Tackling the Emerging Problems in Quantum Computer Architecture

N. Cody Jones
Edward L. Ginzton Laboratory, Stanford University, Stanford, California 94305-4088, USA and
Corresponding author: ncjones@stanford.edu

Quantum computer architecture is a young field of study, and there are many challenges looking forward. We present here several open problems facing the development of large-scale quantum computers. In particular, we examine issues regarding the organization and layout of a quantum computer, the interface with classical computation, and the necessary tradeoffs between engineering resources. Moreover, we consider matters of standardization versus optimization, which each have their merits yet are frequently incompatible.

I. MATTERS INTERNAL TO A QUANTUM COMPUTER

Consider a modern CPU: despite the multitude of different companies and products, the design of processors in any context follows a similar pattern. There is an ALU (arithmetic and logic unit), where most of the mathematical calculations take place. Instruction sequencing and branch prediction control the execution of programs. Data is stored locally in one of several caches, or it is off-loaded to random access memory. There are various input/ouput interfaces to networking, graphics, *etc.*

In comparison, quantum computers have no such pattern today. They are amorphous (and as yet, imaginary) objects which will implement some very interesting algorithms. But there are several important engineering matters yet to be resolved. For example, what is the best compromise between dynamical decoupling (DD) [1, 2] and quantum error correction (QEC) [3]? Both methodologies can mitigate errors in quantum information, but it is not clear what the optimum solution is. QEC is almost certainly essential, but there is a nontrivial interplay between the two, so that some DD is helpful, but too much can be inefficient. Similarly, the basic approach to QEC estimates error with the "KQ" product — but is this overly conservative, and is it possible to use less error correction in some contexts? Additionally, quantum computers will certainly be large and complex devices. As such, device failure is probably not a purely random process, since some components will display a greater tendency to fail than others. Can we characterize device performance, and tailor QEC to handle so-called "soft" faults? Can we perform this process dynamically while the system is operating?

II. IMPROVING THE CLASSICAL/QUANTUM INTERFACE

At present, we imagine quantum computers to be elaborate calculators. They are given a hard computational task and then solve it. However, it may be more useful to envision them as dynamic computing objects which interact with classical computers in complex ways. For example, what is the best way to manage the considerable classical information flows resulting from QEC? How does one thread quantum subroutines into joint classical/quantum programs? Is there an efficient way for a classical compiler to optimize quantum circuits, such as by reducing qubit or gate counts, or by increasing resistance to operation faults? Several important QEC processes are probabilistic, mandating decision-based control in quantum programs. As such, can we develop a method to track logical quantum bits as dynamic resources, which necessarily have finite "lifetime" and logical accuracy?

III. STANDARDIZE OR OPTIMIZE?

The layered framework presented previously [4] was definitely an attempt to standardize the design of quantum computers. The argument for standardization is that it enables different quantum computer designs to be compared, and the design process itself is standardized, which makes the formation and operation of engineering teams more efficient. However, it is not certain that this is the best approach. Many cutting edge computing technologies from vendors like Intel use proprietary designs and methods, which are optimized to a particular purpose or for a particular company's resources. So the question becomes: should the nascent field of quantum computing develop engineering standards, or should each technology evolve as its designers see fit?

---

[1] Viola, L. and Knill, E. *Phys. Rev. Lett.* **90**(3), 037901 Jan (2003).
[2] Ng, H. K., Lidar, D. A., and Preskill, J. *Preprint* arXiv:quant-ph/0911.3202., (2009).
[3] Preskill, J. *Preprint* arXiv:quant-ph/9712048., (1997).
[4] Jones, N. C., Van Meter, R., Fowler, A. G., McMahon, P. L., Kim, J., Ladd, T. D., and Yamamoto, Y. Soon to be on arXiv., (2010).