

Efficient constructions for simulating Multi Controlled Quantum Gates

Stefan Balauca¹[0000-0003-1750-3707] and Andreea Arusoae²[0000-0001-6660-1288]

^{1,2}Faculty of Computer Science, Al. I. Cuza University of Iași, 700506 Iași, Romania
¹stefann.balauca@gmail.com ²andreea.arusoae@uaic.ro

Abstract. Multi Controlled Gates, with Multi Controlled Toffoli as primary example are a building block for a lot of complex quantum algorithms in the domains of discrete arithmetic, cryptography, machine learning, and image processing. However, these gates cannot be physically implemented in quantum hardware and therefore they need to be decomposed into many smaller elementary gates. In this work we analyse previously proposed circuit constructions for MCT gates and describe 6 new methods for generating MCT circuits with efficient costs, less restrictions, and improved applicability.

Keywords: Multi-Controlled Toffoli Gate · Quantum Circuit · Ancilla Qubit · Efficient Quantum Algorithms

1 Introduction

Since the first ideas of quantum computing arose in the 1980s, various mathematicians, physicists, and computer scientists have put huge amounts of work in designing both quantum hardware and software. The applications they have found so far include number theory, cryptography, quantum physics, molecular bio-chemistry, search problems, and machine learning [13, 18]. The algorithms proposed are, at least in theory, far more efficient than the classical ones [7, 15], but very few of them have been actually tested on real hardware [17].

For the moment, quantum computing faces some practical problems, most of which are related to a physical phenomenon named decoherence. This effect means that a quantum system's state can collapse after some time because of interactions with its surroundings [4,5]. In order to reduce decoherence and create feasible quantum computers, we need to cool the system down to a few milli-Kelvins, which requires very expensive equipment [20]. On the other hand, the task of mathematicians and computer scientists is to design efficient and robust algorithms that will need less qubits, quantum gates, and time in order to be executed, since the effects of decoherence scale with each of these parameters.

In this work, we have focused on a specific, yet very important and representative quantum algorithm, the Multi-Controlled Toffoli Gate (MCT). These gates are the quantum equivalent of the logical AND operation, and are necessary in many quantum arithmetic and discrete computing algorithms. [2] has

also shown that MCT gates can be used to simulate any multi-controlled gate, which in turn are widely used in increasingly complex applications (Quantum Simulations [13], Machine Learning [11], Image Processing [19]). Efficient implementations of MCT gates are therefore essential in real near-term quantum computer applications.

The paper is structured as follows: In Section 2 we introduce the most important quantum computing concepts needed for understanding the contents of this work. In Section 3 we describe various implementation ideas previously proposed by the scientific community. In Sections 4.1, 4.2, and 4.3 we expose and analyse the new constructions we propose. In Section 5 we describe the implementation of these constructions using Qiskit [9] and compare the complexities of our ideas with the ones of the existing ideas. We finish this paper by presenting our conclusions and some ideas of future improvements and research.

Notation: Throughout this work we will use the notation C^nX to represent a n -controlled Toffoli gate. In this context, $C^1X = CX$ is the basic CNOT gate and $C^2X = TOFF$ is the original Toffoli gate or 2-controlled NOT.

Note: All figures representing quantum circuits in this work have been realised using the graphical library Quantikz [10].

2 Basic Concepts

2.1 Circuit Costs

In order to understand what an efficient circuit means, we have to describe some cost functions that are usually used. We will refer to these cost functions when analysing the complexity of the circuit.

1. *Total Gate Count:* the total number of elementary gates in circuit;
2. *CX Gate Count:* the number of elementary 2-qubit gates (usually CX);
3. *Circuit Depth:* the number of time steps necessary for executing the circuit;
4. *CX Depth:* circuit depth after all the one-qubit gates have been removed;
5. *Circuit Volume:* a measure introduced by IBM [3] to better represent circuit performance, defined as: number of qubits \times circuit depth.

For the most part of the paper, our focus will be towards the CX depth of the circuit, because it is proportional to the full circuit depth, but it is easier to generalise between different hardware implementations (most of them implement the CX gate directly, or some other two-qubit gate that can be easily transformed in CX, such as CZ: $CX = H \cdot CZ \cdot H$).

2.2 Ancilla Qubits

It is well known that the most efficient solutions many classical and quantum problems make use of extra memory to achieve a speed-up. However, in the case of Quantum Computing, due to its reversible nature, this speed-up comes with a cost: any auxiliary qubit used must return to its initial state after the

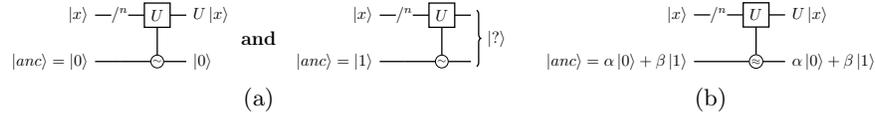


Fig. 1. Ancilla qubits notation: (a) clean, (b) dirty

computation is done, otherwise it will remain entangled to the other qubits, and therefore any operation applied to it may modify the overall state of the system.

In literature, the auxiliary qubits are called **ancilla**, and they can be of two types: *clean* ancillae and *dirty* ancillae.

Clean Ancilla: These are qubits that need to be in the state $|0\rangle$ before the computation is started and returned to this state afterwards. We will represent the usage of clean ancilla by a circle with a single tilde as in Fig. 1a in order to illustrate that only the $|0\rangle$ state is left unchanged.

Dirty Ancilla: These qubits may be in any state before the computation is started and must be returned to the exact same state afterwards. We will represent the usage of dirty ancilla by a circle with a double tilde (Fig. 1b) in order to illustrate that both fundamental states are left unchanged by the gate.

3 Existing Methods (Prior Work)

In this Section we introduce and present the evolution of quantum circuits for simulating multi-controlled gates starting from the first ideas developed simultaneously with the concept of quantum computing.

3.1 First Ideas

In the theoretical model of quantum computations, the reference work which analyses Multi Controlled Gates, with MCT as the primary example is [2]. The authors have shown how one can achieve quantum MCT gates with or without using auxiliary qubits. Some of the algorithms described there are implemented by [9] and we will briefly discuss them here.

No Ancilla: Even though Tomaso Toffoli had proven in 1980 that constructions of multi-controlled gates is impossible in the context of classical reversible computing without the use of auxiliary bits because of parity constraints [16], the ability of generating square roots of quantum gates makes this task possible in the context of quantum computing. However, the construction relies on exploring all the possible states of the control register of qubits in a Gray-code ordering and therefore needs an exponential number of gates. The cost of this construction makes it impractical for a number of control qubits greater than 5.

Since then, other papers have proposed variants of this circuit pretending to have a theoretical linear depth, while still using no ancilla [6, 14]. However, the authors admit that the 2-qubit gates used are not easy to implement on real quantum hardware and would need further decomposition into many basic gates and the overall complexity increases.

Cascading operations: The easiest and most intuitive way of implementing MCT gates is by using classical Boolean logic concepts. These ideas are firstly presented in [16], as classical circuits, and later in [2] in their quantum form. Since the implementations are based on Boolean logic, they require additional qubits to store partial calculation results. Fig. 2 presents the base construction of these circuits by using Toffoli gates in both clean and dirty ancilla setups. It is easy to see how the construction generalises to any number n of control qubits at a cost of using $n - 2$ ancilla qubits and a linear number of total gates. The depth of the circuit is proportional to the number of gates because parallelism cannot be used (operations are done in a sequential order). The shape of the resulting circuits has given this construction the name 'V-chain'.

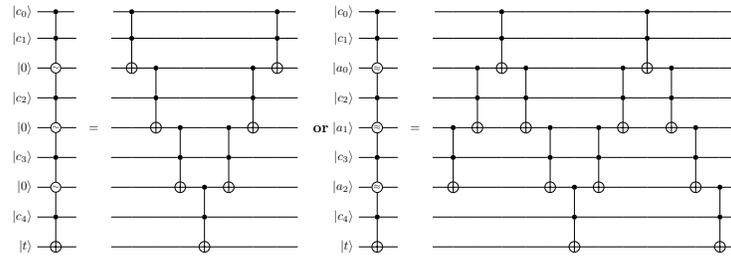


Fig. 2. Structure of the V-chain construction for the case of 5 control qubits and 3 ancilla qubits (left: using clean ancilla, right: using dirty ancilla)

A great use for the dirty ancilla V-chain construction is also presented in Lemma 7.3 from [2]: for creating an arbitrary large multi-controlled X circuit one can use a single ancilla qubit and three or four V-chain constructions with half the number of the initial controls, as illustrated in Fig. 3. The dirty ancilla qubits needed for the V-chain gates are provided by the other half of the circuit which does not participate in the current computation.

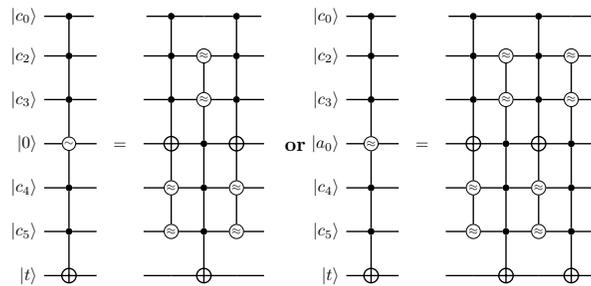


Fig. 3. One Ancilla Construction using dirty ancilla V-chain sub-circuits (left: one clean ancilla, right: one dirty ancilla)

Implemented by Qiskit: The three constructions described above are implemented in [9] with the names `noancilla`, `v-chain` and `recursive` respectively. However, the recursive implementation has a design flaw because the

sub-circuits used are also **recursive** and not **v-chain** as they should be. This makes the overall gate-cost and depth complexities of the circuit to be quadratic rather than linear as presented in [2]. Because of the widely use of Qiskit in many Quantum Computing applications and as a learning environment, lots of students or even specialists might mistakenly believe that a linear construction for multi-controlled gates is not possible by using only one ancilla qubit. In order to prove that is not the case, we have implemented, tested and measured the complexities of these constructions using **v-chain** gates for the sub-circuits (more details in Section 5).

3.2 Logarithmic Depth using Parallelism

In a more recent article [8], authors have described a construction that rearranges the gates of the clean ancilla V-chain circuit in order to achieve a logarithmic depth by making use of parallelism (see Fig. 10a for construction details).

3.3 Relative Phase and Partial Gate Cancelling

An important way of reducing the gate and depth cost of quantum circuits is the usage of gate constructions that have *almost* the same functionality as the gates they are trying to replicate in circumstances where the difference is either irrelevant or can be cancelled by another conjugated gate construction. The most common trick is to use relative phase gates: these gates are identical with their counterparts when taken in absolute value, but differ only by a relative phase applied over some of the states. In matrix form, this is equivalent to a multiplication with a diagonal matrix D whose diagonal entries are of the form $e^{i\varphi}$. We will use the prefix R to denote a relative phase gate, therefore RC^2X will represent a relative phase Toffoli gate.

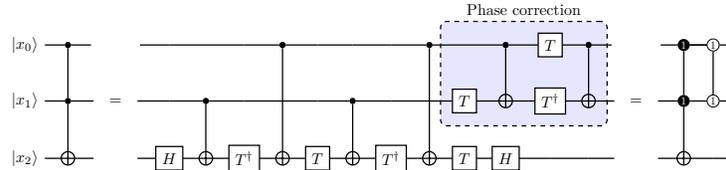


Fig. 4. Decomposition of a Toffoli gate into a Relative Phase Toffoli and a Phase Correction Gate. The circled 1s on the control qubits of the last circuit represent the relative phase. The black circles represent the relative phase induced by the first part of the circuit, and the white circles represent the corresponding phase correction.

Since the relative phase of a state is not observable, these kind of gates have the same effect as the original ones if measurement is applied right after the relative phase gate. However, in more complex circuits the relative phase may interfere with other gates applied and lead to undesired effects. An example of such a gate construction can be observed in Fig. 4.

An advantage of the diagonal matrix is that it can commute with many more operators than an ordinary unitary operator can, and therefore the relative phase

may be cancelled by a conjugated phase applied somewhere later in the circuit. In particular, a relative phase applied on a qubit can be cancelled after applying a gate that has that qubit as a control, as shown in Section III of [12]. Moreover, depending on the circuit shape, a gate and its inverse can be decomposed in pairs of gates that affect different numbers of qubits. If used in a circuit as presented in Fig. 6, we can further reduce the number of gates by eliminating the two corresponding sub-gates that are inverse to each other.

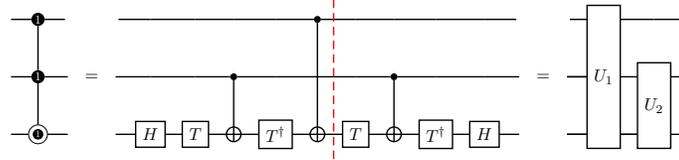


Fig. 5. Decomposition of a relative phase 2-controlled Toffoli gate [12]. The full construction represents the relative phase gate and the dashed red line marks the decomposition of this construction into U_1 and U_2 . The gate U_1 represents the short version of the RC^2X gate and has a depth of 5 with 2 CX gates and 3 single-qubit gates.

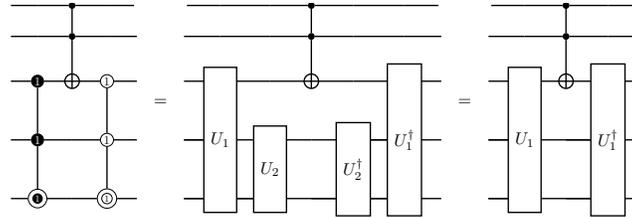


Fig. 6. Usage of conjugated short RC^2X gates in order to reduce the costs of the circuit. The first part of the figure presents the decomposition of the two conjugated RC^2X gates by using the formula $(AB)^\dagger = B^\dagger A^\dagger$, while the second part shows how to apply the gate reduction.

3.4 Using 3-controlled relative phase Toffoli gates

The main contribution of [12] was the introduction of 3-controlled relative phase Toffoli gates with the construction from Fig. 7 which is very cost-efficient, especially when the decomposition is used as in Fig. 6. In this case the cost of a single gate is of 4 CX gates and 6 one-qubit gates. Also, the author proposes a more efficient V-chain construction: all the Toffoli gates in the V-chain construction (Fig. 2) except for the ones at the bottom of the circuit can be replaced by 3-controlled relative phase gates. Moreover, when using dirty ancilla, we can actually use the short version of these gates for all but the two gates at the top. This method doubles the number of total control qubits in the circuit without the need to add more auxiliary qubits. The gate and depth cost of this circuit keep their linear complexities with respect to the number of control qubits n , but the constant factor decreases by 1/3 in the case of the dirty ancilla construction. For the rest of this paper we will refer to these circuits as 'V-chain-2'.

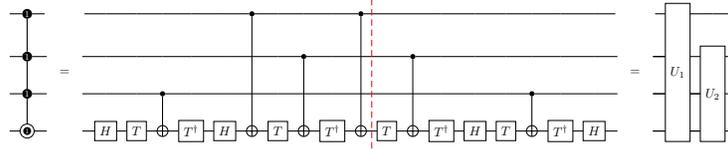


Fig. 7. Decomposition of a relative phase 3-controlled Toffoli gate (RC^3X) [12]. The construction represents the relative phase gate and the dashed red line marks the decomposition into U_1 and U_2 . The gate U_1 represents the short version of the RC^3X gate and has a depth of 10 with 4 CX gates and 6 single-qubit gates.

4 Our Proposed Methods

4.1 One Ancilla Constructions

Starting from the V-chain-2 constructions, we have developed two constructions for the one-ancilla MCT circuits in Fig. 3 that are more efficient than those presented in [2] and [8].

One Clean Ancilla: In the first part of Fig. 3 we can observe that there is a V-chain gate repeated twice. This means that reducing the size of these gates will reduce the overall cost of the circuit. In this construction, we have to split the n control qubits into two groups of m_1 and m_2 qubits each, that will take turns in being control and dirty ancilla for the three V-chain sub-circuits (Fig. 8). Since each of these sub-circuits needs $m/2$ ancilla qubits for their m control qubits, we have two constraints for m_1 and m_2 : $(m_1 - 3)/2 < m_2$ and $(m_2 - 2)/2 < m_1$. Since we want to optimise such that the value of m_1 is minimal, we will reach a configuration with $m_1 = \lceil \frac{n}{3} \rceil$ and $m_2 = \lfloor \frac{2n}{3} \rfloor$.

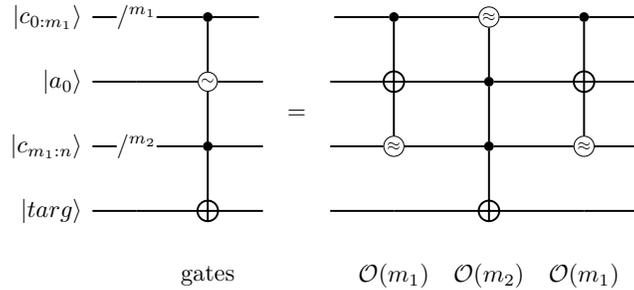


Fig. 8. One clean Ancilla construction with splitting the controls into two groups of m_1 and m_2 qubits respectively. The gate acting on the first group of m_1 qubits appears twice so we need to minimise its cost.

Calculating the cost of this construction we observe a decrease by a little more than $1/3$ over the version proposed in [2] (Details in Section 5).

One Dirty Ancilla: When designing the circuit using only one dirty ancilla, there is only one improvement to make: replace the V-chain gates with V-chain2 ones. Since both the top and bottom part of the circuit contain 2 gates each,

splitting the controls in two groups, m_1 and m_2 does not have any effect on the gate count or circuit depth. In this case we obtain again a decrease by $1/3$ over the version in [2].

4.2 Depth Optimisation with Dirty Ancilla

Since we have observed that the depth of the V-chain-2 circuit that uses $n/2$ dirty ancilla is two times less than the depth of the one-dirty-ancilla construction, we started the search for a general k -dirty-ancilla construction where k varies between 1 and $n/2$. In order to achieve optimal depth in our circuit, we propose the construction in Fig. 9.

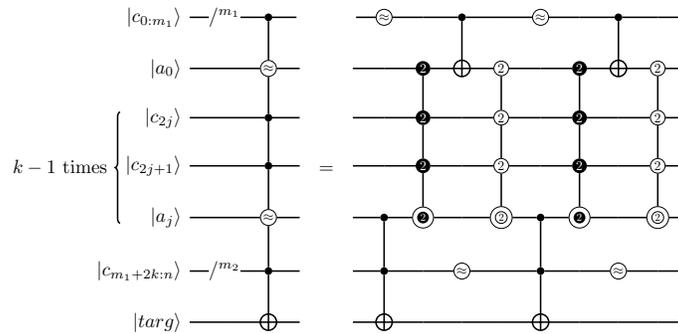


Fig. 9. Construction of a circuit simulating a MCT gate with k dirty ancilla qubits. We use $k - 1$ of the ancilla qubits in the middle part of the circuit to store the results of $2 \times (k - 1)$ pairs of short RC^3X gates. The two groups of m_1 and m_2 qubits at the top and bottom of the circuit alternate between being control qubits or dirty ancillae for the 4 V-chain-2 gates required. We have eliminated the line linking the ancilla qubits with their corresponding gate for clarity.

We mention that the authors of [1] have also proposed a dynamic programming approach to generate an optimal-depth circuit with an arbitrary number of dirty ancilla. However, they have not used the improved version V-chain-2, and therefore the costs of their circuits are higher (details in Section 5).

4.3 Depth Optimisation with Clean Ancilla

It should be clear by now from the discussion in Section 3 that clean ancilla qubits are more useful for reducing the circuit's gate count and depth. However, as stated in Section 1, they are a resource harder to obtain than the dirty ones, so we must always adapt our circuits to the amount of qubits available.

Logarithmic with Half Ancilla: The first idea to reduce the clean Ancilla count is to use relative phase 3-controlled gates as we have already done in the case of dirty ancilla. In order to maintain the logarithmic depth of the circuit, we use the same construction presented in [8], but replace all regular Toffoli gates with relative phase ones, except for the one acting on the target (Fig. 10).

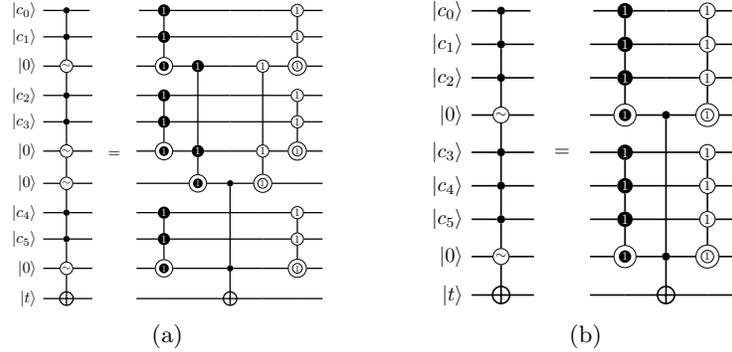


Fig. 10. Logarithmic depth constructions for $n = 6$ control qubits using (a) RC^2X gates and $n - 2$ clean ancilla or (b) RC^3X gates and $n/2$ clean ancilla. The first construction keeps the same depth even if we increase the number of control qubits to $n = 8$ and add two more clean ancillae.

The circuit depth will now be

$$D_3(n) = 2\lfloor \log_3 n \rfloor \times \text{depth}(RC^3X) + \text{depth}(C^2X) = \mathcal{O}(\log n).$$

In comparison, when using 2-controlled relative phase gates, the depth is

$$D_2(n) = 2\lfloor \log_2 n \rfloor \times \text{depth}(RC^2X) + \text{depth}(C^2X) = \mathcal{O}(\log n).$$

We can observe that the constant factor of our construction is higher, since

$$\frac{D_3(n)}{D_2(n)} = \frac{\text{depth}(RC^3X)}{\text{depth}(RC^2X)} \cdot \frac{\log_3 n}{\log_2 n} = \frac{18}{9} \log_3 2 = \log_3 4 \approx 1.26.$$

Depending on the physical implementation of the quantum computer, future engineers should decide whether the $n/2$ additional clean auxiliary qubits required for a $1.3\times$ decrease in depth represent a fair price or not.

Note: The volumes of the two circuits are almost equal, but the one we proposed is 5% smaller:

$$\frac{V_3(n)}{V_2(n)} = \frac{(n + n/2)D_3(n)}{(n + n - 1)D_2(n)} \approx \frac{3}{4} \log_3 4 \approx 0.95.$$

Lower cost with Logarithmic Ancilla: The next idea we present is one that reduces the depth and gate cost of the circuit in exchange for $\mathcal{O}(\log n)$ clean ancilla. The main idea is to recursively add layers to the circuit while using the already used control qubits as dirty ancilla before the restoring phase takes place. Using these qubits as dirty ancilla allows us to use the short version of V-chain-2 gates. The construction is detailed in Fig. 11.

This construction allows us to use j clean ancilla for building a circuit with $n = \mathcal{O}(3^{j-1} \cdot j)$ control qubits, and because of the clever way of using the reduced V-chain gates the depth and gate count of the circuit are around 30% less than in the case of the construction with just one clean ancilla. When compared to

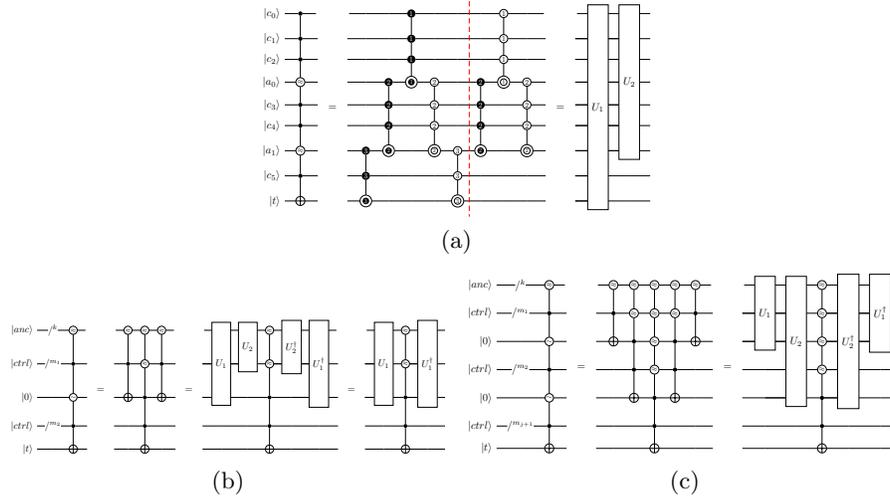


Fig. 11. Construction based on recursively appending layers of V-chain-2 gates. (a) Decomposition of V-chain-2 gate to generate its short form. (b) Usage of short V-chain-2 gates by conjugated gate cancelling: the U_2 and U_2^\dagger gates cancel each other because the gate between them uses the qubits only as dirty ancilla, so it preserves their state. (c) The procedure of recursively adding another layer of short V-chain-2 gates (U_1 and U_1^\dagger): we continue this procedure with smaller gates until we have no more ancilla left. The outer pair of V-chain-2 gates should only use the other $j - 1$ clean ancillae in the circuit and will therefore have $m_1 = 2j - 1$ controls.

the logarithmic depth construction which uses a linear number of clean ancilla, the gate count of this circuit is only higher with 25%, but the depth being linear is definitely a downside. However, in a situation where the number of available clean ancilla is low, this construction might still be preferred since with just 4 ancilla we could simulate circuits with $n \approx 80$ controls.

Sqrt compromise: The logarithmic depth construction could be generalised such that we can use gates with any number of control qubits as the building block. Considering the size of the building block gate to be k controls we would need $2 \cdot \lceil \log_k n \rceil + 1 \doteq 2L + 1$ layers of k -controlled V-chain-2 gates. However this would mean that we need some more auxiliary qubits for the first layer of gates (if we do not have these qubits we may split the first layer in two sub-layers with half of the controls acting as dirty ancilla for the other half). The number of auxiliary qubits needed will then be

$$\#_{clean} = \sum_{i=1}^{L-1} k^i = \frac{k^L - 1}{k - 1} \approx \frac{n}{k - 1}.$$

The depth of the circuit can be calculated as

$$D_k(n) = (2L + 1) * \text{depth}(k\text{-V-chain-2}) = \mathcal{O}(kL).$$

5.2 Dirty Ancilla Constructions

In Fig. 13a we compare the depth of our circuit with the depth obtained by [1] and in Fig. 13b we compare the depth and total gate cost for our circuit. It can be concluded from these comparisons that if we want to minimise the depth of the circuit we can use roughly $k \approx n/3$ dirty ancilla qubits with no great improvement over the $n/2$ ancilla version, while for also minimising the total gate count we need $k \approx n/2$.

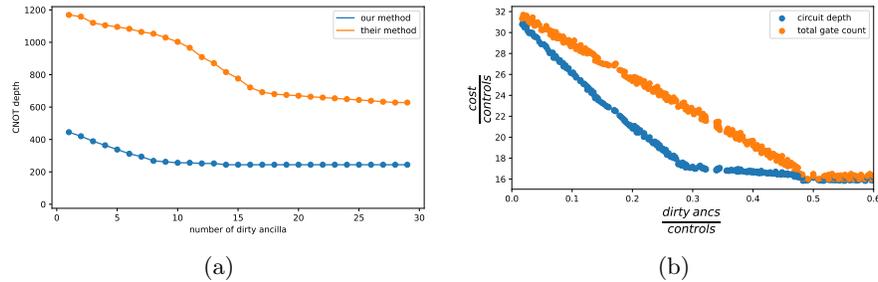


Fig. 13. Circuit costs of the circuit with varying number of dirty ancilla qubits. (a) Comparison between the CX depths of our method and the method proposed in [1] for a circuit with $n = 30$ control qubits. (b) Comparison between the total CX count and the CX depth of the circuit in our method. The horizontal axis shows the ratio between the number of ancilla and number of controls. The vertical axis shows the ratio between the cost and the number of controls. The data used was obtained for values of n between 10 and 100 so that the linear trends can be observed to be the same regardless of the value of n .

5.3 Clean Ancilla Constructions

In Fig. 14 we can observe a CX depth comparison between the methods that make use of clean ancillae. The V-chain-2 method is only shown for comparison, as it should never be used, since it can be replaced by its logarithmic depth equivalent. It is clear from this figure that our methods `one-anc-2` and `log-anc` have better depths than the method `one-anc-1` described in [2].

We can also conclude that the `sqrt` method proves its efficiency for higher values of n , where the circuit depth and ancilla need start to grow slower. When comparing the two logarithmic depth approaches, we can observe the theoretical prediction that our method `log-depth-2` comes at a halved ancilla cost and slightly increased depth, while on the Manhattan Quantum Computer there is no noticeable difference between its depth and that of the original method.

6 Conclusion and Future Work

In this work we have presented and analysed various methods for implementing Multi-Controlled Toffoli gates in Quantum Circuits and have proposed new, more efficient, and versatile constructions.

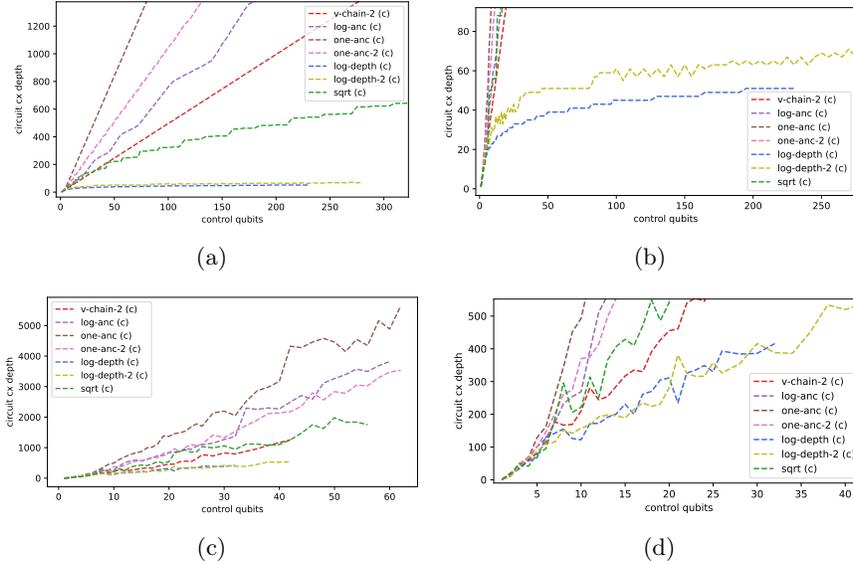


Fig. 14. Comparison between the CX depth of different implementations of MCT circuits with clean ancilla. (a) Comparison between the linear-depth constructions and the `sqrt` construction. (b) Detail of the logarithmic depth constructions: our method has a slightly higher depth, but uses only half as many ancilla. (c) Comparison of the costs on FakeManhattan. (d) Detail of logarithmic depth constructions on FakeManhattan: no noticeable difference between the two methods;

Since the Multi Controlled Gates are widely used in lots of Quantum applications we expect the ideas presented here to help reducing the costs of scalable Algorithms which will one day be run on fault-tolerant Quantum hardware.

While the constructions presented here might not be optimal in any given circumstances, we plan on designing an algorithm that will find the best circuit with respect to a given cost function and resource constraints.

We also plan to search for optimisations of other well-known and widely used Quantum Algorithms such as Quantum Fourier Transform, Arbitrary Permutations, and SAT Grover Oracles.

We invite future hardware engineers to choose the solution best suited for their physical implementation and encourage quantum software engineers to try and improve these constructions.

References

1. Baker, J.M., Duckering, C., Hoover, A., Chong, F.T.: Decomposing quantum generalized toffoli with an arbitrary number of ancilla (2019)
2. Barenco, A., Bennett, C.H., Cleve, R., Divincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. *Phys. Rev. A* **52**(5), 3457–3467 (Nov 1995). <https://doi.org/10.1103/PhysRevA.52.3457>

3. Cross, A.W., Bishop, L.S., Sheldon, S., Nation, P.D., Gambetta, J.M.: Validating quantum computers using randomized model circuits. *Phys. Rev. A* **100**, 032328 (Sep 2019), <https://link.aps.org/doi/10.1103/PhysRevA.100.032328>
4. Dirac, P.A.M.: *The Principles of Quantum Mechanics*. International series of monographs on physics, Clarendon Press (1981)
5. DiVincenzo, D.P.: The physical implementation of quantum computation. *Fortschritte der Physik* **48**(9-11), 771–783 (Sep 2000). [https://doi.org/10.1002/1521-3978\(200009\)48:9/11;771::aid-prop771;3.0.co](https://doi.org/10.1002/1521-3978(200009)48:9/11;771::aid-prop771;3.0.co)
6. Gidney, C.: Using quantum gates instead of ancilla bits (2015), <https://algassert.com/circuits/2015/06/22/Using-Quantum-Gates-instead-of-Ancilla-Bits.html>
7. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. p. 212–219. STOC '96, Association for Computing Machinery, New York, NY, USA (1996), <https://doi.org/10.1145/237814.237866>
8. He, Y., Luo, M.X., Zhang, E., Wang, H.K., Wang, X.F.: Decompositions of n -qubit Toffoli Gates with Linear Circuit Complexity. *International Journal of Theoretical Physics* **56**(7), 2350–2361 (Jul 2017). <https://doi.org/10.1007/s10773-017-3389-4>
9. Héctor Abraham, e.a.: Qiskit: An open-source framework for quantum computing (2019). <https://doi.org/10.5281/zenodo.2562110>
10. Kay, A.: Quantikz (8 2018). <https://doi.org/10.17637/rh.7000520.v4>
11. Krovi, H., Magniez, F., Ozols, M., Roland, J.: Quantum walks can find a marked element on any graph. *Algorithmica* **74**(2), 851–907 (Mar 2015), <http://dx.doi.org/10.1007/s00453-015-9979-8>
12. Maslov, D.: Advantages of using relative-phase toffoli gates with an application to multiple control toffoli optimization. *Physical Review A* **93**(2) (2 2016), <http://dx.doi.org/10.1103/PhysRevA.93.022311>
13. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press (2010). <https://doi.org/10.1017/CBO9780511976667>
14. Saeedi, M., Pedram, M.: Linear-depth quantum circuits for n -qubit toffoli gates with no ancilla. *Phys. Rev. A* **87**, 062318 (Jun 2013), <https://link.aps.org/doi/10.1103/PhysRevA.87.062318>
15. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. pp. 124–134 (1994). <https://doi.org/10.1109/SFCS.1994.365700>
16. Toffoli, T.: Reversible computing. In: de Bakker, J., van Leeuwen, J. (eds.) *Automata, Languages and Programming*. pp. 632–644. Springer Berlin Heidelberg, Berlin, Heidelberg (1980)
17. Vandersypen, L.M.K., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H., Chuang, I.L.: Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* **414**(6866), 883–887 (Dec 2001), <http://dx.doi.org/10.1038/414883a>
18. Yanofsky, N.S., Mannucci, M.A.: *Quantum Computing for Computer Scientists*. Cambridge University Press (2008). <https://doi.org/10.1017/CBO9780511813887>
19. Yao, X.W., Wang, H., et al.: Quantum image processing and its application to edge detection: Theory and experiment. *Physical Review X* **7**(3) (Sep 2017), <http://dx.doi.org/10.1103/PhysRevX.7.031041>
20. Zu, H., Dai, W., de Waele, A.: Development of dilution refrigerators—a review. *Cryogenics* **121**, 103390 (2022), <https://www.sciencedirect.com/science/article/pii/S001122752100148X>