A Layered Architecture for Quantum Computing

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 systematic approach to organizing the resources of a quantum computer. We approach this problem by proposing a layered framework which can be a template for any circuit-model quantum computer. In particular, this framework is *modular* and promotes *fault-tolerance*. Modularity is important for making manageable the unique engineering challenges associated with large-scale quantum computing. Fault-tolerance is essential for a realistic system to scale to problem sizes like Shor's factoring algorithm or the simulation of quantum chemistry.

I. WHY IS QUANTUM COMPUTING INTERESTING?

The field of quantum computing is far too broad to cover in one source, but an excellent introduction can be found in Ref. [1]. Instead, we will motivate why quantum computing is interesting, and in particular, quantum computer *architecture*. First, quantum information in general is believed to contain algorithms and operations not accessible to classical information. Informally, there are programs one can run on a quantum computer which we believe to be simply impossible to ever implement on a traditional electronic computer, because of execution time, system size, *etc.* There are a handful of identified quantum algorithms you might execute, but at this point two stand out as being particularly useful: factoring large numbers and simulating quantum chemistry. The former (Shor's algorithm [2]) is of interest since it can dismantle several prominent cryptographic codes, while the latter [3] is believed to enable unprecedented simulation of chemistry and biology, such as nanostructured materials and the molecular-level design of drugs for pharmacology.

So if quantum computing is a promising field of research, how do we build a quantum computer? Quantum computer architecture addresses this question, which is inherently an engineering problem. For many reasons, quantum computers are more complicated and difficult to design than their classical counterparts. Because the essential physics is often on the atomic scale, these devices are challenging to engineer and more error-prone than conventional electronic circuits. Therefore, large-scale quantum computers will depend critically on *quantum error correction* (QEC) [4]. QEC can enable arbitrarily accurate computation, but the overhead costs in terms of both quantum and classical resources can be very steep, so one must exercise extreme care to design such systems efficiently.

II. ORGANIZING A QUANTUM COMPUTER INTO LAYERS

We propose a method to organize a quantum computer by layers [5], which is a common approach in the engineering of traditional information systems. Each layer serves a specific purpose relative to the other layers and for the system as a whole. In particular, each layer interacts only with its neighbors above and below in a predefined interface. This feature provides modularity, so that one layer can be changed without affecting the others, which is important for large engineering projects. The bottom layer is the Physical layer (Layer 1), which houses the physical components that implement quantum operations. The top is the Application layer (Layer 5), which is where a quantum algorithm like Shor's is executed. The layers between are (in ascending order) Virtualization, Quantum Error Correction (QEC), and Logical. The Virtualization layer constructs fundamental qubits and gates from the raw processes in the Physical layer. These resources are termed "virtual" since they are composed of many different physical components, and the process of virtualization deliberately hides this fact through abstraction. The QEC layer (Layer 3) implements quantum error correction, which is essential in circuit model quantum computing for the accuracy needed in large-scale algorithms like Shor's (*e.g.* error-per-gate $< 10^{-15}$). The job of the Logical layer, which sits between these two, is to translate any arbitrary gate in the Application layer into a sequence of gates from the limited set provided by the QEC layer, such as by the Solovay-Kitaev procedure. We also discuss the use of Pauli frames, which is a simple but highly useful tool in QEC. Taken together, the purpose of Layers 2, 3 and 4 is to mold the physical processes in Layer 1 into a fault-tolerant framework for the Application layer. We motivate this theoretical construct for a quantum computer with a specific example based on the optical control of quantum dots.

---

[1] Nielsen, M. A. and Chuang, I. L. *Quantum Computation and Quantum Information.* Cambridge University Press, 1 edition, October (2000).
[2] Shor, P. W. *SIAM J. Comput.* **26**(5), 1484–1509 (1997).
[3] Lloyd, S. *Science* **273**(5278), 1073–1078 (1996).
[4] Preskill, J. *Preprint* arXiv:quant-ph/9712048., (1997).
[5] N. Cody Jones, *et al.* Soon to be on arXiv., (2010).