On quantum circuits employing roots of the Pauli matrices

Mathias Soeken,1 D. Michael Miller,2 and Rolf Drechsler1
1Institute of Computer Science, University of Bremen, Germany
2Department of Computer Science, University of Victoria, BC, Canada

The Pauli matrices are a set of three 2 × 2 complex Hermitian, unitary matrices. In this article, we investigate the relationships between certain roots of the Pauli matrices and how gates implementing those roots are used in quantum circuits. Techniques for simplifying such circuits are given. In particular, we show how those techniques can be used to find a circuit of Clifford+T gates starting from a circuit composed of gates from the well studied NCV library.

I. INTRODUCTION

The well-studied NCV quantum gate library [1] contains the gates: NOT (X), controlled NOT (CNOT), and both the single controlled square root of NOT as well as its adjoint, denoted V and V†, respectively. In their seminal paper [2], Barenco et al. presented a general result that as one instance shows how a classical reversible Toffoli gate can be realized by five NCV gates as follows:

\[
\begin{array}{l}
\text{NOT} \\
\text{CNOT} \\
V \\
V^\dagger \\
\end{array}
\]

and the adjoint gates S† and T † has also received considerable attention. The Clifford+T library has the advantage over the NCV library with respect to fault-tolerant computing [9].

Recently different synthesis results based on the Clifford+T gate library have been presented [10,11,12]. One common aim in these works is to reduce the so-called T-depth, i.e. the number of T-stages where each stage consists of one or more T or T † gates that can operate simultaneously on separate qubits. In [10] the authors describe a search-based algorithm that finds optimal circuit realizations with respect to their T-depth. One of their circuits that realizes a Toffoli gate is [10, Fig. 13]:

\[
\begin{array}{l}
\text{T} \\
\text{CNOT} \\
\text{S} \\
\text{S}^\dagger \\
\end{array}
\]

This circuit has a T-depth of 3 and a total depth of 10. (Note that the gates surrounded by the dashed rectangle together have a depth of 1 and are drawn in sequence only for clarity.) The approach in [10] produces optimal circuits but since the technique’s complexity is exponential, it is only applicable to small circuits.

The NOT operation is described by the Pauli X matrix [10]. Note that the controlled NOT could be drawn as a controlled X but we use the normal convention of a ⊕ as shown in the figures above. The matrices for the V and V† operations as used in [10] are square roots of the Pauli X matrix. Similarly, the T gate operation is given by a matrix [2] that is the fourth root of the Pauli Z matrix.

The use of gates associated with different Pauli matrices and their roots within the same circuit, as illustrated in [3], motivated us to explore the relations between the Pauli matrices and their roots, and to investigate how the associated gates can be used in constructing quantum circuits. This article presents our findings and demonstrates their applicability in deriving the optimal circuits from [8] from known NCV circuits rather than using exhaustive search techniques.

II. PRELIMINARIES

The three Pauli matrices [10] are given by

\[
\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
\]

The alternate naming X = σ1, Y = σ2, and Z = σ3 is often used and we use it whenever we refer to a specific Pauli matrix.

Matrices describing rotations around the three axes of the Bloch sphere are given by

\[
R_a(\theta) = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \sigma_a
\]

where \(a \in \{1,2,3\}\) with \(\theta\) being the rotation angle and I the identity matrix [10]. Each Pauli matrix specifies a half turn (180°) rotation around a particular axis up to a global phase, i.e.

\[
\sigma_a = e^{i \frac{\theta}{2}} R_a(\pi).
\]

The conjugate transpose of \(R_a(\theta)\) is found by negating the angle \(\theta\) or by multiplying it by another Pauli matrix \(\sigma_b\) from both sides, i.e.

\[
R_a^\dagger(\theta) = R_a(-\theta) = \sigma_b R_a(\theta) \sigma_b
\]
where \( a \neq b \). Note that it does not matter which of the two possible \( \sigma_b \) is used. Since \( \sigma_b \) is Hermitian, we also have \( R_a(\theta) = \sigma_b R_a^\dagger(\theta) \sigma_b \).

Rotation matrices are additive with respect to their angle, i.e., \( R_a(\theta_1) R_b(\theta_2) = R_a(\theta_1 + \theta_2) \), so one can derive the \( k \)-th root of the Pauli matrices as well as their conjugate transpose from (9) as

\[
\sqrt[k]{\sigma_a} = e^{\frac{i\pi}{k}} R_a\left(\frac{\pi}{k}\right) \quad \text{and} \quad \sqrt[k]{\sigma_a^\dagger} = e^{-\frac{i\pi}{k}} R_a^\dagger\left(\frac{\pi}{k}\right). \quad (8)
\]

For brevity, we term these matrices the Pauli roots.

Using (9), the rotation matrices can also be expressed in terms of the roots of the Pauli matrices, i.e.

\[
R_a\left(\frac{\pi}{k}\right) = e^{-\frac{i\pi}{k}} \sqrt[k]{\sigma_a} \quad \text{and} \quad R_a^\dagger\left(\frac{\pi}{k}\right) = e^{\frac{i\pi}{k}} \sqrt[k]{\sigma_a^\dagger}. \quad (9)
\]

Consequently, we can derive

\[
\sqrt[k]{\sigma_a} = e^{\frac{i\pi}{k}} R_a\left(\frac{\pi}{k}\right) \quad \sqrt[k]{\sigma_b} R_a(\theta) \sigma_b \quad \text{and} \quad e^{\frac{i\pi}{k}} \sqrt[k]{\sigma_b} \sigma_b
\]

and analogously \( \sqrt[k]{\sigma_a^\dagger} = e^{-\frac{i\pi}{k}} \sigma_b \sqrt[k]{\sigma_a} \sigma_b \) for \( a \neq b \). This last equation will be a key element in showing how control lines can be removed from controlled gates by using roots of higher degree. A number in brackets above an equal sign indicates the identity applied.

Translation from one Pauli matrix to another is given by

\[
\sigma_a = \rho_{ab} \sigma_b \rho_{ab}
\]

where

\[
\rho_{ab} = \rho_{ba} = \frac{1}{2} (\sigma_a + \sigma_b) = e^{\frac{i\pi}{k}} R_a\left(\frac{\pi}{k}\right) R_b\left(\frac{\pi}{k}\right) R_a\left(\frac{\pi}{k}\right)
\]

with \( a \neq b \) are translation matrices. Further, we define \( \rho_{aa} = I \). Eq. (11) can be extended to the Pauli roots giving

\[
\sqrt[k]{\sigma_a} = \rho_{ab} \sqrt[k]{\sigma_b} \rho_{ab}. \quad (13)
\]

The well-known Hadamard operator is given by the normalized \( 2 \times 2 \) matrix

\[
H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
\]

which can also be expressed as

\[
H = \frac{1}{\sqrt{2}} (X + Z)
\]

or in terms of rotation matrices as

\[
H = e^{i\pi} R_1\left(\frac{\pi}{2}\right) R_3\left(\frac{\pi}{2}\right) R_1\left(\frac{\pi}{2}\right)
\]

In fact, \( H = \rho_{12} = \rho_{21} \) so the Hadamard operator can be used to translate between \( X \) and \( Z \), i.e.

\[
Z = HXH \quad \text{and} \quad X = HZH
\]

Given a unitary \( 2^n \times 2^n \) matrix \( U \), called the target operation, we define four controlled operations

\[
\begin{array}{cccc}
\bullet & U & \bullet & , \quad \bullet & | & \bullet & U & | & \bullet & , \quad \bullet & U & \bullet & , \quad \bullet & | & \bullet & U & | & \bullet
\end{array}
\]

which are described by

\[
\begin{align*}
C_1(U) &= \ket{0}\bra{0} \otimes I_{2^n} + \ket{1}\bra{1} \otimes U, \\
C_2(U) &= I_{2^n} \otimes \ket{0}\bra{0} + U \otimes \ket{1}\bra{1}, \\
C_1^{-1}(U) &= \ket{0}\bra{0} \otimes U + \ket{1}\bra{1} \otimes I_{2^n}, \quad \text{and} \\
C_2^{-1}(U) &= U \otimes \ket{0}\bra{0} + I_{2^n} \otimes \ket{1}\bra{1},
\end{align*}
\]

respectively, where \( \otimes \) denotes the Kronecker product, \( I_{2^n} \) denotes the \( 2^n \times 2^n \) identity matrix, and \( \ket{\cdot} \) and \( \bra{\cdot} \) is Dirac’s bra-ket notation [11]. The latter two operations are referred to as negative controlled operations. As examples, the CNOT gate is \( C_1(X) \) and the Toffoli gate is \( C_1(1(X)) \). Notice that adjacent controlled operations with the same control preserve matrix multiplication, i.e.

\[
\begin{array}{ccc}
\bullet & | & \bullet \quad \text{and} \quad \bullet & | & \bullet
\end{array}
\]

and gates with opposite control polarities commute, i.e.

\[
\begin{array}{ccc}
\bullet & | & \bullet
\end{array}
\]

Finally, if the same target operation is controlled with both polarities, the target operation does not need be controlled, i.e.

\[
\begin{array}{ccc}
\bullet & \text{and} \quad \bullet
\end{array}
\]

The identities in (20)–(22) hold for \( C_2 \) and \( C_2^{-1} \) analogously. Eq. (21) describes a circuit which performs one of two different target operations depending on the value assigned to the control line. Let us consider the special case

\[
C_1^{-1}(U_1)C_1(U_2) = \ket{0}\bra{0} \otimes U_1 + \ket{1}\bra{1} \otimes U_2
\]

which performs \( U_1 \) or \( U_2 \) if the control line is negative or positive, respectively. We call this matrix a case gate and employ a special gate representation shown on the left-hand side of:

\[
\begin{array}{ccc}
\bullet & | & \bullet \quad \text{and} \quad \bullet & | & \bullet
\end{array}
\]

Another interesting circuit equality applies to the Pauli \( Z \) gate and its roots. A positively controlled gate can be flipped without changing its functionality, i.e.

\[
C_1(\sqrt[Z]{2}) = \sqrt[Z]{2} = C_2(\sqrt[Z]{2}).
\]
Using the translation gates, a circuit identity for the other Pauli roots can be derived:

\[ \sqrt{\sigma_a} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{pmatrix} \]

where \( \rho_{3,3} = I \).

Since \( \sqrt{Z} \otimes I = C_1(e^{i\pi} I) \)

\[ C_1(U)C_1(e^{i\pi} I) = C_1(e^{i\pi} U) = C_1(e^{i\pi} I)C_1(U) \]
a root of the Pauli Z gate can be moved across a positive control, i.e.

\[ \sqrt{Z} = \sqrt{Z} \]

Lastly, in the remainder of the paper we will often make use of a circuit identity given in [12, Rule D7]:

\[ \sqrt{\sigma_a} = \sqrt{\sigma_a} \]

III. MAPPING SCHEMES FOR SYNTHESIS

Many of the mappings for translating a given quantum circuit into a circuit in terms of a restricted gate library in fact reduce the number of controls in a controlled gate, as shown for example in [1]. We follow this approach and will provide generic rules in terms of general Pauli roots.

First, we show how a single controlled Pauli root gate can be mapped to a circuit involving uncontrolled Pauli root gates by doubling the index, i.e. \( \sqrt[2]{\sigma_a} \). We start by proving the following lemma.

**Lemma 1.** The following circuit equality holds for \( a \neq b \):

![Diagram](image)

\[ \sqrt{\sigma_a} = \sqrt{\sigma_a} \]

**Proof.** Based on the special structure of \( \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{pmatrix} \), one obtains

\[ \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{pmatrix} \]

which proves (29).

**Corollary 1.** Analogously to Lemma 4 we can derive

![Diagram](image)

**Theorem 1.** If \( a \neq b \), then the following circuit equality holds:

![Diagram](image)

**Proof.** We have

![Diagram](image)

which concludes the proof.

It follows from the proof that the first gate on the right-hand side of (31) commutes with the other three gates treated as a single gate. Using Theorem 1 we can for example derive the identity shown in [8, Fig. 5(b)] where \( S = \sqrt{Z} \):

![Diagram](image)

Note that a root of the Pauli Z gate can be moved across the control of a CNOT but not over the target.

In [2, Lemma 6.1] the circuit identity of which [1] is a particular case is defined for an arbitrary unitary matrix \( U \) on the left-hand side and its square root on the right-hand side. Given that Lemma and the translation gate equality in [10] we can define a new generic mapping that involves the Pauli roots.

**Theorem 2.** The following circuit equality holds:

![Diagram](image)

where \( \rho_{ab} = I \) if \( a = b \).
Proof. Considering the circuit one gets from Lemma 6.1, the Pauli roots are adjusted by inserting translation gates around each of them. Since the translation gates are Hermitian they are involutory and therefore two translation gates in sequence cancel.

One obtains (1) by setting \(a = b = k = 1\) in (3). Note that the circuit identities in both (31) and (32) can be extended by adding a control to the left-hand side and controls to particular gates on the right-hand side. Repeating this process enables the mapping of multiple controlled gates. As the above illustrates, one needs to double the degree of the root in order to remove one control line when not using ancilla lines. As a result, one needs operations \(x/\sigma_n\) in order to represent an uncontrolled Toffoli gate if the only allowed controlled gate is a CNOT.

By combining (32) with (31) a powerful mapping scheme can be obtained for Toffoli gates that is very flexible due to the number of degrees of freedom:

1. In (31) we can move the first gate to the end of the circuit.
2. In (31) the Pauli Z root can be moved over the controls on the upper line.
3. In (31) the \(C_1(\sigma_b)\) operations can be replaced by \(C_2(\sigma_b)\) operations, if \(b = 3\).
4. In (32) the first two lines can be swapped.
5. In (32) the functionality is preserved if all controlled Pauli roots are replaced by their adjoint.
6. The last controlled Pauli root in (32) can be freely moved within the \(\rho_{ab}\) gates.

By exploiting these options, it is possible to derive a \(T\) gate realization for a Toffoli gate of the same quality as the one shown in (3) starting from the mapping in (1) instead of using expensive search techniques. We will demonstrate this by deriving a realization for a more general gate

\[
\begin{align*}
x_1 & \mapsto f(x_1, x_2) \\
x_2 & \mapsto f(x_1, x_2) \\
x_3 & \mapsto x_3 \oplus f(x_1, x_2)
\end{align*}
\]

(33)
in which the target line \(x_3\) is inverted with respect to a control function \(f\) of a particular type. The Toffoli gate in (3) is the special case where \(f = x_1 \land x_2\). Given a unitary matrix \(U\), in this construction, we will use the notation \(U^p\) for \(p \in \{0, 1\}\) where \(U^0 = U\) and \(U^1 = U^\dagger\).

To begin, applying a modified version of the mapping in (1) yields

Next, the controlled \(S\) gates are replaced according to (31) into circuits with uncontrolled \(T\) gates. We implicitly make use of the commutation property (first item in the list of degrees of freedom) to enable the combination of gates.

A \(T\) gate can move over a control line without changing the function, but not over a target. In order to move CNOT gates over other controlled gates, we are making use of (28). The combined \(T^a \cdot T^b\) gate commutes with the block of three gates to the right to it.

This last step reduces the total depth by one.

By setting \(a = c = 0\) and \(b = 1\) one obtains a circuit that realizes the Toffoli gate and has the same characteristics as the circuit in (3), \(i.e.\), a \(T\)-depth of 3 with a total depth of 10. The more complicated derivation of precisely the same circuit as in (3) is given in the appendix.

Depending on how \(a\), \(b\), and \(c\) are assigned values of 0 and 1, we can realize four different circuits as illustrated in the following table where the first row yields a circuit for the Toffoli gate. Note that each row represents two possible assignments, \(e.g.\) setting \(a = c = 1\) and \(b = 0\) also realizes a Toffoli gate. The two possibilities correspond to performing the required rotations in opposite direction around the Bloch sphere.

| Assignment | Control function |
|------------|------------------|
| \(c = a\) and \(b \neq a\) | \(f(x_1, x_2) = x_1 \land x_2\) |
| \(b = a\) and \(c \neq a\) | \(f(x_1, x_2) = x_1 \land \bar{x}_2\) |
| \(b = c\) and \(a \neq c\) | \(f(x_1, x_2) = \bar{x}_1 \land x_2\) |
| \(a = b = c\) | \(f(x_1, x_2) = x_1 \lor x_2\) |

The last entry in the table above can be used to realize \(x_3 \oplus x_1 \land x_2\) by adding a NOT gate to the right of the right-hand \(H\).

Another example is given in Fig. 3 in which the full adder circuit in (8) Fig. 11] is derived starting from two Peres gates, each shown as a Toffoli gate followed by a CNOT. After the NCV expansion has been applied two controlled \(V\) gates cancel and we can reorder the gates before mapping them to controlled \(S\) gates. After the controlled \(S\) gates have been mapped according to (31), the next steps aim to align the \(T\) gates into as few stages as possible by moving the blocking CNOT gates using (28). In that way a \(T\)-depth of 2 is readily achieved. However, in order to get the same total depth as (8) Fig. 11], one needs to perform the not so obvious
be possible for certain Pauli matrices. However, some optimization techniques may only depend on the underlying Pauli root quantum gate libraries. As a result, the size of the circuit does not and not specific to this example. Moreover, other Pauli matrices obey the commutation relation \( \{\sigma_a, \sigma_b\} = 2\delta_{ab}I \) where \( \delta_{ab} = \langle a|I_3|b \rangle \) is the Kronecker delta. Exploiting these relations, the following equation is obtained
\[
[\sigma_a, \sigma_b] + \{\sigma_a, \sigma_b\} = (\sigma_a\sigma_b - \sigma_b\sigma_a) + (\sigma_a\sigma_b + \sigma_b\sigma_a)
\]
\[
= 2i \sum_{c=1}^3 \varepsilon_{abc}\sigma_c + 2\delta_{ab}I \quad (34)
\]
leading to
\[
\sigma_a\sigma_b = i \sum_{c=1}^3 \varepsilon_{abc}\sigma_c + \delta_{ab}I. \quad (35)
\]
Hence, an equation can be derived that expresses a Pauli root \( \sigma_b \) in terms of a Pauli root \( \sigma_a \) and the square root of \( \sigma_c \) for \( \{a, b, c\} = 3, \ i.e. \)
\[
\sqrt[3]{\sigma_b} = \varepsilon_{abc} \left( \sqrt{\sigma_c} \sqrt[3]{\sigma_a} \sqrt[3]{\sigma_c} \right). \quad (36)
\]
Let
\[
P_n = \{\sigma_{j_1} \otimes \cdots \otimes \sigma_{j_n} \mid \{j_1, \ldots, j_n\} \subseteq \{0, 1, 2, 3\}\} \quad (37)
\]
be the Pauli group where \( \sigma_0 = I \), i.e. the set of all \( n \)-fold tensor products of the identity and the Pauli matrices [13]. Following the definition in [8], the Clifford group \( C_n \) is the normalizer [13] of \( P_n \) in the group of all unitary \( 2^n \times 2^n \) matrices \( U(2^n) \), i.e.
\[
C_n = \{U \in U(2^n) \mid \forall P \in P_n : UPU^\dagger \in P_n\}. \quad (38)
\]
The Clifford group can be generated by \( C_1(X), S = \sqrt{Z}, \) and \( H \) [13]. Other matrices can then be derived e.g.
\[
Z = SS, \ X = HSSH, \ Y \quad (39)
\]
\[
SX^3S^\dagger = SHSSHSSS. \quad (39)
\]
\[
FIG. 1. Derivation of the full adder
\]

IV. CLIFFORD GROUPS

The Clifford group is usually defined over \( C_1(X), S, \) and \( H \) in the literature. In this section, we investigate how the Pauli roots relate to Clifford groups and can be utilized in order to express Clifford groups in a general manner. We first derive some useful equalities involving Pauli matrices and their roots.

The Pauli matrices form an orthogonal basis for the real Hilbert space and therefore one Pauli matrix cannot be expressed as a linear combination of the others. However, the Pauli matrices obey the commutation relation
\[
[\sigma_a, \sigma_b] = 2i \sum_{c=1}^3 \varepsilon_{abc}\sigma_c \quad \text{where} \quad \varepsilon_{abc} = \frac{(a-b)(b-c)(c-a)}{2}
\]
is the Levi-Civita symbol and the anticommutation relation \( \{\sigma_a, \sigma_b\} = 2\delta_{ab}I \) where \( \delta_{ab} = \langle a|I_3|b \rangle \) is the Kronecker delta. Exploiting these relations, the following equation is obtained
\[
\sigma_a\sigma_b = i \sum_{c=1}^3 \varepsilon_{abc}\sigma_c + \delta_{ab}I. \quad (35)
\]
where \( W = \sqrt{X} \). This translation is applicable in general and not specific to this example. Moreover, other Pauli roots can be used together with their respective translation matrices. As a result, the size of the circuit does not depend on the underlying Pauli root quantum gate library. However, some optimization techniques may only be possible for certain Pauli matrices.
Other choices are possible. For example, the Clifford group can be generated by e.g. $C_1(X)$, $V$, and $H$, since $S = HVH$.

**Theorem 3.** The gates $C_1(\sigma_a)$, $\sqrt{\sigma_a}$, and $\rho_{ab}$ generate the Clifford group where $a \neq b$.

**Proof.** According to \[13\] one can obtain a second Pauli square root $\sqrt{\sigma_b}$ from $\sqrt{\sigma_a}$ by multiplying it with $\rho_{ab}$ on both sides. The third square root can then be obtained from \[36\].

**Corollary 2.** The gates $C_1(\sigma_a)$, $\sqrt{\sigma_b}$, and $\rho_{ab}$ generate the Clifford group.

Based on the results of this section, circuits built using gates from the Clifford+$T$ library can also be expressed using a more general Clifford group together with a fourth root of a corresponding Pauli matrix.

## V. NEGATOR OPERATIONS

In \[14\] the authors introduce a related gate called the **NEGATOR** $N(\theta)$ which they obtain by replacing $\frac{\pi}{2k}$ by an angle $\theta$ in the matrix representation for the root of Pauli $X$, i.e.

$$N(\theta) = I + i \sin \theta + e^{i\theta} (I - X).$$

This gate can be generalized for all Pauli matrices as

$$N_a(\theta) = I + i \sin \theta + e^{i\theta} (I - \sigma_a),$$

and it can be proven that the matrix is obtained from the corresponding Pauli root by replacing $\frac{\pi}{2k}$ with $\theta$.

**Theorem 4.** It holds, that

$$N_a\left(\frac{\theta}{2}\right) = e^{\frac{i\theta}{2}} R_a(\theta).$$

**Note that** $\sqrt{\sigma_a} = e^{\frac{i\theta}{2}} R_a\left(\frac{\pi}{8}\right)$.

**Proof.** We rewrite (42) to $R_a(\theta) = e^{-\frac{i\theta}{2}} N_a\left(\frac{\theta}{2}\right)$ and after expanding $N_a\left(\frac{\theta}{2}\right)$ we have

$$e^{-\frac{i\theta}{2}} \left( I + i \sin \frac{\theta}{2} e^{\frac{i\theta}{2}} (I - \sigma_a) \right)$$

$$= e^{-\frac{i\theta}{2}} I + i \sin \frac{\theta}{2} I - i \sin \frac{\theta}{2} \sigma_a = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \sigma_a = R_a(\theta)$$

which concludes the proof.

**Corollary 3.** From Theorem \[3\] it can be seen that $C_1(\sigma_a)$, $N_a(\theta)$, and $\rho_{ab}$ generate the Clifford group.

## VI. CONCLUDING REMARKS

This paper has examined relationships between the Pauli matrices and their roots with an emphasis on circuit synthesis and optimization. We have presented a number of useful identities and techniques. We have also shown that by applying those methods it is possible to systematically derive circuits employing Clifford+$T$ gates from NCV circuits without using expensive exhaustive search techniques. Our ongoing work is to incorporate the techniques discussed in this paper into a synthesis algorithm opening the way to the synthesis and optimization of larger Clifford+$T$ gate circuits.

---

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[2] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A 52, 3457 (1995).

[3] W. N. N. Hung, X. Song, G. Yang, J. Yang, and M. A. Perkowski, IEEE Trans. on CAD 25, 1652 (2006).

[4] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, Multiple-Valued Logic and Soft Computing 15, 283 (2009).

[5] Z. Sasanian and D. M. Miller, Multiple-Valued Logic and Soft Computing 18, 83 (2012).

[6] H. Buhrman, R. Cleve, M. Laurent, N. Linden, A. Schrijver, and F. Unger, in *Foundations of Computer Science*, Vol. 27 (IEEE Computer Society, 2006) pp. 411–419.

[7] A. Bocharov and K. M. Store, Phys. Rev. Lett. 109, 100501 (2012).

[8] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, IEEE Trans. on CAD 32, 818 (2013).

[9] P. Selinger, Phys. Rev. A 87, 042302 (2013).

[10] E. U. Condon and P. M. Morse, *Quantum Mechanics* (McGraw-Hill, 1929).

[11] P. A. M. Dirac, Math. Proc. of the Cambridge Phil. Soc. 35, 416 (1939).

[12] M. Soeken and M. K. Thomsen, in *Reversible Computation*, Lecture Notes in Computer Science, Vol. 7948 (Springer, 2013) pp. 196–208.

[13] D. Gottesman, arXiv:quant-ph/9807006v1.

[14] A. De Vos and S. De Baerdemacker, Open Syst. & Inform. Dynam. 20, 1350004 (2013).
Appendix A: Deriving the circuit in (3)

Before we derive the circuit, we show one further identity:

\[ T^\dagger T T^\dagger T = S S \quad \text{(A1)} \]

We start as before by mapping the controlled V and \( V^\dagger \) gates to \( S \) and \( S^\dagger \) gates as follows.

\[ \text{(25)} \]

Replacing the controlled \( S \) and \( S^\dagger \) gates by cascades consisting of uncontrolled \( T \) and \( T^\dagger \) gates yields

\[ \text{(31)} \]

After moving \( T \) gates on the first and second line towards the left and canceling the \( T \) with the \( T^\dagger \) gate on the third line we have

\[ \text{(32)} \]

In this circuit we move a CNOT to the left by applying (28) resulting in

\[ \text{(34)} \]

After moving the selected gates to the left the circuit is as follows:

\[ \text{(35)} \]