

A Novel Environment for Simulation of Quantum Computing

Joanna Patrzyk¹, Bartłomiej Patrzyk¹, Katarzyna Rycerz^{1,2}, Marian Bubak^{1,2,3}

¹ AGH University of Science and Technology, Institute of Computer Science AGH,
Department of Computer Science, al. Mickiewicza 30, 30-059 Kraków, Poland

²ACC Cyfronet AGH, Nawojki 11, 30-950, Kraków, Poland

³ University of Amsterdam, Institute for Informatics, Faculty of Science, Science Park 904, 1098XH
Amsterdam, The Netherlands

emails: {joanna, bartlomiej}@patrzyk.eu, {kzajac, bubak}@agh.edu.pl,

Keywords: quantum computation, quantum computer simulators, development environment, quantum algorithms, SUS survey

1. Introduction

In this paper we describe a quantum computer simulator with an integrated development environment - QuIDE. The goal of this study was to simplify building and testing quantum circuits and algorithms and to design a simulator which will help to learn and understand quantum computation and quantum algorithms in an efficient way. Furthermore, the simulator should support implementation and analysis even the most complicated algorithms. The development environment should provide both: the flexibility of source code edition and ease of graphical building of circuit diagrams.

2. Description of a problem solution

Figure 1 presents shortly the key modules and capabilities of the simulator.

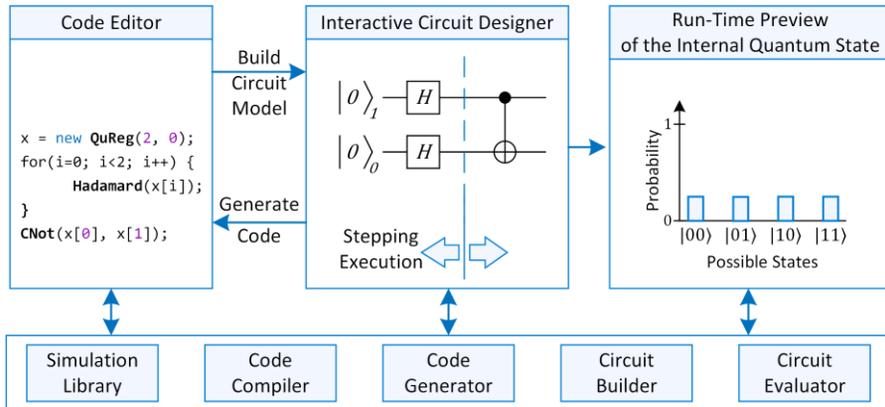


Fig. 1. The key modules and capabilities of QuIDE simulator.

Quantum circuits can be composed with the graphical editor or programmed with the code editor and a user can switch between the code and graphical representation at any time. The simulation environment enables the user to compute the result of the entire circuit or execute it step by step. The simulator is capable of displaying run-time preview of the

internal quantum state. The basic module of the environment is the *Simulation Library* responsible for emulating qubits (quantum bits) and their behavior. The results obtained from the library are presented in the run-time preview. The code from the *Code Editor* can be executed directly using the *Code Compiler* or it can be converted into the graphical circuit representation using the *Circuit Builder*. The circuit can be further executed with the *Circuit Evaluator* or it can be converted into source code using the *Code Generator*. The *Simulation Library* may also be used as a standalone .NET library.

3. Results

We have implemented the most important quantum algorithms: Grover's Database Search, Quantum Fourier Transform, Quantum Teleportation and Shor's Factoring Algorithm [1-3]. Shor's algorithm was implemented in a few optimization variants. The simulator is able to emulate up to 26 qubits in the worst case (mixed states). The example running time for Shor's algorithm was 58 minutes – measured on personal computer when factoring 4087. This operation needed emulating 27 qubits. The results of all implemented algorithms are proven to comply with the corresponding theory.

The simulator was chosen to be the main simulation framework on Quantum Computation course attended by about 100 Computer Science students. At the beginning of the course we measured the usability of QuIDE. The obtained average System Usability Scale (SUS) [4] score was 75 points (for 100 possible), with standard deviation 10. At the same time we measured usability for other quantum simulator – libquantum [5], it received 51 points with standard deviation 16. At the end of the course we repeated the SUS survey for QuIDE and received 76 points with standard deviation 6. The answers were collected from the group of 31 students at the beginning and 53 students at the end.

4. Conclusions and future work

We received comprehensive feedback from the users, the simulator has proven to be useful for didactic purposes. We have successfully used it for testing and comparing quantum algorithms. The most frequently proposed improvement was the support for multiple platforms. Currently, the simulator is available only for Microsoft Windows. In future releases the simulator should be ported to web application.

Acknowledgement. This study was partly supported by the AGH grant no 11.11.230.124 and also by Domain-oriented services and resources of Polish Infrastructure for Supporting Computational Science in the European Research Space – PLGrid Plus project no POIG.02.03.00-00-096/10.

References

1. Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In Proc. of Annual ACM Symposium on Theory of Computing, pages 212-219, 1996.
2. C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters, Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels, Phys. Rev. Lett. 70, pages 1895–1899, 1993
3. Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. In Proc. of the 35th Annual Symposium on Foundations of Computer Science, pages 124-134, 1994.
4. John Brooke Usability evaluation in industry, SUS - a quick and dirty usability scale (CRC Press, Boca Raton, FL), pages 189–194, 1996
5. John B. Butscher and H. Weimer. Simulation eines Quantencomputers, March 2003.