State of the Art in Quantum Computer Architectures

Rodney Van Meter

August 13, 2011

Abstract

Quantum computer architecture as a field remains in its infancy, but carries much promise for producing machines that vastly exceed current classical capabilities, *for certain systems designed to solve certain problems*. It must be recognized that *large systems are not simply larger versions of small systems*. These notes review the fronts on which progress must be made for such systems to be realized: experimental development of quantum computing technologies, and theoretical work in quantum error correction, quantum algorithms, and computer architecture. Key open problems are discussed from both a technical and organizational point of view, and specific recommendations for increasing the vibrancy of the architecture effort are given.

Keywords: quantum computation, quantum error correction, quantum computer architecture

1 Introduction

> When will the first paper appear in *Science* or *Nature* in which the point is the results of a computation, rather than the machine itself? That is, when will a quantum computer *do* science, rather than *be* science?

This question provokes answers ranging from, "Already have," (in reference to analog quantum simulation of a specific Hamiltonian) to "Twenty years," to "Never," – and all these from people actually working in the field. I will try to shed a little light on how such varying answers can arise, and more importantly, how we can change that equation.

This informal set of notes accompanying the FIRST 2011 Quantum Computing Summer School is intended to convey the current state of the art in designing and building large-scale quantum computers, that is, the art of quantum computer architecture. I do not present other major areas of quantum technology such as quantum key distribution (QKD) or quantum repeater networks. I am particularly happy to discuss quantum repeaters if the occasion arises. Naturally, everything written and said is from the point of view of the author/presenter only, and occasionally is over-stated to clarify rhetorical arguments.

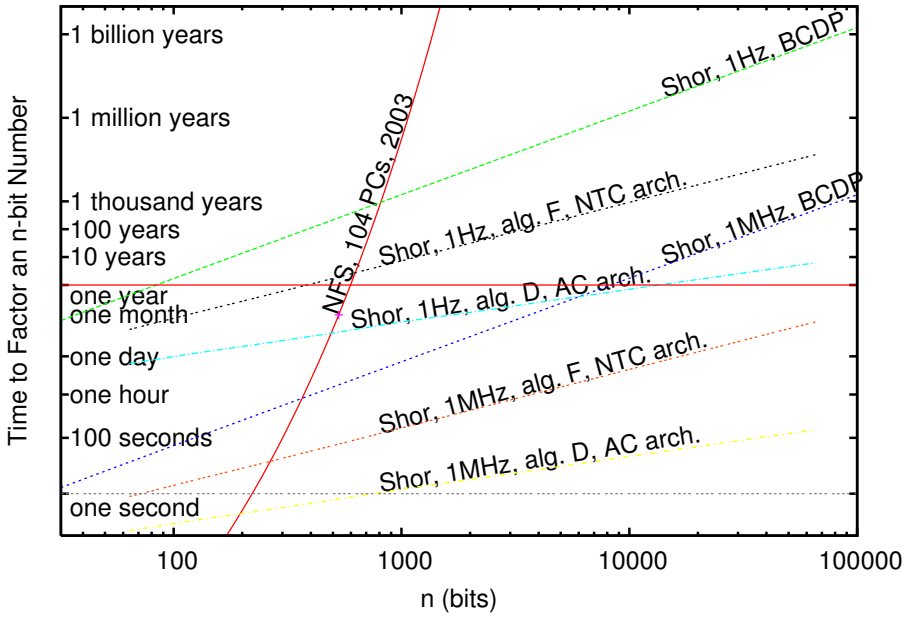

Figure 1: Scaling of the classical number field sieve (NFS), the fastest known classical factoring algorithm [3], versus Shor's algorithms for factoring. The horizontal axis is the length of the number to be factored. The steep line is NFS, while the other lines are various combinations of logical clock speed for a Toffoli gate (1Hz and 1MHz), method of implementing Shor's algorithm (BCDP, D and F), and quantum computer architecture (NTC and AC). The assumed capacity of a machine in this graph is $2n^2$ logical qubits. From [4].

These notes (and my talk) assume that readers are familiar with the basics of quantum computing covered over the last several days [1, 2]: I will use qubit, gate, controlled-NOT, ion trap, Josephson junction, Bell pair, etc. without pausing to define them. Likewise, I will discuss architectural concepts without preamble, but with a somewhat lower expectation of familiarity.

Fig. 1 demonstrates the impact that architecture can have on the bottom-line viability of a quantum computer. While the vernacular press often talks of factoring large numbers "in seconds" using a quantum computer, in reality it is not even possible to discuss the prospective performance of a system without knowing the physical and logical clock speed, the topology of the interconnect among the qubits, the number of logical qubits available in the system, and the details of the algorithmic implementation, including how well it is tuned to match the architecture.

To actually do science with a quantum computer, we must have coordinated advances on several fronts. Fig. 2 shows a conceptual stack of subfields that must all contribute to a complete, useful machine. Machines will be built to run specific algorithms that exist in certain computational classes inaccessible to classical computers.

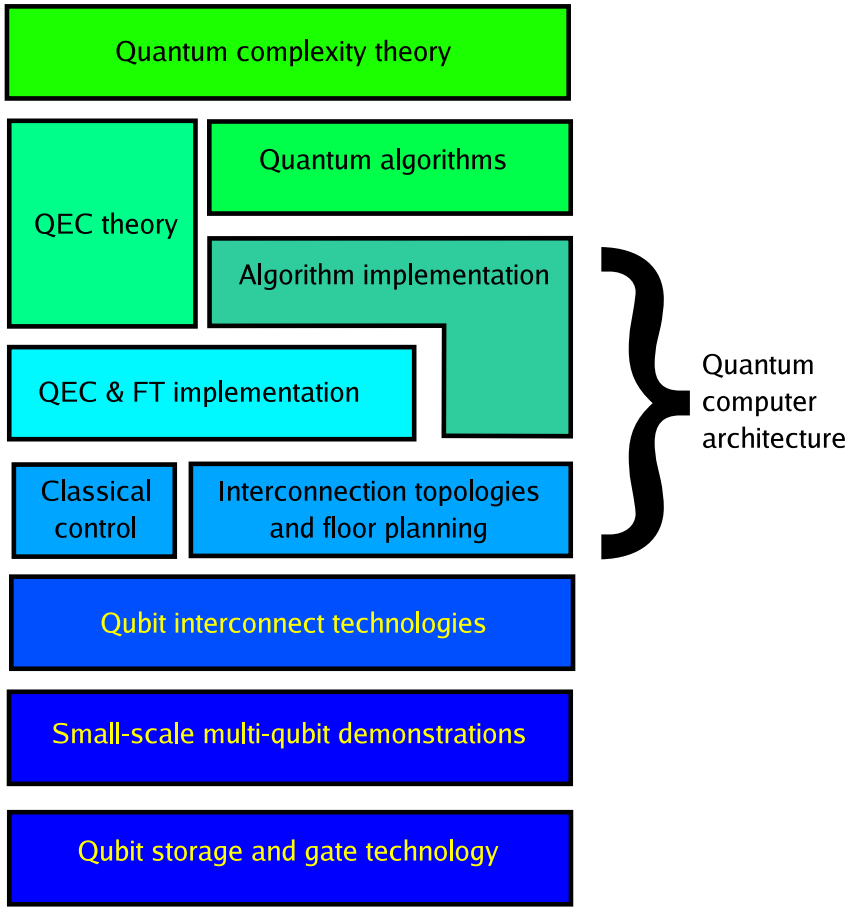

Figure 2: Quantum computer architecture among some subfields of quantum computation. QEC is quantum error correction; FT is fault tolerant. From [4].

Without these algorithms, there will be no economic incentive to build and deploy machines. Without error correction (both theory and practice), the machines cannot run for any useful length of time. At the "bottom" of the stack lie the qubit storage, gate and transport/interconnect technologies, which are providing useful proving grounds for quantum theory itself, but are of interest here primarily as the foundation for building large-scale quantum computers. Both the top and bottom of this stack are heavily populated with brilliant, dedicated researchers.

The middle layers of this figure, constituting the realm of quantum computer architecture, are less heavily populated. Designing an architecture is the process of choosing the metrics for success, defining the desired performance and the costs for components so that they may be balanced as desired. Functionality is divided into subsystems, and the interfaces between subsystems defined, along with corresponding promises and assumptions about behavior or conditions maintained.

This bridge is critical, and an area with much room for creative, rigorous work. A microprocessor is much more than simply a collection of transistors, and a complete system is much more than simply a microprocessor. Likewise, quantum computers will be much more than simply uniform collections of qubits.

Quantum computer architecture has much to learn from classical computer architecture, but with a few important caveats. Foremost among these is that the delicate nature of quantum information demands that memory elements be very active; in most conceptions, a quantum computer will not make a strong distinction between memory elements and computational elements. Second, long wires or long-distance connections inside of a quantum computer are non-existent, requiring nearest-neighbor cellular automaton-like transfer of data, or are at best poor quality, requiring much effort to transfer even a single qubit from place to place using purification and teleportation. Thus, the *principles* can be applied, but the *answers* arrived at will likely differ substantially from classical von Neumann architectures.

In the rest of this set of notes, I discuss (anecdotally rather than comprehensively) recent progress on experiments, quantum error correction, algorithms, and architectures. This process has been made substantially easier by the appearance in the last few years of excellent reviews on each topic except architecture [5, 6, 7, 8, 9, 10, 11, 12, 13]. Immodestly, I will say that although I'm not completely happy with it, the best architecture-focused survey I'm aware of is part of my own Ph.D. thesis [4].

2 Recent Experimental Progress

Experimental progress continues apace, with new results reported regularly on the four main technical issues: memory lifetime, single-qubit gates, controllable two-qubit gates, and single-shot measurement. A couple worth highlighting:

- The startup company D-Wave has demonstrated quantum effects in their adiabatic quantum computing machine that uses Josephson junction flux qubits [14]. They have reported in journal papers a *yield* of 40% of their qubits being functional; more recently, I have been told that yield has increased to 75% on their chip containing 128 qubits. This means that, among other things, they have been

advancing the state of the art in scaling of classical control.

- Rainer Blatt's group in Austria has demonstrated two important advances using ion traps: 14-qubit entanglement [15] and multiple rounds of an error correction-*like* circuit [16]. The error correction experiment is phase errors only, does not use single-shot measurement and classical feed-forward (as production systems are likely to do), and repeatedly encodes and decodes the logical qubit, rather than keeping the logical qubit and performing true correction for multiple rounds. While these experiments are impressive, it is worth noting that they were performed on a linear Paul trap, which is not scalable. However, Chris Monroe, Dave Wineland, Ike Chuang and others are building non-linear ion traps with good scaling properties.

Beyond these, the *transmon* qubit (another type of Josephson junction qubit, held in a millimeters-long cavity to improve qubit coupling and suppress noise), nitrogen vacancy (NV) diamond, and solid-state nanophotonics are generating a great deal of interest.

Even so, it's really rather unfair to single out any particular technology or group, since progress has a broad foundation across dozens of groups and dozens of technologies worldwide. The best recent survey is the Nature paper by Ladd *et al.* [5], which, despite the title, is about technologies rather than architectures.

Beyond this, I leave the survey of technologies up to the other professors in the group, who have filled you in on many aspects of the technologies.

One final point on technologies – despite the fact that we are dealing with quantum effects, most of these technologies use devices that are enormous compared to transistors in modern computer chips, which really *are* reaching down to atomic scales, with channel lengths of less than a hundred times the Si lattice cell. For QC devices, ion traps are limited to spacing of perhaps tens of microns for RF and optical control. Nanophotonic systems likewise will require components tens of microns across, to accommodate the 1.5μm wavelength light that is desirable for telecommunications and silicon optics. Superconducting flux qubits require a current ring tens of microns across. All of these technologies result in qubits that are macroscopic, or nearly so, with areal densities a million times less than computer chips. This fact will have enormous impact on large-scale architectures, as we will see below.

3 Recent Progress in Error Correction

Here, likewise, the recent news is good. Probably the most promising approach in error correction is *surface code* computation, which grew out of highly abstract work by Alexei Kitaev and others on topological computing. Robert Raussendorf and collaborators created the 2-D and 3-D versions suitable for solid-state and photonic systems, respectively [17, 18]. Austin Fowler has worked tirelessly to extend the practicality of these results, and he and David Wang have shown numerically that **the *threshold*** [1] **is at**

[1] The threshold is the physical error rate at which applying error correction gives a net improvement in logical error rate. It is widely believed that implementation of large-scale systems becomes practical only when the physical error rate betters the threshold by one to two orders of magnitude.

least 1.4% [19], treating (as is customary) single-qubit, two-qubit, measurement, and memory errors as having the same error rates. Surface code, of course, is not the only viable approach to quantum error correction; earlier Calderbank-Steane-Shor (CSS) approaches have yielded recent progress, as in Bacon-Shor codes [6].

In both CSS and surface code systems, direct execution of a Toffoli gate is not possible. A specially-prepared ancilla state must be used. The state is prepared using magic state injection and distillation. While this process is not believed to be the Achilles' heel in logical error rate, consumption of these states may very well dominate the workload of the machine, and our own recent work has assumed that 75% of the machine is dedicated to their production. Others have referred to this need as running quantum applications "at the speed of data," that is, producing the generic ancilla states rapidly enough that they are not the performance bottleneck [20].

In the course of discussing error correction, I probably have used the word *stabilizer* several times. A stabilizer of a quantum state is an operator for which the state is an eigenvector. A shorthand notation was developed by Daniel Gottesman allowing fairly compact descriptions of quantum states and the operation of quantum error correction [21] [2].

In establishing the error correction requirements for a particular application, we talk in terms of the product KQ, where Q is the number of logical qubits required for the algorithm, and K is the circuit depth in units of some basic gate. For the algorithm to have a good probability of successfully producing an answer, the system must be engineered to have a net logical gate error rate $p_e \ll KQ$ [22].

Beyond what we think of as "standard" quantum error correction, other techniques can be used to partially decouple the qubits from the environment. Decoherence-free subspaces encode a logical qubit in the phase difference of two physical qubits, suppressing the effect of certain types of noise [8]. Composite pulse techniques originally developed for NMR can be used to suppress systematic gate errors [9]. Spin-echo and dynamic decoupling techniques similarly can be used to partially reverse the impact of system effects on memory, going with the error for a while and against it for a while, canceling out the effect [23].

Finally, purification techniques in a sense implement very simple error correction on a known state, typically a Bell state [7]. Purification is valuable in low-resource environments, such as hardware-constrained repeater networks, and as part of the system substrate on which other error correction techniques are built.

4 Recent Progress in Applications

Here, unfortunately, the news is more mixed: it has been widely believed that Shor's factoring algorithm [24] and quantum simulation [12, 13] (that is, simulation of other quantum systems, in either a digital or analog fashion) will provide the two driving reasons to build machines, but the size and speed of a machine needed to run the former has been consistently misunderstood, and the machine to run the latter continues to

[2] For a quick undergraduate-level introduction to stabilizers, see the notes available at `http://aqua.sfc.wide.ad.jp/for-students.html`.

grow as investigations into algorithmic precision and error management have pushed up the minimum size of machine necessary.

Our own recent results have suggested a physical machine size of billions of qubits to factor a 2,048-bit number, even using the promising surface code [25]. In a more unexpected result, Ken Brown's group (Georgia Tech) has shown that even certain quantum simulation algorithms expected to be computationally tractable on quantum computers are turning out to have dismayingly large resource requirements [26]. However, the field of quantum simulation is varied, and it is not immediately clear (to me, at least) how broadly applicable this result is. Alán Aspuru-Guzik's group at Harvard is certainly forging ahead with interesting work. It's worth emphasizing that these simulators remain attractive; they are simply going to take more computational resources (hence, more calendar years to develop and more dollars to deploy) than originally hoped.

The news is not all bad, however. Bacon and van Dam [10] and Mosca [11] have published surveys covering work in quantum random walks, games, and group theory, as well as quantum simulation. Our understanding of *how* to design new quantum algorithms that (at least asymptotically) outperform classical ones continues to grow, though the number of people who can put the concepts into practice remains small.

Here, I am fond of saying that we are still in the time of Babbage, trying to figure out what Knuth, Lampson and Torvalds will do with a quantum computer. While we can't economically justify building machines without applications, I fully expect that the field of algorithms will develop in unexpected ways once we put machines in the hands of actual programmers.

5 Recent Progress in Architectures

Very few groups are working on truly large-scale architectures. These groups include my own collaboration with Austin Fowler (Melbourne) and Thaddeus Ladd and Cody Jones from Yoshi Yamamoto's group (Stanford/NII), the Chuang-Chong-Oskin-Kubiatowicz orbit, and the Nemoto-Munro-Devitt (NII) group with which Thaddeus, Austin and I sometimes collaborate. Austin has also done work on the scalability of Josephson junction systems [27]. Ike Chuang has also worked with students Krysta Svore and Andrew Cross, and Prof. Igor Markov of Michigan (who is a VLSI/VAD guy with much-needed expertise, in my opinion). Jungsang Kim (Duke) and Ken Brown (Ga. Tech) likewise are pushing into the field. Andrew Steane is essentially a lone, occasional European outpost of architectural thinking [28]. Outside of this list, I can't recall having seen any papers that provide an estimate of how physically large and how fast a system will need to be in order to calculate post-classical results.

Sec. 4 gave an inkling of the work in architectures; while it might have seemed out of order there, it is difficult to tease apart the intertwined issues of the *workload* for a system, and the design of the system itself. In fact, workload, technology, error correction and architecture must all come together to create a complete system. Most of us have been using Shor's algorithm as a benchmark, both for its importance and clarity, and because the arithmetic and quantum Fourier transform on which it is founded are valuable building blocks for other algorithms, as well.

Fig. 1 shows several lines for different logical clock rates. The conversion factor from physical gate rate to logical gate rate depends on a number of architectural features, as well as the code itself. For ion trap-based QLA used in the Clark/Brown paper [26, 29], a 10μsec physical gate results in a 1.6msec error correction cycle time using one level of the Steane [[7,1,3]] code, and 260msec for two levels. Without digging back into the architecture papers, I'm uncertain of the ion movement time assumed, and the exact conversion factor to logical gates. That architecture has also been improved, reducing the area required for the ion traps by over an order of magnitude [30].

Our "racetrack" nanophotonic quantum dot architecture using the surface code might be one to two orders of magnitude faster than QLA with the concatenated Steane code for many algorithms. In our architecture [25], the surface code lattice refresh time (a full cycle of measuring all stabilizers) is $\sim 50\mu$sec, far slower than the 100psec assumed for physical gates. A Toffoli gate will require $\sim$ 50msec – a factor of a thousand from lattice refresh to logical gate, largely due to the need to distill special ancillae. The lattice refresh time in this architecture is limited by contention for access to on-chip waveguides. The QuDOS architecture [31] might be a hundred times faster, largely due to increased parallelism in measuring the stabilizers for the many plaquettes in the surface. Thus, with a change of physical technology, error correction mechanism, and architecture, we may gain 3-4 orders of magnitude in performance.

In the early-mid 2000s, Oskin, Chong and Kubiatowicz and their students worked with Ike Chuang on large-scale ion trap designs, and published a series of progressively better, more detailed designs, including descriptions of their architecture software toolkits [29, 30, 32, 33, 20, 34, 35, 36, 37, 38]. This group collectively became dormant as students graduated, funding ended and faculty took leaves of absence. However, in the last year or so, I am told that at least some of them have received funding (the recent iARPA Quantum Computer Science Program that set *extraordinarily* ambitious goals for improving architectures, I believe) and taken up the torch again.

Somewhat more recently, Jungsang Kim has been working on the use of micromirrors to guide the control laser beams necessary for e.g. ion trap and quantum dot systems [39, 31]. This kind of research demonstrates the necessity of complete solutions, and how many problems must collectively be solved before large-scale systems can be deployed.

Integrated nanophotonics provide the route to getting optics off of a massive lab bench and into easily scalable systems. Perhaps the leading lab in experiment is Jeremy O'Brien's (Bristol); by far the most complete, interesting architecture is that of Simon Devitt, working in Kae Nemoto's group [40]. This system uses a multi-input, multi-output module that entangles photons. Modules are collected into boards, and the system is organized into stacks of planes in three phases perpendicular to each other. The overall system creates a 3-D cluster state for the surface code. Simon has also implemented the real-time error processing necessary for the system in enough detail to determine that the classical half of the machine is a tractable engineering problem; as far as I am aware, this is the first solid demonstration of this fact [41] [3].

[3] Prof. Nemoto has graciously loaned me their animations of the proposed architecture to show during this workshop, but not given me permission to share copies.

Finally, I want to emphasize that much of the work in architecture revolves around interconnects. Floor planning of the device layout, minimizing crossing of both classical control lines and quantum waveguides as well as layer-to-layer vias is critical. Off-chip communication will be somewhere between slow and abysmal; nevertheless, it still seems to be inevitable and I remain committed to the multicomputer concept [4, 42]. Our racetrack architecture, in fact, recognizes that even within a single chip, the fidelity of achievable two-qubit operations will vary depending on distance, and has explicitly incorporated varying amounts of purification [25].

6 A Roadmap

As far as I am aware, the comprehensive ARDA-supported Quantum Information Science and Technology Roadmapping Project's roadmap has not been updated since 2004 [43]. (Perhaps it is now classified, though that would seem to defeat the purpose of a roadmap.) Certainly, its 2002 goal of "develop[ing] by 2012 a suite of viable emerging-QC technologies of sufficient complexity to function as quantum computer-science test-beds in which architectural and algorithmic issues can be explored," will not be met as originally envisioned.

7 Open Problems/The Path to Accelerated Progress

Naturally, since I am an architect, I think the work that I do is important; it is critical to recognize that *large-scale systems are more than simply larger versions of small systems*. Here, I suggest a few technical problems that must be solved, and a few ways in which the community can strengthen its ability to solve them.

7.1 Key Technical Problems

Any quantum computing researcher will tell you that memory lifetime, single- and two-qubit gate fidelity, and single-shot measurement need more work. A few, mostly interested in quantum repeaters, will tell you that transferring qubits from optical to solid-state, one wavelength to another, or electron spin to nuclear spin, is important. Indeed, all of these are critical. They are also being addressed by many programs from funding agencies around the world.

Considering the issues of quantum computer architecture results in a slightly different focus. I would summarize the important problems in quantum computer architecture using the following keywords: heterogeneity, interconnects, integration, defects, and tools.

- **Heterogeneity.** Some researchers have been investigating the use of both electron and nuclear spins in a single system. This process is important. It is equally important to consider *structural* heterogeneity, in operation capability or time, interconnect speed or fidelity, etc. Most researchers are not (yet) considering separate storage and operation areas, but this may be possible under some circumstances.

- **Integration.** Increasing numbers of on-chip devices will require improved on-chip integration of control circuits and multiplexing of I/O pins, as well as getting away from multiple rack-mount units to control each individual qubit. This will require *substantial* investment in *systems* engineering to control meso-scale systems – tens to low hundreds of qubits. Experimental laboratories likely will have to hire professional staff to do extensive control circuit design.

- **Interconnects.** While we remain far from the limits of monolithic designs, many of the solid-state systems under investigation use devices that are enormous compared to classical transistors. Integration levels will likely reach hundreds to low thousands of devices in a single chip, but *reaching millions to billions of devices in a system will require interconnects that remain only on the drawing board*.

- **Defects.** Fabrication will inevitably be an imperfect process. Architectures and error correction systems need to be designed to work around defective qubit storage sites. Qubits may be declared defective because they fail to correctly trap the correct state variable carrier (e.g., single electron), because memory lifetime is short or gate control imprecise, or because they fail to couple properly to other qubits. Likewise, even if two qubits are declared functional, their coupling may be declared dead.

- **Tools.** Further investment in automated tools for co-design of hardware and compilation of software is critical; in particular, working with the assumption that systems will be internally heterogeneous. Large-scale designs are difficult to create and evaluate without these tools.

7.2 Meta-Problems

So, what is it that's missing from the ecosystem, and where and how do we get it?

- It seems to be time to *begin demanding Moore's Law-like improvements in system capacity*. The iARPA program seems to be headed in this direction. While I am very much in favor of continued strong support for university-based research, it is worth asking if the university is the right place for the large engineering teams necessary. Startup company? Larger consortium such as those that run observatories and colliders? National lab? Big company?

- *Developing a sense of community is critical.* Creating a shared understanding (including vocabulary, concepts and important problems) among the physics and CS theory, algorithm designer, physics experiment, and architecture communities is critical, but few journals or conferences actually provide a good venue. The occasional International Conference on Quantum Error Correction (run by Daniel Lidar and Todd Brun of USC) and ACM's *Journal of Emerging Technologies in Computing Systems* are perhaps two of the best, but even they are not particularly high visibility. Among physics journals, the high-impact *New Journal of Physics* and almost invisible *International Journal of Quantum Information* have been somewhat receptive to architecture-focused papers. The

classical architecture community has accepted occasional papers to the *International Symposium on Computer Architecture* and *International Symposium on Microarchitecture*, both of which are *highly* competitive and prestigious.

- To alter the focus, reviewers (of papers and funding proposals) should *demand realistic estimates of logical Toffoli gate time*, assuming error correction, for some target logical fidelity.
- Even more ambitious, I recommend *demanding realistic estimates of application performance*.

References

[1] H. Kimble et al. Quantum computing. Technical report, JASON, July 1996.

[2] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[3] A. Lenstra, E. Tromer, A. Shamir, W. Kortsmit, B. Dodson, J. Hughes, and P. Leyland. Factoring estimates for a 1024-bit RSA modulus. In *AsiaCrypt 2003*, Lecture Notes in Computer Science, New York, 2003. Springer-Verlag.

[4] Rodney Doyle Van Meter III. *Architecture of a Quantum Multicomputer Optimized for Shor's Factoring Algorithm*. PhD thesis, Keio University, 2006. available as arXiv:quant-ph/0607065.

[5] T.D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J.L. O'Brien. Quantum computers. *Nature*, 464:45–53, March 2010.

[6] Simon J. Devitt, Kae Nemoto, and William J. Munro. The idiots guide to quantum error correction. arXiv:0905.2794v2 [quant-ph], 2009.

[7] W. Dür and H.J. Briegel. Entanglement purification and quantum error correction. *Rep. Prog. Phys.*, 70:1381–1424, 2007.

[8] Daniel A. Lidar and K. Birgitta Whaley. *Irreversible Quantum Dynamics*, volume 622 of *Lecture Notes in Physics*, chapter Decoherence-Free Subspaces and Subsystems. Springer, 2003.

[9] Lieven M.K. Vandersypen and Isaac Chuang. NMR techniques for quantum computation and control. *Rev. Modern Phys.*, 76:1037, 2004.

[10] Dave Bacon and Wim van Dam. Recent progress in quantum algorithms. *Communications of the ACM*, 53(2):84–93, February 2010.

[11] M. Mosca. Quantum algorithms. *Arxiv preprint arXiv:0808.0369*, 2008.

[12] Iulia Buluta and Franco Nori. Quantum Simulators. *Science*, 326(5949):108–111, 2009.

[13] Katherine L. Brown, William J. Munro, and Vivien M. Kendon. Using quantum computers for quantum simulation. *Entropy*, 12(11):2268–2307, 2010.

[14] MW Johnson, MHS Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, AJ Berkley, J. Johansson, P. Bunyk, et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[15] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximilian Harlander, Wolfgang Hänsel, Markus Hennrich, and Rainer Blatt. 14-qubit entanglement: Creation and coherence. *Phys. Rev. Lett.*, 106(13):130506, Mar 2011.

[16] Philipp Schindler, Julio T. Barreiro, Thomas Monz, Volckmar Nebendahl, Daniel Nigg, Michael Chwalla, Markus Hennrich, and Rainer Blatt. Experimental repetitive quantum error correction. *Science*, 332(6033):1059–1061, 2011.

[17] Robert Raussendorf and Jim Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Physical Review Letters*, 98:190504, 2007.

[18] Robert Raussendorf, Jim Harrington, and Kovid Goyal. Topological fault-tolerance in cluster state quantum computation. *New Journal of Physics*, 9:199, 2007.

[19] David S. Wang, Austin G. Fowler, and Lloyd C. L. Hollenberg. Surface code quantum computing with error rates over 1%. *Phys. Rev. A*, 83(2):020302, Feb 2011.

[20] N. Isailovic, M. Whitney, Y. Patel, and J. Kubiatowicz. Running a quantum circuit at the speed of data. In *International Symposium on Computer Architecture*, pages 177–188. IEEE, 2008.

[21] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, May 1997.

[22] Andrew M. Steane. Overhead and noise threshold of fault-tolerant quantum error correction. *Physical Review A*, 68:042322, 2003.

[23] M. J. Biercuk, A. C. Doherty, and H. Uys. Dynamical decoupling sequence construction as a filter design problem. *Journal of Physics B*, 2011. to appear; arXiv:1012.4262v2 [quant-ph].

[24] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. 35th Symposium on Foundations of Computer Science*, pages 124–134, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[25] Rodney Van Meter, Thaddeus D. Ladd, Austin G. Fowler, and Yoshihisa Yamamoto. Distributed quantum computation architecture using semiconductor nanophotonics. *International Journal of Quantum Information*, 8:295–323, 2010. preprint available as arXiv:0906.2686v2 [quant-ph].

[26] Craig R. Clark, Tzvetan S. Metodi, Samuel D. Gasster, and Kenneth R. Brown. Resource requirements for fault-tolerant quantum simulation: The ground state of the transverse Ising model. *Phys. Rev. A*, 79(6):062314, Jun 2009.

[27] Austin G. Fowler, William F. Thompson, Zhizhong Yan, Ashley M. Stephens, and Frank K. Wilhelm. Long-range coupling and scalable architecture for superconducting flux qubits. cond-mat/0702620, February 2007.

[28] Andrew M. Steane. How to build a 300 bit, 1 Gop quantum computer. http://arxiv.org/abs/quant-ph/0412165, December 2004.

[29] Tzvetan S. Metodi, Darshan D. Thaker, Andrew W. Cross, Frederic T. Chong, and Isaac L. Chuang. A quantum logic array microarchitecture: Scalable quantum data movement and computation. In *Proceedings of the 2005 International Symposium on Microarchitecture (MICRO-38)*, 2005.

[30] Darshan D. Thaker, Tzvetan Metodi, Andrew Cross, Isaac Chuang, and Frederic T. Chong. CQLA: Matching density to exploitable parallelism in quantum computing. In *Computer Architecture News, Proc. 33rd Annual International Symposium on Computer Architecture* [44].

[31] N. Cody Jones, Rodney Van Meter, Austin G. Fowler, Peter L. McMahon, Jungsang Kim, Thaddeus D. Ladd, and Yoshihisa Yamamoto. A layered architecture for quantum computing using quantum dots. arXiv:1010.5022v1 [quant-ph], October 2010.

[32] Mark G. Whitney, Nemanja Isailovic, Yatish Patel, and John Kubiatowicz. A fault tolerant, area efficient architecture for Shor's factoring algorithm. In *Proc. 36th Annual International Symposium on Computer Architecture*, June 2009.

[33] Dean Copsey, Mark Oskin, Tzvetan Metodiev, Frederic T. Chong, Isaac Chuang, and John Kubiatowicz. The effect of communication costs in solid-state quantum computing architectures. In *Proceedings of the fifteenth annual ACM Symposium on Parallel Algorithms and Architectures*, pages 65–74, 2003.

[34] Nemanja Isailovic, Mark Whitney, Yatish Patel, John Kubiatowicz, Dean Copsey, Frederic T. Chong, Isaac L. Chuang, and Mark Oskin. Datapath and control for quantum wires. *ACM Transactions on Architecture and Code Optimization*, 1(1), March 2004.

[35] Nemanja Isailovic, Yatish Patel, Mark Whitney, and John Kubiatowicz. Interconnection networks for scalable quantum computers. In *Computer Architecture News, Proc. 33rd Annual International Symposium on Computer Architecture* [44].

[36] Mark Oskin, Frederic T. Chong, Isaac L. Chuang, and John Kubiatowicz. Building quantum wires: The long and short of it. In *Computer Architecture News, Proc. 30th Annual International Symposium on Computer Architecture*. ACM, June 2003.

[37] Steven Balensiefer, Lucas Kregor-Stickles, and Mark Oskin. An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures. In *Proc. 32nd Annual International Symposium on Computer Architecture*, June 2005.

[38] Dean Copsey, Mark Oskin, Frederic T. Chong, Isaac Chuang, and Khaled Abdel-Gaffar. Memory hierarchies for quantum data. In *1st Workshop on Non-Silicon Computing*, 2002.

[39] Jungsang Kim and Changsoon Kim. Integrated optical approach to trapped ion quantum computation. *Quantum Information and Computation*, 9(2), 2009.

[40] S. J. Devitt, A. G. Fowler, A. M. Stephens, A. D. Greentree, L. C. L. Hollenberg, W. J. Munro, and Kae Nemoto. Architectural design for a topological cluster state quantum computer. *New Journal of Physics*, 11:083032, 2009.

[41] Simon J. Devitt, Austin G. Fowler, Todd Tilma, William J. Munro, and Kae Nemoto. Classical processing requirements for a topological quantum computing system. *International Journal of Quantum Information*, 2010.

[42] Rodney Van Meter, W. J. Munro, Kae Nemoto, and Kohei M. Itoh. Arithmetic on a distributed-memory quantum multicomputer. *ACM Journal of Emerging Technologies in Computing Systems*, 3(4):17, January 2008.

[43] ARDA. *A quantum information science and technology roadmap*, v2.0 edition, April 2004.

[44] ACM. *Computer Architecture News, Proc. 33rd Annual International Symposium on Computer Architecture*, June 2006.