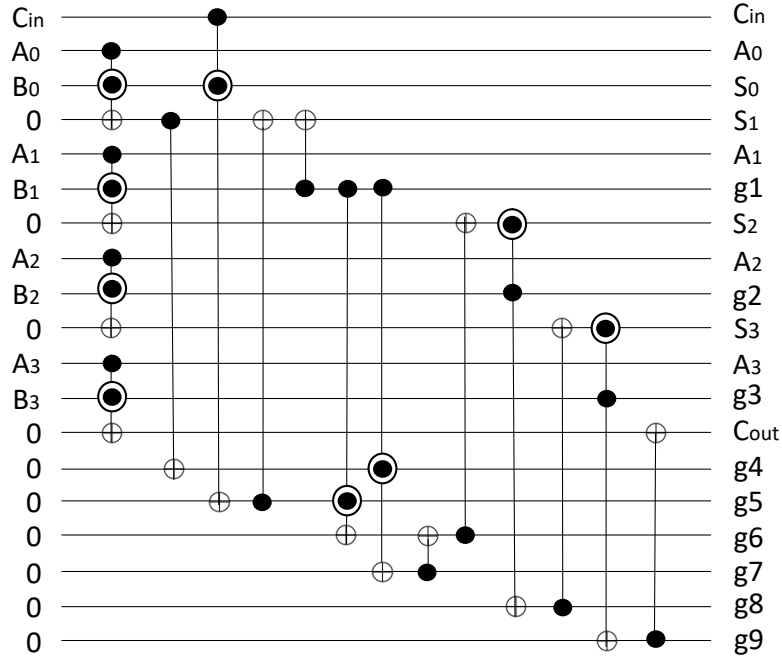


Figure 24: Example of the second design of a carry-lookahead adder proposed in [91] for the case $N = 4$. It is built using *CNOT* and Peres gates. A_i and B_i are the numbers to be added, C_{in} and C_{out} are the carry in and the carry out respectively, S_i are the digits of the sum, and g_i are garbage outputs.

575 A reversible adder presented in [92] improves the quantum cost and delay of [89] 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 5Δ . 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 [3] would increase this number to $2 \times 11N + (N + 1)$.
 580 For instance, for the case $N = 4$, this circuit would have a quantum cost of 93, whereas the adders of [91] and [89] have 61 and 55 respectively.

585 Focusing on fault-tolerance, Thapliyal et al. presented in [93] four adders based on their own design of [89] but optimizing the number of T gates at the cost of increasing the rest of the metrics. To achieve this optimization they

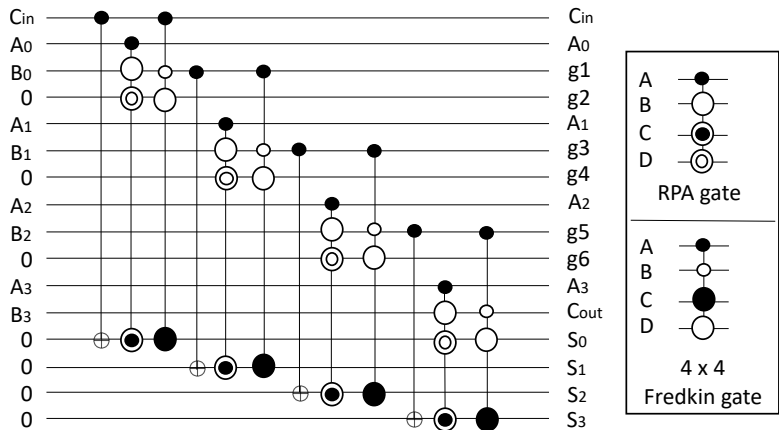


Figure 25: Carry-lookahead adder proposed in [92], for the case $N = 4$. It is built using 4×4 RPA and Fredkin gates. A_i and B_i are the numbers to be added, C_{in} and C_{out} are the carry in and the carry out respectively, S_i is the sum, and g_i are garbage outputs.

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 [89] 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 [31] (Fig. 4) achieves the best value. The proposals of [53] and [50] have a quantum cost of 5, one more than [31], but [53] can also act as a subtractor, so the extra value is justified in this case. [10] has a quantum cost of 6, and [51] has the highest quantum cost, 14. The quantum cost of [51] 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 [51] which has 3. Only [51] presents garbage outputs.

Adder	Quantum cost	Delay Δ	Ancilla inputs	Garbage outputs	Adder/subtractor
Kaur et al. [51]	14	14	3	3	Yes
Nielsen et al. [10]	6	6	1	0	
Yamashita et al. [50]	5	5	1	0	
Sarma et al. [53]	5	5	1	0	Yes
Hung et al. [31]	4	4	1	0	

Table 5: Comparative evaluation of half adders.

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 optimised adder in terms of quantum cost, delay and garbage output is [66][a], with 6, 4Δ and 0 respectively. The adder of [64] 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 [58], but it has a quantum cost and a delay higher than the average (14 and 14Δ respectively). [55], [64], [67], [69], [74], [75], [76] and [77] present garbage outputs, so their quantum cost and delay would be higher if they were

uncomputed [3]. The best adder/subtractor is the proposed in [66][b] as it has no garbage outputs, and it has the best quantum cost and delay among the adder/subtractors, 8 and 5Δ respectively. It is followed by [75] with a quantum cost of 10 and delay 10Δ , but with 3 garbage outputs. Therefore, the adder of [53], which has a quantum cost of 15 , would be a better option than [75] as it has no garbage outputs. Finally, focusing on the fault tolerant adders, [74] and [69] are the most optimised options. [74] presents lower values of quantum cost and delay, and both have the same number of auxiliary inputs. Moreover, [74] 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 [66][a] (the best full adder) results in a ripple-carry adder which is worse than, for instance, the improved ripple-carry adder presented in [83]. 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 supports C_{in} or not.

In terms of auxiliary inputs, only [58] and [65] have them. The adder of [58] has only 1 , whereas the adder of [65] employs $4N$. In terms of quantum cost, [65] 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 [28] improves the quantum cost of [65], but only when $N < 10$. [83] is the third best adder in these terms. However, both [28] and [83] do not support C_{in} . In terms of delay, the circuit of [86] gets the best value, $9N + 5\Delta$, followed by the non fault-tolerant adder of [28], which has $10N - 4\Delta$. Finally, we can highlight that no circuit has garbage outputs.

The adder presented by Gidney [29] 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 [28] does not improve the quantum cost nor the delay with

Adder	Quantum cost	Delay Δ	Ancilla inputs	Garbage outputs	Adder/ subtractor	Fault tolerant
Saligram et al. [76]	18	18	5	6	Yes	
Yamashita et al. [50]	15	15	3	0		
Sarma et al. [53]	15	10	1	0	Yes	
Cuccaro et al. [58]	14	14	0	0		
Hung et al. [31]	12	12	3	0		
Nagamani et al. [65]	12	12	1	0		
Mitra et al. [69]	11	11	2	3		Yes
Kumar et al. [77]	11	11	4	4	Yes	
Wang et al. [63]	10	8	1	0		
Rangaraju et al. [75]	10	10	1	3	Yes	
Singh et al. [67]	8	8	1	1		
Bhagyalakshmi et al. [55]	8	8	1	1		
Thapliyal [66][b]	8	5	2	0	Yes	
Zhou et al. [74]	8	7	2	2		Yes
Maslov et al. [64]	6	4	1	1		
Thapliyal[66][a]	6	4	1	0		

Table 6: Comparative evaluation of full adders.

respect to the previous one, but it represents a substantial improvement in terms of ancilla inputs.

Adder	Quantum cost	Delay Δ	Ancilla inputs	Garbage outputs	C_{in}	Fault tolerant
Li et al. (2) [28]	$35N - 25$	$16N - 3$	0	0		Yes
Takahashi et al. [59]	$26N - 29$	$24N - 27$	0	0		
Gidney [29]	$18N - 2$	$15N - 5$	N	0		Yes
Cuccaro et al. [58]	$17N - 12$	$10N$	1	0		
Takahashi et al. (2) [82]	$15N - 9$	$13N - 7$	0	0		
Thapliyal et al. [86]	$15N - 6$	$9N + 5$	0	0	Yes	
Thapliyal et al. (2) [83]	$13N - 8$	$11N - 4$	0	0		
Li et al. (1) [28]	$13N - 10$	$10N - 4$	0	0		
Nagamani et al. [65]	$12N$	$10N$	$4N$	0	Yes	

Table 7: Comparative evaluation of ripple-carry adders.

5.4. Carry-lookahead Adders

655 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 [91], and [92]. It is not useful to compare such circuits with those which have uncomputed their garbage outputs since uncomputing these outputs
660 following [3] would increase the quantum cost, delay and auxiliary inputs [33]. On the other hand, a 4×4 Fredkin gate is used in the case of [92]. 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

665 the carry-lookahead adders for the sake of clarity). Considering the remaining
non fault-tolerant adders, [90] and [89], it can be concluded that [89] 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 [93], they
do not improve any of the metrics from [30] to the previous adders. However,
670 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
675 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 some-
680 thing 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, [28]
and [83] for the case without C_{in} and [65] and [86] 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, [89] is the most opti-
685 mized 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. [28],[29]
(ripple-carry adders), and the four adders of [93] (carry-lookahead adders), are
the best exponents in this new stage of optimization.

6. Conclusions

690 In this work, a revision on the state-of-the-art reversible adders has been car-
ried out. First, appropriate metrics have been considered for the measurement
and comparison of quantum circuits. Second, the adders have been classified

Adder	Quantum cost	Delay Δ	Ancilla inputs	Fault-tolerance
Thapliyal et al. [93][a]	$> 40N$	$O(\log N)$	$4N - 2W(N)$ $-2\log(N)$	Yes
Thapliyal et al. [93][b]	$> 40N$	$O(\log N)$	$2N - W(N)$ $-\log(N) + 1$	Yes
Thapliyal et al. [93][c]	$> 40N$	$O(\log N)$	$4N - 2W(N)$ $-2\log(N)$	Yes
Thapliyal et al. [93][d]	$> 40N$	$O(\log N)$	$2N - W(N)$ $-\log(N) + 1$	Yes
Draper et al. [90]	$28N - 15W(N)$ $-15\log(N) - 6$	$\log N + \log N/3$ $+7$	$5N/4$	
Thapliyal et al. [89]	$26N - 15W(N)$ $-15\log(N - 4)$	$\log N + \log N/3$ $+2$	$5N/4$	

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 [93] 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.

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
695 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
700 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
705 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
710 exemplify some of the most remarkable adders for some values on the number of inputs.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science throughout Project RTI2018-095993-B-I00, by the Regional Ministry of
715 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.

720 **References**

- [1] R. Landauer, Irreversibility and heat generation in the computing process, IBM J. Res. Dev. 5 (3) (1961) 183–191.
- [2] Y. Lecerf, Machines de Turing réversibles, Comptes Rendus Hebdomadaires des Séances de L'academie des Sciences 257 (1963) 2597–2600.
- 725 [3] C. H. Bennett, Logical reversibility of computation, IBM journal of Research and Development 17 (6) (1973) 525–532.
- [4] T. Toffoli, Reversible computing, in: Proceedings of the 7th Colloquium on Automata, Languages and Programming, Springer-Verlag, Berlin, Heidelberg, 1980, pp. 632–644.
- 730 [5] E. Fredkin, T. Toffoli, Conservative logic, International Journal of Theoretical Physics 21 (3-4) (1982) 219–253.
- [6] E. Cohen et al., All-optical design for inherently energy-conserving reversible gates and circuits, Nature communications 7 (2016) 11424.
- [7] Anamika, R. Bhardwaj, Reversible logic gates and its performances, in: 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018, pp. 226–231.
- 735 [8] J. Chaves et al., Energy efficient QCA circuits design: simulating and analyzing partially reversible pipelines, Journal of Computational Electronics 17 (1) (2018) 479–489.
- 740 [9] S. L. Sahu L., Kumar U., 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, 2019, pp. 229–238.
- [10] M. A. Nielsen, I. L. Chuang, Quantum computation and quantum information: 10th anniversary edition (2011).
- 745

- [11] L. Zhou et al., Quantum technique for access control in cloud computing ii: Encryption and key distribution, *Journal of Network and Computer Applications* 103 (2018) 178–184.
- [12] D. Zhang et al., Novel optimized link state routing protocol based on quantum genetic strategy for mobile learning, *Journal of Network and Computer Applications* 122 (2018) 37–49.
- [13] N. M. Linke et al., Experimental comparison of two quantum computing architectures, *Proceedings of the National Academy of Sciences* 114 (13) (2017) 3305–3310.
- [14] K. Michielse et al., Benchmarking gate-based quantum computers, *Computer Physics Communications* 220 (2017) 44 – 55.
- [15] C. Neill et al., A blueprint for demonstrating quantum supremacy with superconducting qubits, *Science* 360 (6385) (2018) 195–199.
- [16] F. Arute et al., Quantum supremacy using a programmable superconducting processor, *Nature* 574 (7779) (2019) 505–510.
- [17] Y. Shi, Both Toffoli and controlled-not need little help to do universal quantum computing, *Quantum Info. Comput.* 3 (1) (2003) 84–92.
- [18] J. J. Vartiainen et al., Implementing Shor’s algorithm on Josephson charge qubits, *Phys. Rev. A* 70 (2004) 012319.
- [19] A. G. Fowler et al., Implementation of Shor’s algorithm on a linear nearest neighbour qubit array, *Quantum Info. Comput.* 4 (4) (2004) 237–251.
- [20] P. W. Shor, 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, 1994, pp. 124–134.
- [21] R. L. Rivest et al., A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM* 21 (2) (1978) 120–126.

- [22] W. Diffie, M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory* 22 (6) (1976) 644–654.
- 775 [23] A. Pavlidis, D. Gizopoulos, Fast Quantum Modular Exponentiation Architecture for Shor’s Factorization Algorithm, *Quantum Information & Computation* 14 (7) (2014) 649–682.
- [24] L. K. Grover, 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, 1996, pp. 212–219.
- 780 [25] S. E. Venegas-Andraca, Quantum walks: a comprehensive review, *Quantum Information Processing* 11 (5) (2012) 1015–1106.
- [26] E. F. Combarro et al., A quantum algorithm for the commutativity of finite dimensional algebras, *IEEE Access* 7 (2019) 45554–45562.
- 785 [27] E. F. Combarro et al., Quantum walks for the determination of commutativity of finite dimensional algebras, *Journal of Computational and Applied Mathematics* 354 (2019) 496–506.
- [28] M. Mohammadi et al., Efficient quantum arithmetic operation circuits for quantum image processing, *Science China Physics Mechanics & Astronomy* 63 (2020) 1–13.
- 790 [29] C. Gidney, Halving the cost of quantum addition, *Quantum* 2 (74) (2018) 10–22331.
- [30] M. Mohammadi et al., On figures of merit in reversible and quantum logic designs, *Quantum Information Processing* 8 (4) (2009) 297–318.
- 795 [31] W. N. Hung et al., 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) (2006) 1652–1663.

- [32] H. Thapliyal, N. Ranganathan, Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs, ACM Journal on Emerging Technologies in Computing Systems (JETC) 6 (4) (2010) 14.
- [33] F. Orts et al., An optimized quantum circuit for converting from sign-magnitude to two's complement, Quantum Information Processing 18 (11) (2019) 332.
- [34] B. P. Lanyon et al., Simplifying quantum logic using higher-dimensional hilbert spaces, Nature Physics 5 (2) (2009) 134.
- [35] K. Lemr et al., Experimental implementation of optimal linear-optical controlled-unitary gates, Physical review letters 114 (15) (2015) 153602.
- [36] D. S. Steiger et al., ProjectQ: an open source software framework for quantum computing, Quantum 2 (49) (2018).
- [37] D. Große et al., Exact synthesis of elementary quantum gate circuits for reversible functions with don't cares, in: 38th International Symposium on Multiple Valued Logic (ismvl 2008), IEEE, 2008, pp. 214–219.
- [38] D. M. Miller et al., Mapping NCV circuits to optimized Clifford+T circuits, in :International Conference on Reversible Computation, Springer, 2014, pp. 163–175.
- [39] X. He et al., The Mapping and Optimization Method of Quantum Circuits for Clifford+T Gate, Journal of Applied Mathematics and Physics, 7 (11) (2019) 2796–2810.
- [40] S. Harris, D. Harris, Digital design and computer architecture: arm edition, Morgan Kaufmann, 2015.
- [41] D. Chatterjee, A. Roy, A transmon-based quantum half-adder scheme, Progress of Theoretical and Experimental Physics 2015 (9).
- [42] G. A. Barbosa, Quantum half-adder, Physical Review A 73 (2006) 052321.

- 825 [43] S. Srivastava et al., Quantum half-adder boolean logic gate with a nano-graphene molecule and graphene nano-electrodes, *Chemical Physics Letters* 667 (2017) 301–306.
- [44] C. Wu, C. Cain, A non-qubit quantum adder as one-dimensional cellular automaton, *Physica E: Low-dimensional Systems and Nanostructures* 59
830 (2014) 243–247.
- [45] G. Dridi et al., The mathematics of a quantum Hamiltonian computing half adder boolean logic gate, *Nanotechnology* 26 (34) (2015) 344003.
- [46] L. Eloie et al., Quantum arithmetics via computation with minimized external control: The half-adder, *Physical Review A* 97 (2018) 062321.
- 835 [47] A. Peres, Reversible logic and quantum computers, *Physical review A* 32 (6) (1985) 3266.
- [48] E. P. A. Akbar et al., Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology, *Microelectronics Journal* 42 (8) (2011) 973–981.
- 840 [49] K. Batish et al., Comparative analysis for performance evaluation of full adders using reversible logic gates, in: 2018 International Conference on Intelligent Circuits and Systems (ICICS), IEEE, 2018, pp. 126–132.
- [50] S. Yamashita et al., DDMF: An efficient decision diagram structure for design verification of quantum circuits under a practical restriction, *IEICE Transactions on Fundamentals of Electronics, Communications and*
845 *Computer Sciences* 91 (12) (2008) 3793–3802.
- [51] P. Kaur, B. S. Dhaliwal, Design of fault tolerant full adder/subtractor using reversible gates, in: 2012 International Conference on Computer Communication and Informatics, IEEE, 2012, pp. 1–5.
- 850 [52] B. Parhami, Fault-tolerant reversible circuits, in: 2006 Fortieth Asilomar Conference on Signals, Systems and Computers, IEEE, 2006, pp. 1726–1729.

- [53] R. Sarma, R. Jain, Quantum gate implementation of a novel reversible half adder and subtractor circuit, in: 2018 International Conference on Intelligent Circuits and Systems (ICICS), IEEE, 2018, pp. 72–76.
855
- [54] B. Balaji et al., Full adder/ subtractor using reversible logic, International Journal of Pure and Applied Mathematics 120 (6) (2018) 437–446.
- [55] H. Bhagyalakshmi, M. Venkatesha, An improved design of a multiplier using reversible logic gates, International journal of engineering science and technology 2 (8) (2010) 3838–3845.
860
- [56] A. B. Khlopov et al., Reversible logic synthesis by iterative compositions., in: Iwls, 2002, pp. 261–266.
- [57] J. Bruce et al., 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 2002, IEEE, 2002, pp. 83–88.
865
- [58] S. A. Cuccaro et al., D. P. Moulton, A new quantum ripple-carry addition circuit, arXiv preprint quant-ph/0410184.
- [59] Y. Takahashi, N. Kunihiro, A linear-size quantum circuit for addition with no ancillary qubits, Quantum Information & Computation 5 (6) (2005) 440–448.
870
- [60] Y. Takahashi, N. Kunihiro, A fast quantum circuit for addition with few qubits, Quantum Information & Computation 8 (6) (2008) 636–649.
- [61] A. Trisetarso, R. Van Meter, Circuit design for a measurement-based quantum carry-lookahead adder, International Journal of Quantum Information 8 (05) (2010) 843–867.
875
- [62] R. V. Meter et al., Arithmetic on a distributed-memory quantum multi-computer, ACM Journal on Emerging Technologies in Computing Systems (JETC) 3 (4) (2008) 2.

- 880 [63] F. Wang et al., Improved quantum ripple-carry addition circuit, *Science China Information Sciences* 59 (4) (2016) 042406.
- [64] D. Maslov et al., C. Negrevergne, Quantum circuit simplification and level compaction, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27 (3) (2008) 436–444.
- 885 [65] A. Nagamani et al., Design of optimized reversible binary adder/subtractor and BCD adder, in: *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, 2014, pp. 774–779.
- [66] H. Thapliyal, Mapping of subtractor and adder-subtractor circuits on reversible quantum gates, in: *Transactions on Computational Science XXVII*, Springer, 2016, pp. 10–34.
- 890 [67] V. P. Singh, M. Rai, Verilog design of full adder based on reversible gates, in: *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall)*, IEEE, 2016, pp. 1–5.
- [68] M. Valinataj et al., Novel low-cost and fault-tolerant reversible logic adders, *Computers & Electrical Engineering* 53 (2016) 56–72.
- 895 [69] S. K. Mitra, A. R. Chowdhury, Minimum cost fault tolerant adder circuits in reversible logic synthesis, in: *2012 25th International Conference on VLSI Design*, IEEE, 2012, pp. 334–339.
- [70] M. Islam et al., Efficient approaches for designing fault tolerant reversible carry look-ahead and carry-skip adders, *MASAUM Journal of Basic and Applied Sciences* 1 (3) (2009) 354–360.
- 900 [71] M. Haghparast, K. Navi, Design of a novel fault tolerant reversible full adder for nanotechnology based systems, *World Applied Sciences Journal* 3 (1) (2008) 114–118.
- 905 [72] M. S. Islam et al., 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, 2009, pp. 396–401.

- [73] F. Dastan, M. Haghparast, A novel nanometric fault tolerant reversible divider, International Journal of Physical Sciences 6 (24) (2011) 5671–5681. 910
- [74] R. G. Zhou et al., Novel designs for fault tolerant reversible binary coded decimal adders, International Journal of Electronics 101 (10) (2014) 1336–1356.
- [75] H. Rangaraju et al., Low power reversible parallel binary adder/subtractor, International journal of VLSI design and Communication Systems 1 (3) 915 (2010) 23–34.
- [76] R. Saligram, T. Rakshith, 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, 2013, pp. 1–6. 920
- [77] P. K. Kumar et al., Optimal design of reversible parity preserving new full adder/full subtractor, in: 2017 11th International Conference on Intelligent Systems and Control (ISCO), IEEE, 2017, pp. 368–373.
- [78] B. Bhuvana, K. B. VS, Design of reversible adders using a novel reversible bkg gate, in: 2016 Online International Conference on Green Engineering and Technologies (IC-GET), IEEE, 2016, pp. 1–6. 925
- [79] D. Krishnaveni et al., Design of an efficient reversible 8x8 Wallace tree multiplier, World Applied Sciences Journal 20 (8) (2012) 1159–1165.
- [80] H. Rohini, S. Rajashekar, Design of reversible logic based basic combinational circuits, Communications on Applied Electronics 5 (9) (2016) 38–43. 930
- [81] M. S. Islam et al., Synthesis of fault tolerant reversible logic circuits, in: 2009 IEEE Circuits and Systems International Conference on Testing and Diagnosis, IEEE, 2009, pp. 1–4.

- [82] Y. Takahashi et al., Quantum addition circuits and unbounded fan-out,
935 Quantum Information and Computation 10 (9) (2010) 872–890.
- [83] H. Thapliyal, N. Ranganathan, Design of efficient reversible logic-based
binary and bcd adder circuits, ACM Journal on Emerging Technologies in
Computing Systems (JETC) 9 (3) (2013) 17.
- [84] V. Vedral et al., Quantum networks for elementary arithmetic operations,
940 Physical Review A 54 (1) (1996) 147.
- [85] M. Skoneczny et al., Reversible fourier transform chip, in: 2008 15th Inter-
national Conference on Mixed Design of Integrated Circuits and Systems,
IEEE, 2008, pp. 281–286.
- [86] H. Thapliyal, N. Ranganathan, A new reversible design of bcd adder, in:
945 2011 Design, Automation & Test in Europe, IEEE, 2011, pp. 1–4.
- [87] H. Thapliyal, N. Ranganathan, Design of efficient reversible binary sub-
tractors based on a new reversible gate, in: 2009 IEEE computer society
Annual symposium on VLSI, IEEE, 2009, pp. 229–234.
- [88] S. Kotiyal, Design exploration and application of reversible circuits in
950 emerging technologies, Ph.D. thesis, Computer science and engineering de-
partment. University of South Florida (2016).
- [89] H. Thapliyal et al., Progress in reversible processor design: a novel method-
ology for reversible carry look-ahead adder, in: Transactions on Computa-
tional Science XVII, Springer, 2013, pp. 73–97.
- [90] T. G. Draper et al., A logarithmic-depth quantum carry-lookahead adder,
955 arXiv preprint quant-ph/0406142.
- [91] M. Rahmati et al., Novel designs of a carry/borrow look-ahead
adder/subtractor using reversible gates, Journal of Computational Elec-
tronics 16 (3) (2017) 856–866.

- 960 [92] N. J. Lisa, H. M. H. Babu, Design of a compact reversible carry look-ahead adder using dynamic programming, in: 2015 28th International Conference on VLSI Design, IEEE, 2015, pp. 238–243.
- [93] H. Thapliyal et al., T-count and Qubit Optimized Quantum Circuit Designs of Carry Lookahead Adder, arXiv preprint arXiv:2004.01826.