

Mapping NCV Circuits to Optimized Clifford+ T Circuits

D. Michael Miller¹, Mathias Soeken^{2,3}, and Rolf Drechsler^{2,3}

¹ Dept. of Computer Science, University of Victoria, Victoria, BC, Canada V8W 3P6
mmiller@cs.uvic.ca

² Institute of Computer Science, University of Bremen, 28359 Bremen, Germany
{msoeken,drechsle}@informatik.uni-bremen.de

³ Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany

Abstract. The need to consider fault tolerance in quantum circuits has led to recent work on the optimization of circuits composed of Clifford+ T gates. The primary optimization objectives are to minimize the T -count (number of T gates) and the T -depth (the number of groupings of parallel T gates). These objectives arise due to the high cost of the fault tolerant implementation of the T gate compared to Clifford gates. In this paper, we consider the mapping of a circuit composed of NOT, Controlled-NOT and square-root of NOT (NCV) gates to an equivalent circuit composed of Clifford+ T gates. Our approach is heuristic and proceeds through three phases: (i) mapping a circuit of NCV gates to a Clifford+ T circuit; (ii) optimization of the placement of the T gates in the Clifford+ T circuit; and (iii) optimization of the subcircuits between T gate groupings. The approach takes advantage of earlier work on the optimization of NCV circuits. Examples are presented to show the approach presented here compares well with other approaches. Our approach does not add ancilla lines.

1 Introduction

Quantum circuits are an important model of quantum computation and there is thus considerable interest in the synthesis and optimization of such circuits [9, 11, 15]. Recently, there has been particular interest in circuits composed of Clifford+ T gates [2, 3, 14] where a major objective is to minimize the number of T gates and particularly the T -depth of the circuit. This is motivated by the importance of fault tolerance in quantum computations [5, 18] and by the fact the cost of the fault tolerant implementation of a T gate can exceed the cost of implementing a Clifford gate by a factor of 100 or more [2].

Previously, there has been work on the optimization of NCV circuits [4, 13]. In this paper, we consider the mapping of an NCV circuit to an equivalent circuit composed of Clifford+ T gates with particular emphasis on optimizing T -count and T -depth. The approach is heuristic but as examples will show, the approach compares well with other methods. In particular, we compare our method to circuits produced by the matroid partitioning approach described by Amy *et al.* [2] which optimizes T -count and T -depth. Our approach is comparable for those parameters and yields lower circuit depth in certain cases. We do not consider the addition of ancilla lines in this work.

Table 1. Gate definitions

Type	Symbol	Matrix	Diagram	Type	Symbol	Matrix	Diagram
NOT	N	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	\oplus	CNOT	C	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	
<i>controlled</i> V ($V = \sqrt{N}$)	V	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1+i}{2} & \frac{1-i}{2} \\ 0 & 0 & \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix}$		<i>controlled</i> V^\dagger	V^\dagger	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1-i}{2} & \frac{1+i}{2} \\ 0 & 0 & \frac{1+i}{2} & \frac{1-i}{2} \end{pmatrix}$	
Hadamard	H	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	\boxed{H}				
T gate	T	$\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}$	\boxed{T}	T gate $^{-1}$	T^\dagger	$\begin{pmatrix} 1 & 0 \\ 0 & e^{-\frac{i\pi}{4}} \end{pmatrix}$	$\boxed{T^\dagger}$
Phase	S	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	\boxed{S}	Phase $^{-1}$	S^\dagger	$\begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$	$\boxed{S^\dagger}$

The rest of the paper is organized as follows. Section 2 provides the background for the work. Our NCV to Clifford+ T mapping approach is described in Section 3 and examples are given in Section 4. The paper concludes with suggestions for further work in Section 5.

2 Background

We assume the reader is familiar with the basics of quantum circuits and their representations and only provide the notation and specifics required for this work. A full review of the background can be found in the literature, *e.g.* [11].

Definition 1. A quantum circuit is a model of quantum computation representing a sequence of quantum operations. Each operation is represented by a quantum gate and the circuit is a cascade of gates where the circuit lines represent the qubits (quantum bits) of a quantum system. A quantum circuit has no fanout or feedback.

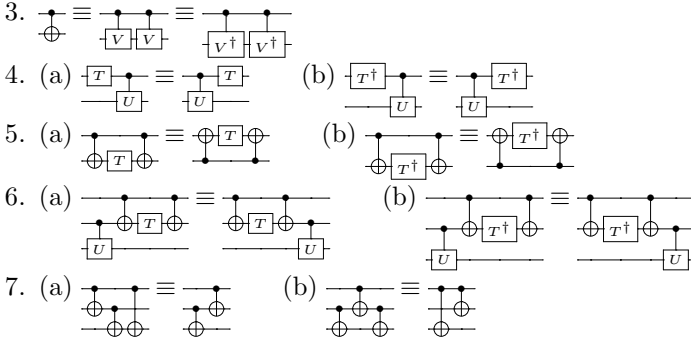
The gates used in quantum circuits are commonly defined by unitary matrices and we do so in Table 1 which shows the gates used in this paper. U will denote an arbitrary quantum operation. Note that since the matrices of interest are unitary, the adjoint (denoted by \dagger) is the inverse.

The C , V and V^\dagger gates are controlled gates and the operation is applied to the target line if, and only if, the control line (indicated by a \bullet) is 1.

The Toffoli gate [11] is like the CNOT except it has two controls which must both be 1 for the target to be inverted. This has been generalized to multiple-control Toffoli (MCT) gates [4] which have a number of controls all of which must be 1 for the target to be inverted.

Quantum Gate Properties. The following quantum gate properties are important in this work.

- (a) $H = H^{-1}$ (b) $S = TT$ (c) $S^\dagger = T^\dagger T^\dagger$
- (a) (b)



We refer to the structures in Property 5 as CTC structures where the T can be a T or T^\dagger gate. Note the three gates in the structure have a common target and the two CNOTs have a common control. Applying Property 5(a) or (b) will be referred to as *flipping* the CTC structure.

Property 6 is somewhat surprising and provides considerable flexibility in moving gates. It is a direct consequence of Properties 4 and 5.

Property 7 gives two important identities for reducing the number of CNOT gates in a circuit. The applicability of Property 7(a) was illustrated in [17]. Note the equivalence shown is a particular example. The general rule is that interchanging 2 CNOTs which share a qubit as target for one and control for the other introduces a third CNOT with control and target from the unshared lines of the initial pair of gates. Property 7(b) is a direct consequence of 7(a). These properties have not been widely used in the literature but are in fact quite effective as will be shown below.

In this paper, we refer to three types of circuits:

- MCT: which are composed of MCT gates which includes NOT, CNOT and Toffoli gates;
- NCV: which are composed of N , C , V and V^\dagger gates; and
- Clifford+ T : which are composed of the Clifford gates N , C , H , S and S^\dagger , together with T and T^\dagger gates.

The following define key characteristics used in evaluating circuits.

Definition 2. A circuit level is defined as a sub-sequence of gates in a circuit that can be applied in parallel. We assume in this work that two or more gates can operate at the same level if they operate on disjoint qubits and can be grouped together in the circuit.

Definition 3. Circuit depth is the number of levels in the circuit.

Definition 4. The T -count of a Clifford+ T circuit is the total number of T and T^\dagger gates in the circuit.

Definition 5. The T -depth of a Clifford+ T circuit is the number of levels in the circuit that contain one or more T or T^\dagger gates.

3. If g is a V or V^\dagger gate with target t :
 - (a) If $t \notin \mathcal{H}$ add H, S, T gates as described in 2(a)-(d) above substituting t for c .
 - (b) Put t into set \mathcal{H} .
4. If g is a V gate,
 - (a) Add 1 to \mathcal{J}_t and add 1 to \mathcal{J}_c .
 - (b) Replace gate g with three gates CNOT- T^\dagger -CNOT using control c and target t .
5. Else if g is a V^\dagger gate,
 - (a) Subtract 1 from \mathcal{J}_t and subtract 1 from \mathcal{J}_c .
 - (b) Replace gate g with three gates CNOT- T -CNOT using control c and target t .
6. Else add H, S, T gates as described in 2(a)-(d) above substituting t for c .
7. Set g to the next gate to the right in the circuit. If there is no such gate the procedure is done.
8. Go to step 2.

To illustrate the application of this algorithm, consider the reversible full adder circuit in Fig. 1(a) [11]. Each Toffoli-CNOT pair is in fact a Peres [11] gate. Figure 1(b) is an optimal NCV realization found by expanding the two Peres gates and then canceling two gates [10]. The Clifford+ T circuit shown in Fig. 1(c) illustrates key features of the initial expansion process. The T gates placed on the controls of the three V gates (1,2,3) are positioned as far right as possible. Note that a T and T^\dagger gate cancellation coming from gates 3 and 5 occurs on line d and the T gates from gates 1 and 2 combine to form an S gate which is again placed as far right as possible.

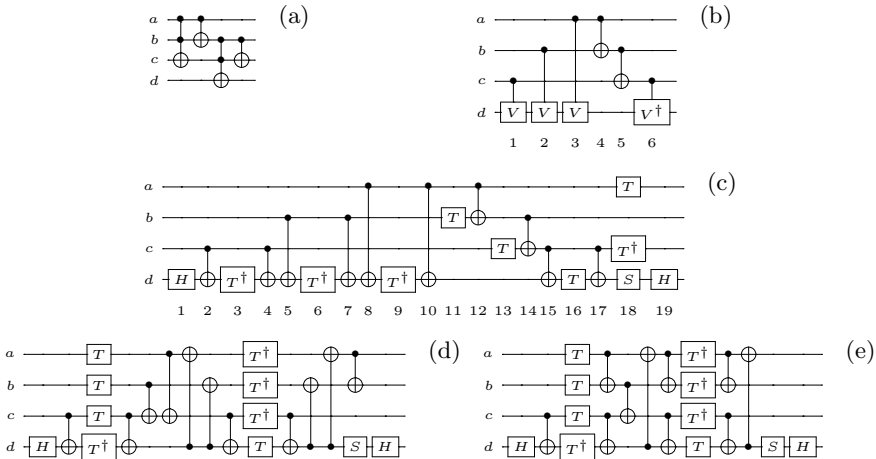


Fig. 1. Full adder

T Gate Parallelization. T gate parallelization means moving T and T^\dagger gates, each on a unique circuit line, to the same circuit level so that they can operate in parallel. The objective is to reduce the T -depth of the circuit. We consider three types of T and T^\dagger gate moves:

- I. A T or T^\dagger gate that can be moved to the desired position unimpeded, *i.e.* it can pass over all intervening gates with no additional gate movement or alteration.
- II. Moving the T or T^\dagger gate to the desired position requires moving one or more CNOTs and/or requires that a CTC group be flipped.
- III. A sequence of Type I and Type II moves is required to reposition a T or T^\dagger gate.

Our approach to T gate parallelization is outlined below. Note that this algorithm addresses only Type I and Type II moves. Type III moves are applied afterwards. Several illustrative examples are given in the next section.

T Gate Parallelization Algorithm. Let n be the number of qubits (lines) in the circuit.

1. Start at the leftmost CTC structure in the circuit.
2. Let p be the position of the T or T^\dagger in the CTC structure.
3. Set $X1_i$ to be the position of the leftmost T gate that can be moved to position p on line i by a Type I move. Set $X1_i = \emptyset$ if no such T gate exists.
4. Set $X2_i$ to be the set of the positions for all T gates that can be moved to be in position p on line i (the set may be empty). Note that in doing this each CTC structure is entered twice as it can be moved flipped or unflipped, *i.e.* the T or T^\dagger can end up on one of two possible lines.
5. Let t be the target line and c be the control line in the CTC being considered. If $X1_t \neq \emptyset$ and $X1_c = \emptyset$, flip the CTC structure; otherwise set $X1_t = \emptyset$.
6. For each line i , if $X1_i \neq \emptyset$ move the T or T^\dagger gate from position $X1_i$ to position p on line i (a Type I move) and set $X2_i = \emptyset$.
7. For each line i , if $X2_i \neq \emptyset$ move the T or T^\dagger gate from position j to position p on line i (a Type II move) where j is the position of the leftmost gate identified in $X2_i$. Remove j from any other $X2_k$, $k > i$.
8. Go right to the next CTC structure and go to step 2. If there is no such structure, the procedure is done.

As an example, applying our T gate parallelization approach to the circuit in Fig. 1(c) yields the circuit in Fig. 1(d). First, the left parallelization is accomplished by moving T gates from positions 18, 11 and 13 left on lines a , b and c respectively using Type I moves. Second, the right parallelization requires the gate groups (5,6,7) and (8,9,10) to be flipped and then gates 8 and 9 have to be shifted left. The group (15,16,17) then has to be shifted into position by a Type II move. To do that requires gate 14 to be moved left which requires gate 14 be moved past gate 12 (recall that gate 13 has already been moved to the left parallelization). Moving gate 14 past gate 12 creates the CNOT($a;c$) gate in Fig. 1(d). Lastly, the T^\dagger gate in position 18 is moved left to join the parallelization.

CNOT Reduction. A circuit composed solely of CNOTs is a linear reversible circuit. As can be seen in Fig. 1, a Clifford+T circuit typically has a number of linear subcircuits. It is thus useful to consider optimization of a circuit of CNOT gates.

As shown in Patel *et al.* [12], this problem can be expressed in terms of operations on a matrix over GF(2). In particular, the operation of a CNOT gate can be expressed as a matrix row operation. In particular, the operation of a CNOT with control α and target β corresponds to replacing row β by the mod-2 sum of rows α and β .

For example, consider the 6 CNOT gates between the two T gate parallelizations in Fig. 1(d). Starting from a 4×4 identity matrix and applying (α, β) row additions in the order (c, d) , (b, c) , (a, c) , (d, a) , (d, b) , (c, d) , corresponding to the 6 CNOTs gates from the adder, yields the matrix in (2). This matrix represents the functionality of the 6 CNOTs. The problem here is to determine if there is a more efficient CNOT circuit performing the same functionality.

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \quad (2)$$

One approach is to use Gaussian elimination over GF(2). This is typically not optimal. Patel *et al.* [12] have given a method that is typically more effective in terms of CNOT count but is not always optimal. For example, applying their method to the matrix in (2) yields a circuit that can be further reduced using Property 7. Also, the Patel method was not designed to take advantage of levelizing CNOTs, *i.e.* positioning two or more CNOTs into one circuit level.

CNOT Optimization Algorithm. Given a 0-1 matrix M , let M_i denote the i^{th} row of M and let $|M_i|$ denote the number of 1's in that row. Our method proceeds as follows:

1. Set $X = \emptyset$. (Note: The set X is key to making choices that allow levelization.)
2. If M is the identity matrix, go to step 10.
3. For all possible pairs, find the pair (α, β) such that $\alpha \notin X$, $\beta \notin X$ and $|M_\beta| - |M_\beta + M_\alpha|$ is maximal, and there is a 1 in the β position in $|M_\beta + M_\alpha|$ where the addition is modulo 2.
4. If the maximum value found in step 3 is 0, apply the Patel *et al.* method [12] to M to complete the solution.
5. If no (α, β) is found in step 3, set $X = \emptyset$ and go to step 2.
6. If an (α, β) is found in step 3, add a CNOT with control α and target β to the solution. Note that the gates are generated in order from right to left.
7. Replace M_β by $M_\beta + M_\alpha$ where the addition is modulo 2.
8. Add α and β to X .
9. Return to step 2.
10. Property 7 is systematically applied to further reduce the number of CNOTs.

Applying the above method to the matrix in (2) yields the (α, β) sequence: (a, b) , (c, d) , (d, a) , (b, c) , (a, b) , (c, d) which corresponds to the CNOT sequence shown in Fig. 1(e). Note that while there are 6 CNOTs in both circuits,

the sequence in Fig. 1(d) requires 5 levels, while the sequence in Fig. 1(e) requires only 3. Using CNOT reduction, the CNOTs at the right end of Fig. 1(d) can also be reduced as shown in Fig. 1(e) saving one level in the circuit. The circuit in Fig. 1(e) is the same as identified in [3, 16].

4 Examples

In this section we provide a number of examples to illustrate the application of our approach. In particular, these examples illustrate certain aspects of optimizing T gate parallelization not illustrated in the full adder example.

Toffoli Gate with Two Controls. It is well-known [4] that the 2-control Toffoli gate depicted in Fig. 2(a) can be realized by 5 NCV gates as shown in Fig. 2(b). Figure 2(c) is found by expanding the V and V^\dagger gates in Fig. 2(b) to Clifford+ T gates. Note that the H gates that would fall between gates 1 and 3 and between 3 and 5 cancel as do a T from the expansion of gate 1 and a T^\dagger from the expansion of gate 3 on line c .

The next step is T gate parallelization. Gates 1e and 5d can be moved to be above gate 1c using Type I moves since there are no intervening CNOT targets or H gates. Similarly, gate 3d can be moved above 3b as a Type I move. Gate 5b is moved above 3b by a Type II move. In particular, the gate group (5a,5b,5c) is flipped and then the gates are moved into the positions shown in Fig. 2(d).

The circuit in Fig. 2(d) has T -count 7, T -depth 3 and 12 levels. The number of levels can be reduced by shifting gate 3d left; flipping the gate group (3a,3b,3c); and then shifting gate 5e left. This yields the circuit in Fig. 2(e) which has 11 levels since gates 5a and 3d can be combined in a single level. Note that no CNOT optimization is possible for this circuit. The T -count of 7 and T -depth of 3 are optimal but a circuit with 10 levels has been found previously [16].

The Function a2x [2]. The function a2x can be realized by 2 Toffoli gates as shown in Fig. 3(a). An NCV realization is shown in Fig. 3(b). Note that the realizations for the Toffoli gates have been arranged so that a V from the left Toffoli combines with a CNOT from the right Toffoli to yield the V^\dagger gates marked by the arrow. This reduction is possible since a CNOT can be replaced by two identical V^\dagger gates which then leads to a V - V^\dagger cancellation. Using the initial expansion and T gate parallelization techniques in the manner described in the above examples yields the circuit in Fig. 3(c).

Figure 3(d) shows two CNOT optimizations. The 3 leftmost CNOTs in Fig. 3(c) are reduced to 2. The 3 CNOTs between the right two T gate parallelizations are also reduced to 2.

Figure 3(d) also shows the insertion of 2 identical CNOTs at positions 3 and 4. This is possible since they functionally cancel. The reason for the insertion is so that the gate group (4,5,6) can be flipped and then moved to the positions shown in Fig. 3(e) with the effect of reducing the T -depth by 1. There is also a resulting CNOT reduction shown in Fig. 3(e). The gates in position 1 and 2 have been interchanged which results in the gate from position 3 being eliminated. Our final circuit has T -count 12, T -depth 4 and circuit depth 20. The circuit in [2] also has T -count 12 and T -depth 4. It has 23 levels.

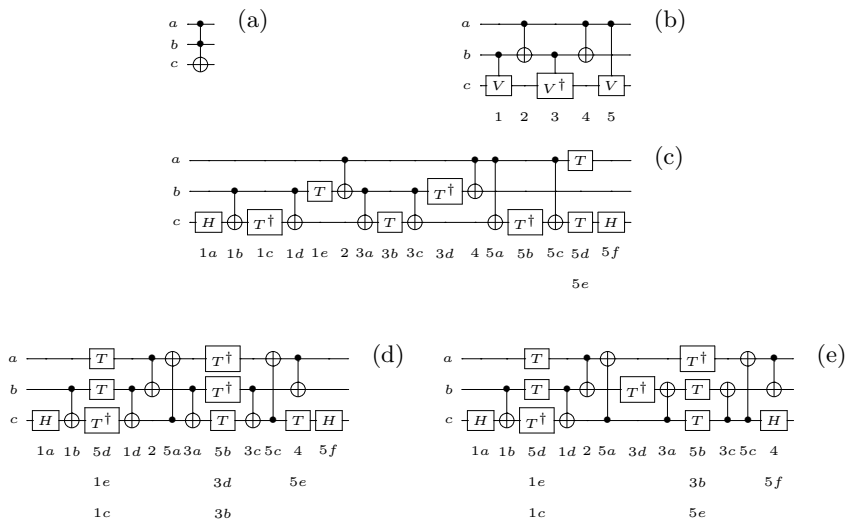


Fig. 2. 2-control Toffoli gate

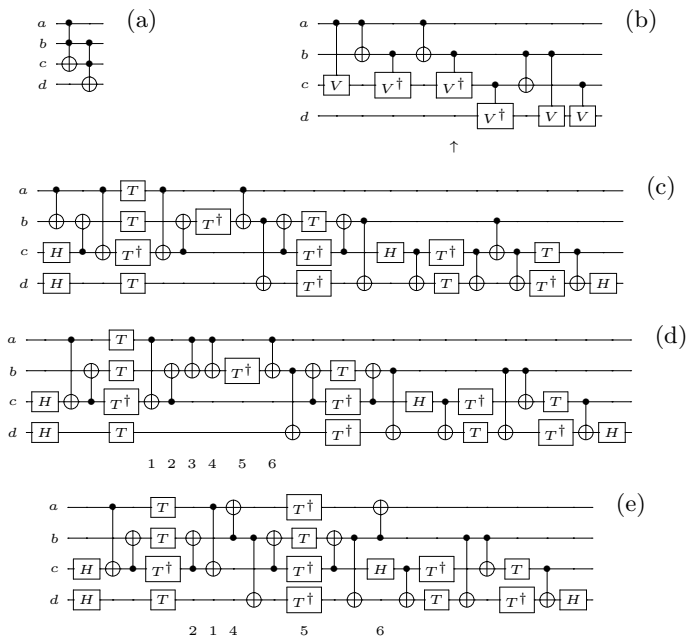


Fig. 3. $a2x$

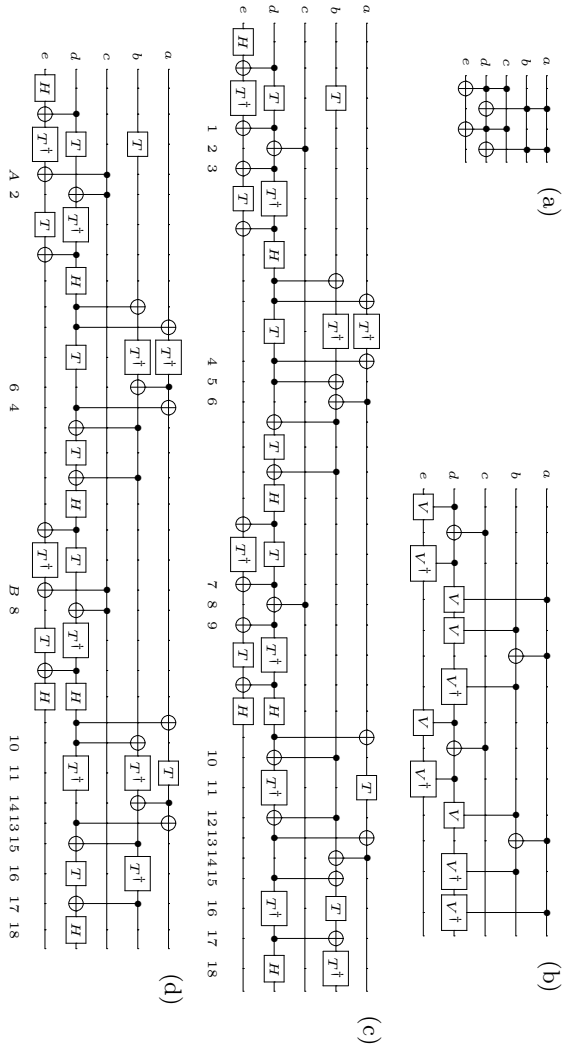


Fig. 4. a33x

The Function $a3x$ [2]. The Toffoli gate circuit for $ax3$ is shown in Fig. 4(a). Note that this circuit implements the 3-control Toffoli gate $TOF(a, b, c; e)$ using d as an ancillary line. The mapping to NCV gates we use is shown in Fig. 4(b). Note that the mapping for the 2 Toffoli gates with target e are chosen so that a total of 4 NCV gates, 2 from each Toffoli, cancel, so each Toffoli maps to 3 gates. Likewise, the 2 Toffoli gates with target d are mapped so that 2 CNOTs cancel so that each of these Toffoli gates is mapped to 4 gates. The V/V^\dagger assignment for the rightmost Toffoli is chosen so that there are 3 V and 3 V^\dagger on line d which heightens the opportunity for cancellations during the expansion process. This mapping is, up to reordering, the NCV realization given in [13].

Figure 4(c) shows the Clifford+ T circuit after T gate parallelization is almost complete. This circuit has T -depth 9. However by doing the following in order: flipping the gate group at (10,11,12); moving the T^\dagger from position 16 to position 11; flipping the gate group at (15,16,17) and then moving the T^\dagger from position 18 to position 16, the T -depth can be reduced to 8. Note that gates (1,2,3) in Fig. 4(c) become gates A,2 in Fig. 4(d). Likewise, gates (4,5,6) become 6,4 and gates (7,8,9) become B,8. These changes are the result of CNOT reductions.

Our final circuit has T -count 18 and T -depth 8 with circuit depth 33. Amy *et al.* [2] give a circuit for $a3x$ with T -count 16 and T -depth 8. Their circuit has circuit depth 40. Which of the two circuits is truly better will likely be technology dependent. The fact they have the same T -depth makes the reduction from 40 to 33 circuit levels attractive.

The Function 3_17. Our final example is known as 3_17 [19]. The NCV circuit is shown in Fig. 5(a). Fig. 5(b) shows the circuit after the Type I and Type II

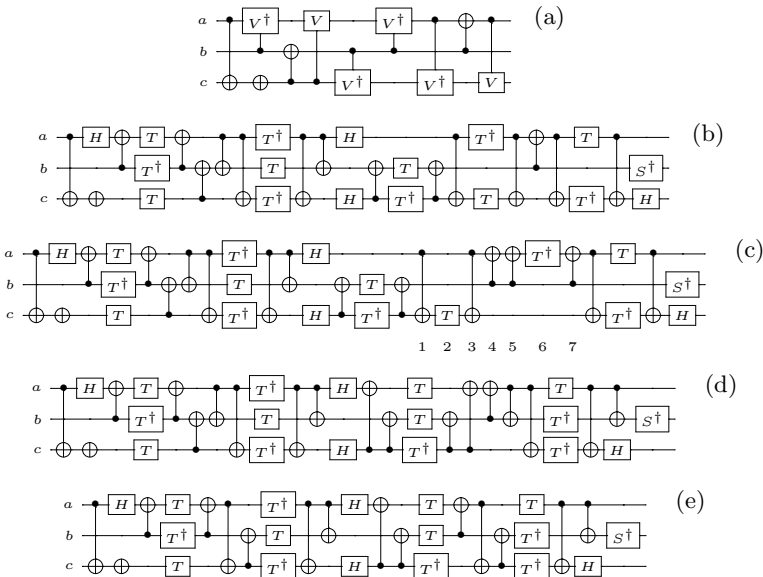


Fig. 5. 3_17

moves in the T gate parallelization phase. The circuit in Fig. 5(c) is derived from the one in Fig. 5(b) by moving the T^\dagger gate into position 6 and adding the two CNOTs in positions 4 and 5. The gate group (1,2,3) can be flipped and moved left and the gate group (5,6,7) can be flipped and moved right. Doing both of those operations yields the circuit in Fig. 5(d) with T -depth 4 rather than 5. CNOT reduction then yields the final circuit, Fig. 5(e).

5 Conclusion and Future Work

In this paper, we have presented a method for mapping an NCV circuit to a Clifford+ T circuit. Several examples were presented to illustrate the approach. The results show that the method is promising as it produces results comparable to earlier methods.

More work is required to determine how effective the method will be for larger NCV circuits. The methods presented are heuristic and further study is required to determine if the choices made are in fact the most effective. It would be most interesting to look at the integration of our methods with the matroid partitioning approach due to Amy *et al.* [3], particularly our CNOT optimization approach.

In this paper, we have assumed that gates can be applied in parallel, *i.e.* at the same circuit level, if they involve different qubits. Depending on the technology, a different assumption may apply. For example, the *nearest neighbour* constraint requires that the target and control for a controlled gate must be adjacent circuit lines [7]. In that case, the CNOT optimization phase of our approach must be modified or another appropriate nearest neighbour CNOT optimization technique [1, 6, 8] can be applied. Other constraints may apply for other technologies and again the CNOT optimization phase must be appropriately adjusted. The initial expansion and T -gate parallelization phases are not affected.

H gates are particularly problematic for our approach since they put limits on the movement of other gates in the circuit and can affect the ability of our methods to optimally parallelize T gates. Work to limit or to judiciously place H gates would be valuable. In addition, it is not clear that minimal NCV circuits are the best starting point since non-minimality may in some instances lead to better H gate placement.

Acknowledgment. This work was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

References

1. AlFailakawi, M., AlTerkawi, L., Ahmad, I., Hamdan, S.: Line ordering of reversible circuits for linear nearest neighbour realization. *Quantum Inf. Process.* 12, 3319–3339 (2013)
2. Amy, M., Maslov, D., Mosca, M.: Polynomial-time T -depth optimization of Clifford+ T circuits via matroid partitioning, arXiv:quant-ph/1303.2042v2 (2013)
3. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans. on CAD* 32(6), 818–830 (2013)

4. 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 (1995)
5. Buhrman, H., Cleve, R., Laurent, M., Linden, N., Schrijver, A., Unger, F.: New limits on fault-tolerant quantum computation. In: *Foundations of Computer Science*, vol. 27, pp. 411–419. IEEE Computer Society (2006)
6. Chakrabarti, A., Sur-Kolay, S., Chaudhury, A.: Linear nearest neighbor synthesis of reversible circuits by graph partitioning. CoRR, arXiv:1112.0564v2 (2012)
7. DiVincenzo, D.P., Bacon, D., Kempe, J., Burkard, G., Whaley, K.B.: Universal quantum computation with the exchange interaction. *Nature* 408, 339–342 (2000)
8. Khan, M.H.A.: Cost reduction in nearest neighbour based synthesis of quantum Boolean circuits. *Engineering Letters* 16, 1–5 (2008)
9. Lukac, M.: *Quantum Inductive Learning and Quantum Logic Synthesis*. BiblioLabsII (2011)
10. Maslov, D., Dueck, G.W., Miller, D.M., Negrevergne, C.: Quantum circuit simplification and level compaction. *IEEE Trans. CAD* 27(3), 436–444 (2008)
11. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information*. Cambridge University Press (2000)
12. Patel, K., Markov, I.L., Hayes, J.P.: Optimal synthesis of linear reversible circuits. *Quantum Information and Computation* 8(3&4), 282–294 (2008)
13. Sasanian, Z., Miller, D.M.: Mapping a multiple-control Toffoli gate cascade to an elementary quantum gate circuit. *Multiple-Valued Logic and Soft Computing* 18(1), 83–98 (2012)
14. Selinger, P.: Quantum circuits of T -depth one. *Phys. Rev. A* 87, 042302 (2013)
15. Shende, V.V., Bullock, S.S., Markov, I.L.: Synthesis of quantum logic circuits. *IEEE Trans. on CAD* 25(6), 1000–1010 (2006)
16. Soeken, M., Miller, D.M., Drechsler, R.: Quantum circuits employing roots of the Pauli matrices. *Phys. Rev. A* 88, 042322 (2013)
17. Soeken, M., Thomsen, M.K.: White dots *do* matter: Rewriting reversible logic circuits. In: Dueck, G.W., Miller, D.M. (eds.) RC 2013. LNCS, vol. 7948, pp. 196–208. Springer, Heidelberg (2013)
18. Weinstein, Y.S.: Non-fault tolerant T -gates for the $[7,1,3]$ quantum error correction code. *Phys. Rev. A* 87, 032320 (2013)
19. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: An online resource for reversible functions and reversible circuits. In: *Int’l Symp. on Multi-Valued Logic*, pp. 220–225 (2008), RevLib is available at www.revlib.org