

ARTICLE

Received 31 Jan 2013 | Accepted 30 Aug 2013 | Published 3 Oct 2013

DOI: 10.1038/ncomms3524

Requirements for fault-tolerant factoring on an atom-optics quantum computer

Simon J. Devitt¹, Ashley M. Stephens¹, William J. Munro^{1,2} & Kae Nemoto¹

Quantum information processing and its associated technologies have reached a pivotal stage in their development, with many experiments having established the basic building blocks. Moving forward, the challenge is to scale up to larger machines capable of performing computational tasks not possible today. This raises questions that need to be urgently addressed, such as what resources these machines will consume and how large will they be. Here we estimate the resources required to execute Shor's factoring algorithm on an atom-optics quantum computer architecture. We determine the runtime and size of the computer as a function of the problem size and physical error rate. Our results suggest that once the physical error rate is low enough to allow quantum error correction, optimization to reduce resources and increase performance will come mostly from integrating algorithms and circuits within the error correction environment, rather than from improving the physical hardware.

¹National Institute for Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. ²NTT Basic Research Laboratories, NTT Corporation, 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198, Japan. Correspondence and requests for materials should be addressed to S.J.D. (email: devitt@nii.ac.jp).

The prospect of an entirely new industry based on quantum mechanics has motivated technological development and led to a much better understanding of the principals governing our universe at the atomic scale. For quantum technology, experimental progress has been pronounced^{1–6}. Not only has a fledgling industry based on quantum key distribution emerged^{7,8}, but many experimental groups now routinely demonstrate the creation, manipulation and measurement of multiple qubits in various physical systems with increasingly higher accuracy⁹. The goal of developing a viable, large-scale quantum computer is now in sight, due in part to recent theoretical progress. Techniques for fault-tolerant quantum error correction, necessary to deal with imperfect physical components, have been substantially refined^{10–12}. Adaptation of these techniques to the physical restrictions of quantum hardware has led to multiple architecture designs, each one representing a pathway towards a future quantum computer^{13–21}.

Although a large-scale quantum computer is still years away, it is now possible to make qualitative and quantitative predictions about the performance and required hardware resources for such a computer. Some estimates consider architectures based on specific physical systems^{20–24}, an essential aspect in resource analysis. However, these estimates omit a full prescription for executing the algorithms in question. Others consider promising error-correction codes and circuits, such as post-selection¹⁰ and topological models¹², yet do so without reference to particular architectures or applications. Above the physical device level, there are a number of implementation layers required to reliably execute a quantum algorithm. By careful choice of all technological elements and the integration of all implementation layers, a complete analysis is now possible.

A full account of the resources required for fault-tolerant quantum computation must consider several factors. Because physical components will fail and introduce errors, an error-correction code must be chosen and then adapted to any restrictions imposed by the hardware. Physical error rates must be suppressed below the fault-tolerant threshold of the chosen code. Next, the code restricts the set of logical gates that can be directly applied to encoded data. Each gate in the high-level quantum algorithm must be decomposed into a universal set of fault-tolerant primitives. To realize these universal primitives, ancillary states and protocols are typically required to enact teleported gates that could otherwise not be directly applied to the encoded data²⁵. Each of these steps increases the total qubit/time overhead and must be carefully integrated in a way that ensures all steps are counted.

The precise details of how resources are calculated depend on the properties of the architecture in question, the techniques used for fault-tolerant error correction and the circuit-based description of the desired algorithm. One of the most promising error correction techniques is topological error correction; defined over a large three-dimensional cluster state of qubits¹². This error correction technique, despite the fact that it is well suited to large-scale architectures, has only briefly been studied in regards to how a large-scale algorithm will actually be implemented.

In a scalable quantum computer, high-level quantum programming languages will be required to control quantum algorithms^{26–29}, and effective methods will be required to synthesize logical circuits for these algorithms^{30–32}. Logical circuits will need to be compiled to a restricted set of fault-tolerant operations available in the topological model. Finally, circuits will need to be expressed in terms of physical operations in the quantum computer—in the topological model, this requires the construction of compact braided logic. These steps have a direct impact on the physical resources needed for computation, and a concerted effort to understand these requirements has only

just begun³³. Typically, estimates consider the number of required gates in the high-level quantum algorithm and the basic amount of ancillary space needed for additional fault-tolerant protocols^{20–22}. However, these estimates provide only a partial analysis. Error correction codes inevitably suffer from constraints that need to be taken into account; for example, the interaction of qubits required by the actual algorithm with qubits needed for ancillary protocols. The scheduling and routing of these ancillary protocols are often overlooked when estimating resources and are likely to dramatically affect the resource estimates^{34,35}.

The compatibility of the topological model with hardware architectures has been demonstrated in Devitt *et al.*¹⁸, Yao *et al.*¹⁹, Meter *et al.*²⁰ Jones *et al.*²¹ In our complete analysis, we consider an atom-optics architecture^{18,36}, based on the photonic module³⁷. The photonic module is a relatively simple device that allows an atomic qubit to mediate the generation of photonic entanglement. The three-dimensional cluster state that supports topological error correction is created by an array of these devices. Decomposition of each logical gate into a series of physical operations in this architecture is straightforward, and hence all geometry and connectivity constraints at both logical and physical levels can be explicitly included in the analysis.

The desired algorithm, Shor's algorithm, is a comparatively simple application compared with other problems solvable by a quantum computer^{24,38}. More importantly, it has a rich history of theoretical development and a number of explicit circuit constructions^{39–41}. In the present study, we begin with a suitable circuit from the literature and further compile the circuit to one that is fault-tolerant and error corrected. We choose an explicit construction amenable to the system design described above. However, to run the circuit, we still have to take the geometric constraints at the logical level into account. Even though scheduling at the physical level is taken care of by the topological model, scheduling and arrangement of gates and ancillary operations within the logical space impacts performance. This step is largely unexplored and leaves substantial room for optimization. The ability of an error-corrected computer to achieve the optimal circuit size at the logical level is dependent on adapting to these constraints, hence estimates should be made with care. Our results show that factoring a 1,024-bit number requires ~ 2.3 years of computational time and ~ 1.9 billion quantum devices. These resources are higher than other estimates; however, the numbers are an accurate reflection of the quantum circuit and the error correction model. Our results illustrate that the sophisticated techniques in quantum compilation must be developed and that these techniques can significantly reduce the resources required to execute large-scale quantum algorithms.

Results

Preliminaries. In the topological cluster state model, a three-dimensional cluster forms the effective Hilbert space in which computation takes place^{12,42}. In the architecture, a photonic cluster state is continuously prepared from non-entangled photons by the hardware. Logical qubits are introduced as pairs of defects in the cluster. Defects are created in the cluster by measuring physical qubits in the Z-basis¹². An entangling gate is realised by braiding pairs of defects. Logical errors occur when chains of physical errors connect or encircle defects, which is made less likely by increasing the circumference of defects and by increasing their separation. Physical qubits in the bulk of the cluster, those not associated with defects, are measured in the X-basis. These measurements reveal the endpoints of chains of errors, from which the most likely set of errors can be inferred. To

estimate physical resources, we are ultimately interested in the size of the three-dimensional cluster state required to execute Shor's algorithm. As the algorithm is executed at the logical level, it is useful to introduce a scale factor that essentially encapsulates the overhead associated with error correction¹². A logical cell is defined as a three-dimensional volume of the cluster that has an edge length of $d + d/4$ unit cells, where d is the distance of the error-correction code. Defects have circumference of d unit cells and are separated by d unit cells (illustrated in Fig. 1).

Shor's algorithm. There are a number of different circuit implementations for this algorithm^{39–41}, which assume that arbitrary sets of qubits can be simultaneously entangled without any penalty related to their separation. This is not the case in the topological model, as gates are realised by braiding defects, which occupy physical space in the cluster. The scheduling required to implement multiple gates over long distances in the same time step is non-trivial and will likely add significant overhead.

A simpler approach is to use a circuit that has already been modified to require only nearest neighbour gates in some restricted geometry. We have chosen to use the Beauregard circuit, which is a linear nearest neighbour (LNN) construction with swap gates used to rearrange qubits in the circuit^{43,44}. This circuit is not as efficient as others, but its explicit LNN construction means it can be applied in the topological model without modification. The Beauregard circuit performs modular exponentiation in Fourier space, which is usually approximated, while modular exponentiation based on reversible Boolean circuits is exact⁴⁵. However, the complicated nature of fault-tolerant protocols makes the manual design and optimization of Boolean addition intractable. With logical qubits arranged in a line, the circuit to factor an L -bit number requires $Q = 2L$ qubits and has depth $K = 32L^3$ to the leading order. The circuit is not inherently robust to errors⁴⁶, requiring an error rate per gate $\sim 10^{-1}/KQ = 10^{-1}/64L^4$ to ensure a 90% chance of success.

Topological circuits. The Methods section details the steps that decompose Shor's algorithm into a valid set of operations that can

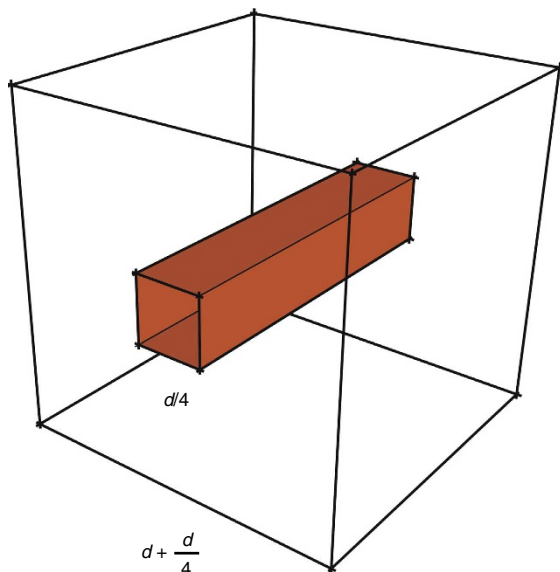


Figure 1 | A logical cell. The size of topological quantum circuits are expressed in terms of logical cells, giving an error correction-independent measure. The lengths are expressed in terms of unit cells of the cluster state. The defect is the coloured region within the cell. Pairs of defects form a logical qubit.

be performed in the topological model. We can translate this decomposed circuit into a sequence of braids (a topological circuit) in the three-dimensional cluster state. In the Methods section, we detail the geometric structures of all the valid topological operations; initialization, measurement, state injection, teleportation and the CNOT gate. Decomposing Shor's algorithm with respect to these gates converts the algorithm into a sequence of $R_z(\pi/8)$ rotations, which dominate the resource costs of the algorithm. This is the logical gate that will be designed. Shown in Fig. 2 is the topological circuit for the logical $R_z(\pi/8)$ rotations with one and two levels of concatenated state distillation. The full details underlying the topological circuit in Fig. 2 can be found in Supplementary Figs S1–S8 and the Supplementary Discussion. The topological circuits are compressed manually into cuboids such that they can be stacked tightly in both the spatial and temporal directions in the cluster. The algorithmic qubits (the qubits specified in the Beauregard circuit) are the defects marked as green (two defects per algorithmic qubit occupying a cross-sectional area of two logical cells). Immediately above each algorithmic qubit is an empty region of the cluster—this space is used for CNOT gates needed in the Beauregard circuit and swap gates required by the Beauregard circuit. Because the layout of algorithmic qubits in the cluster mimics the LNN aspect of the Beauregard circuit, no optimization is required to achieve the original circuit depth. Above the empty region is the distillation space for $|Y\rangle$ states, required to implement a $R_z(\pi/4)$ correction for each applied $R_z(\pi/8)$ gate and for Hadamard operations. Below the algorithmic qubits is the distillation space for $|A\rangle$ states, Fig. 3 illustrates.

At one level of concatenation, each algorithmic qubit has a dedicated $|A\rangle$ and $|Y\rangle$ state distillery. As the algorithmic layer is linear, these distilleries connect from above and below in the cluster, where direct connections in the topological model correspond to teleported gates¹². At two levels of concatenation, the topological circuit encapsulates four algorithmic qubits. The first concatenation level has physical injection points for low fidelity $|A\rangle$ and $|Y\rangle$ states, and the size of the defects is half of what is required at the algorithmic layer. This reduction in size and separation of defects for the first concatenation level reflects the fact that distillation circuits have a residual error. If the error of an injected state at the physical level is $O(10^{-3}-10^{-4})$, then implementing full-strength error correction for these circuits is redundant, as the residual error from distillation will always dominate. At the second layer of concatenation, the residual error becomes commensurate with the required logical error needed for computation. Therefore, after the first layer of concatenation, defects are expanded and separated to the same size and distance as in the required error correction for the algorithm. Additionally, at the second level of the $|A\rangle$ state distillation circuit, we require the distilled $|Y\rangle$ state at the first level of concatenation. The appropriate circuits are placed in the relevant free space in the cluster, as required.

The application of corrective $R_z(\pi/4)$ rotations for $|A\rangle$ state distillation and the probabilistic nature of the circuits themselves are compensated for at the second level of concatenation by placing extra distilleries in the cluster (see the Supplementary Discussion section). For $|Y\rangle$ state distillation, there is sufficient space for one extra circuit adjacent to the second-level circuit, to compensate for any one failure at level one. For $|A\rangle$ state distillation, there is space for two extra circuits within the cuboid to compensate for a given circuit failure. These circuit failures occur with probability of $O(p)$, where p is the error associated with the injected states. Given the additional level one circuits that we have incorporated into the gate and assuming p is of $O(10^{-3}-10^{-4})$, we will have too many failures at level one with a probability of $O(10^{-5}-10^{-7})$ for $|Y\rangle$ states and of

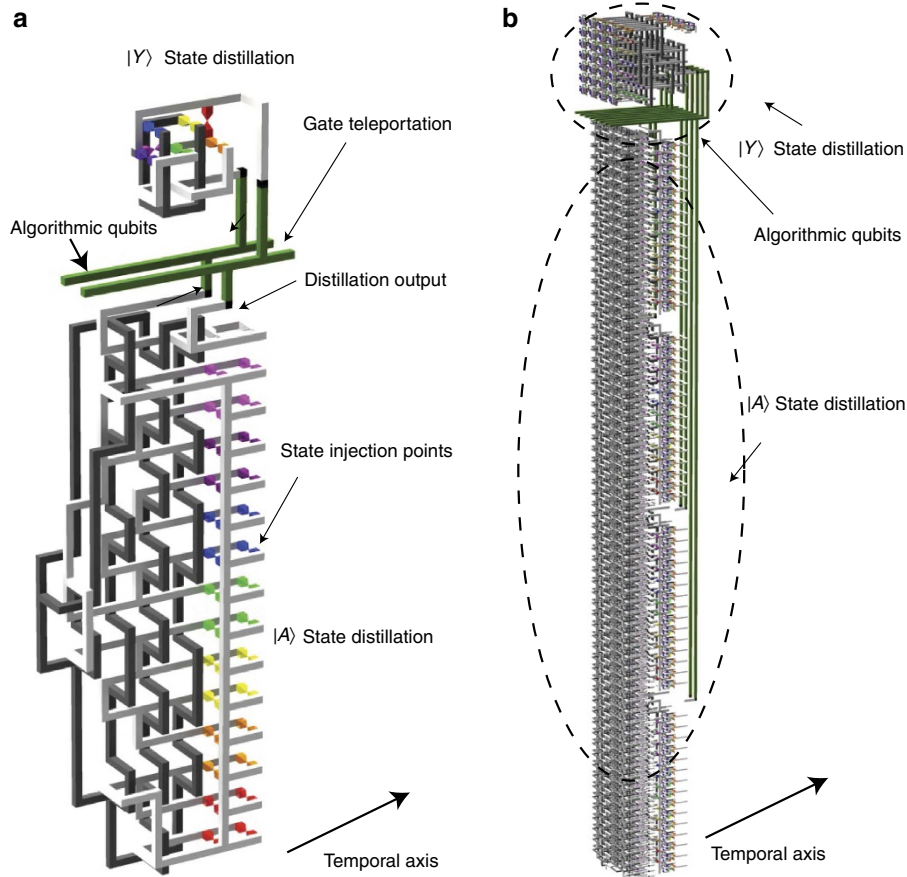


Figure 2 | Topological circuits for a $R_z(\pi/8)$ rotation in the topological cluster. Panel **a** shows one level of concatenated state distillation, whereas panel **b** is for two levels. The temporal axis in the cluster is illustrated. For a detailed explanation of these constructions see Supplementary Figs S3–S8 and the Supplementary Discussion. Qubits that are part of the algorithmic circuit for Shor’s algorithm are illustrated in green and represent a small fraction of the total circuit. The white and dark structures represent primal and dual defects in the topological cluster whereas the coloured pyramids are injection points corresponding to the circuit elements illustrated in Supplementary Fig. S2. The volumes and depths of these two circuits are $V = \{210, 1386\}$ and $D = \{5, 9\}$ logical cells, respectively.

$O(10^{-7} - 10^{-9})$ for $|A\rangle$ states, respectively. Therefore, we expect that we will not have sufficient first-level states every $10^5 - 10^7$ logical gates. Although these failures lead to an increase in circuit depth, they occur infrequently enough to be neglected. Finally, at the second level of state distillation, a total of 15 first-level circuits for corrective $|Y\rangle$ states, needed by the second-level $|A\rangle$ state circuit, are used. The probability that not enough level-one $|Y\rangle$ states are available is given by $15p/2^{15}$. This occurs when all 15 $R_z(\pi/8)$ corrections are needed, and a single level-one $|Y\rangle$ distillation failure occurs, and if p is of $O(10^{-3})$, this is also of $O(10^{-7})$. Corrective $|Y\rangle$ states, needed with a probability of 0.5, for the logical $R_z(\pi/8)$ gate are located above the algorithmic layer.

The total logical volume of cluster for one and two levels of state distillation can be calculated explicitly. For one level of concatenation, each $R_z(\pi/8)$ gate occupies a volume of $V = 5 \times 21 \times 2$ cells with a depth along the temporal axis of the cluster of $D = 5$ cells and a cross-sectional area of $A = 21 \times 2$. For two levels of concatenation, the volume is $V = 8 \times 77 \times 9/4$ with a depth along the temporal axis of $D = 9$ and a cross-sectional area of $A = 77 \times 2/4$. The factor of $1/4$ accounts for the fact that the cuboid represents four gates.

Cluster volume. To determine the total size of the cluster state, we need to know the amount of error correction and state

distillation required. Each logical gate requires $\Lambda \times V$ logical cells. The factor of Λ comes from the decomposition of each gate at the algorithmic level into $R_z(\pi/8)$ gates⁴⁷. This decomposition is detailed in the Methods section. Hence, the failure probability of such a gate needs to be

$$1 - (1 - p_f)^{\Lambda V} \leq \frac{1}{640L^4}, \quad (1)$$

where p_f is the failure rate of a logical cell. The right hand side sets the target failure rate for the logical gates in the circuit for Shor’s algorithm. For standard depolarizing noise, we can estimate the failure probability of a single logical volume of the cluster as $p_f \approx C_1(C_2 p/p_{th})^{\lfloor (d+1)/2 \rfloor}$, where d is the distance of the code, p is the physical error rate, p_{th} is the threshold error rate ($\approx 0.62\%$), $C_1 \approx 0.13$, and $C_2 \approx 0.61$ (ref. 12). Assuming that $p_f \ll 1$ and $1/640L^4 \ll 1$, the distance required to achieve the target error rate is

$$d \geq \left\lceil \frac{2 \log(640C_1L^4\Lambda V)}{\log(p_{th}) - \log(C_2 p)} - 1 \right\rceil. \quad (2)$$

Here we use the fact that the residual error after l levels of state distillation should be below the error rate of a logical cell, such that $7^{(3^l-1)/2} p^{3^l} \leq p_f$ and $35^{(3^l-1)/2} p^{3^l} \leq p_f$ for $|Y\rangle$ states $|A\rangle$ states, respectively. These conditions determine the level of state distillation required. Only for very large L ($L \gtrsim 2^{16}$) or for high values of P ($\gtrsim 0.001$) does state distillation require three levels.

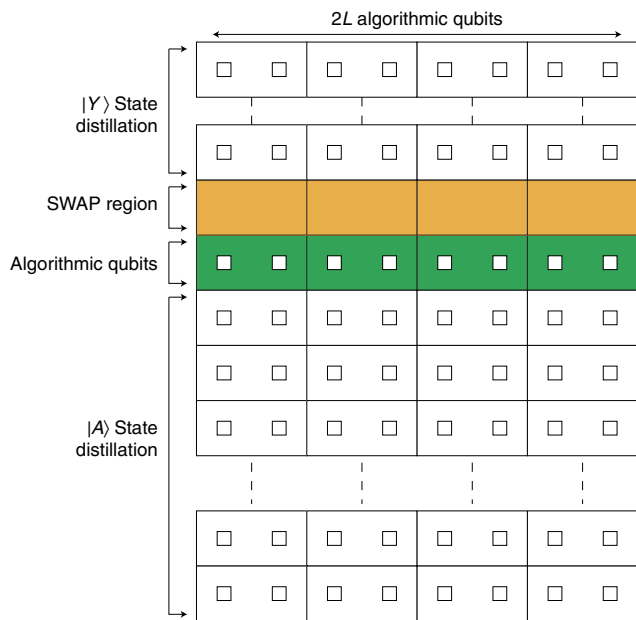


Figure 3 | Layout of logical qubits for Shor's algorithm. The layout includes the necessary cluster space for $|A\rangle$ and $|Y\rangle$ state distillation. The small squares represent the defects that exist within each larger box defining a cross-sectional area of a logically encoded qubit. The region for algorithmic qubits is green, a SWAP region for performing gates between algorithmic qubits does not contain any defects and is illustrated in orange whereas the regions devoted to state distillation are illustrated in white.

The volume and depth at this level was extrapolated from the level-two circuits to be $V = 10,000$ and $D = 15$.

Finally, we can specify the properties of the entire cluster state. The cluster contains $4L \times A$ logical cells. The total cross-sectional area of the cluster is $5Ld \times 5dA/4$ physical unit cells. The third dimension of the cluster represents the temporal axis, and its length determines the computational time. The depth of a single logical gate is $\Lambda \times D$, and the depth of a single $R_z(\pi/8)$ gate is $\Lambda D \times (5d/4)$. Therefore, the total depth of the cluster is $(32L^3\Lambda D) \times (5d/4)$.

Physical resources. In the architecture, photonic modules are used to prepare the cluster state as well as to initialize and measure single photons³⁶. There is a one-to-one mapping between the cross-sectional size of the three-dimensional cluster and the number of required modules. For a cluster with a cross-sectional area of $N_1 \times N_2$ physical unit cells, a total of $(2N_1 + 1)(2N_2 + 1)$ optical lines exist. Of these, $(2N_1 + 1)(2N_2 + 1)/2$ lines require two modules for photon detection and $(2N_1 + 1)(2N_2 + 1)/2$ lines require four modules. All optical lines require one module as a probabilistic photon source. The number of modules required to prepare the cluster state is $2(N_1 + 2)(N_2 + 1) + 2(N_2 + 2)(N_1 + 1)$ ¹⁸. This gives a total number of modules equal to $(12 + 14N_1 + 14N_2 + 20N_1N_2)$, where $N_1 = 5Ld$ and $N_2 = 5d/4A$. In addition to the number of modules, we can specify the physical size of the computer and its runtime. The dimensions of the computer are $S_x = 5LdM$ and $S_y = 5dMA/4$, where $M \times M$ is the surface area of a photonic module (with depth $< M$)¹⁸. The physical depth of the computer is $S_z \leq 2Tc_f$, where c_f is the speed of light in fibre and T is the time required to prepare a single layer of the cluster state¹⁸, corresponding to the operational speed of the photonic module. This depth is governed by the optical lines that recycle photons

from the detectors to the sources³⁶. The time required to run the algorithm is $32L^3\Lambda D \times 5d/4 \times 2T$.

Figure 4 shows the runtime of the algorithm, the total number of photonic modules and the dimensions of the computer as functions of the physical error rate and the problem size. Here we have assumed that $p_{th} = 0.62\%$ (refs 12,48), $M = 100 \mu\text{m}$ and $T = 10 \text{ ns}$ (ref. 49). Contour lines in Fig. 4 indicate when the completion time is 1 year, when the total number of photonic modules is one billion and when the cross-sectional dimensions are 1 m. With a physical error rate an order of magnitude below the threshold and without any further algorithmic improvements, the largest problem size that can be completed within a year is $L \approx 810$.

Discussion

A recent milestone was the factorization of a 768-bit RSA modulus over several years using the number field sieve classical factoring method⁵⁰. Hence, these results show the clear advantage of quantum computation. They seem not to demonstrate a significant increase in the processing power of quantum computers. Our results give a comfortable upper bound for the resource requirements using explicit circuit constructions in the topological model. The time required to factor a 1024-bit number in this analysis is 2.3 years with 1.9 billion photonic modules required to prepare the cluster. An interesting question is how these numbers can be compared with the fundamental circuit used in this analysis. As we mentioned in the introduction, there are different techniques to construct a circuit for Shor's algorithm. The overhead associated with error correction should be calculated based on the resource costs associated with the same non-error-corrected circuit. For instance, the circuit using Toffoli-based quantum adders⁴⁵ and/or using an approximate quantum fourier transform⁵¹ could reduce the overall physical resources; however, the error correction overhead is highly non-trivial. In this analysis, the basic circuit requires a computational depth of $32L^3$ and $2L$ qubits. For physical gate times of 10 ns, for $L = 1,024$, the error correction overhead amounts to a factor of 2.3×10^5 in time and 9.1×10^5 in terms of qubits (modules). These numbers are based on a physical error rate an order of magnitude below threshold. This overhead can be significantly reduced by optimisations unrelated to the fundamental hardware. This is illustrated by the fact that decreasing the error rate by an order of magnitude results in a computational speed-up to 1.2 years. The same speed-up can be achieved by compacting the topological circuits shown here by $\sim 45\%$ along the temporal axis of the cluster.

There have been several other resource estimates made for various quantum computer architectures using concatenated and topological error correction. Thaker *et al.*²² estimated that to factor a 1,024-bit number on an architecture based on trapped ions would take around 25 days. Van Meter *et al.*²⁰ estimated that a 2048-bit number on a distributed architecture based on quantum dots would take around 400 days. Jones *et al.*²¹ recently improved the latter estimate to around 10 days by using a monolithic array of dots and increasing the speed of error correction. New results in superconducting designs suggest a factoring time for a 2,000-bit number of slightly less than 1 day⁵². The differences in these estimates arise due to how the algorithm is implemented. In particular, more resource-efficient techniques are utilized in these results, which need to be explicitly integrated within the topological model.

All resource estimates, including ours, illustrate that a large fraction of the overhead arises from the need to prepare ancillary states. Other results assume sufficient space within the computer such that ancillary protocols can be completed rapidly enough so that the depth of the algorithmic circuit is unchanged. This could

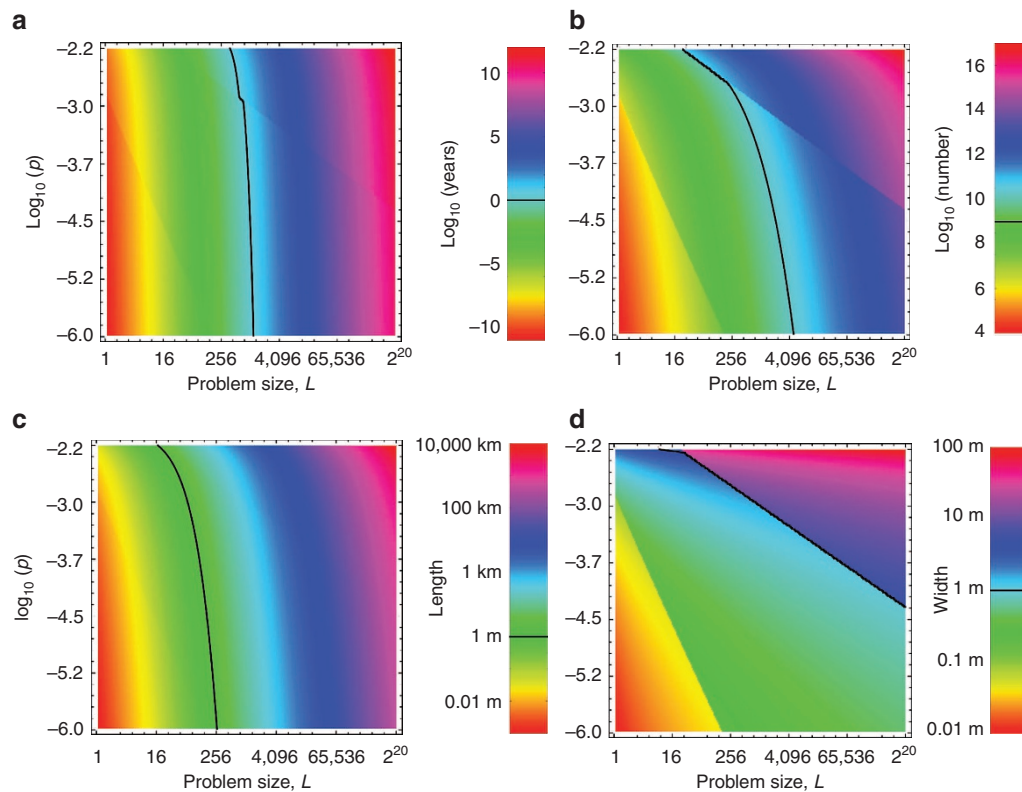


Figure 4 | Physical resources required for factoring. Each panel illustrates the resources required to factor an L -bit number with an error rate p . Panel **a** shows the computational time to complete factoring assuming a $T=10$ ns operational time for the photonic module. The solid line is where factoring takes 1 year. Panel **b** shows the number of photonic modules with the solid line showing where factoring requires one billion modules. Panels **c** and **d** illustrate the computer size, assuming each photonic module has a cross-sectional size of $100 \times 100 \mu\text{m}$. The solid line is when the length or width of the computer reaches 1 m. The discontinuities represent points where the concatenation level of state distillation increases.

be of significant benefit. However, the appropriate routing of these ancillary protocols needs to be explicit. How distillation circuits are interfaced with data qubits needs to be detailed, and which protocols are utilized needs to be analysed. Estimates from Jones *et al.*,²¹ and Fowler *et al.*⁵² use the most optimal circuit for Shor's algorithm^{39,45}; however, this circuit has not yet been adapted to the geometric constraints of the topological cluster. Until an appropriate construction is presented for the topological cluster, it is difficult to assume that the circuit size will remain unchanged. If such a circuit design is presented, then we anticipate immediate reductions in resources. Previous results also assume that various subcomponents of a fault-tolerant implementation can be applied without space/time penalty. There have been many results published optimizing various components in a fully error-corrected quantum algorithm^{53–56}. However, each of these results has been derived in isolation, some have not been converted into the topological model and none have been carefully integrated together. This is the primary challenge of topological compilation. Subcomponents may be efficient, but the success of a large-scale computation requires delicate integration. Our results illustrate that there is a significant gap between optimistic resource estimates and those performed using explicit circuit constructions.

It is clear that algorithmic compilation is a necessity before a quantum computer is actually built. Reducing the burden on experimental development is ultimately a function of how we realize abstract algorithms. Our results illustrate that there is still much work to be done. Although the topological model is promising, its ultimate success is dependent on continual efforts

to integrate all necessary protocols in a way that minimizes the number of devices and the time required to execute an algorithm.

Methods

Valid operations in the Topological cluster. There are five primitive, fault-tolerant operations allowed in the topological model: measurement, initialization, state injection, the two-qubit CNOT and the teleported phase rotations $R_z(q)$, $q = \{\frac{\pi}{4}, \frac{\pi}{8}\}$. These are illustrated in Fig. 5. There are two types of defects that can be created, primal and dual. Braided logic operations can only occur between defects of opposite type and the geometric structures for X- and Z-basis initialization and measurement are interchanged depending on the type of defect.

Gate decomposition. As in the case of all error-corrected models of quantum computation, the topological model does not allow all gate operations to be directly applied in a fault-tolerant manner. At the logical level, only preparation of the states $|+\rangle$ and $|0\rangle$, X and Z gates, measurement in the X and Z bases, and the CNOT gate can be directly applied. Swap gates are achieved by deforming the trajectory of the defects. To complete a universal set, we add the $R_z(\pi/8)$ and $R_z(\pi/4)$ rotations¹². The $R_z(\pi/4)$ rotation is constructed using the $[[7,1,3]]$ Steane code to distil an appropriate ancillary state. This construction of the $R_z(\pi/4)$ rotation is more resource efficient than two $R_z(\pi/8)$ rotations, which require distillation protocols based on the $[[15,1,3]]$ Reed-Muller code. To apply these gates, we perform a teleportation operation with the ancillary states $|A\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$ and $|Y\rangle = (|0\rangle + i|1\rangle)/\sqrt{2}$. Each time we attempt the $R_z(\pi/8)$ gate, there is a 50% chance that a $R_z(\pi/4)$ correction is required.

To ensure that the error rate of the R_z rotations is sufficiently low, the states $|A\rangle$ and $|Y\rangle$ must be of sufficient fidelity. As these ancillary states are prepared in the cluster via injection protocols¹², state distillation is used to increase the fidelity of the ancillary states²⁵, consuming multiple $|A\rangle$ or $|Y\rangle$ states with a lower fidelity. This process can be concatenated until the desired fidelity is reached. If p_l is the error probability of the state after l levels of state distillation, then $p_{l+1}^A = 35(p_l^A)^3$ and $p_{l+1}^Y = 7(p_l^Y)^3$ for $|A\rangle$ and $|Y\rangle$ states, respectively²⁵. Each distillation circuit is probabilistic with a failure probability of $O(p)$. Supplementary

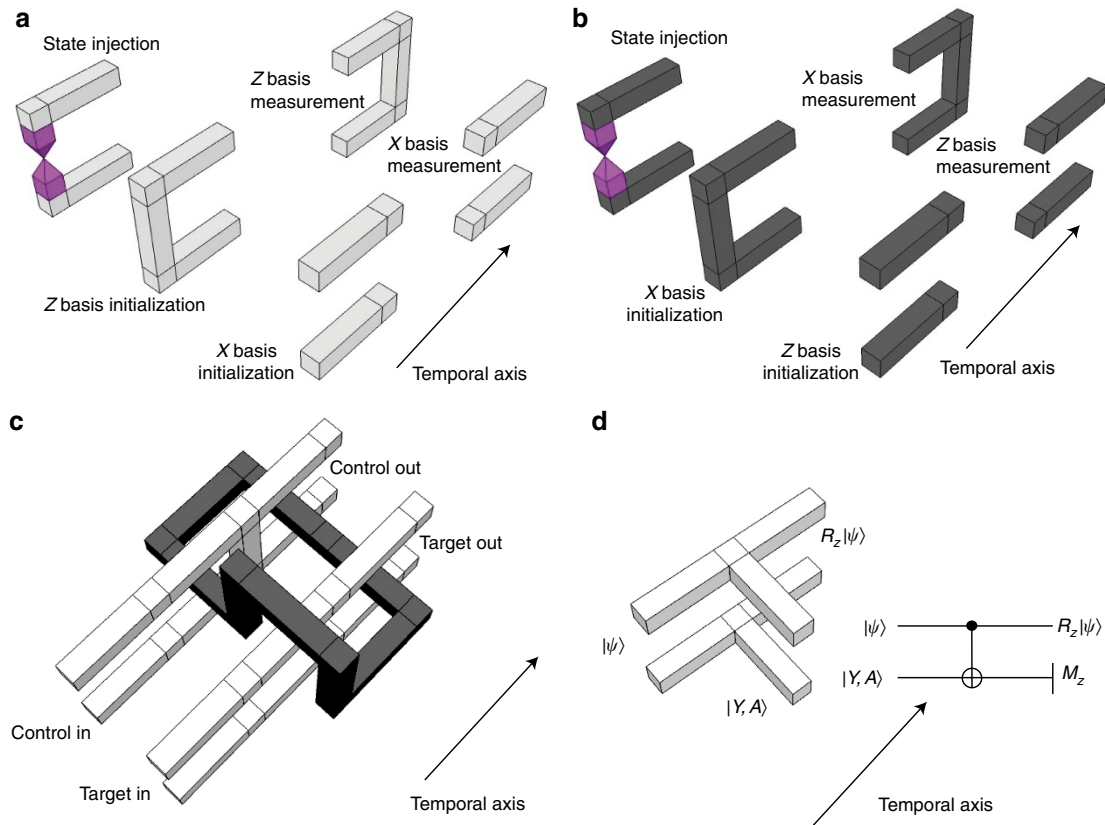


Figure 5 | Examples of basic operations used to construct braid sequences. (a) Primal defects (white) in a horseshoe shape are used to prepare a logical qubit in the state $|0\rangle$ and to measure a logical qubit in the Z-basis. An arbitrary state can be prepared by measuring one of the physical qubits (shown in pink) in a rotated basis as the defects are created. (b) Dual defects (dark) in a horseshoe shape are used to prepare a logical qubit in the state $|+\rangle$ and to measure a logical qubit in the X-basis. (c) A CNOT gate can be achieved by braiding a pair of dual defects (prepared in the state $|+\rangle$) with three pairs of primal defects (the control qubit, the target qubit and an extra qubit prepared in the state $|0\rangle$)¹⁴. (d) Teleported $R_z(\frac{\pi}{8})$ rotations can be achieved by attaching the relevant ancillary state to the data qubit¹².

Fig. S2 illustrates the quantum circuits for these distillation protocols, and Supplementary Fig. S1 illustrates the compacted topological structures.

Given our set of logical gates, which now includes the $R_z(\pi/8)$ rotation, we decompose the circuit for Shor’s algorithm into an appropriate sequence of these operations. To estimate an upper bound on the number of gates required, we will (pessimistically) assume every gate is a non-trivial phase rotation. These rotations must be approximated by a sequence of logical gates found using the Solovay–Kitaev algorithm⁵⁷. ~50% of the gates in this sequence⁵⁸ are $R_z(\pi/8)$ rotations, which are the most resource intensive of our logical gates. For simplicity, we will (pessimistically) assume all of these gates are $R_z(\pi/8)$ rotations.

Recent algorithms, which can efficiently calculate these been developed^{47,59,60}. These sequences are restricted to arbitrary Z-axis rotations, $R_z(0)$, and arbitrary axis rotations are achieved with standard Euler angle decompositions and additional Hadamard gates. Fortunately, the nature of the Beauregard circuit is such that only arbitrary Z-axis rotations are required (The Quantum Fourier transform can be decomposed into arbitrary Z-axis rotations and CNOTs). We utilize the algorithm of Kliuchnikov *et al.*⁵⁹, as it has demonstrated the best quantum scaling to date. Each gate in Shor’s algorithm is approximated by a sequence of $\Lambda = 2 \times (3.21 \log_2(640L^4) - 6.93)$ rotations. The factor of two comes from the fact that the scaling reported in Kliuchnikov *et al.*⁵⁹ is only the number of $R_z(\pi/8)$ rotations that are needed for approximating an arbitrary rotation. This constitutes approximately half of the sequence, with the additional half consisting mainly of Hadamard operations. Although we assume that the entire sequence consists of $R_z(\pi/8)$ gates, explicitly constructing fault-tolerant Hadamard operations has only a marginal influence on resources. For further details, see Supplementary Figs S9–S11 and the Supplementary Discussion.

References

1. Hanson, R. & Awschalom, D. Coherent manipulation of single spins in semiconductors. *Nature* **453**, 1043–1049 (2008).
2. Press, D., Ladd, T. D., Zhang, B. & Yamamoto, Y. Complete quantum control of a single quantum dot spin using ultrafast optical pulses. *Nature* **456**, 218–221 (2008).

3. Politi, A., Matthews, J. & O’Brien, J. Shor’s quantum factoring algorithm on a photonic chip. *Science* **325**, 1221 (2009).
4. Blatt, R. & Wineland, D. Entangled states of trapped atomic ions. *Nature* **453**, 1008–1015 (2008).
5. Pla, J. *et al.* A single-atom electron spin qubit in Silicon. *Nature* **489**, 541–545 (2012).
6. Lucero, E. *et al.* Computing prime factors with a Josephson phase qubit quantum processor. *Nat. Phys.* **8**, 719–723 (2012).
7. MagiQ www.magiqtech.com (2012).
8. idQuantique www.idquantique.com (2012).
9. Ladd, T. D. *et al.* Quantum Computers. *Nature* **464**, 45–53 (2010).
10. Knill, E. Quantum computing with realistically noisy devices. *Nature* **434**, 39–44 (2005).
11. Bacon, D. Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A* **73**, 012340 (2006).
12. Raussendorf, R., Harrington, J. & Goyal, K. Topological fault-tolerance in cluster state quantum computation. *New J. Phys.* **9**, 199 (2007).
13. Taylor, J. *et al.* Fault-tolerant architecture for quantum computation using electrically controlled semiconductor spins. *Nat. Phys.* **1**, 177–183 (2005).
14. Metodi, T., Thaker, D., Cross, A., Chong, F. & Chuang, I. A general purpose architecture layout for arbitrary quantum computations. *Proc. SPIE* **5815**, 91–102, (2005).
15. Hollenberg, L., Greentree, A., Fowler, A. & Wellard, C. Two-Dimensional Architectures for Donor-Based Quantum Computing. *Phys. Rev. B* **74**, 045311 (2006).
16. Fowler, A. *et al.* Long-range coupling and scalable architecture for superconducting flux qubits. *Phys. Rev. B* **76**, 174507 (2007).
17. Stock, R. & James, D. A scalable, high-speed measurement based quantum computer using trapped ions. *Phys. Rev. Lett.* **102**, 170501 (2009).
18. Devitt, S. *et al.* Architectural design for a topological cluster state quantum computer. *New J. Phys.* **11**, 083032 (2009).
19. Yao, N. *et al.* Scalable architecture for a room temperature solid-state quantum information processor. *Nat. Comm.* **3**, 800 (2012).

20. Meter, R. V., Ladd, T., Fowler, A. & Yamamoto, Y. Distributed quantum computation architecture using semiconductor nonophotonics. *Int. J. Quant. Inf.* **8**, 295–323 (2010).
21. Jones, N. C. *et al.* A layered architecture for quantum computing using quantum dots. *Phys. Rev. X* **2**, 031007 (2012).
22. Thaker, D., Metodi, T., Cross, A., Chuang, I. & Chong, F. Quantum memory hierarchies: efficient designs to match available parallelism in quantum computing. *ACM SIGARCH Computer Architecture New* **34**, 376–390 (2006).
23. Steane, A. How to build a 300 bit, 1 Giga-operation quantum computer. *Quant. Inf. Comp.* **7**, 171–183 (2007).
24. Clark, C. R., Metodi, T. S., Gasster, S. D. & Brown, K. R. Resource requirements for fault-tolerant quantum simulation: the ground state of the transverse Ising model. *Phys. Rev. A* **79**, 062314 (2009).
25. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A* **71**, 022316 (2005).
26. Selinger, P. Towards a quantum programming language. *Math. Struct. Comp. Sci.* **14**, 527–586 (2004).
27. Altenkirch, T. & Grattage, J. A functional quantum programming language. Logic in Computer Science, LICS 2005, Proceedings 20th Annual IEEE Symposium 26–29 June 2005, 249–258 (2005).
28. Giles, B. Programming with a quantum stack. *Masters Thesis, Univers. Calgary* (2007).
29. Green, A. S., Lumsdaine, P. L., Ross, N. J., Selinger, P. & Valiron, B. An Introduction to Quantum Programming in Quipper. Preprint at <http://arxiv.org/abs/1304.5485> (2013).
30. Shende, V., Bullock, S. & Markov, I. Synthesis of quantum logic circuits. *IEEE Trans. on Computer-Aided Design* **25**, 1000–1010 (2006).
31. Patel, K., Markov, I. & Hayes, J. Optimal synthesis of linear reversible circuits. *Quant. Inf. Comp.* **8**, 282–294 (2008).
32. Amy, M., Maslov, D., Mosca, M. & Roetteler, M. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. Preprint at <http://arxiv.org/abs/1206.0758> (2012).
33. iARPA. Quantum Computer Science (QCS) Program www.iarpa.gov/Programs/sso/QCS/qcs.html (2013).
34. Whitney, M., Isailovic, M., Patel, Y. & Kubiawicz, J. Automated generation of layout and control for quantum circuits. Proc. 4th International Conference on Computing Frontiers. doi:10.1145/1242531.1242546 (ACM, New York, NY, USA, 2007).
35. Maslov, D., Falconer, S. & Mosca, M. Quantum circuit placement. *IEEE Trans. Comp.-Aided Design Integr. Circuits Syst* **27**, 752–763 (2008).
36. Devitt, S., Stephens, A., Munro, W. & Nemoto, K. Integration of highly probabilistic sources into optical quantum architectures: perpetual quantum computation. *New J. Phys.* **13**, 095001 (2011).
37. Devitt, S. *et al.* The photonic module: an on-demand resource for photonic entanglement. *Phys. Rev. A* **76**, 052312 (2007).
38. Jones, N. C. *et al.* Simulating chemistry efficiently on fault-tolerant quantum computers. *New J. Phys.* **4**, 115023 (2012).
39. Vedral, V., Barenco, A. & Ekert, A. Quantum networks for elementary arithmetic operations. *Phys. Rev. A* **54**, 147–153 (1996).
40. Gossett, P. Quantum carry-save arithmetic. Preprint at <http://arxiv.org/abs/quant-ph/9808061> (1998).
41. Beauregard, S. Circuit for Shor's algorithm using $2n + 3$ qubits. *Quant. Inf. Comp.* **3**, 175–185 (2003).
42. Raussendorf, R., Harrington, J. & Goyal, K. A fault-tolerant one-way quantum computer. *Ann. Phys.* **321**, 2242–2270 (2006).
43. Fowler, A., Devitt, S. & Hollenberg, L. Implementation of Shor's algorithm on a linear nearest neighbour qubit array. *Quant. Inf. Comp.* **4**, 237–251 (2004).
44. Kutin, S. Shor's algorithm on a nearest-neighbor machine. Preprint at <http://arxiv.org/abs/quant-ph/0609001> (2006).
45. Cuccaro, S., Draper, T., Kutin, S. & Moulton, D. A new quantum ripple-carry addition circuit. Preprint at <http://arxiv.org/abs/quant-ph/0410184> (2004).
46. Devitt, S., Fowler, A. & Hollenberg, L. Robustness of Shor's Algorithm. *Quant. Inf. Comp.* **6**, 616–629 (2006).
47. Kliuchnikov, V., Maslov, D. & Mosca, M. Asymptotically optimal approximations of single qubit unitaries by Clifford and T circuits using a constant number of ancillary qubits. *Phys. Rev. Lett.* **110**, 190502 (2013).
48. Barrett, S. & Stace, T. Fault-Tolerant quantum computation with very high threshold for loss errors. *Phys. Rev. Lett.* **105**, 200502 (2010).
49. Su, C.-H., Greentree, A., Munro, W., Nemoto, K. & Hollenberg, L. High Speed quantum gates with cavity quantum electrodynamics. *Phys. Rev. A* **78**, 062336 (2008).
50. Kleinjung, T. *et al.* Factorization of a 768-bit RSA Modulus. *Proceedings of the 30th Annual Conference on Advances in Cryptology. CRYPTO'10.* (ed. Rabin, T.) 333–350 (Springer-Verlag, Berlin, Heidelberg, 2010).
51. Coppersmith, D. *An Approximate Fourier Transform Useful in Quantum Factoring.* IBM Research Report No. RC19642, *quant-ph/0201067* (1996).
52. Fowler, A., Mariantoni, M., Martinis, J. & Cleland, A. Surface codes, towards practical large-scale quantum computation. *Phys. Rev. A* **86**, 032324 (2012).
53. Fowler, A. & Devitt, S. A bridge to lower overhead quantum computation. *arxiv:1209.0510* (2012).
54. Fowler, A. Time-optimal quantum computation. Preprint at <http://arxiv.org/abs/1210.4626> (2012).
55. Jones, N. C. Multilevel distillation of magic states for quantum computing. *Phys. Rev. A* **87**, 042305 (2012).
56. Bravyi, S. & Haah, J. Magic state distillation with low overhead. *Phys. Rev. A* **86**, 052329 (2012).
57. Dawson, C. & Nielsen, M. The Solovay-Kitaev Algorithm. *Quant. Inf. Comp.* **6**, 81–95 (2006).
58. Fowler, A. Constructing arbitrary Steane code single logical qubit fault-tolerant gates. *Quant. Inf. Comp.* **11**, 867–873 (2011).
59. Kliuchnikov, V., Maslov, D. & Mosca, M. Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T Circuits. Preprint at <http://arxiv.org/abs/1212.6964> (2012).
60. Selinger, P. Efficient Clifford + T approximations of single-qubit operators. Preprint at <http://arxiv.org/abs/1212.6253> (2012).

Acknowledgements

We thank N.C. Jones and A.G. Fowler for valuable discussions. This work was partly supported by the Quantum Cybernetics (MEXT) project and FIRST programme in Japan.

Author contributions

S.J.D. conceived the idea. All authors were responsible for primary calculations and drafting of the manuscript.

Additional information

Supplementary Information accompanies this paper at <http://www.nature.com/naturecommunications>

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

Reprints and permission information is available online at <http://npg.nature.com/reprintsandpermissions/>

How to cite this article: Devitt, S. J. *et al.* Requirements for fault-tolerant factoring on an atom-optics quantum computer. *Nat. Commun.* **4**:2524 doi: 10.1038/ncomms3524 (2013).