Quantum Interpreted Circuits (QuIC): Rapidly Simulating Quantum Algorithms

Teik Guan Tan and **Jianying Zhou**, ISTD, Singapore University of Technology & Design

Introduction

QuIC (pronounced "quick") is a Quantum interpreter and simulator whose design was inspired by Terry Rudolph's "Q is for Quantum" [1]. The objective for developing QuIC is to promote the understanding of basic Quantum circuits through a hands-on, responsive approach. QuIC is targeted at Computer-Science users wanting to realize Quantum algorithms quickly and conveniently, without needing to consider the underlying physical structure of the Quantum computer.

QuIC is written in C POSIX and currently runs as a command-line tool that takes in a sequence of text strings to represent the ordering of gates that operate on the qubits sequentially.

Features

QuIC provides a simple command-line interface with the following features:

- *Efficient*. Able to support > 20 Qubits simulation through dynamic Just-in-time (JIT) allocation of memory resources.
- *Universal*. Includes Not (X), Identity (I), multi-Qubit Controlled-Not (C^{n}N) and Hadamard (H) gates as a universal gate set [2].
- *Interactive*. Has helper functions to print intermediate results, filter away unwanted qubits and trigger off required interferences.
- *Multi-platform*. Code is portable on multiple platforms, such as Windows, Mac, UNIX, Android (see Fig 1) and runs as a standalone binary.

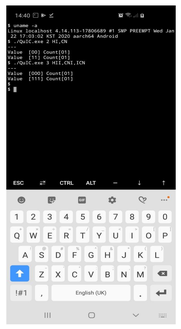

Figure 1: QuIC running in Termux terminal on Android

In Figure 1, we simulate the 2-Qubit Bell [3] state and 3-Qubit Greenberger–Horne–Zeilinger (GHZ) [4] state in a Termux (www.termux.com) terminal app running on a Samsung S10e Android phone.

QuIC Execution

QuIC takes in 2 parameters in the command-line, namely <num_qubits> and <QuIC_string>. The <num_qubits> represents the starting number of qubits in the circuit while <QuIC_string> represents the ordered list of text strings which indicate the gates for each sequence of simulation. We illustrate QuIC's command line usage through 2 examples: Figure 2 creating the 2-qubit Bell state [3], and Figure 3 simulating the balanced Deutsche-Jozsa circuit [5].

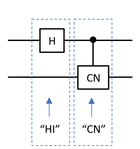

Figure 2: Circuit for Bell State.
QuIC string = `HI,CN`

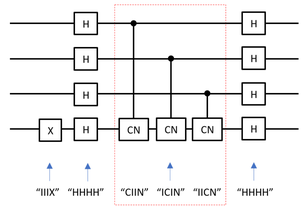

Figure 3: Circuit for balanced Deutsche-Jozsa.
QuIC string = `IIIX,HHHH,CIIN,ICIN,IICN,HHHH`

We next proceed to simulate a 3-qubit Grover [6] algorithm with the marking Oracle looking for "111" (in red) using the following command:

```
QuIC 4 HHHI,IIIX,IIIH,CCCN,IIIH,IIIX,HHHI,
XXXI,IIHI,CCNI,IIHI,XXXI,HHHI,IIIX,IIIH,CCCN,
IIIH,IIIX,HHHI,XXXI,IIHI,CCNI,IIHI,XXXI,HHHI
```

Qubit 4 in this case is used as the ancillary qubit.

And QuIC returns the following result:

```
Value -[0000] Count[256], Value -[0010] Count[256]
Value -[0100] Count[256], Value -[0110] Count[256]
Value -[1000] Count[256], Value -[1010] Count[256]
Value -[1100] Count[256], Value  [1110] Count[2816]
```

The result shows that "111" has the highest probability of being measured.

Finally, we implement Simon's algorithm [7] (See Figure 4) on QuIC to perform the Kuwakado-Morii key recovery attack [8] on Even-Mansour ciphers. Table 1 shows the execution times for key sizes ranging from 6 bits to 11 bits.

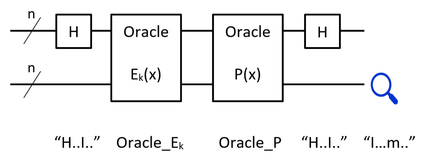

Figure 4: Circuit for Simon's algorithm in Kuwakado-Morii [8] key recovery attack on Even-Mansour ciphers.

Key Size (bits)	No. of Qubits used	Time taken by QuIC for 1 round	Probability of finding key	Estimated time for key recovery
6	12	0.001 sec	2.7596 %	0.036 sec
7	14	0.122 sec	1.5625 %	7.808 sec
8	16	0.401 sec	0.7458 %	53.734 sec
9	18	8.391 sec	0.3951 %	35 min 22.923 sec
10	20	71.343 sec	0.1742 %	11 hr 22 min 30.882 sec
11	22	1 hr 31 min 47.072 sec	0.0992 %	> 64 days

Table 1: QuIC execution timings for Simon's Algorithm run on an Intel Core I5-8250U 8th Gen machine with 8GB RAM

Conclusion and Next Steps

The QuIC is a lightweight, easy-to-use, yet powerful Quantum interpreter and simulator. We have run QuIC on different platforms including on an Android mobile phone. Beyond the basic algorithms such as Bell and Deutsche-Jozsa, we are also able to simulate complicated and computationally-intensive Quantum algorithms, including realizing Kuwakado-Morii's attack which uses Simon's algorithm to carry out key recovery on a 11-bit Even-Mansour cipher through the use of 22 qubits. Moving forwards, we see 3 directions in using QuIC for cryptography and cryptanalysis research:

- *Distributed computing*. QuIC currently runs serially on a single thread within a standalone process. The execution can be optimized to run in a multi-threaded, distributed environment to yield greater performance, and support more qubit-hungry algorithms.
- *Gate fuzzing*. Since the QuIC string can be arbitrarily constructed and executed, QuIC can enable users to deploy Quantum gates in various unrestricted combinations to discover or disprove the existence of Quantum algorithms to solve classical hard problems.
- *Oracle development*. We can extend the QuIC language to support more complex Quantum oracles which are needed in Grover's and Simon's algorithms. These can then be used for cryptanalysis or to find possible weaknesses in cryptographic codes.

References

[1] T. Rudolph, 2017. Q is for Quantum. *ISBN: 9780999063507*

[2] D. Aharonov, 2003. A simple proof that Toffoli and Hadamard are quantum universal. *arXiv preprint quant-ph/0301040*.

[3] S.L. Braunstein, A. Mann, and M. Revzen. 1992, Maximal violation of Bell inequalities for mixed states. *Physical Review Letters*, *68*(22), p.3259.

[4] D.M. Greenberger, M.A. Horne, and A. Zeilinger, 1989. Going beyond Bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe* (pp. 69-72). Springer, Dordrecht.

[5] D. Deutsch, and R. Jozsa, 1992. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, *439*(1907), pp.553-558.

[6] L.K. Grover, 1997. Quantum mechanics helps in searching for a needle in a haystack. *Physical review letters*, *79*(2), p.325.

[7] D. Simon, 1994, November. On the power of quantum computation. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (pp. 116-123).

[8] H. Kuwakado, and M. Morii., 2012, October. Security on the quantum-type Even-Mansour cipher. In *2012 International Symposium on Information Theory and its Applications* (pp. 312-316). IEEE.

Acknowledgments

This research was partially supported by MOE AcRF Tier 2 grant (MOE2018-T2-1-111).

Further information

QuIC is available at github.com/tanteikg/QuIC
Please contact the authors at teikguan_tan@mymail.sutd.edu.sg if you have any questions or would like to collaborate.