3-qubit Toffoli gate implementation with 4 CNOT gates

Samir Lipovaca

slipovaca@aol.com

Abstract

This paper presents a small advancement on Problem 4.4b from the Nielsen and Chuang textbook, demonstrating the construction of the Toffoli gate using single-qubit gates and four CNOT gates.

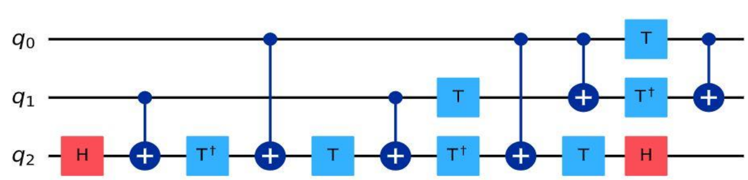
Introduction

Theorem 1 in reference [2] states that a circuit consisting of CNOT gates and one-qubit gates that implements an n-qubit Toffoli gate without ancillae requires at least 2n CNOT gates. For n=3, that would mean at least 6 CNOT gates. Problem 4.4b from the Nielsen and Chuang textbook asks about the minimal Toffoli gate construction. As per reference [3], there are at least two different ways to construct the 3-qubit Toffoli gate out of one-qubit gates and CNOT gates. One construction requires 8 CNOT gates while a more efficient construction requires 6 CNOT Gates. The current IBM qiskit implementation of the 3-qubit Toffoli gate [4] uses 6 CNOT gates.

2. A Toffoli, or controlled-controlled-not gate (`ccx`), is a three-qubit gate. Given that our basis gate set includes only single- and two-qubit gates, this operation must be decomposed. However, it is quite costly:

```
qc = QuantumCircuit(3)
qc.ccx(0, 1, 2)
qc.decompose().draw("mpl")
```

Output:



For every Toffoli gate in a quantum circuit, the hardware may execute up to six CNOT gates and a handful of single-qubit gates. This example demonstrates that any algorithm making use of multiple Toffoli gates will end up as a circuit with large depth and will therefore be appreciably affected by noise.

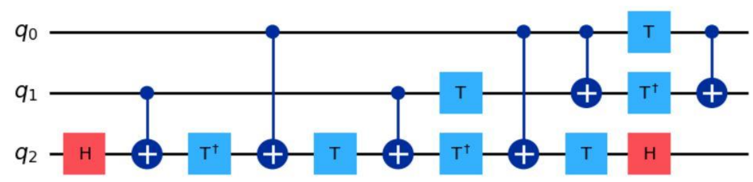
- The current qiskit (2.0.0) implementation of the Toffoli gate (with 6 CNOT gates)

In [3]:

```
%matplotlib
from qiskit import QuantumCircuit
qc = QuantumCircuit(3)
qc.ccx(0, 1, 2)
qc.decompose().draw("mpl")
```

```
Using matplotlib backend: tkagg
```

Out[3]:



As a counterexample to the Theorem 1 in reference [2], in this paper we present a small variation on this 6 CNOT gates construction of the 3-qubit Toffoli gate that uses only 4 CNOT gates.

3-qubit Toffoli gate implementation with 4 CNOT gates

Figure 1 displays the 3-qubit Toffoli gate implementation consisting of 4 CNOT gates and one-qubit gates.

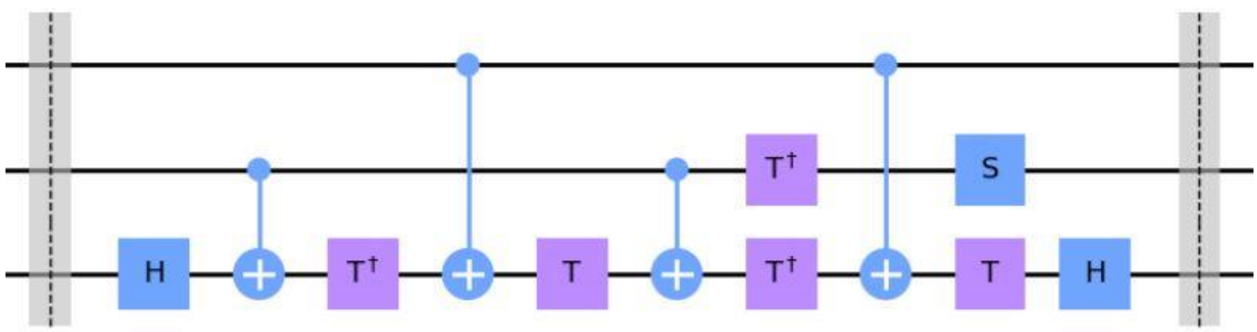


Figure 1

A thorough mathematical analysis of this circuit, confirming the Toffoli gate truth table, is included in this notebook. Figure 2 displays the final 3-qubit output state.

$$\begin{pmatrix} INPUT-STATE & OUTPUT-STATE \\ |000> & |000> \\ |100> & |100> \\ |010> & e^{i\pi/4}|010> \\ |001> & |001> \\ |110> & e^{i\pi/4}|110> \\ |101> & |101> \\ |011> & -e^{i3\pi/4}|111> \\ |111> & -e^{i3\pi/4}|011> \end{pmatrix}$$

Qubits

$$q_0, q_1$$

are control qubits while

$$q_2$$

is the target qubit.

The final output state of the 3-qubits is

$$\Psi_{OUT} = (1/\sqrt{2})^3(|000> +|100> +e^{i\pi/4}|010> +|001> +e^{i\pi/4}|110> +|101> -e^{i3\pi/4}|011> -e^{i3\pi/4}|111>)$$

Figure 2

In the ToffoliGateExploration_qiskit_0_18_1.ipynb notebook, we successfully execute 3-qubit Toffoli gate with 4 CNOT gates on the qasm_simulator.

Figure 3

The ToffoliGateExploration_qiskit_2_0_0.ipynb notebook proves the implementation can be executed on a real IBM quantum computer (ibm_torino). Interestingly, respective transpiled circuit uses control Z gates.

In [5]:
```
# Transpile the ideal circuit to a circuit that can be
# directly executed by the backend
transpiled_circuit = transpile(circuit, backend)
transpiled_circuit.draw('mpl', style="iqp")
```

Out[5]:

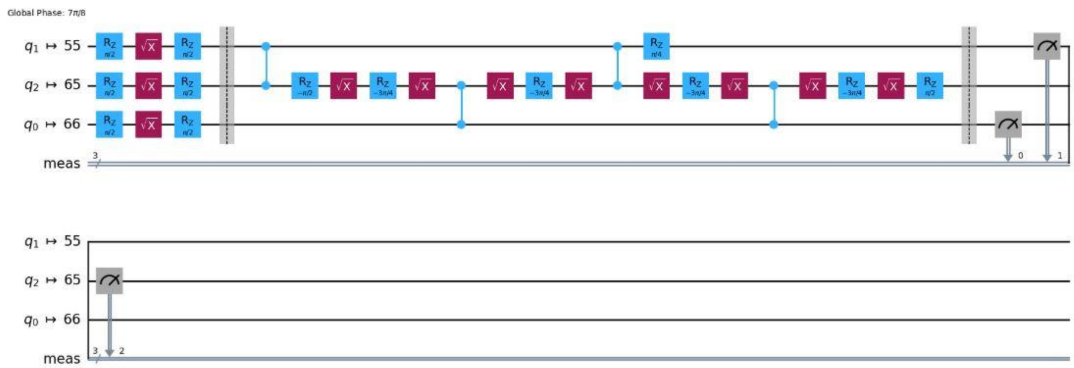
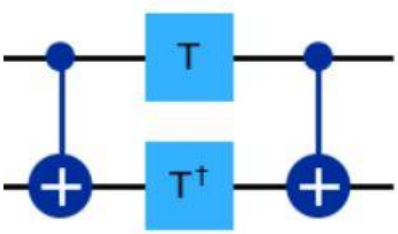


Figure 4

Discussion and Conclusions

In this short paper, we present a variation of the standard implementation of the 3-qubit Toffoli gate, which typically uses 6 CNOT gates. Our implementation uses only 4 CNOT gates because a below portion of the quantum circuit in the standard implementation does not alter the states of the respective qubits and can be removed.



The phases generated by T and T dagger cancel each other out, and two consecutive CNOT gates do not effect any change in the qubit states.

It appears that we could replace the T dagger S gates combination for the middle qubit with just a T gate. This change would lead to a quantum circuit that differs from the standard implementation by omitting the aforementioned portion of the quantum circuit. The transpiled version of our circuit, executed on a real IBM quantum computer (ibm_torino), uses 4 control Z gates, indicating that we have made some progress on Problem 4.4c from the Nielsen and Chuang textbook as well. Further research is needed to determine if additional reductions in the number of required CNOT gates are possible and also if our 4 CNOT gates implementation is optimal.

References

[1] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information,

Cambridge University Press (2000).

[2] [0803.2316] On the CNOT-cost of TOFFOLI gates

[3] Mermin ND. Frontmatter. In: *Quantum Computer Science: An Introduction*. Cambridge University Press; 2007:i-vi.

[4] https://docs.quantum.ibm.com/guides/transpiler-stages#translation-stage

All notebooks can be downloaded from this repository: ToffoliGateExploration/README.md at main · samlip-blip/ToffoliGateExploration