A 2D Nearest-Neighbor Quantum Architecture for Factoring

Paul Pham
University of Washington
Quantum Theory Group
Box 352350, Seattle, WA 98195, USA,
ppham@cs.washington.edu,
`http://www.cs.washington.edu/homes/ppham/`

Krysta M. Svore
Microsoft Research
Quantum Architectures and Computation Group
One Microsoft Way, Redmond, WA 98052, USA
ksvore@microsoft.com,
`http://research.microsoft.com/en-us/people/ksvore/`

December 13, 2012

This document responds to comments by Referee 1, which were received on November 30, 2012. These comments are quoted and responded to below.

1 General Comments

> In this paper, the authors present in detail a new version of the modular exponentiation component of Shor's algorithm, with attention to the constraints of a 2-D planar graph of moderate degree for qubit connectivity. They use teleportation-based fanout to move qubits around within the machine. The modular addition uses a novel method, depending upon carry-save arithmetic and a small triangular lattice as its unit cell.
>
> Perhaps the most novel part of the arithmetic is the approach to modular multiplication. In a traditional modular multiplication of two numbers x and y, the n^2 bit-wise products $x_i y_j$ are calculated and laid out in a trapezoid shifted to give the correct column (power of two), such that each entry is $2^i 2^j x_i y_j$, which of course can be represented by a single bit in the right place. Then the columns

are added, creating a 2n-bit number, which then must be further operated on to perform the modulo N operation. (N is assumed to be an n-bit number with the high-order bit being a one.)

In contrast, in this approach, each of the n^2 entries is a full n-bit number, $(2^i 2^j \bmod m)x_i y_j$. By adding those numbers using their modular circuit, the full $xy \bmod N$ value can be calculated directly. As proposed, this requires $O(n^3)$ bits (qubits) in the register. By combining the n^2 partial results in a log-depth tree structure, the depth for a modular multiplication becomes $O(\log n)$ times the constant depth of their 3-to-2 carry-save modular operation.

Up through section 5, I was prepared to recommend nearly immediate acceptance of the paper. The writing is clear and elegant and the technical work both valuable and polished. In sections 6 and 7, however, I have some doubts about the technical work, and the writing seems a bit more rushed, and the tail of the paper is not yet satisfactory.

Most importantly, it is disappointing that the authors have not produced a more complete estimate of the number of qubits (resources) required, as well as the actual circuit depth. A rough estimate, at least, should be very achievable given the level of detail already developed in the paper. The authors mention this as future work, but without it, the value of the paper is substantially diminished, and it does not seem unreasonable to expect it to appear here.

2 Other significant technical comments:

The authors appear not to have noticed that half of the n^2 numbers in their multiplier require only a single bit. As long as $i + j \leq n$, the modulo operation results in the same number, allowing us to avoid using a full n-bit number: $(2^i 2^j \bmod m)x_i y_j = 2^i 2^j x_i y_j$.

3 Section 2.2

"this 'consumes' the cat state" – this sentence is confusing.

We meant that the cat state remains entangled with the original source qubit and its fanned-out, entangled copies. We have since discovered via personal conversation with Aram Harrow and Dan Browne that this statement is no longer true. It is possible to "un-fanout" and therefore disentangle the cat state from the soruce and target qubits, allowing us to reuse this state.

4 Section 3

> Choi and Van Meter were not the first to consider 2-D architectures; most of the solid-state proposals and even some of the ion trap proposals worked with a 2-D layout. Kielpinski's 2002 proposal might have been the first 2-D architecture. Working out exact algorithms on the structures came later, but papers by Kubiatowicz's group and Chong/Oskin clearly included at least some level of work on the movement of qubits in a planar system, albeit with less attention to the abstraction of logical qubit connectivity.

We've added a reference to the 2002 Nature paper by Kielpinski et al.
KRYSTA TODO: Can you add references to quantum architecture papers by Kubi/Chong/Oskin on 2D implementations, since you are probably more familiar with those works?

> I'm not sure a modular adder is "extended" to do modular exponentiation. "Composed" or "used", perhaps?

KRYSTA TODO

> Your citation of Gossett has no year.

KRYSTA TODO

> "all other factoring implementations"? That's a rather broad characterization. Cleve and Watrous long ago proposed using a parallel reduction tree of multiplications before the QFT. Van Meter and Itoh investigated in detail the tradeoffs in resource consumption for doing this.

We will clarify by referring to all other nearest-neighbor factoring implementations. By "implementations," we mean a concrete mapping to an architecture, such as those given by [Fowler et al. 2004] and [Kutin 2006]. While we do acknowledge the Cleve and Watrous paper, which gives similar parallel results to those in the Kitaev-Shen-Vyalyi book, both assume arbitrary connectivity of qubits.

> As far as I am aware, no one has worked out the details of Draper's transform adder taking into account the need to do Solovay-Kitaev decomposition. This may add a very large factor to the execution time.

Agreed, we are not aware of anyone working out compilation of the Draper transform adder to a universal gate set, such as the Clifford group plus $\pi/8$ gates. It's now known how to compile more efficiently than Solovay-Kitaev using more time and space, but it would still not be more efficient than the Gossett adder.

Do you think the Zalka approximate multiplier approach actually works?

We think it is likely that something similar to the Zalka approximate multiplier actually works in practice for the majority of input values, if not the exact implementation described in the Zalka and Kutin papers. However, a rigorous theoretical argument, or empirical verification by simulation, has not yet appeared in the literature.

While it's okay to include a "forward reference" to a forthcoming paper of your own that carries more detail, you can't ask us to "refer to" it!

I don't think the BKP "exact" circuit guarantees a result, does it? Shor's own original algorithm only probabilistically gives the correct answer, and that probability is a matter of some debate in the literature. Papers by Fowler (2004), Miquel (1996), Garcia-Mata (2007) and others provide different estimates.

The journal style will ultimately dictate this, but when the bibliographic labels in the text are alphabetic, the references are usually ordered alphabetically.

5 Section 4

It would be worth pointing out that qubit 0 is the low-order qubit.

"At the level of bits, a CSA..." this sentence is awkward, reword.

Fig. 4 shows the layout, but it doesn't exactly match the circuit of Fig. 3. Having the exact circuit to accompany Fig. 4 would be useful.

6 Section 5

It feels somewhat like the phrasing on constant depth is a bit misleading in this section. Please reread and make sure it is easy for the reader to follow your claims.

Proof of Lemma 1: "O.ur" typo.

A little attention to classical versus quantum addends in this section might help the reader.

Fig. 5: labeling the lines themselves as "Layer 1", "Layer 2", etc. might help the reader.

"at no layer generates" –¿ "no layer generates"?

Fig. 6: Swap the left and right ends of this figure to make it correspond to Fig. 5 as closely as possible, and point out this correspondence by also labeling the "layers" and the time axis here. "FANOUT RAIL" – of which variable(s)?

My estimate here is that spatial resources are 29n qubits, temporal resources are 12n Toffoli gates. Is that about right?

7 Section 6

If you are worried that "quantum number" will mislead some readers, would "quantum integer" be better?

Bottom of 6.1 is the place to expand the discussion to incorporate the above on the size of partial products and their impact on resource utilization.

Your white tiles are hard to see on my printout.

Important: Your second and third rules under "black tiles" appear to conflict. Reword to be precise.

Bottom of Sec. 6: more discussion of the size and number of addends and resource consumption is needed.

Fig. 10: In earlier figures, arrows were used only to indicate addend motion via teleportation, correct? A different symbol to indicate actual multiplication would make the figure clearer. Otherwise, asserting that an arrow itself is "log n depth" will confuse the reader. The tree approach used in this figure is not original, and in fact dates to the 2000 paper by Cleve and Watrous, at least.

We do not claim that this tree structure is original, merely that it is now possible to do the necessary communication (teleportation and fan-out) in constant depth. That is, the tree structure before was a theoretical construction, and now based on the previous nearest-neighbor implementations of modular arithmetic, correspond to an actual physical tree structure if this architecture were to be fabricated and observed to run over time.

Sections 7, 8 and 9 should be extended with more discussion, and actual resource consumption figures. Some of the papers cited in Fig. 11 (which ought to be a table, not a figure) provide detailed estimates on resources (depth and width), as do Beckman et al. (1996, uncited here, but should be) and Van Meter and Itoh. Both of those latter papers offer several configurations that may make direct comparison tricky.

Overall, as noted, this paper will be a valuable contribution to the literature once these issues are addressed.