

Resource estimation and verification in Q#

Mathias Soeken
October 13, 2020

Majority-of-three $f = \langle x_1 x_2 x_3 \rangle = [x_1 + x_2 + x_3 \geq 2]$

Task: Implement $Q\#$ operation that maps $|x_1 x_2 x_3\rangle |y\rangle \mapsto |x_1 x_2 x_3\rangle |y \oplus f\rangle$

Truth table

i	x_3	x_2	x_1	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

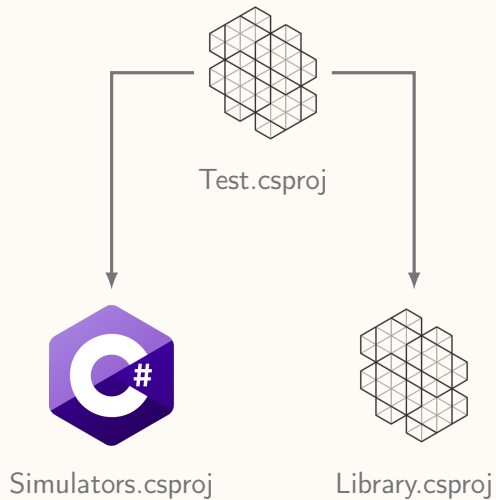
Boolean expression

$$f = x_1 x_2 \oplus x_1 x_3 \oplus x_2 x_3$$

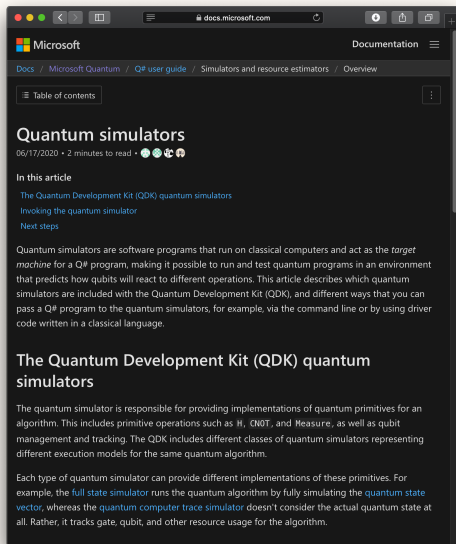
Alternative Boolean expression

$$f = (x_1 \oplus x_2)(x_2 \oplus x_3) \oplus x_2$$

Project organization



QDK simulators




The screenshot shows a web browser window displaying the Microsoft Quantum QDK user guide. The page is titled "Quantum simulators" and is part of the "Simulators and resource estimators" section. The breadcrumb trail is "Docs / Microsoft Quantum / Q# user guide / Simulators and resource estimators / Overview". The page includes a "Table of contents" link and a "In this article" section with links to "The Quantum Development Kit (QDK) quantum simulators", "Invoking the quantum simulator", and "Next steps". The main text explains that quantum simulators are software programs that run on classical computers and act as the target machine for a Q# program. It also introduces the Quantum Development Kit (QDK) quantum simulators and their role in providing implementations of quantum primitives for an algorithm.

Microsoft Documentation

Docs / Microsoft Quantum / Q# user guide / Simulators and resource estimators / Overview

Table of contents

Quantum simulators

06/17/2020 • 2 minutes to read • 

In this article

- [The Quantum Development Kit \(QDK\) quantum simulators](#)
- [Invoking the quantum simulator](#)
- [Next steps](#)

Quantum simulators are software programs that run on classical computers and act as the *target machine* for a Q# program, making it possible to run and test quantum programs in an environment that predicts how qubits will react to different operations. This article describes which quantum simulators are included with the Quantum Development Kit (QDK), and different ways that you can pass a Q# program to the quantum simulators, for example, via the command line or by using driver code written in a classical language.

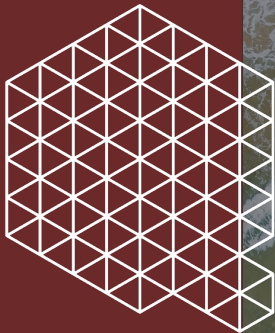
The Quantum Development Kit (QDK) quantum simulators

The quantum simulator is responsible for providing implementations of quantum primitives for an algorithm. This includes primitive operations such as `H`, `CNOT`, and `Measure`, as well as qubit management and tracking. The QDK includes different classes of quantum simulators representing different execution models for the same quantum algorithm.

Each type of quantum simulator can provide different implementations of these primitives. For example, the [full state simulator](#) runs the quantum algorithm by fully simulating the [quantum state vector](#), whereas the [quantum computer trace simulator](#) doesn't consider the actual quantum state at all. Rather, it tracks gate, qubit, and other resource usage for the algorithm.

Resource estimation

Implementation	CNOT	Clifford	T	T Depth	Qubits
Ints	120	30	84	56	5
CCNOTs	30	6	21	15	4
TruthTable	14	3	8	6	4
CNOTTransformation	15	2	7	5	4



Resource estimation and verification in Q#

Mathias Soeken
October 13, 2020