# Quantum Circuits for Functionally Controlled NOT Gates

Mathias Soeken  Martin Roetteler
Microsoft Quantum, Redmond, United States

***Abstract*—We generalize quantum circuits for the Toffoli gate presented by Selinger [1] and Jones [2] for functionally controlled NOT gates, i.e., $X$ gates controlled by arbitrary $n$-variable Boolean functions. Our constructions target the gate set consisting of Clifford gates and single qubit rotations by arbitrary angles. Our constructions use the Walsh-Hadamard spectrum of Boolean functions and build on the work by Schuch and Siewert [3] and Welch et al [4]. We present quantum circuits for the case where the target qubit is in an arbitrary state as well as the special case where the target is in a known state. Additionally, we present constructions that require no auxiliary qubits and constructions that have a rotation depth of 1.**

## I. Background and Motivation

Finding quantum circuit implementations of subroutines that are given as classical functions is a problem that occurs in many contexts. Examples includes Shor's algorithm for factoring and dlogs [5], Grover's search algorithm [6], quantum walk algorithms [7], the Harrow-Hassidim-Lloyd algorithm for solving linear equations [8], [9], and quantum simulation methods [10], [11]. In this paper, we consider the case where the classical functions are provided as a Boolean function, i.e., we are interested in finding quantum circuits that implement the unitary

$$U_f : |x\rangle|y\rangle|0^\ell\rangle \mapsto |x\rangle|y \oplus f(x)\rangle|0^\ell\rangle, \tag{1}$$

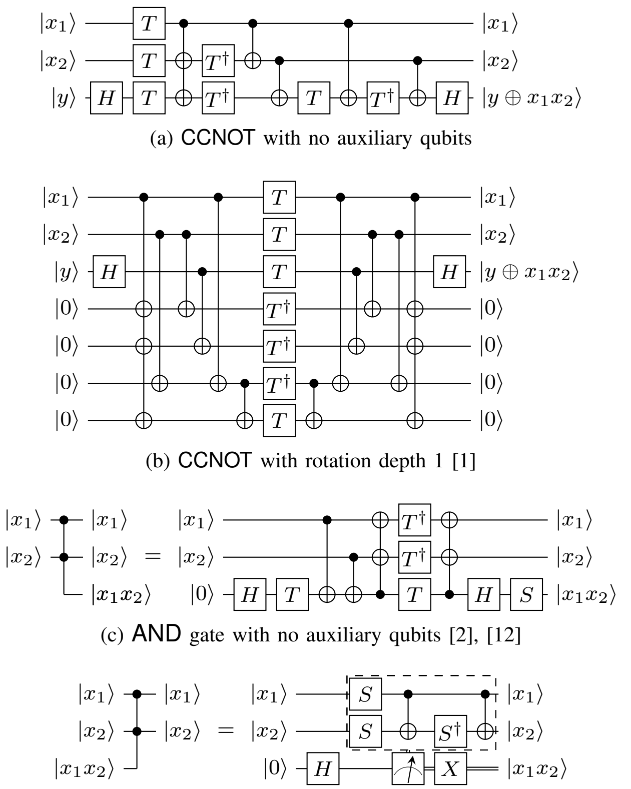
where $f(x)$ is a Boolean function over $n$ variables $x = x_1, \dots, x_n$.

As the target gate set we consider the universal gate set Clifford+$R_1$, which is generated by the $\mathsf{CNOT}$ gate, the Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right)$, and unitaries $R_1(\frac{k\pi}{2^j}) = \mathrm{diag}(1, e^{\mathrm{i}k\pi 2^{-j}})$ as well as their adjoints for arbitrary non-negative integers $j$ and $k$. We also allow classical control based on intermediate measurement outcomes. The well-known Clifford+$T$ gate library is a special case in which $j = 2$, since $T = R_1(\frac{\pi}{4})$ and $T^\dagger = R_1^\dagger(\frac{\pi}{4}) = R_1(-\frac{\pi}{4})$. A *rotation stage* is a set of $R_1$ gates in a circuit that can be executed in parallel, and the *rotation depth* of a circuit is the smallest number of rotation stages in it. Note that for some angles, an $R_1$ gate is a Clifford gate, e.g., $R_1(\frac{\pi}{2}) = S$ and $R_1(\pi) = Z$; such gates are ignored in the depth computation.

Our main result are six different constructions for the implementation of $f$ , depending on different contexts of the target qubit as well as trade-offs between circuit depth and number of qubits. Specifically, we distinguish the cases in which $|y\rangle$ is an arbitrary quantum state, the case in which $|y\rangle = |0\rangle$, and the case in which $|y\rangle = |f(x)\rangle$. For each of these three cases we show one construction where $\ell = 0$, i.e., no auxiliary qubits are required, but the circuits have rotation depth $O(2^n)$, and we show one construction where $\ell = O(2^n)$, but the resulting circuits have rotation depth 1.

Several previous works have presented constructions for $U_f$, in particular for the special case in which $f(x_1, x_2) = x_1 \land x_2$, for which $U_f$ is also referred to as Toffoli gate. We refer to $U_f$ in this case as $\mathsf{CCNOT}$. For instance, Selinger has shown a construction for $\mathsf{CCNOT}$ with rotation depth 1 and $\ell = 4$ [1], depicted in Fig. 1(a). Moreover, Jones has shown that fewer rotation gates are required to implement $\mathsf{CCNOT}$ when the target is in state $|0\rangle$ or $|f(x)\rangle$ [2]; we refer to the operation in these cases as $\mathsf{AND}$ and $\mathsf{AND}^\dagger$, respectively. The quantum circuits are shown in Figs. 1(c) and (d) based on the constructions by Gidney [12]; also using the diagrammatic notation for the $\mathsf{AND}$ and $\mathsf{AND}^\dagger$ gates proposed in that reference. A measurement operation is required in the latter case. The constructions for these two cases can be leveraged to realize a $\mathsf{CCNOT}$ with an arbitrary target state and $\ell = 1$ [2]. Schuch and Siewert presented a constructive algorithm to find quantum circuits that perform arbitrary controlled phase-shift operations $U_{\vec{\theta}} : |x\rangle \mapsto e^{-\mathrm{i}\theta_x}|x\rangle$ on $n$ qubits using only $\mathsf{CNOT}$ and $R_z$ gates, where $\vec{\theta}$ is a $2^n$-element column vector of real rotation angles [3]. This operation generalizes the functionally controlled $Z$ gate, for which all values in $\vec{\theta}$ are either 0 or $\pi$. Constructions for the functionally controlled $Z$ gate can be used to implement functionally controlled $\mathsf{NOT}$ gates by surrounding the target qubit with $H$ gates. The authors use the Hadamard transform to translate the input vector $\vec{\theta}$ into rotation angles for the $R_z$ gates in the circuit. The construction has been rediscovered by Welch et al. [4], and improved by using less $\mathsf{CNOT}$ gates by exploiting Gray codes. An explicit construction for functionally controlled $\mathsf{NOT}$ gates is described in [13]. Constructions that exploit the fact that the target qubit is in state $|0\rangle$ or $|f(x)\rangle$ and aim at finding a good trade-off between number of Toffoli gates and the number of auxiliary qubits $\ell$ is presented in [14].

We follow the usual convention of using quantum circuits as computational models to describe quantum computations [15], e.g., as already used in Fig. 1. We also use a textual description for quantum circuits. Assuming that qubits are indexed by a distinct set of integers, we write $U_i$ to mean that a single qubit unitary is applied to the qubit with index $i$, we write $\mathsf{CNOT}_{i,j}$ when a $\mathsf{CNOT}$ gate is applied with control qubit $i$ and target qubit $j$, and we use '$\circ$' for sequential composition of unitaries



(a) CCNOT with no auxiliary qubits

(b) CCNOT with rotation depth 1 [1]

(c) AND gate with no auxiliary qubits [2], [12]

(d) AND$^\dagger$ gate with no auxiliary qubits [2], [12]

Fig. 1. Clifford+$T$ implementations for the CCNOT gate.

(note that sequential composition is read from left to right, as in the diagrammatic notation, and not from right to left as in matrix multiplication). We use $C^\dagger$ to denote the reverse circuit of $C$, i.e., the reverse sequence where each unitary is replaced by its adjoint. Finally, we use the notation $[C]_i$ to mean that circuit $C$ is only performed if the measurement outcome of qubit $i$ is 1 when measured in the $Z$ basis. Note that this formalism does not explicitly describe which computations are performed in parallel, but we assume that gates are parallelized in a way such that the rotation depth is minimized. As an example, the circuit in Fig. 1(d) can be described as $H_2 \circ [S_0 \circ S_1 \circ \mathsf{CNOT}_{0,1} \circ S_1^\dagger \circ \mathsf{CNOT}_{0,1} \circ X_2]_2$, assuming qubit indexes $0, 1$, and $2$ for the qubits from top to bottom.

*Other notation used in the paper.:* For a bitstring $x = x_1 \dots x_n$, let $\mu(x) = \sum_{i=1}^n x_i$ be the sideways sum of $x$, also called Hamming weight. Further, let $\rho x$ be the ruler function of $x$, which is the largest integer $k$ such that $2^k \mid x$, where $x \neq 0$. Furthermore, we define $\rho 0 = \infty$. For two bitstrings $x$ and $y$ of the same length, let $x \oplus y$ be the bitwise XOR operation. When convenient, we write $\bar{1}$ in place of $-1$.

## II. General Case: Arbitrary Target Quantum State

The proposed constructions make use of Gray codes and the Hadamard transform, which we review in the beginning of this section.

*Gray codes.:* Let $v_0, v_1, \dots, v_{2^n-1}$ be a cyclic binary Gray code that traverses all $n$-bit strings, i.e., $\mu(v_k \oplus v_{(k+1) \bmod 2^n}) = 1$ and $v_k \neq v_l$ when $k \neq l$. In other words, the sequence forms a Hamiltonian cycle on the $n$-dimensional hypercube. Without loss of generality, we further assume that $v_0 = 0 \dots 0$. With this start value, the sequence is uniquely determined by integers $\delta_0, \delta_1, \dots, \delta_{2^n-1}$, such that $v_{(k+1) \bmod 2^n} = v_k \oplus 2^{\delta_k}$. Note that for the standard Gray binary code, we have $\delta_k = \rho(k+1)$ for $0 \le k < 2^n - 1$ and $\delta_{2^n-1} = n - 1$.

**Example 1.** *For $n = 2$, a cyclic binary Gray code is $v_0 = 00, v_1 = 01, v_2 = 11, v_3 = 10$, or alternatively expressed as $\delta_0 = 0, \delta_1 = 1, \delta_2 = 0, \delta_3 = 1$.*

*Hadamard transform.:* The Hadamard transform $H_n$ on $n$ variables is a $2^n \times 2^n$ integral matrix which can be recursively defined as $H_n = H_1 \otimes H_{n-1}$, where $H_1 = \left(\begin{smallmatrix} 1 & 1 \\ 1 & \bar{1} \end{smallmatrix}\right)$. For an $n$-variable Boolean function $f(x_1, \dots, x_n)$ we define the $2^n$-element column vector

$$\hat{f} = ((-1)^{f(0,\dots,0,0)}, (-1)^{f(0,\dots,0,1)}, \dots, \\ (-1)^{f(1,\dots,1,0)}, (-1)^{f(1,\dots,1,1)})^T, \quad (2)$$

which contains all truth values of $f$, where 0 and 1 are encoded as 1 and $-1$, respectively. The vector $\hat{f}$ is also called the (Walsh-)Hadamard spectrum of $f$ [16]. Multiplying $H_n\hat{f}$ results in a column vector $s = (s_0, \dots, s_{2^n-1})^T$ whose elements are called the spectral coefficients of $f$.

**Example 2.** *For $f(x_1, x_2) = x_1x_2$, we have $\hat{f} = (1, 1, 1, -1)^T$ and*

$$H_2\hat{f} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \bar{1} & 1 & \bar{1} \\ 1 & 1 & \bar{1} & \bar{1} \\ 1 & \bar{1} & \bar{1} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ \bar{1} \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \\ -2 \end{pmatrix}.$$

**Construction 1** (General case, no aux. qubits)**.** *Let $f$ be an $n$-variable Boolean function and let $s = (s_0, \dots, s_{2^n-1})^T$ be its spectral coefficients. Assuming that the control qubit for variable $x_i$ has index $i - 1$ and the target qubit has index $n$, the circuit*

$$H_n \circ S_n \circ \bigcirc_{i=0}^{n-1} C_i \circ C \circ H_n \quad (3)$$

*where*

$$C_i = \bigcirc_{k=0}^{2^i-1} (R_1(\theta_{2^i+v_k})_i \circ \mathsf{CNOT}_{\delta_k,i})$$

*and*

$$C = \bigcirc_{k=0}^{2^n-1} \left(R_1^\dagger(\theta_{v_k})_n \circ \mathsf{CNOT}_{\delta_k,n}\right)$$

*implements $U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$ without any auxiliary qubits, where $\theta_j = \frac{s_j\pi}{2^{n+1}}$ and $\mathsf{CNOT}_{-1,0} = I_0$ (this case occurs once in subcircuit $C_0$ when $k = 0$.)*

For each value $1 \le k < 2^n$, with binary expansion $k = (b_nb_{n-1} \dots b_1)_2$, we apply the gate $R_1(\theta_k)$. This phase must be applied to a computational state corresponding to $x_1^{b_1} \oplus \dots \oplus x_n^{b_n}$, which can be constructed using CNOT

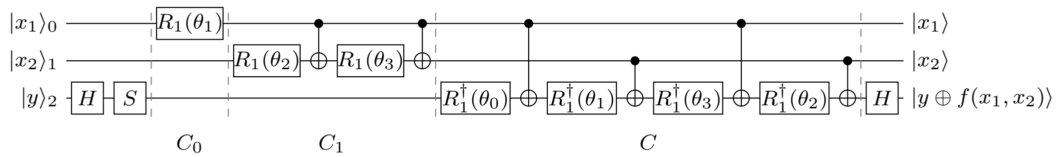


Fig. 2. Example circuit for Construction 1 where $n = 2$. The subscripts on the input qubit label indicate the qubit index.

gates [17], [3], [18], [19]. In order to reduce the number of required CNOT gates, the subcircuit $C_i$ applies them using a cyclic binary Gray code [4], for all values $k$ in which the leading 1 is at position $b_{i+1}$, i.e., $2^i \leq k < 2^{i+1}$. Since the Gray code is cyclic, the computational state of the qubits remains unchanged and corresponds to the input qubits after each application of subcircuit $C_i$. The last subcircuit $C$ prepares all computational states corresponding to all linear combinations of the input qubits together with the target qubit. In this case, the adjoint of the corresponding rotation gate is applied. Why these rotation angles correspond to the spectral coefficients of $f$ is explained and proven in Appendix A.

**Example 3.** *Fig. 2 shows the circuit from Construction 1 when* $n = 2$.

Note that the $S$ gate may be merged with the $R_1^\dagger(\theta_0)$ gate as $R_1^\dagger(\theta_0 + \frac{\pi}{2})$. When $f = x_1 x_2$ and when moving all gates as far as left as possible, one obtains the circuit in Fig. 1(a).

**Construction 2** (General case, depth 1)**.** *Let* $f(x_1, \dots, x_n)$ *be a Boolean function and let* $s = (s_0, s_1, \dots, s_{2^n-1})^T$ *be its spectral coefficients. The circuit we construct has* $2^{n+1} - 1$ *qubits indexed from* 1 *to* $2^{n+1} - 1$. *Assuming that the control qubit for variable* $x_i$ *has index* $2^{i-1}$, *that the target qubit has index* $2^n$, *and that all other indexes are assigned to the auxiliary qubits, the circuit*

$$H_{2^n} \circ S_{2^n} \circ C_1 \circ C_2 \circ R \circ C_2^\dagger \circ C_1^\dagger \circ H_{2^n}, \tag{4}$$

*where*

$$C_1 = \bigcirc_{\substack{3 \leq k < 2^{n+1} \\ \mu k \neq 1}} \mathit{CNOT}_{\rho k, k}, \; C_2 = \bigcirc_{\substack{3 \leq k < 2^{n+1} \\ \mu k \neq 1}} \mathit{CNOT}_{k - \rho k, k}$$

*and*

$$R = \bigcirc_{k=1}^{2^n-1} R_1(\theta_k)_k \circ \bigcirc_{k=0}^{2^n-1} R_1^\dagger(\theta_k)_{2^n+k}$$

*implements* $U_f : |x\rangle|y\rangle|0^\ell\rangle \mapsto |x\rangle|y \oplus f(x)\rangle|0^\ell\rangle$, *where* $\ell = 2^{n+1} - n - 2$ *and* $\theta_k = \frac{s_k \pi}{2^{n+1}}$. *Note that all rotation gates in* $R$ *can be executed in parallel, since they are all applied to different qubits.*

The auxiliary qubit with index $k = (b_{n+1} \dots b_1)_2$ is used to prepare the computational state of the linear combination $x_1^{b_1} \oplus \dots \oplus x_n^{b_n} \oplus y^{b_{n+1}}$. Subcircuit $C_1$ initializes the auxiliary qubits with $\rho k$, i.e., the trailing 1 in $k$ and subcircuit $C_2$ copies over the remaining bits using $k - \rho k$. The order guarantees that this is always possible.

**Example 4.** *Fig. 3 shows the circuit from Construction 2 when* $n = 2$.

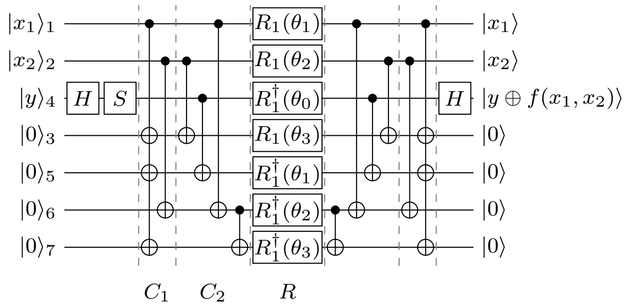


Fig. 3. Example circuit for Construction 2 where $n = 2$. The subscripts on the input qubit label indicate the qubit index.

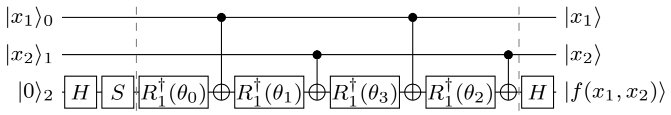


Fig. 4. Example circuit for Construction 3 where $n = 2$. The subscripts on the input qubit label indicate the qubit index.

As in Construction 1, the $S$ gate may be merged with the $R_1^\dagger(\theta_0)$ gate as $R_1^\dagger(\theta_0 + \frac{\pi}{2})$. When $f = x_1 x_2$ and after merging the $S$ gate, one obtains the circuit in Fig. 1(b).

## III. Special Case: Target Quantum State is $|0\rangle$ or $|f(x)\rangle$

In this section, we show constructions for generalizations of AND and AND$^\dagger$, i.e., unitary operations in which the target quantum state is known to be $|0\rangle$ or $|f(x)\rangle$, respectively.

**Construction 3** (Target $|0\rangle$, no aux. qubits)**.** *Let* $f$ *be an* $n$-*variable Boolean function and let* $s = (s_0, \dots, s_{2^n-1})^T$ *be its spectral coefficients. The circuit*

$$H_n \circ S_n \circ \bigcirc_{k=0}^{2^n-1} \left( R_1^\dagger(\theta_{v_k})_n \circ \mathit{CNOT}_{\delta_k, n} \right) \circ H_n \tag{5}$$

*implements* $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ *without any auxiliary qubits, where* $\theta_j = \frac{s_j \pi}{2^{n+1}}$. *Note that this equals* $H_n \circ S_n \circ C \circ H_n$, *where* $C$ *is as in* (3) *from Construction 1.*

If the target qubit is in state $|0\rangle$, we only need to apply $R_1$ gates to computational states that involve the target qubit. Therefore, almost half of the rotation gates can be saved, in fact all subcircuits $C_i$, which are required in Construction 1, can be omitted if the target qubit is known to be in state $|0\rangle$.

**Example 5.** *Fig. 4 shows the circuit from Construction 3 when* $n = 2$.

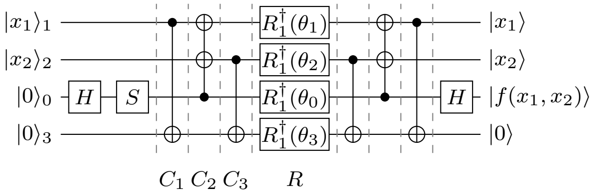


Fig. 5. Example circuit for Construction 4 where $n = 2$. The subscripts on the input qubit label indicate the qubit index.

Note that the circuit in Fig. 3 is quite different from the circuit implementing AND in Fig. 1(c). This is because the circuit in Fig. 1(c) aims at reducing the rotation depth, which is optimal in that case when no auxiliary qubits are used. In order to achieve the better rotation depth, more CNOT gates are required. Similarly, a CCNOT gate with rotation depth 3 is possible, when using more CNOT gates as in Fig. 1(a) [20]. However, requiring to distribute $2^n$ rotation gates over $n+1$ qubits asymptotically still leads to an exponential worst case complexity for the rotation depth.

**Construction 4** (Target $|0\rangle$, depth 1)**.** *Let $f$ be an $n$-variable Boolean function with spectral coefficients $(s_0, \dots, s_{2^n-1})^T$. Assuming that control qubit $x_i$ is indexed $2^{i-1}$ and that the target qubit has index $0$, the circuit*

$$H_0 \circ S_0 \circ C_1 \circ C_3 \circ C_2 \circ R \circ C_2^\dagger \circ C_3^\dagger \circ C_1^\dagger \circ H_0, \qquad (6)$$

*where*

$$C_1 = \bigcirc_{\substack{3 \le k < 2^n \\ \mu k \ne 1}} \mathsf{CNOT}_{\rho k, k},\ C_3 = \bigcirc_{i=0}^{n-1} \mathsf{CNOT}_{0, 2^i},$$

$$C_2 = \bigcirc_{\substack{3 \le k < 2^n \\ \mu k \ne 1}} \mathsf{CNOT}_{k - \rho k, k},\ R = \bigcirc_{k=0}^{2^n-1} R_1^\dagger(\theta_k)_k$$

*implements $U_f : |x\rangle|0\rangle|0^\ell\rangle \mapsto |x\rangle|f(x)\rangle|0^\ell\rangle$, where $\ell = 2^n - n - 1$ and $\theta_j = \frac{s_j \pi}{2^{n+1}}$.*

One possible way to get a circuit with rotation depth 1 is to take the circuit from Construction 2 and remove all $R_1$ gates that are on qubits with indexes smaller than $2^n$. However, this requires unnecessarily many auxiliary qubits, since only roughly half as many rotations are needed as in Construction 2. Subcircuits $C_1$ and $C_2$ play the same role as in Construction 2, and subcircuit $C_3$ ensures to add the target qubit to all the linear combinations of input qubits in the computational states.

**Example 6.** *Fig. 5 shows the circuit from Construction 4 when $n = 2$.*

**Construction 5** (Target $|f(x)\rangle$, no aux. qubits)**.** *Let $f$ be an $n$-variable Boolean function with spectral coefficients $(s_0, \dots, s_{2^n-1})^T$. Assuming qubit index $i-1$ for variable $x_i$ and qubit index $n$ for the target qubit, the circuit*

$$H_2 \circ \left[ \bigcirc_{i=0}^{n-1} C_i \circ X_n \right]_n, \qquad (7)$$

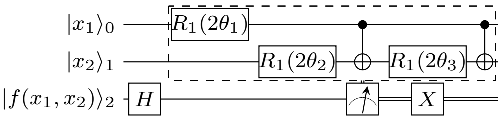


Fig. 6. Example circuit for Construction 5 where $n = 2$. The subscripts on the input qubit label indicate the qubit index.

*where*

$$C_i = \bigcirc_{k=0}^{2^i-1} \left( R_1(2\theta_{2^i + v_k})_i \circ \mathsf{CNOT}_{\delta_k, i} \right).$$

*implements $U_f : |x\rangle|f(x)\rangle \mapsto |x\rangle|0\rangle$ without any auxiliary qubits, where $\theta_j = \frac{s_j \pi}{2^{n+1}}$ and $\mathsf{CNOT}_{-1,0} = I$ (this case occurs once in subcircuit $C_0$ when $k = 0$.)*

While Construction 3 only requires subcircuit $C$ from Construction 1, in this case subcircuit $C$ can be omitted, but the subcircuits $C_i$ are required to perform a phase correction in case of a positive measurement outcome. Note, however, that the angle is doubled in subcircuit $C_i$ compared to Construction 1. A larger rotation angle typically affects the resource costs of an error-corrected rotation gate positively in a fault-tolerant quantum computer.

**Example 7.** *Fig. 6 shows the circuit from Construction 5 when $n = 2$.*

When $f = x_1 x_2$ and when moving all gates as far to the left as possible, one obtains the circuit in Fig. 1(d).

**Construction 6** (Target $|f(x)\rangle$, depth 1)**.** *Let $f$ be an $n$-variable Boolean function with spectral coefficients $(s_0, \dots, s_{2^n-1})^T$. Assuming that control qubit $x_i$ is indexed $2^i$ and that the target qubit has index $0$, the circuit*

$$H_0 \circ \left[ C_1 \circ C_2 \circ \bigcirc_{k=1}^{2^n-1} R_1(2\theta_k)_k \circ C_2^\dagger \circ C_1 \circ X_0 \right], \qquad (8)$$

*where*

$$C_1 = \bigcirc_{\substack{3 \le k < 2^n \\ \mu k \ne 1}} \mathsf{CNOT}_{\rho k, k},\ C_2 = \bigcirc_{\substack{3 \le k < 2^n \\ \mu k \ne 1}} \mathsf{CNOT}_{k - \rho k, k}.$$

*implements $U_f : |x\rangle|f(x)\rangle|0^\ell\rangle \mapsto |x\rangle|0\rangle|0^\ell\rangle$, where $\ell = 2^n - n - 1$ and $\theta_j = \frac{s_j \pi}{2^{n+1}}$.*

The subcircuits $C_1$ and $C_2$ play the same role as in the general case in Construction 2, however, only computational states of linear combinations involving the control lines need to be prepared in order to apply the conditional phase correction in case of a positive measurement result. As in Construction 5 the rotation angles are doubled.

**Example 8.** *Fig. 7 shows the circuit from Construction 6 when $n = 2$.*

## IV. Summary and Conclusions

We have described several constructions to implement a NOT gate controlled by some Boolean function $f(x)$, without auxiliary qubits or with rotation depth 1, both for the general case in which the target qubit is in an arbitrary state and for

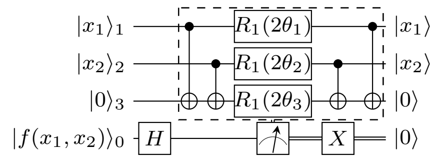


Fig. 7. Example circuit for Construction 6 where $n = 2$. The subscripts on the input qubit label indicate the qubit index.

the special cases in which the target qubit is $|0\rangle$ or $|f(x)\rangle$. Q# [21] implementations are available for Constructions 1, 3, and 5,[1] as well as for Constructions 4 and 6 when $f$ is the $n$-ary AND function.[2]

The presented constructions assume worst case complexity, i.e., all spectral coefficients are nonzero. Spectral coefficients that are 0 correspond to identity gates in the constructions which may enable further gate cancellation or savings in auxiliary qubits. Similarly, some spectral coefficients lead to rotation gates that are Clifford gates, and therefore do not contribute to the rotation depth. It would be interesting to investigate dedicated constructions that take such spectral coefficients into account. Further, an analysis of the distribution of spectral coefficients over all Boolean functions can help to better estimate the average complexity of the constructions.

The constructions have exponential worst case complexity either in the rotation depth, when no auxiliary qubits are allowed, or in the number of qubits, when the rotation depth is 1. Pebbling strategies (see, e.g., [22], [23], [24]) can be used to find tradeoffs in the constructions, e.g., one might fix the number of auxiliary qubits and then minimize for the rotation depth within this limit. Further, quantum circuits might need to obey some layout constraints due to the qubit coupling in the targeted physical quantum computer. Instead of applying additional algorithms as a post-process, the layout information can be an additional parameter to the construction algorithm.

## Appendix

### A. Relationship between spectral coefficients and angles

In this section, we prove the relationship between the spectral coefficients of the control function and the rotation angles that were used in the constructions above. Let $f(x_1,\dots,x_n)$ be an $n$-variable Boolean function, and let $U_f : |x\rangle|x_{n+1}\rangle \mapsto |x\rangle|x_{n+1} \oplus f(x)\rangle$, with $x = x_1,\dots,x_n$. We choose $x_{n+1}$ in place of $y$ for a more convenient notation in the proof. Now let $g(x_1,\dots,x_{n+1}) = x_{n+1} \land f(x_1,\dots,x_n)$ be an $(n+1)$-variable function, and let $\hat{g} = (g_0,\dots,g_{2^{n+1}})^T$ be the function values in $\{1,-1\}$-coding as in (2). We use the following Lemma from [13].

**Lemma 1.** *Given $f$ and $\hat{g}$ as above, we have*

$$U_f = (I_{2^n} \otimes H) \cdot D \cdot (I_{2^n} \otimes H),$$

*where $D = \operatorname{diag}(g_0,\dots,g_{2^{n+1}})$ and $I_{2^n}$ is the $2^n \times 2^n$ identity matrix.*

Using the phase polynomial representation, we can express the action of $D$ as

$$D : |x_1,\dots,x_{n+1}\rangle \mapsto \prod_{k=1}^{2^{n+1}} e^{\mathrm{i}\theta_k p_k(x_1,\dots,x_{n+1})}|x_1,\dots,x_{n_1}\rangle,$$

where $\theta_k = \frac{\pi s'_k}{2^{n+1}}$ and $p_k = x_1^{b_1} \oplus \cdots \oplus x_{n+1}^{b_{n+1}}$ when $k = (b_{n+1}\dots b_1)_2$ [3], [4], [19]. Here $(s'_0,\dots,s'_{2^{n+1}-1})^T$ are the spectral coefficients of $g$.

**Theorem 1.** *Let $s = (s_0,\dots,s_{2^n-1})^T$ be the spectral coefficients of $f$. Then*

$$s'_k = 2^n[k \bmod 2^n = 0] + (-1)^{[k \geq 2^n]} s_{k \bmod 2^n},$$

*for $0 \leq k < 2^{n+1}$, where $[\cdot]$ is the Iverson bracket.*

*Proof.* Let $\hat{f} = (f_0,\dots,f_{2^n-1})^T$ as in (2). Since $g = x_{n+1} \land f$, note that $\hat{g} = (1,\dots,1,f_0,\dots,f_{2^n-1})^T$. Since $s' = H_{n+1}\hat{g}$ and $H_{n+1} = \left(\begin{smallmatrix} H_n & H_n \\ H_n & -H_n \end{smallmatrix}\right)$, the upper $2^n$ entries of $s'$ are $H_n\vec{1} + H_n\hat{f}$ and the lower $2^n$ entries of $s'$ are $H_n\vec{1} - H_n\hat{f}$, where $\vec{1}$ is a $2^n$ column vector in which all entries are 1. The row sums of $H_n$ are 0 for all but the first row, where the row sum is $2^n$. Therefore, $H_n\vec{1} = (2^n, 0, \dots, 0)^T$. □

The lower entries of $s'$ being negative explains the adjoint operation of the rotation gates in the constructions. Further, the offset of $2^n$ in $s'_{2^n} = 2^- s_0$ explains the additional $S$ gate in the constructions.

*Acknowledgements:* We thank Nathan Wiebe, Vadym Kliuchnikov, and Thomas Häner for discussions and valuable feedback.

## References

[1] P. Selinger, "Quantum circuits of $T$-depth one," *Physical Review A*, vol. 87, p. 042302, 2013.

[2] N. C. Jones, "Low-overhead constructions for the fault-tolerant Toffoli gate," *Physical Review A*, vol. 87, no. 2, p. 022328, 2013.

[3] N. Schuch and J. Siewert, "Programmable networks for quantum algorithms," *Physical Review Letters*, vol. 91, no. 027902, 2003.

[4] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, "Efficient quantum circuits for diagonal unitaries without ancillas," *New Journal of Physics*, vol. 16, no. 033040, pp. 1–15, 2014.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Symposium on Theory and Computing*, 1996, pp. 212–219.

[7] J. Kempe, "Quantum random walks - an introductory overview," *Contemporary Physics*, vol. 44, no. 4, pp. 307–327, 2003.

[8] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, p. 150502, 2009.

[9] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, "Preconditioned quantum linear system algorithm," *Physical Review Letters*, vol. 110, no. 25, p. 250504, 2013.

[10] D. W. Berry, A. M. Childs, and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *Foundations of Computer Science*, 2015, pp. 792–809.

[11] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, "Exponential improvement in precision for simulating sparse Hamiltonians," in *Symposium on Theory and Computing*, 2014, pp. 283–292.

[1] see github.com/microsoft/quantum, sample *oracle-synthesis*

[2] see github.com/microsoft/quantumlibraries, library functions `ApplyAnd` and `ApplyLowDepthAnd`

[12] C. Gidney, "Halving the cost of quantum addition," *Quantum*, vol. 2, p. 74, 2018.

[13] M. Soeken, F. Mozafari, B. Schmitt, and G. De Micheli, "Compiling permutations for superconducting QPUs," in *Design, Automation and Test in Europe*, 2019, pp. 1349–1354.

[14] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, "Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization," *Quantum*, vol. 3, p. 208, 2019.

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

[16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[17] C. M. Dawson, H. L. Haselgrove, A. P. Hines, D. Mortimer, M. A. Nielsen, and T. J. Osborne, "Quantum computing and polynomial equations over the finite field $\mathbb{Z}_2$," *Quantum Information and Computation*, vol. 5, no. 2, pp. 102–112, 2005, arXiv preprint arXiv:quant-ph/0408129.

[18] M. Amy, D. Maslov, and M. Mosca, "Polynomial-time $T$-depth optimization of Clifford+$T$ circuits via matroid partitioning," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 33, no. 10, pp. 1476–1489, 2014.

[19] M. Amy and M. Mosca, "T-count optimization and Reed-Muller codes," *IEEE Trans. on Information Theory*, vol. 65, no. 8, pp. 4771–4784, 2019, arXiv preprint arXiv:1601.07363.

[20] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 32, no. 6, pp. 818–830, 2013.

[21] K. Svore, A. Geller, M. Troyer, J. Azariah, C. Granade, B. Heim, V. Kliuchnikov, M. Mykhailova, A. Paz, and M. Roetteler, "Q#: Enabling scalable quantum computing and development with a high-level DSL," in *Real World Domain Specific Languages Workshop*, 2018, pp. 7:1–7:10.

[22] C. H. Bennett, "Time/space trade-offs for reversible computation," *SIAM Journal on Computing*, vol. 18, no. 4, pp. 766–776, 1989.

[23] R. Královic, "Time and space complexity of reversible pebbling," in *Conf. on Current Trends in Theory and Practice of Informatics*, 2001, pp. 292–303.

[24] G. Meuli, M. Soeken, M. Roetteler, N. Bjørner, and G. De Micheli, "Reversible pebbling game for quantum memory management," in *Design, Automation and Test in Europe*, 2019, pp. 288–291.