



OPEN

Factoring 51 and 85 with 8 qubits

Michael R. Geller & Zhongyuan Zhou

SUBJECT AREAS:

QUANTUM
INFORMATION

QUBITS

Received
21 August 2013

Accepted
3 October 2013

Published
28 October 2013

Correspondence and
requests for materials
should be addressed to
M.R.G. (mgeller@uga.
edu)

Department of Physics and Astronomy, University of Georgia, Athens, Georgia 30602, USA.

We construct simplified quantum circuits for Shor's order-finding algorithm for composites N given by products of the Fermat primes 3, 5, 17, 257, and 65537. Such composites, including the previously studied case of 15, as well as 51, 85, 771, 1285, 4369, ... have the simplifying property that the order of a modulo N for every base a coprime to N is a power of 2, significantly reducing the usual phase estimation precision requirement. Prime factorization of 51 and 85 can be demonstrated with only 8 qubits and a modular exponentiation circuit consisting of no more than four CNOT gates.

Shor's prime factoring algorithm¹ reduces the factorization of a product $N = pp'$ of distinct odd primes p and p' to that of finding the order r of $a \bmod N$ for a randomly chosen base a coprime to N (with $1 < a < N$), which can be performed efficiently with a quantum computer. The standard implementation² factors a b -bit number with $3b$ qubits using a circuit of depth $O(b^3)$; alternative modular exponentiation circuits can be used to reduce either the space (qubit number)³ or time⁴ requirements. The case $N = 15$, which has the simplifying property that all orders are powers of 2, has been demonstrated experimentally by several groups^{2,5-8}. Recent experiments have also factored $N = 21$ ^{9,10} and 128¹¹.

In this paper we consider the application of Shor's algorithm to products of special primes of the form

$$p_k \equiv 2^{2^k} + 1 \quad \text{with } k = 0, 1, 2, 3, 4. \quad (1)$$

Explicitly,

$$p = 3, 5, 17, 257, \text{ and } 65537. \quad (2)$$

Fermat proposed that numbers of the form $2^{2^k} + 1$ for any $k = 0, 1, 2, \dots$, (called Fermat numbers) are prime; however it is now known that the Fermat numbers with $5 \leq k \leq 32$ are not prime, and it is not known whether there are additional primes of this form for larger values of k .

Products of the form

$$\begin{aligned} N = p_k p_{k'} &= (2^{2^k} + 1)(2^{2^{k'}} + 1), \quad \text{with } k, k' \in \{0, 1, 2, 3, 4\} \quad \text{and } k \neq k' \\ &= 15, 51, 85, 771, 1285, 4369, 196611, 327685, 1114129, \quad \text{and } 16843009 \end{aligned} \quad (3)$$

have the special property that the order of $a \bmod N$ for every base a coprime to N is a power of 2. This follows from Euler's theorem,

$$a^{\phi(y)} \bmod y = 1, \quad (4)$$

where y is a positive integer, $\phi(y)$ is the number of positive integers less than y that are coprime to y , and $\gcd(a, y) = 1$. When p and p' are odd primes, all $pp' - 1$ positive integers less than pp' are coprime to pp' except for the $p - 1$ multiples of p' and the $p' - 1$ multiples of p , and these exceptions are distinct, so

$$\phi(pp') = pp' - 1 - (p - 1) - (p' - 1) = (p - 1)(p' - 1). \quad (5)$$

(This result also follows from Euler's product formula.) Thus,

$$a^{(p-1)(p'-1)} \bmod pp' = 1. \quad (6)$$

Recall that the order r of $a \bmod N$ is the smallest positive integer x satisfying $a^x \bmod N = 1$; therefore for a composite of the form (3),

$$\phi(N) = (p_k - 1)(p_{k'} - 1) = 2^{2^k + 2^{k'}} \quad (7)$$

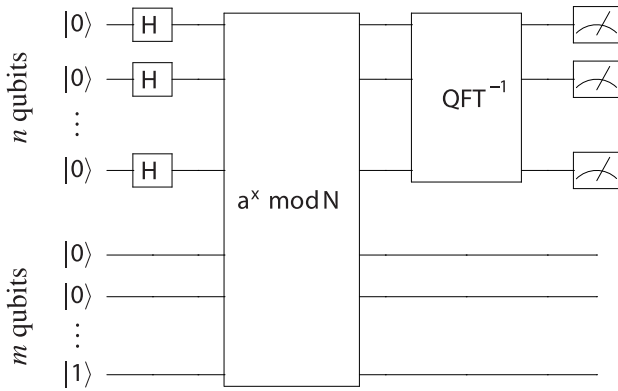


Figure 1 | Basic quantum circuit for order finding. Here $n = 2b$ and $m = b$, where $b \equiv \lceil \log_2 N \rceil$ is the number of bits in N .

must be a *multiple* of r . Because r must be an integer, we conclude that for any $1 < a < N$ with $\text{gcd}(a, N) = 1$, r is a power of 2 as well.

Results

The standard² order-finding circuit is shown in Fig. 1. The first register has n qubits and the second has m . The modular exponentiation operator in Fig. 1 acts on computational basis states as

$$|x_1 x_2 \cdots x_n\rangle \otimes |0 \cdots 1\rangle \rightarrow |x_1 x_2 \cdots x_n\rangle \otimes |a^x \bmod N\rangle, \tag{8}$$

where

$$x = \sum_{j=1}^n 2^{n-j} x_j. \tag{9}$$

After the inverse quantum Fourier transform, measurement of the first register is done in the diagonal basis. The probability to observe the value

$$x \in \{0, 1, \dots, 2^n - 1\} \tag{10}$$

is

$$\text{prob}(x) = \frac{\sin^2(\pi r x A / 2^n)}{2^n A \sin^2(\pi r x / 2^n)}, \tag{11}$$

where r is the order and A is the number of distinct values of x such that $a^x \bmod N$ has the same value (this is approximately $2^n/r$). This probability distribution has peaks at integer values of x near

$$j \times \frac{2^n}{r} \quad \text{with } j = 0, 1, \dots, r-1. \tag{12}$$

The number of qubits n in the first register is chosen to enable reliable extraction of the value of r in (12), which depends on whether or not r is a power of 2. In actual applications of Shor’s algorithm this will not be known, of course, as the point of the quantum algorithm is to determine r . In this usual situation, measurement will yield (with $\text{prob} > 4/\pi^2$) an x satisfying

$$\left| \frac{x}{2^n} - \frac{j}{r} \right| \leq \frac{1}{2^n} \quad \text{with } j \in \{0, 1, \dots, r-1\}. \tag{13}$$

By choosing $n = 2b$ qubits in the first register, where $b \equiv \lceil \log_2 N \rceil$, we are guaranteed that j/r will be a (continued fraction) convergent of $x/2^n$. However, for the family of composites $N = (2^{2^k} + 1)(2^{2^k} + 1)$ considered here, all bases have orders

$$r = 2^\ell \quad \text{with } \ell \in \{1, 2, 3, \dots, \ell_{\max}\}, \tag{14}$$

where the value of ℓ_{\max} is discussed below. In this case

$$A = \frac{2^n}{r} \tag{15}$$

and the peaks (12) in (11) occur at integral values

$$x = 0, 2^{n-\ell}, 2 \times 2^{n-\ell}, \dots, (r-1) \times 2^{n-\ell}. \tag{16}$$

Therefore, as long as we have

$$n = \ell_{\max} \tag{17}$$

qubits in the first register we will be able to determine r , possibly after a small number of repetitions. The simplest way to extract r from x here (assuming $x \neq 0$) is to simplify the ratio

$$\frac{x}{2^n} \tag{18}$$

down to an irreducible fraction, which will yield both j and r [recall (12)] unless they have happen to have a common factor.

Next we discuss the value of ℓ_{\max} (which determines the largest order $2^{\ell_{\max}}$) for a given composite N . We do not have an explicit formula for ℓ_{\max} . However, when N is a product of distinct odd primes, r can be as large as $\phi(N)/2$ (ref. 12), so for an N of the form (3) we have the bound [see (7)]

$$\ell_{\max} \leq 2^k + 2^{k'} - 1. \tag{19}$$

For example, in the case of $N = 51$ ($k = 0, k' = 2$), the largest order is $2^4 = 16$, and the upper bound is realized. However for $N = 85$ ($k = 1, k' = 2$), it is not (the largest order present is 16, not 32).

The second register stores the values of

$$a^x \bmod N \in \{0, 1, \dots, N-1\} \tag{20}$$

and therefore normally requires b qubits. However, for a given a , only r of these values are distinct. Thus we can use fewer than b qubits. This simplification, while not essential, has been used in all gate-based factoring demonstrations to date. The reduction amounts to computing a table of values of $a^x \bmod N$ classically for a given base a , constructing a corresponding quantum circuit, and ignoring or eliminating unused qubits in the second register. We note that in addition to being unscalable, this method of constructing the modular exponentiation operator implicitly or explicitly uses the value of the order r , i.e., the answer which the quantum computation is supposed to determine; we discuss this issue further below.

In this work we will adopt an equivalent—but perhaps more systematic and transparent—modular exponentiation circuit construction: We follow the output of $a^x \bmod N$ by a second transformation

$$\begin{matrix} 1 & 0 \\ a \bmod N & 1 \\ a^2 \bmod N & \rightarrow 2 \\ \vdots & \vdots \\ a^{r-1} \bmod N & r-1 \end{matrix}, \tag{21}$$

which maps the r distinct values of $a^x \bmod N$ to $0, 1, \dots, r-1$. In (21) we assume that $1 < a < N$. We refer to this classical pre-processing of $a^x \bmod N$ as *compression*. Compression does not adversely affect the operation of the order-finding circuit, but reduces m from b to ℓ_{\max} in a systematic manner (and generalizes the “full compilation” method used in Ref. 6.)

Any set of r distinct non-negative integers—in any order—could be used for the output of the compression map (21). However the choice employed here, and indicated in (21), is especially simple because it can be compactly written as

$$a^x \bmod N \rightarrow x \bmod r(a). \tag{22}$$

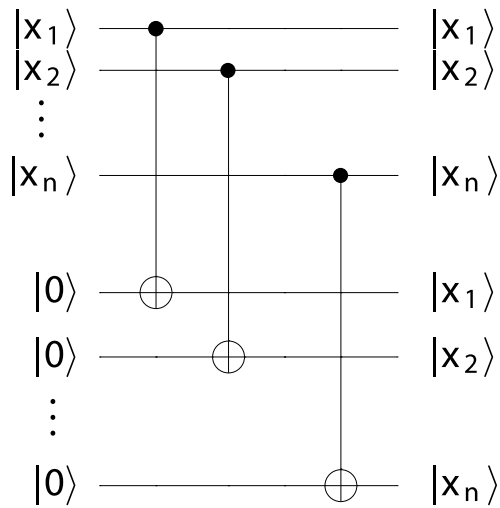


Figure 2 | Circuit to copy the first register to the second.

Then, after changing the initial state of the second register from $|00 \dots 1\rangle$ to $|00 \dots 0\rangle$, we have, instead of (8), the compressed modular exponentiation operation

$$|x\rangle \otimes |0 \dots 0\rangle \rightarrow |x\rangle \otimes |x \bmod r\rangle. \quad (23)$$

The operation (23) without the modulo r is just the bit-wise COPY shown in Fig. 2 and the effect of the modulo r is to only copy the $\log_2 r$ least significant bits.

In conclusion, we require ℓ_{\max} qubits in each register, for a total of $2\ell_{\max}$ qubits. ℓ_{\max} can either be computed classically or the bound (19) can be used. We note that the space requirements can be further reduced by using iterative phase estimation^{13–15}, but with an increase in circuit depth. This might be useful for ion-trap and optical realizations but probably not for superconducting qubits.

We provide explicit quantum circuits for the cases of $N = 51$ and 85. In both cases $\ell_{\max} = 4$ (the largest order is 16), so we require $n = 4$ qubits in the first register and $m = 4$ in the second, for a total of 8 qubits. This is significantly fewer than the $3b$ required for general b -bit numbers ($b = 6$ when $N = 51$ and $b = 7$ when $N = 85$). It is also fewer than the $2b + 3$ qubits required by Beauregard³.

After the compression discussed above, only four different circuits are needed to cover all $N = 51$ and $N = 85$ cases, because there are four possible orders. The assignments are listed in Tables I and II, and the circuits are given in Figs. 3a–d.

Circuits for the remaining composites 771, ..., 16843009 follow from the method described above, and require no more than $2\ell_{\max}$ qubits. From (19) we find that the total number of qubits required in these cases is bounded by

$$771 (16), 1285 (18), 4369 (22), 196611 (32), \\ 327685 (34), 1114129 (38), 16843009 (46). \quad (24)$$

Table I | $N = 51$ quantum circuits. The base marked by an asterisk satisfies $a^{r/2} = -1 \pmod N$ and will result in a factorization failure in the classical post-processing analysis

base a	circuit
16, 35, 50*	Fig. 3a
4, 13, 38, 47	Fig. 3b
2, 8, 19, 25, 26, 32, 43, 49	Fig. 3c
5, 7, 10, 11, 14, 20, 22, 23, 28, 29, 31, 37, 40, 41, 44, 46	Fig. 3d

Discussion

Given the considerable interest in experimental demonstrations of Shor's algorithm, it is reasonable to ask what constitutes a "genuine" demonstration of this important algorithm, and whether the cases presented here should be considered as such. In our opinion a genuine implementation should use no knowledge of the value of the order r —including whether or not it is a power of two—because the objective of the quantum stage of the algorithm is to calculate r . Therefore we do not regard the factorization of products of Fermat primes to be genuine implementations of Shor's algorithm. Moreover, such special cases can be efficiently factored classically, by comparing N against a list of products of these primes.

However we do view the circuits presented here as quasi-legitimate implementations of quantum order finding, and in our view they are still interesting for this reason. In particular, each eight-qubit circuit presented here is able to detect periods of two, four, eight, and sixteen, so there are failure modes where an incorrect period could be observed. But these genuine order-finding instances are nongeneric cases from the perspective of Shor's algorithm. Note that in this work we have simplified the modular exponentiation circuits to reduce their depth. It is also possible to implement uncompiled versions, which do not make any use of the value of r and which would constitute a fully genuine implementation of order-finding (but not of factoring). The main point of this work, that the number of qubits required in the first register is greatly reduced for composites in the series (3), applies to either approach.

Smolin, Smith, and Vargo¹⁶ recently addressed the question of what should constitute a genuine factoring demonstration by simplifying the entire order-finding circuit for any product of distinct odd primes down to only *two* qubits. This is possible by implementing the phase estimation iteratively^{13–15} (or the Fourier transform semiclassically¹⁷), and by choosing only bases a with order two. Smolin *et al.*¹⁶ show that *with knowledge of the factors*, it is always possible to find an order-two base, and they provide an algorithm for doing so. The circuit of Smolin *et al.* does not constitute a genuine implementation of Shor's algorithm either. However, the focus of our work is different than that of Ref. 16, as the circuits presented here are still quasi-legitimate implementations of order finding, and we do not make explicit use of the factors in simplifying the circuits.

Finally, we note that the $r = 16$ cases (Fig. 3d) result in a uniform probability distribution for observing computational basis states $|x\rangle$ after measurement of the first register, which would also result from

Table II | $N = 85$ quantum circuits. Bases marked by an asterisk satisfy $a^{r/2} = -1 \pmod N$ and result in factorization failures in the classical post-processing analysis

base a	circuit
16, 69, 84*	Fig. 3a
4, 13*, 18, 21, 33, 38*, 47*, 52, 64, 67, 72*, 81	Fig. 3b
2, 8, 9, 19, 26, 32, 36, 42, 43, 49, 53, 59, 66, 76, 77, 83	Fig. 3c
3, 6, 7, 11, 12, 14, 22, 23, 24, 27, 28, 29, 31, 37, 39, 41, 44, 46, 48, 54, 56, 57, 58, 61, 62, 63, 71, 73, 74, 78, 79, 82	Fig. 3d

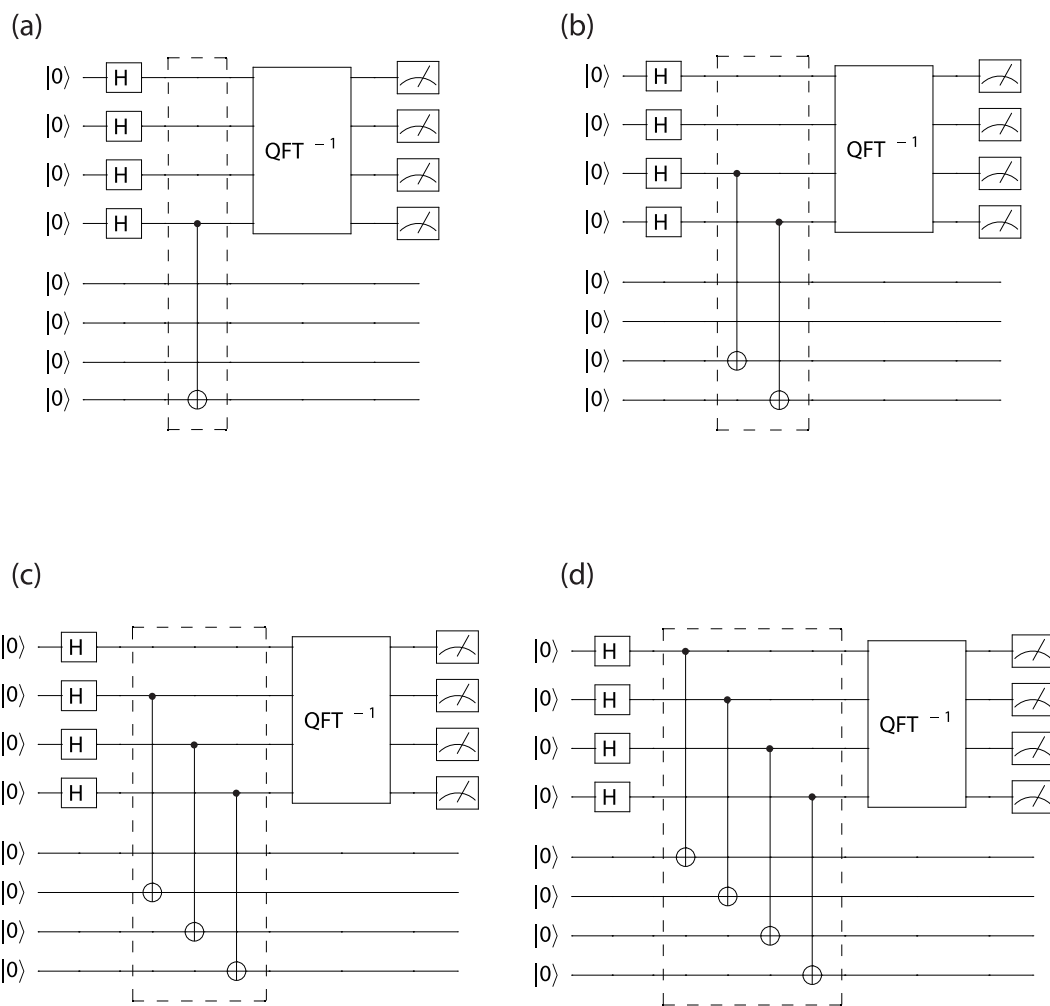


Figure 3 | Quantum circuits for factoring 51 and 85. Note the modification of the input to the last qubit of the second register compared with Fig. 1. The circuits inside dashed boxes are the compressed modular exponentiation operations discussed in the text. Note that the CNOT gates here can be executed in parallel.

an unintended, purely decohering action of the CNOT gates (we thank Alexander Korotkov for this observation). One method of verifying that the circuit is functioning correctly is to perform tomography on the final state. A simpler method, however, is to change the

input of the second register from $|0\rangle^{\otimes 4}$ to $|+\rangle^{\otimes 4}$, as shown in Fig. 4. If the gates are purely decohering, this will not change the output of the first register upon measurement. But if the CNOTs are acting ideally, the entire compressed modular exponentiation operator now acts as the identity [because $|+\rangle \equiv 2^{-1/2}(|0\rangle + |1\rangle)$ is an eigenvector of the NOT gate] and can be effectively dropped from the circuit, leading to an observation of the final state $|0000\rangle$ with unit probability.

In conclusion, we have shown that the simple and well-studied case of factoring $N = 15$ is the first in a series of cases

$$15, 51, 85, 771, 1285, 4269, \dots \quad (25)$$

that have all orders equal to a power of two and that can be factored with fewer resources than that of other products with the same number of bits.

Methods

The results in Tables I and II are found by classically computing the orders r for all bases $1 < a < N$ satisfying $\text{gcd}(a, N) = 1$. Cases where $a^{r/2} \equiv -1 \pmod N$ lead to a failure of Shor’s algorithms and are marked by an asterisk. The operation (23) is then implemented by a quantum circuit as described above.

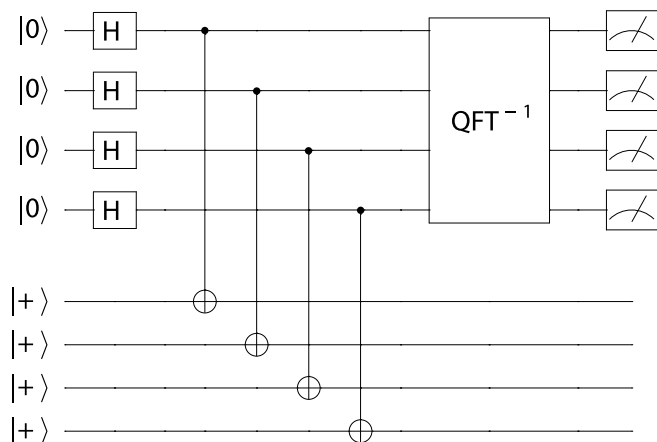


Figure 4 | Changing the input states on the second register to verify coherent operation of the CNOT gates.

1. Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **26**, 1484–1509 (1997).



2. Vandersypen, L. M. K. *et al.* Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature (London)* **414**, 883–887 (2001).
3. Beauregard, S. Circuit for Shor's algorithm using $2n + 3$ qubits. *Quant. Info. Comput.* **3**, 175–185 (2003).
4. Zalka, C. Fast versions of Shor's quantum factoring algorithm. arXiv:quant-ph/9806084.
5. Lu, C.-Y., Browne, D. E., Yang, T. & Pan, J. W. Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits. *Phys. Rev. Lett.* **99**, 250504 (2007).
6. Lanyon, B. P. *et al.* Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement. *Phys. Rev. Lett.* **99**, 250505 (2007).
7. Politi, A., Matthews, J. C. F. & O'Brien, J. L. Shor's Quantum Factoring Algorithm on a Photonic Chip. *Science* **325**, 1221–1221 (2009).
8. Lucero, E. *et al.* Computing prime factors with a Josephson phase qubit quantum processor. *Nat. Phys.* **8**, 719–723 (2012).
9. Peng, X. *et al.* Quantum Adiabatic Algorithm for Factorization and Its Experimental Implementation. *Phys. Rev. Lett.* **101**, 220405 (2008).
10. Martin-Lopez, E. *et al.* Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nat. Photonics* **6**, 773–776 (2012).
11. Xu, N. *et al.* Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System. *Phys. Rev. Lett.* **108**, 130501 (2012).
12. Gerjuoy, E. Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers. *Am. J. Phys.* **73**, 521–540 (2005).
13. Mosca, M. & Ekert, A. The hidden subgroup problem and eigenvalue estimation on a quantum computer. *Lect. Notes Comput. Sci.* **1509**, 174–188 (1999).
14. Parker, S. & Plenio, M. B. Efficient Factorization with a Single Pure Qubit and $\log N$ Mixed Qubits. *Phys. Rev. Lett.* **85**, 30493052 (2000).
15. Dobsicek, M., Johansson, G., Shumeiko, V. & Wendin, G. Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark. *Phys. Rev. A* **76**, 030306(R) (2007).
16. Smolin, J. A., Smith, G. & Vargo, A. Oversimplifying quantum factoring. *Nature (London)* **499**, 163–165 (2013).
17. Griffiths, R. B. & Niu, C.-S. Semiclassical Fourier Transform for Quantum Computation. *Phys. Rev. Lett.* **76**, 32283231 (1996).

Acknowledgements

This research was funded by the US Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the US Army Research Office grant No. W911NF-10-1-0334. All statements of fact, opinion, or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI, or the US Government.

We thank Joydip Ghosh for useful discussions and for finding a mistake in a previous version of this manuscript.

Author contributions

M.G. and Z.Z. developed the theory, designed the circuits, and wrote the manuscript. All authors reviewed the manuscript.

Additional information

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Geller, M.R. & Zhou, Z. Factoring 51 and 85 with 8 qubits. *Sci. Rep.* **3**, 3023; DOI:10.1038/srep03023 (2013).



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0>