Received May 3, 2021; revised March 12, 2022; accepted March 22, 2022; date of publication March 30, 2022;
date of current version May 6, 2022.

Digital Object Identifier 10.1109/TQE.2022.3163624

On the Realistic Worst-Case Analysis of Quantum Arithmetic Circuits

ALEXANDRU PALER[1,3,4], OUMAROU OUMAROU[2], AND ROBERT BASMADJIAN[2]

[1] Aalto University, 02150 Espoo, Finland
[2] Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany
[3] University of Texas at Dallas, Richardson, TX 75080 USA
[4] Transilvania University, 500036 Brasov, Romania

Corresponding author: Alexandru Paler (e-mail:alexandrupaler@gmail.com)

The work of Alexandru Paler was supported in part by a Fulbright Fellowship and in part by Google Faculty Research awards.

ABSTRACT We provide evidence that commonly held intuitions when designing quantum circuits can be misleading. In particular, we show that 1) reducing the T-count can increase the total depth; 2) it may be beneficial to trade controlled NOTs for measurements in noisy intermediate-scale quantum (NISQ) circuits; 2) measurement-based uncomputation of relative phase Toffoli ancillae can make up to 30% of a circuit's depth; and 4) area and volume cost metrics can misreport the resource analysis. Our findings assume that qubits are and will remain a very scarce resource. The results are applicable for both NISQ and quantum error-corrected protected circuits. Our method uses multiple ways of decomposing Toffoli gates into Clifford+T gates. We illustrate our method on addition and multiplication circuits using ripple-carry. As a byproduct result, we show systematically that for a practically significant range of circuit widths, ripple-carry addition circuits are more resource-efficient than the carry-lookahead addition ones. The methods and circuits were implemented in the open-source QUANTIFY software.

I. INTRODUCTION

For the foreseeable future, quantum computing will be performed in very resource-restricted environments, where the number of qubits (e.g., hardware) is the biggest constraint. Practical problems (e.g., quantum chemistry) are solved by executing a quantum circuit. The goal is to use the smallest possible amount of hardware for executing a computation, error-corrected or not. When assuming only circuit widths, the decisions are straightforward, but circuit depth has to be considered too. A circuit's depth is indicative for the execution time of the computation and its width is the number of qubits required.

Recent works concerned with resource estimations of fault-tolerant computations assumed that Clifford operations have negligible cost and that the runtime of a quantum computer is dominated by the cost of executing non-Clifford gates [2], [23]. However, this is not necessarily a realistic assumption. In asymptotic worst-case estimations, constant factors are insignificant. Nonetheless, as we will show in this article, the depth could be underestimated by up to 1/3. Such ratios impact, for example, the distance of the code required to protect the quantum error-corrected (QECC) version of the computation.

It is assumed that the Toffoli+H gate set is at a higher level than the Clifford+T one. The Clifford+T to Toffoli+H compilation is not being realized in the literature yet. This work focuses on the optimization potential when translating circuits from the Toffoli+H to the Clifford+T gate set. This kind of translation has, for the moment, a very high classical cost because very large circuits take a lot of time and energy to be compiled and optimized [19]. The Clifford+T gate set is very often used for preparing QECC circuits. To this end, the Clifford+T form of Toffoli gates has received considerable attention with respect to QECCs.

When departing from the asymptotic method, how should realistic worst-case resource estimations be performed? We argue that significant optimizations can be achieved by making appropriate Toffoli gate decomposition choices: We improved the resource analysis of the state-of-the-art arithmetic circuits from [13] and [16]. We argue that there is more potential in carefully choosing the Toffoli gate decomposition when automatically compiling circuits. We find out the following.

- T-count optimization can be detrimental to a circuit's depth: Reducing T-count can increase the overall depth.

- Due to low connectivity and complexity of noisy intermediate-scale quantum (NISQ) circuit compilation, it may be useful to replace circuit controlled NOTs (CNOTs) with measurements;
- Ripple-carry arithmetic is more resource-efficient than carry-lookahead for particular width ranges (cf. ripple-carry-based multiplier has the width of a carry-lookahead adder).

This work is structured as follows. Section II introduces the circuit types analyzed in this work, as well as the Toffoli gate decompositions. Section III describes the methodology of how we optimize resources of the arithmetic circuits. Section IV presents the contributions. We conclude by formulating future work related to the automatic optimization of large-scale quantum circuits.

II. BACKGROUND

Quadratic speedups seem to be not sufficient for achieving a quantum computing advantage. This observation was first made by Draper *et al.* [4] and then extended, for example, by [23]. The optimization of arithmetic circuits for applications with an exponential speedup becomes even more important. Due to their logarithmic depth, carry-lookahead adders started receiving increased attention. Recent examples are [16], [28], which consider the adaptation of the circuits to both NISQ and surface QECC. The cost of a circuit is generally considered being determined either by the number of T gates (T-count, for QECC protected circuits, [4]), or the number of CNOT gates (CNOT-count, for NISQ circuits [14]).

A. GATE PARALLELISM

We make the following realistic assumptions. First, gate parallelism is possible, but T state distillations are sequential in time (one at a time). *There is a clear distinction between magic state distillation and T gate application*, and the readers may refer to Section A in the Appendix. In this article, we maintain gate parallelism, but assume distillations are sequential. This is because parallelizing distillations is truly a luxury with respect to the available hardware: One will choose to operate additional logical qubits instead of executing more distillations in parallel. This does not mean that T gates cannot be executed in parallel: T states may be distilled and stored in a queue when T gates are not used [18].

Second, single-control-multiple-target CNOT gates have depth 1. This kind of CNOT parallelism is not necessarily always possible with all hardware platforms. However, the lack of CNOT parallelism is one of the least problems, because of the restricted connectivity: Not all NISQ machines support all-to-all connectivity between qubits. Connectivity plays a significant role in the success of executing a quantum circuit: The more the better. In general, all-to-all (logical) qubit connectivity exists in QECC-protected circuits. There are hardware proposals where connectivity is better than 2D nearest neighbor, as available in superconducting circuits [9], [31].

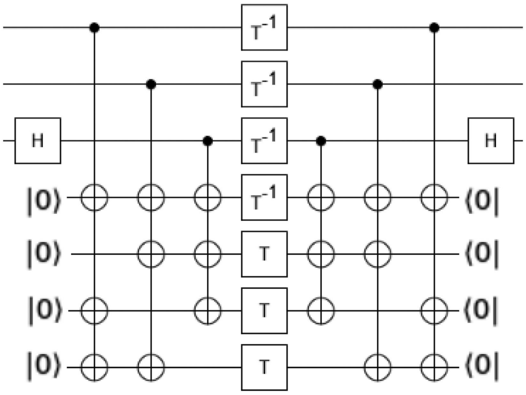

FIGURE 1. Four ancillae T-depth 1 (4AT1) Toffoli gate decomposition. The upper three wires are for the Toffoli gate.

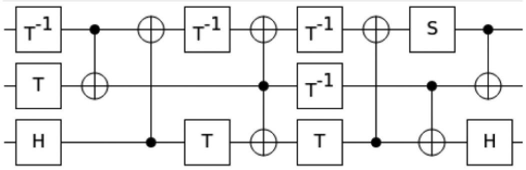

FIGURE 2. Zero ancilla T-depth 3 (0AT3) Toffoli decomposition [24]. This circuit has a depth of 9 compared to depth 10 presented in [13].

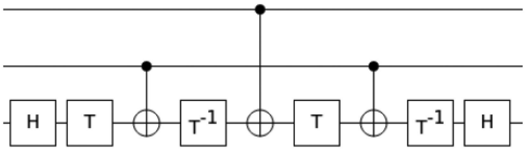

FIGURE 3. RT3. A relative phase Toffoli decomposition with three CNOT gates. Running this circuit in reverse is called IRT3, and is the same as RT3.

B. TOFFOLI GATE DECOMPOSITIONS

The literature includes two types of Toffoli gate decompositions: 1) the exact ones having 7 T gates and a various number of ancillae; 2) the relative phase decompositions using 4 T gates and various numbers of CNOTs and ancillae. In the latter case, the number of CNOT gates in the decomposition influences the implemented relative phase, and there cannot be less than three CNOTs [25]. The relative phase Toffoli gate is also known as the Margolus gate, or the simplified Toffoli gate. It has been presented in different formulations, for example, by [5], [7], [11], and [24]. The work of [11] mentions that there is a relation between T-count and CNOT-count in the Toffoli gate decompositions and conjectures that the optimization of quantum circuits could benefit from using it. The standard Toffoli (ST) gate decomposition is the one from [15] (see Fig. 12 in the Appendix).

The inverse of the relative phase Toffoli gate has been implemented in two manners. The first is by running in reverse the Clifford+T decomposition of the gate. The second

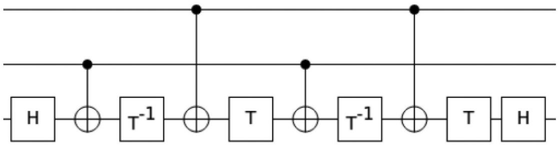

FIGURE 4. RT4. A relative phase Toffoli decomposition with four CNOT gates. Running this circuit in reverse is called IRT4.

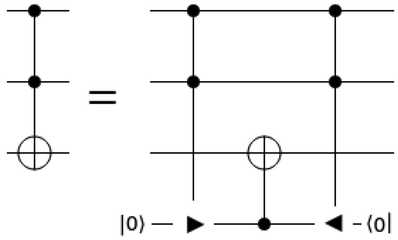

FIGURE 5. Simple method for replacing exact Toffoli gates with relative phase Toffoli gates. The double-controlled gates with a triangle are relative phase Toffoli gates. These come in pairs, and the second gate is uncomputing the ancilla initialized in |0⟩. We call this circuit *optimize depth beneficial* (ODB) when the first gate is replaced with one from Table 1 and the last gate is replaced with a measurement-based uncomputation. When both gates are replaced with one of the decompositions from Table 1, the notation is, for example, *ST/ST*. This scheme is also known as the Bennett trick.

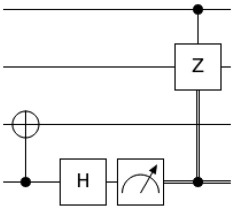

FIGURE 6. Measurement-based uncomputation for relative phase Toffoli gates. This circuit can be used to replace the CNOT and the second relative phase gate from Fig. 5.

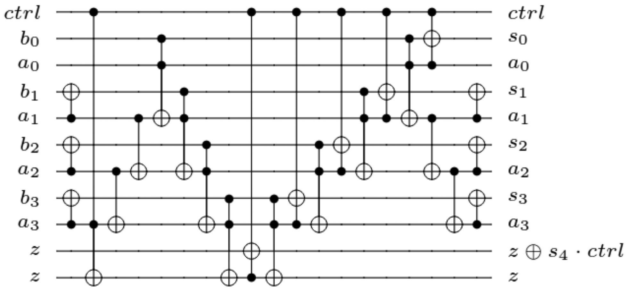

FIGURE 7. Four qubit-controlled adder according to [13].

implementation is a measurement-based circuit applied when the target of the relative phase gate is treated like an ancilla to reset the ancilla and to correct the wrong phase on the control wires. In the Appendix, we show that the same uncomputation circuit can be used for multiple types of relative phase Toffoli gates.

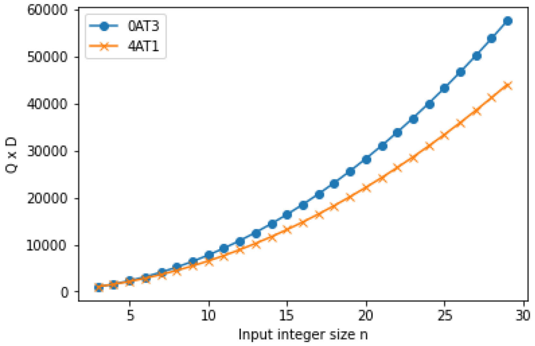

FIGURE 8. KQ (depth × width) as a function *n* for the control-adder when decomposed. OAT3 and 4AT1 refer to the zero ancilla T-depth three- and four-ancillae T-depth one Toffoli decompositions, respectively.

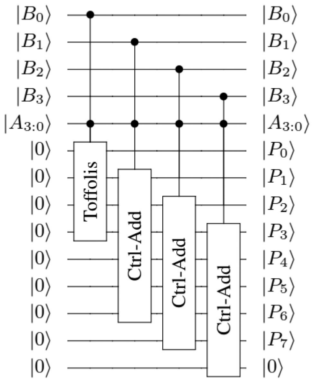

FIGURE 9. Four qubit multiplier according to [13].

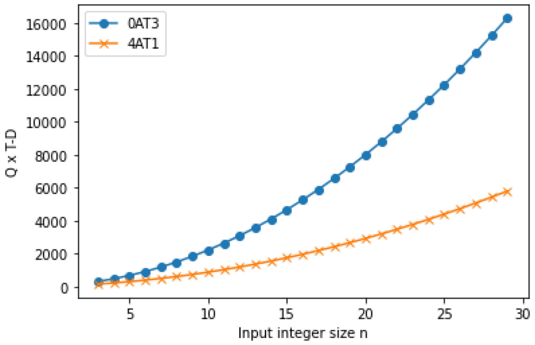

FIGURE 10. KQ$_T$ (T-depth × width) as a function *n* for the control-adder when decomposed. OAT3 and 4AT1 refer to the zero ancilla T-depth three- and four-ancillae T-depth one Toffoli decompositions, respectively.

C. QUANTUM ARITHMETIC CIRCUITS

Quantum addition circuits can be classified [22] at least into 1) ripple-carry [30] and 2) carry-lookahead [3]. The first have a smaller width but are deeper, whereas the latter are wider and shallower. Carry-lookahead adders are very often used in classical computers, and have a logarithmic depth at the expense of introducing more ancillae: $\mathcal{O}(4n)$ width. Although carry-lookahead seems more expensive than a ripple-carry

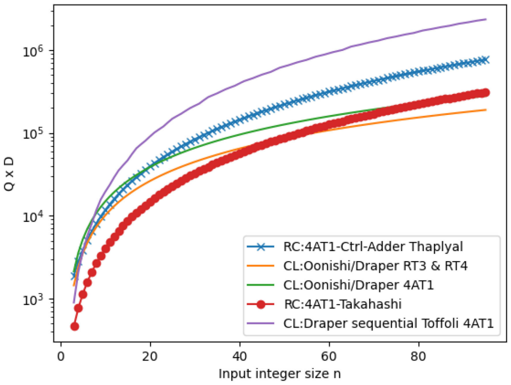

FIGURE 11. Comparison between ripple-carry (RC) and carry lookahead (CL) using the KQ cost metric that is the product between depth and width. Blue and orange lines with markers are RC. It can be observed that RC is more efficient than CL: (a) up to 50 qubits when CL uses relative phase Toffoli gate decompositions (green); (b) 96 qubits when CL uses exact Toffoli decompositions (red). The absolute unrealistic worst case (magenta) is when CL is decomposed with 4AT1 such that T gates are sequential due to the low distillation rate of T states. This effectively forces all Toffoli gates to be sequential and the logarithmic depth of the adder is lost.

TABLE 1. Costs of the Different Toffoli Gate Decompositions. For 0AT3, we consider with respect to the arithmetic circuits the depth of the circuit from [13], although 0AT3 can have depth 9. RT3 and RT4 do not include the uncomputation (depth 4). ST is the standard Toffoli decomposition from [15], and AND is the relative phase Toffoli gate from [5].

$Tof.Dec.$	$Depth$	$CNOT_c$	T_d	T_c
4AT1 (Fig. 1)	7	16	1	7
0AT3 (Fig. 2)	10 (9)	7	3	7
RT3 (Fig. 3)	9	3	4	4
RT4 (Fig. 4)	10	4	4	4
ST (Fig. 12)	13	6	6	7
AND	9	6	2	4

circuit, there has been so far no exhaustive quantitative analysis between the two adder approaches in the literature. In this article, we perform such an analysis.

Our analysis considers the ripple-carry [13] and the carry-lookahead [16], [28] adders. The first introduced a controlled-adder (ripple-carry) that has a total width of $2n + 3$ qubits and a depth of $5n - 1$. The second optimized the carry-lookahead circuit from [3] by replacing exact Toffoli gates with relative phase Toffoli gates similar to how this was realized by [28].

III. METHODS

We replace Toffoli gates with exact (Figs. 1 and 2) or relative phase (Figs. 3 and 4) Clifford+T decompositions. We either replace single or pairs of Toffoli gates (e.g., Fig. 5). We will call *optimize depth beneficial* (ODB) the replacement circuit when the first relative phase Toffoli is replaced with one of the decompositions from Table 1 and the second with the measurement-based uncomputation.

The replacement method from Fig. 5 was used, for example, in [16] and [28]. The replacement 1) guarantees the correctness of the resulting circuit without laying it out and verifying it; 2) introduces, however, an ancilla which controls a CNOT on the initial target of the Toffoli. This is effectively the method used by [5] as well: Two Toffoli gates which 1) share the control wires and 2) their target qubit is used during its entire lifetime only as control by further operations, can be simplified to a relative phase gate and measurement-based uncomputation.

Our method is based on the investigation on the adder and multiplier circuits, and choosing the best way to compile the circuit based on its overall depth, T-depth, and width (e.g., number of qubits). We use QUANTIFY [17] to count exactly the gates and determine the depth of the circuits. The work of [16] analyzes the role of Toffoli gate decompositions for improving carry-lookahead adder circuits. Such circuits have logarithmic depth but their width is almost double compared to the ripple-carry ones. It is interesting to analyze the trade-off between these two types of arithmetic circuits and to determine which one is most compatible with computers where quantum hardware is definitely the limiting factor to scalability.

The work of Gidney [6] appeared in parallel and independent to our efforts and analysis. The author noticed the comparison potential opened by the method from [16] and performed a quantum resource analysis of the space-time volume of surface code-protected quantum circuits. Compared to [6], we offer an exhaustive and systematic comparison of the resources required by the adders, and do not focus directly on surface code volumes because of the reasons discussed in Section IV-E.

IV. RESULTS

T-counts are usually reduced because of the requirements of the QECCs. For NISQ purposes, the T gate does not play any special role in NISQ and is treated on the same footing as any other single qubit gate. The CNOT-count is more important because of the high error rates associated with two qubit gates. We will show that relative phase Toffoli gates can be used for reducing the CNOT-count, too.

This section presents the results of optimizing the controlled-adder and multiplier from [13] using different decompositions [24]. We have chosen these circuits because of their modularity, low resources, and representative ripple-carry structure. For the circuits, we derive formulas to express the depth D, the T-depth T_d, and the number of qubits (e.g., wires) Qub. The expressions are used afterward for the trade-off analysis of using different Toffoli gate decompositions.

A. REDUCING T-COUNT CAN INCREASE DEPTH

A circuit's depth increases whenever the gate parallelism is lost, for example, due to suboptimal baseline decompositions. In other words, although it looks like the T-count and the depth have been reduced, only the T-count is reduced but the depth is actually increased.

This is the case for [16] where the authors have chosen ST with a depth of 13. They replaced ST/ST decompositions with RT3/IRT3 and RT4/IRT4 for optimizing their circuits.

There are two alternative replacements which would have generated different results. First, if the authors would have used 0AT3 (depth 9) instead of ST, their depth optimizations would have been minimal. In particular, the total depth of RT3/IRT3 equals to the one of 0AT3/0AT3, but RT4/IRT4 (total 20) increases actually the depth by two for each pair of replaced Toffoli gates (total 18 with 0AT3/0AT3). Without further gate-level circuit optimizations, the circuit has a 10% increase in depth. The second scenario is when T-count reductions are generating an increased T-depth. If in [16] they would have used 4AT1 as baseline, the T-depth would have actually increased for Toffoli gate pairs (from 2 to 8), and the total depth would have increased, as well.

The third example is a particularly inefficient replacement when ODB is used for a single Toffoli (instead of pair), and taking 0AT3 as baseline. The T-count is reduced but the total depth increases from 9 (0AT3) to $12 = 9 + 1$ (CNOT) + 2 [Hadamard and controlled-Z gate (CZ)].

The third example is of practical importance, because it shows that whenever ODB is used for circuit optimization, at least 30% of the total depth may be generated by measurements and corrective CZ gates (the uncomputation circuit from Fig. 6 represents 1/3 of the total depth of ODB). In case the highly parallel 4AT1 would have been used, the resulting ratio between measurements and depth would be significantly higher and close to 1/2.

B. TRADING CNOT FOR MEASUREMENTS IS BENEFICIAL

The ODB scheme is not considered being NISQ compatible, because physical measurement gates have high error rates. We show that, as long NISQ circuit compilers have the efficiencies observed, for example, in [27], replacing CNOTs with measurements can be beneficial for depth and total circuit error rate.

We assume that measurements have a *40 times* higher probability of failure compared to CNOTs (assuming single qubit gates with errors about 0.1%, two qubit gates ten times higher at about 1%, such that a worst case is having measurements almost random at 40% error rate). This may be a very pessimistic overestimation of the error rates for some NISQ architectures such as [31]. Furthermore, we assume that CNOTs have an overhead: NISQ chips have a reduced connectivity and these gates have to be routed/compiled to the underlying hardware. Next, we show that it could be a good idea to replace at least 40 physical CNOTs with a measurement.

Connectivity of the hardware plays an important role. Toffoli gate decompositions seem to be close to compatibility with 2-D nearest neighbor interactions. However, for the example in Figs. 7 and 9, the compilation of the very long range Toffoli gates will be expected to significantly increase the total depth. Moreover, the CNOT gate set is not native for ion traps and has to be compiled to MS gates [12]. Although the translation between CNOT and MS is direct, the optimization of the circuit gate counts and depths is not.

We consider a best case CNOT overhead of five physical CNOTs. This is to say that a circuit's CNOT is compiled to five CNOTs on the NISQ device. The CNOT overhead value was estimated after calculating the characteristic path length (CPL, cf. Appendix) for the graphs of the most common NISQ devices. The CPL for Sycamore is 5, and Hummingbird has a CPL of almost 8. According to [27], circuits which are structurally similar to Toffoli circuits (called TFL circuits in [27]) get compiled with an increased depth by a factor between 5 and 20 depending on the used compiler. Thus, we consider the pessimistic and optimistic cases for the failure rate of measurements and CNOT overhead, respectively.

Whenever *pairs of Toffoli gates* can be replaced, one of the Toffoli gates is replaced with a measurement-based uncomputation, and this is effectively the ODB scheme from Fig. 5. If the pair is 0AT3/0AT3, and counting the middle CNOT too, there are 15 CNOTs in total. If the ST decomposition would have been used, the total would have been 13 CNOTs. Using the ODB circuit with RT3 reduces the number of CNOTs to 5, because we assume the worst case that CZs are always applied. Thus, the ODB circuit has cut by 10 the CNOT-count per pair of Toffoli gate pair (eight in case of ST).

Reducing 8–10 circuit CNOTs means that about 40–50 physical CNOTs are replaced with a measurement gate ($8 \times 5 = 40$). If the measurement error rate is lower, it may be possible to directly replace any Toffoli with a relative phase one. As a result, there may be situations where the ODB scheme is compatible with NISQ circuits. This could be the case for topologies with low values of CPL.

C. LOWER DEPTH CONTROLLED-ADDER

In the previous section, if we would have used 4AT1 (16 CNOTs) instead of 0AT3 (7 CNOTs), the CNOT-count optimization would have been even more dramatic. One should not consider 4AT1 for arbitrary circuits without making sure that it is a realistic worst case: Comparing ODB against 4AT1 would skew the magnitude of the optimization. When designing circuits and estimating the worst case, the estimations should not be too pessimistic, but realistic. We will show that this was not the case for the ripple-carry arithmetic circuits from [13].

We assume that the number of wires (e.g., qubits/width) cannot be reduced in a ripple-carry adder, and the next optimization goal is the depth. All the Toffoli gates in the adder are sequential and not parallel. This property favors Clifford+T circuits with shorter depths even at the cost of additional ancillae. Because the Toffoli gates are sequential, the ancillae can be reused without losing any Toffoli gate parallelism (there is no parallelism anyway). The original work of [13] used the 0AT3 decomposition, but we propose to use the 4AT1 decomposition. The extra four ancillae can be reused in favor of a shorter depth. Further optimizations may be possible by using ODB like in [5], [11] and [16].

For n-bit integers, the adder has: 1) $3n + 2$ sequential Toffoli gates; 2) $n - 1$ parallel CNOT gates at the beginning of the circuit which contribute to the depth by 1 only, 3) $n-2$ sequential CNOT gates in the first half of the circuit.

As a result, the sequence of CNOT gates has an overall depth of $n-1$. At the end of the circuit, these CNOT gates are used again to reset the qubit of the first input integer to its original value and one of the CNOTs is parallel with the last Toffoli gate. Hence, the total depth of the CNOT gates is $2(n-1)-1$ and the depth of the adder circuit is the sum of the depth of the Toffoli gates and the CNOT gates.

Since the Toffoli gates are all sequential, the T-depth equals the T-depth of the used Toffoli decomposition multiplied by the number of Toffolis, which is $3n+2$. Concerning the width, a constant number of ancillae will be added to the original width of $2n+3$, namely the number of ancillae in the used Toffoli decomposition. The general depth, T-depth, and total number of qubits formulas are the following:

$$D_{\text{add}} = (3n+2)D_t + 2n - 3 \tag{1}$$

$$T_{\text{add}} = (3n+2)T_d \tag{2}$$

$$Qub_{\text{add}} = 2n+3+A \tag{3}$$

$$CNOT_{\text{count}} = 2(2n+3) + C(3n+2) \tag{4}$$

where D_t, T_d, A, and C are the depth, T-depth, additional ancillae, and the CNOT-count of the chosen Toffoli decomposition, respectively. After replacing C with the values from the $CNOT_c$ column of Table 1, we obtain the following:

$$CNOT_{4\text{AT}1} = 52n + 26 \tag{5}$$

$$CNOT_{0\text{AT}3} = 25n + 8. \tag{6}$$

Our choice of the 4AT1 decomposition performs better than 0AT3, because it reduces the circuit depth by approximately 30% and the T-depth by 66.6% at the cost of four additional qubits only (cf. the ratio between the depth and T-count of the decompositions in Table 1).

D. MULTIPLIER USING HYBRID DECOMPOSITIONS

The multiplier from [13] is built using the controlled-adder from Section IV-C. The multiplier includes 1) a sequence of n Toffoli gates; and 2) a succession of $n-1$ controlled-adders. There are $n-1$ adders, and each adder has $3n+2$ Toffoli and $2n-3$ CNOTs. They contribute to the depth of the multiplier by $n+(3n+2)(n-1)$ and $(2n-3)(n-1)$. For the general case, we have

$$D_{\text{mult}} = (3n^2-2)D_t + (n-1)(2n-3) \tag{7}$$

$$T_{\text{mult}} = (3n^2-n-2)T_d \tag{8}$$

$$Qub_{\text{mult}} = 4n+1+A \tag{9}$$

where D_t, T_d, and A have the same meaning like in Section IV-C. Note that the sequence of n Toffoli gates at the beginning is considered not to be parallel in (7). This is because, it cannot be ensured that after the decomposition, those still remain parallel. Hence, the choice of the correct decomposition plays an essential role in the optimization, as we will show it next.

The structure of the multiplication circuit allows us to consider two distinct Toffoli gate decompositions, one type for each region. We will use the formulas from Section IV-C to determine the costs of the multiplier. We decompose the first set of parallel n Toffoli gates using the 0AT3 decomposition. We maintain the parallelism of these gates without introducing ancillae. If we use a Toffoli decomposition with ancillae, we then have to introduce a linear number of ancillae to maintain the parallelism. Otherwise, we introduce a constant number of ancillae but increase the depth to linear. The 0AT3 is an optimal choice in this case since we maintain the parallelism (a constant depth of 10) and don't introduce any ancilla. The second part of the circuit consisting of $n-1$ controlled-adder is decomposed using 4AT1, similar to how it was performed in Section IV-C.

Due to the parallel decomposition of the first n Toffoli gates, the corresponding T-depth is constant and equals 3. The T-depth of the rest of the circuit is equal to the product between the number of controlled-adders, $n-1$, and the T-depth of the 4AT1. Lastly, we add four ancillae to the original width of the multiplier when undecomposed. One can observe that ripple-carry multiplier has the same width as a carry-lookahead adder, namely $\mathcal{O}(4n)$ [13], [16] (cf. (12) for multiplier width).

$$D_{\text{mult}} = 10 + 7(3n^2-n-2) + (2n-3)(n-1) \tag{10}$$

$$T_{\text{mult}} = 3 + (3n^2-n-2) \tag{11}$$

$$Q = 4n+1+4. \tag{12}$$

E. AREA AND SPACE-TIME VOLUME VERSUS WORST CASE

The worst case space-time volume of large computations is not trivial to estimate correctly. This holds even when distillations are sequentialized like in [18]. The best option is to compile the space-time volumes using [20] and [21], schedule the distillation procedures [18] and then check the resulting worst case depth. However, those compilers take the circuit-level description as input, such that worst-case volume estimations are as good as the worst-case circuit-level estimations. Furthermore, there are different tricks that can be applied to worsen or improve the volumes or other volume-related costs, such that space-time volume estimations may be misleading. We illustrate this with a simple example, in the following.

We analyze the applicability of the KQ metric [16]: *the product of the number of qubits and the depth of the circuit*. There are variations of the metric such as KQ_{CX} for the CNOT-depth, and KQ_T for the T-depth.

We are interested in the relevance of KQ_T. While comparing Figs. 8 with 10, we notice the drastic improvements (blue vs. orange—our choice) when comparing KQ_T with the generic KQ (the distance between blue and orange lines is not drastic). The KQ_T metric does not necessarily reflect the amount of improvements or degradation of the adopted decomposition methods.

The same effect will be obtained when the worst-case space-time volume of surface code computations is optimized when considering the T-count as an approximation of the depth after using ODB. Clifford gates and measurements have to be accounted when estimating the area and space-time volumes of a circuit. Otherwise, optimization results are misinterpreted.

F. RIPPLE-CARRY VERSUS CARRY-LOOKAHEAD

Considering the fact that distillations are sequential, and that qubits are very scarce, ripple-carry is with respect to hardware more efficient than carry-lookahead. Nevertheless, despite the fact that the carry-lookahead has logarithmic depth, there has to be a width range for which ripple-carry is with respect to the KQ metric also efficient. We use the following scenarios to compare the two different adders.

1) RC: 4AT1 (Thaplyal Ctrl-Adder) is the controlled-adder discussed in Section IV-C from [13] which we decompose using 4AT1.
2) RC: 4AT1 (Takahashi) from [26] is the adder used in the construction of the controlled-adder, and we decompose it with 4AT1 too.
3) CL: RT3 and RT4 (Oonishi/Draper) is the carry-lookahead from [3] adder decomposed with the relative phase decompositions like in [16].
4) CL: 4AT1 (Oonishi/Draper) is the original adder from [3] which we penalize with four ancillae per Toffoli (4AT1) in order to minimize the T-depth.
5) CL: 4AT1 (Draper with all Toffoli gates sequence) is the absolute unrealistic worst case when taking the 4AT1 adder and executing it in a very resource-restricted environment where a single distillation can be executed at a time—this is to show that circuit designs have to be adapted to the environment.

Fig. 11 illustrates the obtained results of our analysis, where the X-axis denotes the size of the integer and the Y-axis presents the KQ (width times depth) metric. Among the five different scenarios, “CL:4AT1” of Draper is not efficient with respect to KQ metric for more than eight qubits. Practically, as expected, carry-lookahead does not seem viable for less than eight qubits.

In realistic worse-case scenarios, where hardware is scarce and distillations can be performed only sequentially, ripple-carry addition (“RC:4AT1” of Takahashi) is more efficient for up to approximately 50 qubits.

The carry-lookahead adder has on the order of $10n$ Toffoli gates [3] which are executed to a high level of parallelism. It is very unfortunate if the circuit is compiled in such a way that distillations need to be sequentialized. This is the case for RT3 and RT4. The extreme situation is when all T-gate parallelism is lost because of sequential distillations.

Regarding the QECC cost of the adders, one should consider that connecting distillations to the main computation [8], [10], [21] uses also hardware. Another aspect is that the volume of the distillation procedures could be further lowered, or in the extreme case may be even embedded into empty regions of the main computation space-time volume. Moreover, distillation space-time volume costs are a function of the total space-time, but which is difficult to estimate correctly (see Section IV-E). Therefore, the plot from Fig. 11 should be seen as a recommendation.

V. CONCLUSION

The constants in asymptotic worst-case estimations play a role when computing a circuit’s execution time or failure rate. We have showed that relative phase Toffoli gate decompositions are optimal in specific contexts. Inappropriate usage of optimizations may result in worsening other costs in unexpected ways. More precisely, reducing T-depth increases the depth of the circuit and we exemplified this using the carry-lookahead adder from [16]. Compiling CNOTs to NISQ architectures can be very costly and we showed this through a simple analysis of chip topologies and worst-case measurement error rates. We showed that when using measurement-based uncomputations, at least one-third of the circuit’s depth is occupied by classical processing.

Our argument is based on a structural analysis of the arithmetic circuits and selecting the appropriate Toffoli decomposition. We collected some of the most used Toffoli gate decompositions and described these in terms of gate depths and counts. In the Appendix, we show that practically for compilation purposes, there is actually a continuum of Toffoli gate decompositions that can be used. Subcircuits, such as Toffoli gate decompositions, should be seen as having a maleable structure that can be adapted for optimization purposes. On the one hand, the presented results may seem obvious if one makes reasonable assumptions and builds correct worst cases. On the other hand, the results are not that obvious—we were able to improve state-of-the-art arithmetic circuits such as the ones from [13] and [16]. This is not to say that those circuits are inefficient, but to argue that there is more potential in carefully choosing the Toffoli gate decomposition when automatically compiling circuits.

Future work will include a focus on investigating how the techniques from [29], where the AND uncompute was replaced with its reverse circuit instead of the measurements-based circuit, influence the optimality of the compiled space-time volume. We assume that the NISQ version of the circuits from [28] will benefit from measurements-uncompute instead of the reverse AND.

In this article, we lowered the depth of a controlled-adder by replacing the Toffoli gate decomposition. Using the controlled-adder, we showed that area and volume cost metrics can be sometimes misleading. We went one step further and reduced the resources needed for multiplication by using two types of Toffoli gate decompositions. Finally, we illustrated that for up to 50-bit numbers, ripple-carry adders are more resource-efficient than carry-lookahead.

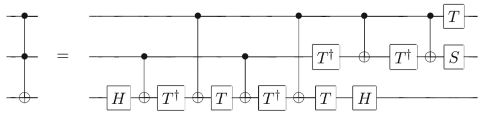

FIGURE 12. Standard Toffoli gate decomposition from [15]. The depth is 13, but considering CNOT parallelism and by commuting some of the T gates, the depth can be reduced to 11. It can be seen that this is the RT4 gate decomposition, followed by an implementation of a controlled-S gate that uses three T gates. The last S gate can be removed if the phase of the previous controlled-S is adapted.

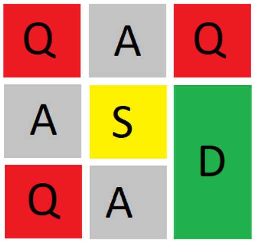

FIGURE 13. Example of a logical qubit layout where computations protected by the surface code implemented by lattice surgery are implemented. Each patch is abstracting a set of physical qubits. The red patches marked with Q are used to store logical (computational) qubits, the grey A patches are for ancillae, the yellow S patch is where distilled states are stored. In this layout a single state can be stored, and more could be stored if the size of the yellow region would span a multiple of single patches. The green D region spans multiple patches and is where states are distilled before being stored in the S region.

APPENDIX

A. DISTILLATION VERSUS GATE APPLICATION

Distillations are conceptually not the same as the application of the gate that uses the distilled state. This is a common misconception, but the reality of compilation is that distillations are performed separately from the gate applications.

Sequential distillations do not imply that the corresponding gates are sequential. From the perspective of error-corrected circuit layouts, such as the one from Fig. 13 which is influenced by the ones presented in [10], distillations (sequentialized or not) are performed in one or multiple regions specialized for this particular task. Each time a state has been successfully distilled, it is either being stored in a queue or used immediately. Gate parallelism is achieved if the average distillation rate is higher than the average consumption rate—this is most of the times possible, because T gates are not uniformly distributed along the timeline of the circuit. There are bursts of T gates being executed (e.g., seven per Toffoli in some cases), but also regions dominated by Clifford gates and measurements (cf. Section IV-B where the unerror-corrected case was discussed, but the starting observation is valid for surface codes, too). Moreover, the size of the S region can be adapted to buffer the maximum number of necessary distilled states.

B. NISQ CONNECTIVITY ANALYSIS

We consider the most relevant hardware topologies proposed and realized in practice. We take into account graphs having different number of nodes (e.g., qubits). For instance, 20 nodes for the case of a regular grid and IBM's Tokyo each having 4 rows and 5 columns, 54 nodes for the case of IBM's Rochester and Google's Sycamore, and 64 nodes for IBM's Hummingbird structure.

To provide an evidence on the most appropriate topology in practice regarding the abovementioned structures, we adopt the CPL metric. Such a metric is used in the literature to assess qualitatively the characteristics of network topologies. CPL indicates the average shortest distance (lowest value of 1 and largest value of n) between any pair of nodes in the network. Briefly, it is calculated by finding for each node of the network 1) the shortest path to all other nodes, and using this information to 2) calculate the average of the shortest paths of the corresponding node to all other nodes. Then, the average of the shortest paths of each node is summed up to calculate the overall average shortest distance of the whole network.

Among the two structures with 20 nodes, IBM's Tokyo (thanks to the additional connectivity) has a CPL value of 2.25 against 3 for the regular grid. Regarding the two topologies with 54 nodes, Sycamore has the edge over Rochester with a CPL of 4.98 and 7.39 for the former and latter, respectively. Finally, Hummingbird having a similar structure as the one of Rochester, however with 10 nodes more, has a CPL of 7.89.

To justify our belief that Tokyo's structure has more connectivity than the others, we adopted the second metric of clustering coefficient (CC) from the literature. Such a metric has a value between 0 and 1 and is used to denote the probability that the neighborhoods of each node in the network are connected to each other. For the abovementioned five structures, we found out that Tokyo has a CC of 0.47 (i.e., 50% of the neighbors are connected with each other), whereas the others have a CC of 0.

Based on those results, we can notice that the regular grid, IBM's Tokyo, and Google's Sycamore have similar characteristics with respect to the CPL. To assess this, we configured a regular grid of 56 nodes (7 rows and 8 columns) and obtained a CPL of 5. Since we showed above that Tokyo, thanks to the additional link between nodes, has a smaller CPL value than the regular grid, this leads us to the conclusion that among the abovementioned five topologies, Tokyo has a slight edge over regular grid and Sycamore structures, and has almost the half of the CPL with respect to Rochester and Hummingbird.

C. CONTROLLED ADDER WITH RELATIVE PHASE

We report costs when making very inefficient replacements of *single* Toffoli gates with pairs of relative phase Toffoli gates. This is because it could happen that for particular NISQ architectures, it makes more sense to use seemingly inefficient decompositions in order to reduce the mapping/routing overhead per CNOT. In the following, we consider that the controlled-adder used the parallel 0AT1 decomposition.

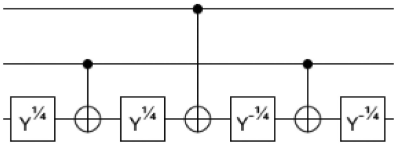

FIGURE 14. Relative phase Toffoli gate presented in [1] uses Y rotations instead of Z rotations of $\pi/4$ (T gates).

When using RT3/IRT3 in the controlled adder, each of the $3n + 2$ exact Toffoli gates costs 7 CNOT gates, 8 T-gates, and has a depth of 19. In a similar fashion, the RT4 costs 8 CNOT gates, 8 T-gates, and a depth of 21. The total number of T-gates, when using RT3 or RT4 in the adder, yields a better cost compared to the standard Toffoli decomposition.

$$T_{\text{count}} = 4 \times (3n + 2) = 12n + 8.$$

As for the number of CNOT, each Toffoli when decomposed with RT3/IRT3 and RT4/IRT4 contributes with 5 and 6, respectively. Furthermore, we have $2(2n - 3)$ CNOT gates from the original adder. The resulting CNOT counts are

$$CNOT_{\text{RT3}} = (3 + 1 + 3)(3n + 2) + 2(2n - 3) = 25n + 8$$

$$CNOT_{\text{RT4}} = (4 + 1 + 4)(3n + 2) + 2(2n - 3) = 31n + 12.$$

Compared to the 4AT1 [cf. (5)], we reduce around one half of the total number of CNOT gates when using the RT3 and RT4 instead of 4AT1.

$$\frac{CNOT_{\text{RT3}}}{CNOT_{\text{4AT1}}} = \frac{25n + 8}{52n - 26} \sim 50\%$$

$$\frac{CNOT_{\text{RT4}}}{CNOT_{\text{4AT1}}} = \frac{31n + 12}{52n - 26} \sim 60\%.$$

Even more interesting, compared to the original circuit from [13], using RT3/RT4, the CNOT count is not reduced at all. So we can make a very inefficient replacement that introduces T gates and the CNOT-count is still not changed.

$$\frac{CNOT_{\text{RT3}}}{CNOT_{\text{0AT3}}} = \frac{25n + 8}{25n + 8} \sim 100\%.$$

D. RELATIVE PHASE TOFFOLI: CIRCUIT IDENTITIES

Computations with relative phase Toffoli gates can be uncomputed in a measurement-based manner. In the following, we show that the relative phase Toffoli gate described in [5] is equivalent to 1) the one presented by [11], and 2) the original presented by [1] (i.e., Figs. 14, 16–28). The ancilla is uncomputed after the controlling a NOT gate. If the ancilla would be initialized in an arbitrary state, then for each relative-phase Toffoli gate, there would be a distinct uncomputation circuit. However, most of the times, the ancilla is initialized in $|0\rangle$, such that the same uncomputation circuit, namely the one from [5], can be used for other inverse relative phase Toffoli gate uncomputations.

The following circuit equivalence can also be shown by looking at the matrices of the relative phase Toffoli gates

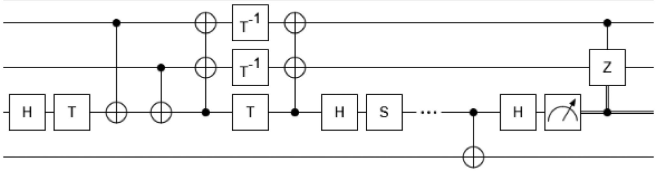

FIGURE 15. To derive the measurement pattern of other relative phase Toffoli gates, we start from the circuit proposed in [5]. The first region of the circuit that is applied to the upper three qubits implements the relative phase Toffoli gate. The CNOT between the third and fourth qubits is copying the bit to the target qubit of the Toffoli gate. Uncomputation of the ancilla starts at the rightmost H gate on the third wire.

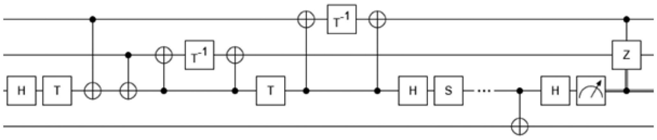

FIGURE 16. CNOTs and the T gates are moved such that parallelism is lost.

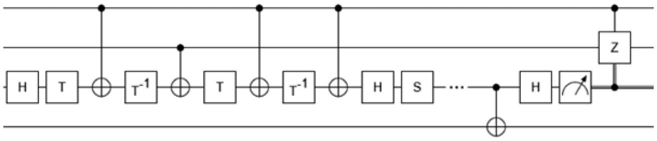

FIGURE 17. CNOTs and the T gate are flipped between the wires. This is possible due to the diagonal nature of the Z rotation gates. Afterward, two CNOTs cancel.

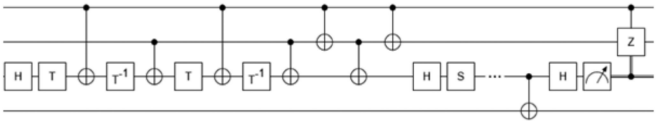

FIGURE 18. One of the CNOTs is replaced with four other CNOTs. This transformation is similar to approaches used in the linear nearest neighbor compilation of quantum circuits.

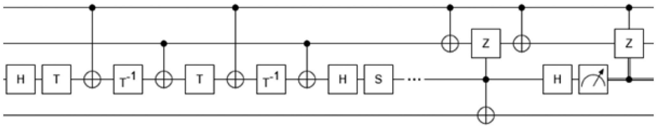

FIGURE 19. Three of the previous four CNOTs are commuted through the H and S gates. The result is that the CZ and two CNOTs can be commuted past the CNOT that copies the bit information to the Toffoli target. Consequently, the uncomputation circuit.

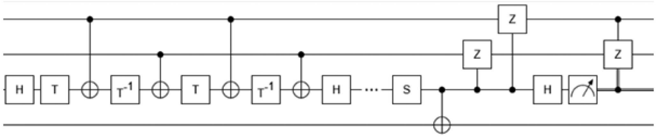

FIGURE 20. After using CNOT and H circuit identities, the result is that the third qubit controls the application of two CZ gates. However, considering that the qubit is initialized to $|0\rangle$, it can be $|1\rangle$ (with a relative phase) only iff the upper two qubits are $|1\rangle$. In this situation, whenever the two CZs are applied, the state is actually left unchanged. Therefore, the CZs can be removed.

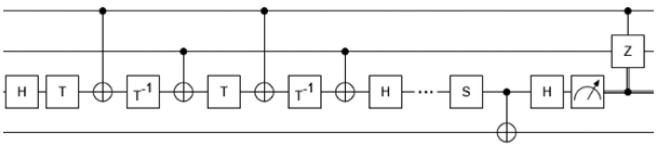

FIGURE 21. RT4 relative phase circuit with the simple uncomputation from Fig. 15.

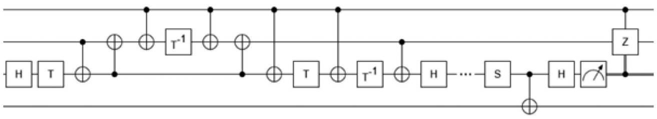

FIGURE 22. Two CNOTs are inserted before the leftmost T^{-1}; the leftmost CNOT is commuted to the right, and the CNOTs between the second and third qubits are flipped together with the T^{-1}.

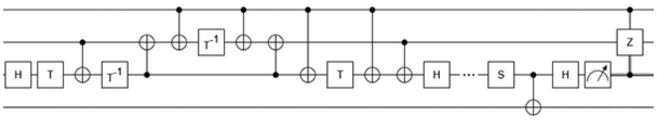

FIGURE 23. Rightmost T gates are commuted through the *long range* CNOT gates. One of the T gates is commuted to the leftmost possible position after commuting on the third wire with other CNOT controls.

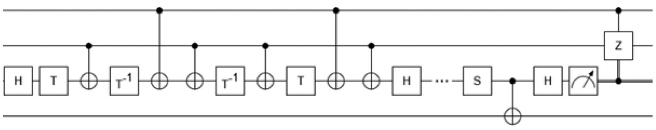

FIGURE 24. After flipping back CNOTs and the T^{-1} gate.

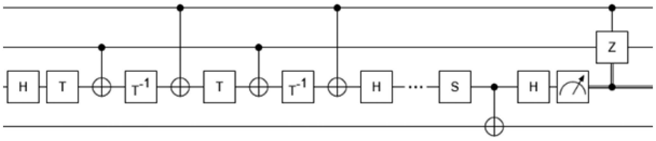

FIGURE 25. Commute the pair of T/T^{-1} with the short-range CNOTs, and cancel two CNOTs afterward.

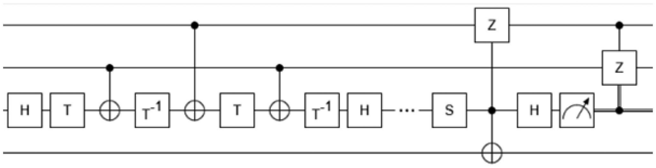

FIGURE 26. Commute a long-range CNOT through the H and S gates. The resulting CZ is controlled by the third wire.

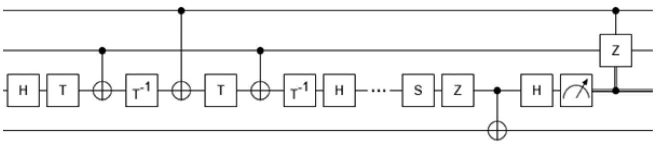

FIGURE 27. Using a similar argument to Fig. 21, the CZ can be removed and be replaced with a single Z gate.

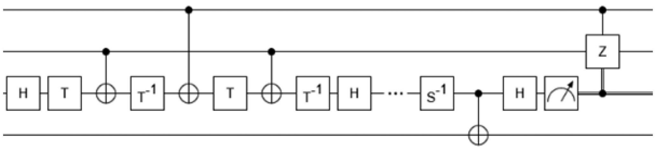

FIGURE 28. RT3, after S and Z gates are replaced with S^{-1}.

and considering that when the ancilla (third qubit) is initialized to $|0\rangle$, the resulting state vectors are equal. However, we derive the circuits, in order to highlight the potential of automatic optimization of circuits using a large dictionary of Clifford+T decompositions of relative phase Toffoli gates.

ACKNOWLEDGMENT

The authors thank Vlad Gheorghiu for suggestions about how to improve the manuscript.

REFERENCES

[1] A. Barenco *et al.*, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, 1995, Art. no. 3457, doi: 10.1103/PhysRevA.52.3457.

[2] S. Chakrabarti, R. Krishna Kumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, "A threshold for quantum advantage in derivative pricing," *Quantum*, vol. 5, p. 463, 2021, doi: 10.22331/q-2021-06-01-463.

[3] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "Alogarithmic-depth quantum carry-lookahead adder," *Quantum Inf. Comput.*, vol. 6, no. 4, pp. 351–369, 2006.

[4] V. Gheorghiu and M. Mosca, "Benchmarking the quantum cryptanalysis of symmetric, public-key and hash-based cryptographic schemes," 2019, *arXiv:1902.02332*, doi: 10.48550/arXiv.1902.02332.

[5] C. Gidney, "Halving the cost of quantum addition," *Quantum*, vol. 2, p. 74, 2018, doi: 10.22331/q-2018-06-18-74.

[6] C. Gidney, "Quantum block lookahead adders and the wait for magic states," 2020, *arXiv:2012.01624*, doi: 10.48550/arXiv.2012.01624.

[7] C. Jones, "Low-overhead constructions for the fault-tolerant Toffoli gate," *Phys. Rev. A*, vol. 87, Feb. 2013, Art. no. 022328, doi: 10.1103/PhysRevA.87.022328.

[8] Y. Lin, B. Yu, M. Li, and David Z. Pan, "Layout synthesis for topological quantum circuits with 1-D and 2-D architectures," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1574–1587, Aug. 2018, doi: 10.1109/TCAD.2017.2760511.

[9] N. M. Linke *et al.*, "Experimental comparison of two quantum computing architectures," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3305–3310, 2017, doi: 10.1073/pnas.1618020114.

[10] D. Litinski, "A game of surface codes: Large-scale quantum computing with lattice surgery," *Quantum*, vol. 3, p. 128, 2019, doi: 10.22331/q-2019-03-05-128.

[11] D. Maslov, "Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization," *Phys. Rev. A*, vol. 93, Feb. 2016, Art. no. 022311, doi: 10.1103/PhysRevA.93.022311.

[12] D. Maslov, "Basic circuit compilation techniques for an ion-trap quantum machine," *New J. Phys.*, vol. 19, no. 2, 2017, Art. no. 023035, doi: 10.1088/1367-2630/aa5e47.

[13] E. Muñoz-Coreas and H. Thapliyal, "Quantum circuit design of a T-count optimized integer multiplier," *IEEE Trans. Comput.*, vol. 68, no. 5, pp. 729–739, May 2019, doi: 10.1109/TC.2018.2882774.

[14] B. Nash, V. Gheorghiu, and M. Mosca, "Quantum circuit optimizations for NISQ architectures," *Quantum Sci. Technol.*, vol. 5, no. 2, 2020, Art. no. 025010, doi: 10.1088/2058-9565/ab79b1.

[15] A. M. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. 10th ed. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[16] K. Oonishi, T. Tanaka, S. Uno, T. Satoh, R. V. Meter, and N. Kunihiro, "Efficient construction of a control modular adder on a carry-lookahead adder using relative-phase Toffoli gates," *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 3100518, doi: 10.1109/TQE.2021.3136195.

[17] O. Oumarou, A. Paler, and R. Basmadjian, "Quantify: A framework for resource analysis and design verification of quantum circuits," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2020, pp. 126–131, doi: 10.1109/ISVLSI49217.2020.00032.

[18] A. Paler and R. Basmadjian, "Clifford gate optimisation and T gate scheduling: Using queueing models for topological assemblies," in *Proc. IEEE/ACM Int. Symp. Nanoscale Archit.*, 2019, pp. 1–5, doi: 10.1109/NANOARCH47378.2019.181305.

[19] A. Paler and R. Basmadjian, "Energy cost of quantum circuit optimisation: Predicting that optimising Shor's algorithm circuit uses 1 GWh," *ACM Trans. Quantum Comput.*, vol. 3, no. 1, pp. 1–14, 2022, doi: 10.1145/3490172.

[20] A. Paler and A. G. Fowler, "OpenSurgery for topological assemblies," in *Proc. IEEE Globecom Workshops*, 2020, pp. 1–4, doi: 10.1109/GCWkshps50303.2020.9367489.

[21] A. Paler, A. G. Fowler, and R. Wille, "Synthesis of arbitrary quantum circuits to topological assembly: Systematic, online and compact," *Sci. Rep.*, vol. 7, no. 1, pp. 1–16, 2017, doi: 10.1038/s41598-017-10657-8.

[22] R. Rines and I. Chuang, "High performance quantum modular multipliers," 2018, *arXiv:1801.01081*, doi: 10.48550/arXiv.1801.01081.

[23] R. Yuval *et al.*, "Compilation of fault-tolerant quantum heuristics for combinatorial optimization," *PRX Quantum*, vol. 1, no. 2, 2020, Art. no. 020312, doi: 10.1103/PRXQuantum.1.020312.

[24] P. Selinger, "Quantum circuits of T-depth one," *Phys. Rev. A*, vol. 87, no. 4, 2013, Art. no. 042302, doi: 10.1103/PhysRevA.87.042302.

[25] G. Song and A. Klappenecker, "Optimal realizations of simplified Toffoli gates," *Quantum Inf. Comput.*, vol. 4, no. 5, pp. 361–372, Sep. 2004.

[26] Y. Takahashi, S. Tani, and N. Kunihiro, "Quantum addition circuits and unbounded fan-out," *Quantum Inf. Comput.*, vol. 10, no. 9, pp. 872–890, Sep. 2010.

[27] B. Tan and J. Cong, "Optimality study of existing quantum computing layout synthesis tools," *IEEE Trans. Comput.*, vol. 70, no. 9, pp. 1363–1373, Sep. 2021, doi: 10.1109/TC.2020.3009140.

[28] H. Thapliyal, E. Muñoz-Coreas, and V. Khalus, "Quantum circuit designs of carry lookahead adder optimized for T-count T-depth and qubits," *Sustain. Comput., Inform. Syst.*, vol. 29, 2020, Art. no 100457, doi: 10.1016/j.suscom.2020.100457.

[29] H. Thapliyal, E. Muñoz-Coreas, and V. Khalus, "Quantum carry lookahead adders for NISQ and quantum image processing," 2021, *arXiv:2106.04758*, doi: 10.48550/arXiv.2106.04758.

[30] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Phys. Rev. A*, vol. 54, no. 1, p. 147, 1996, doi: 10.1103/PhysRevA.54.147.

[31] K. Wright *et al.*, "Benchmarking an 11-qubit quantum computer," *Nature Commun.*, vol. 10, no. 1, pp. 1–6, 2019, doi: 10.1038/s41467-019-13534-2.