

High-Threshold Low-Overhead Fault-Tolerant Classical Computation and the Replacement of Measurements with Unitary Quantum Gates

Benjamin Cruikshank^{1,2} and Kurt Jacobs^{1,2,3}

¹*U.S. Army Research Laboratory, Computational and Information Sciences Directorate, Adelphi, Maryland 20783, USA*

²*Department of Physics, University of Massachusetts at Boston, Boston, Massachusetts 02125, USA*

³*Hearne Institute for Theoretical Physics, Louisiana State University, Baton Rouge, Louisiana 70803, USA*

(Received 23 February 2016; published 21 July 2017)

von Neumann’s classic “multiplexing” method is unique in achieving high-threshold fault-tolerant classical computation (FTCC), but has several significant barriers to implementation: (i) the extremely complex circuits required by randomized connections, (ii) the difficulty of calculating its performance in practical regimes of both code size and logical error rate, and (iii) the (perceived) need for large code sizes. Here we present numerical results indicating that the third assertion is false, and introduce a novel scheme that eliminates the two remaining problems while retaining a threshold very close to von Neumann’s ideal of $1/6$. We present a simple, highly ordered wiring structure that vastly reduces the circuit complexity, demonstrates that randomization is unnecessary, and provides a feasible method to calculate the performance. This in turn allows us to show that the scheme requires only moderate code sizes, vastly outperforms concatenation schemes, and under a standard error model a unitary implementation realizes universal FTCC with an accuracy threshold of $p < 5.5\%$, in which p is the error probability for 3-qubit gates. FTCC is a key component in realizing measurement-free protocols for quantum information processing. In view of this, we use our scheme to show that all-unitary quantum circuits can reproduce any measurement-based feedback process in which the asymptotic error probabilities for the measurement and feedback are $(32/63)p \approx 0.51p$ and $1.51p$, respectively.

DOI: 10.1103/PhysRevLett.119.030503

The problem of performing classical computing reliably with unreliable logic gates is referred to as fault-tolerant classical computation (FTCC). The first method for realizing FTCC was devised by von Neumann, who called it *multiplexing* [1]. It achieves what may be the highest possible error threshold (the maximum stable componentwise error rate), but has hitherto been viewed as impracticable. This is due to an apparent need for high redundancy (number of fundamental components required to construct a noise-free logical gate), the need to continually connect and reconnect bits “at random” at a potentially large spatial separation, and the difficulty of both analytically calculating the performance for moderate code sizes and simulating the performance in the low-error regimes required for reliable computation [1–9]. The field of probabilistic cellular automata was partially motivated by addressing the second problem, but has not, to date, produced a complete and feasible FTCC scheme [10–21]. A second method for FTCC was developed more recently in the context of quantum computing, and involves “concatenating” error-correction codes and logic gates [22–46]. However, the concrete FTCC schemes developed using concatenation require high redundancy and connections between widely separated code bits, and have not to-date achieved the high thresholds of multiplexing schemes [47,48].

We are interested in FTCC here primarily because of its central role in the question of the importance of

measurements in realizing physical processes and control protocols. From a fundamental point of view, measurements play no special role in physical processes: all dynamics generated by measurement and feedback processes (including those involving postselection [49]) can be reproduced by unitary evolution [52]. Consequently the utility of measurements in any physical protocol arises only from technological constraints. For fault-tolerant quantum computation (FTQC), in which all high-threshold schemes to date employ measurements [29,38,44,53], the constraint is a fixed error probability p for all quantum gates and measurements.

Interest in measurement-free (or measurement-light) quantum computing [54–58] is motivated by the fact that implementing large numbers of measurements on many qubits requires an additional technological overhead beyond that of unitary circuits. To this end schemes have been devised in which measurements can be noisy and/or slow [54,55], and quite recently automata-based methods were introduced for eliminating both measurements and the high processing overhead involved in correcting surface codes [57,58]. Here our purpose is to determine the ability of unitary circuits to perform the functions of arbitrary measurement procedures.

For the purposes of FTQC (or any quantum process subjected to errors) the physical addition provided by measurements is *amplification*: measurements are defined as producing a result that can be processed on a classical

digital computer. The resulting error-free classical processing is the sole advantage of measurements. As a result, the question of the importance of measurement to quantum processes is intimately related to how well mesoscopic gates (those with error p) can perform such error-free classical processing, and thus to the fundamental limits of FTCC.

Our first main result is an explicit scheme for FTCC with unitary gates that largely solves the long-standing problems with von Neumann’s multiplexing method, while achieving almost the same high threshold. We then apply our FTCC protocol to the problem of using unitary circuits to reproduce a general measurement and feedback process. Our second main result is that, with a threshold of $p = 2.8\%$, unitary circuits can do this and achieve effective measurement and feedback errors close to p . It should be noted that this result does not imply alone that a given measurement-based protocol can be replaced by a unitary one; additional technological factors, such as the time taken by the processing circuits, may also play an important role in an implementation. This result does show that unitary circuits can effectively realize high-fidelity measurements, and that perfect classical processing within quantum mesoscopic circuits is entirely feasible. They also furnish a new tool for understanding the power of unitary protocols.

Before we present our error-correction scheme, it is worth discussing the key issues with von Neumann’s method in more detail. von Neumann’s scheme uses a repetition code, and corrects errors in the code by applying a “majority counting” gate to triples of code bits. The output of this gate is a single bit whose value is that of the majority of the three input bits. The output bit is then copied to produce three bits that are the corrected versions of the input bits. As noted above, one of the primary problems with the scheme is the need to reduce correlated errors by randomly selecting triples across multiple repetitions of the error correction. This virtually prohibits the use of fixed wiring for the gate interconnections due to the complexity of the resulting circuits. Furthermore, an appeal to truly “random” fixed connections in a computational circuit implies the need for unique random reconnections at every correction step of a computation of any length, something that is clearly infeasible. A possible solution is to dynamically reconnect the gates at each correction step. While such a process cannot be used in a fundamental theory of fault tolerance (the logic circuits that generate and store the new connections will introduce further errors), it could be used to implement FTCC with mesoscopic circuits by using error-free macroscopic classical computers to perform the dynamical reconnection. Nevertheless, doing so requires the significant additional overhead of complex classical control circuits that allow any triple of code bits to interact at any time. Our solution completes the theory of multiplexing by eliminating randomness altogether and allowing error correction to be performed by a small set of fixed connections within the code, which are recycled over

a short sequence. We note that our scheme has a single restriction over von Neumann’s, which is that the code size must be a power of 3.

A further advantage we provide over randomized multiplexing is that with multiplexing the logical error rate for moderate, realistic code sizes can be obtained only by numerical simulation. This is problematic because (i) a simulation must generate at least tens of logical failures to obtain any accuracy, and (ii) for realistic computing applications the error probability per logical bit must be very small (e.g., 10^{-12}). As a result the problem would be challenging even for modern large-scale parallel machines.

We now describe our FTCC scheme, beginning with the logic gates from which it is built.

Elementary 3-bit gates.—We define a MAJ1 gate (the name deriving from “majority”) as von Neumann’s 3-bit majority counting gate, described above. We define a gate AMP (short for “amplification”) that takes in one bit and outputs three copies of it. Finally, we define a gate MAJ3 as a MAJ1 that has three outputs, being the usual MAJ1 output bit and two additional copies of it. Thus the MAJ3 is a MAJ1 followed by an AMP. We must be able to implement all these gates unitarily, so we present explicit unitary versions of them in Fig. 1. These unitary versions are shown in terms of CNOT and Toffoli gates [59], but our error model treats the MAJ1 and AMP as elementary 3-bit unitary processes.

Error-correcting scheme.—The logical values of bits on which we wish to do computation are stored in a simple repetition code of size 3^{n+1} where n is a non-negative integer called the *level* of the code. These bits can be arranged in a hypercube of dimension $n + 1$ with side length 3. Error correction is now achieved by applying n parallel MAJ3 gates along each of the $n + 1$ axes in sequence. This implies that the logical value can be equivalently thought of as stored in a network of 3^n MAJ3 gates that interact sequentially along each axis of an n -dimensional hypercube [49].

Performance of the error-correcting circuit.—Our error model is defined by assigning, in the standard way [29,60], a total intrinsic error probability, p , to each of the 3-bit unitary quantum gates MAJ1 and AMP. To determine the performance of the error-correction circuit we need the probability, ϵ , that there is an error in at least one of the output lines of the MAJ3. Given the quantum error model,

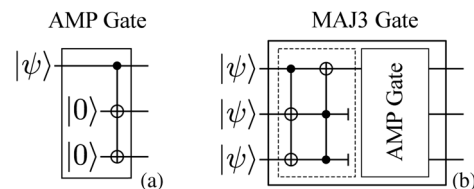


FIG. 1. Unitary constructions for the (a) AMP and (b) MAJ3 gates defined in the text. The unitary version of AMP copies the input qubit in the computational basis by applying a controlled-NOT (CNOT) gate from this input to each of two qubits prepared in the 0 state. The dashed box is the MAJ1 gate.

along with an additional error of $(2/3)p$ in each output to account for errors in the connecting wires and reset locations, we obtain a strict overestimate for ϵ to be $(52/21)p \approx 2.3p$ [61]. We need to obtain the steady-state logical error probability, p_{ss} , that is maintained by repeated applications of the correction circuit [62]. Calculating p_{ss} is nontrivial; however, straightforward simulations are impractical as discussed above. We are able to calculate p_{ss} for $n=2$ and $n=3$ (codes with 27 and 81 bits, respectively) by mapping the error dynamics to a jump process that has only 3 effective states for $n=2$ and 7 for $n=3$ (details are given in the Supplemental Material [49]). In Fig. 2 we plot p_{ss} as a function of ϵ for $n=2$ and 3. We see from these plots that the error decreases doubly exponentially with n for $n=2$ and 3, and so we expect this to continue for larger values of n . This shows a (gatewise) redundancy which scales as 3^n . In comparison, the redundancy for typical concatenation schemes is 21^n [47,48]. We note that for $p=0.4\%$ and $n=3$ one has $p_{ss}=1.5 \times 10^{-11}$; thus, small values of n would likely suffice for applications. We obtain a lower bound on the threshold for $n=3$, shown in Fig. 3(a), to be $\epsilon \approx 15\%$, very close to von Neumann's value of $1/6$.

Performance of von Neumann's multiplexing.—In the inset in Fig. 2 we compare the performance of von Neumann's scheme to ours for a code size of 81 bits. As noted above, this calculation is limited to relatively large error rates. We see that the performance of von Neumann's scheme oscillates about ours, and thus achieves similar performance at high logical error rates. This provides support for the conjecture that a randomized scheme should perform at least as well as ours. Nevertheless, one cannot merely extrapolate to small error rates, as made especially clear by the oscillatory behavior.

Threshold for universal computation.—Reliable MAJ1 and AMP gates can be used for universal computation.

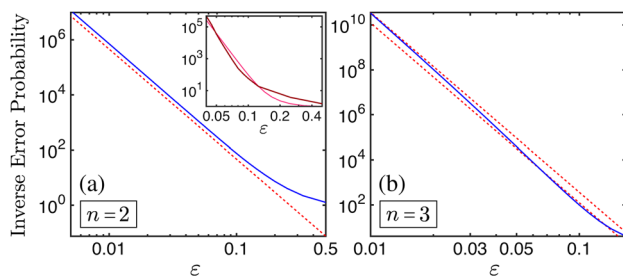


FIG. 2. (a) Solid line: The inverse of the logical error probability, p_{ss} , for our error correction circuit with a 27-bit code. Dashed line: $1/p_{ss}$ for a hypothetical concatenation scheme with a threshold of $1/6$ [49,59]. (b) Solid line: $1/p_{ss}$ for our scheme with an 81-bit code. Dashed lines: $1/p_{ss}$ for hypothetical concatenation schemes with thresholds of $1/6$ (upper line), and $1/7$ (lower line). (No such concatenation schemes are known to date.) The inset shows the performance of von Neumann's scheme (dark line) against the new scheme (light line) for 81 bits, for a range of ϵ in which it is feasible to simulate.

A MAJ1 gate can be used as an AND or OR gate. An AMP gate is composed of CNOTs, which can be used as NOT gates and/or simulated line splitting. (Details can be found in Refs. [1,49].) The most complex construction in our scheme is a coded MAJ1 gate, which consists of a transversal application of MAJ1 gates on three coded bits. The bitwise error rate in an error correcting network at threshold is $1/2$. Since the output of a three-input computational gate is necessarily noisier than any one of the inputs, we must have input errors less than $1/2$, so the componentwise threshold for universal computation must be smaller than $1/6$. We recognize the threshold as the basic error rate at which error rates in the outputs of the MAJ1 gates are equal to $1/2$. Taking into account the steady-state bitwise error rate of our coded input bits, we find the threshold for universal computation to be $p=5.5\%$, or $\epsilon \approx 12.7\%$ [49].

Scaling of wire length with code size.—A crucial issue for fault-tolerant computation is how the wire length required by the correction and computation circuits increases with code size. Since it is reasonable to suggest that fundamental error rates will increase exponentially with the wire length (the distance between interacting gates or bits), an error correction scheme must be compatible with a compact wiring. Part of our solution is the observation, noted above, that using our method code sizes of no more than $n=4$ (and likely $n=3$) can be expected to be sufficient for any application. If we reroute the wiring from the gate outputs to the inputs, the full error correction circuit for $n=3$ can be executed with a cube of 27 MAJ3 gates. The rerouting need only span the cube separately in each direction, giving a wire length of 2 (in units of the distance between adjacent gates). The transversal AND gate for logic between coded bits, when we place three code cubes in a row, requires wires of length 3. For $n=4$ there are also very efficient arrangements. For a single coded bit

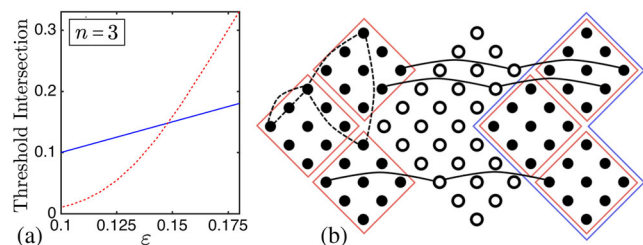


FIG. 3. (a) The solid line is ϵ and the dashed line is the resulting error probability for a code with $n=3$. The point at which these lines cross gives the threshold. (b) Here we show the wiring lengths required for a code with $n=4$ (243 bits). Each dot and circle represents a MAJ3 gate, so that the squares indicate a cube of 27 MAJ3 gates (an $n=3$, 81-bit code) viewed from the top. The blue line encircles three of these cubes that form a single $n=4$ code block. The curved solid lines are examples of wires required to implement an AND gate on the two code blocks consisting of black dots. The dashed lines are examples of wires required to implement an error correction on an individual code block.

we now require three cubes of 27 MAJ3 gates for error correction. In Fig. 3(b) we show that, arranging these three cubes in an “L” configuration, the longest wires required for universal computation have length $3\sqrt{2}$. Only a moderate increase in overall distance is therefore required to implement FTCC. One could alternatively use an array of static qubits rather than static gates. Under this architecture, the error-correction circuits are similarly compact; each qubit needs to interact with only $2(n+1)$ others [63].

Replacing measurements with unitary circuits.—The scheme for efficient FTCC presented above allows us to obtain unitary circuits that perform the role of measurement and feedback processes, and do so with very low error rates. First, we note that the role of every measurement in any physical protocol involves no more than (i) classical processing of the measurement result, and (ii) at some point in the process, the use of the processed result to apply an operation to a quantum system. We can assume the feedback operation is unitary without loss of generality. We will also assume, without significant loss of generality, that the quantum systems involved are qubits.

To reproduce the action of a “black box” that implements measurement and feedback our unitary circuit must (i) encode the qubits to be measured (the input qubits) so that the classical information they contain can be processed by our FTCC protocol, (ii) perform the processing, (iii) use the processed information (the output qubits), which is stored as a repetition code, to apply a unitary gate to one or more “target” qubits. Steps (i) and (iii) correspond to the processes of measurement and feedback, respectively. In step (i), any error in the encoded logical bits introduced by the encoding procedure is precisely equivalent to the measurement error. For step (iii), since we must use the state of a single qubit as a control for the feedback operation, any error by which this qubit deviates from the encoded logical output bit is simply an additional probability of error over that of the feedback applied by a classical device.

To encode the information stored in the computational basis of a qubit we use a circuit consisting of AMP gates. An AMP gate is used to make three copies of the initial qubit in the computational basis, and then each of these is tripled again by feeding it into an AMP gate. By repeating this process we produce 3^{n+1} qubits that constitute the bits of our repetition code. The key quantity of interest is the probability, p_{enc} , that the resulting code fails to correctly reflect the state of the bit contained by the initial qubit (more precisely, the probability that the code bits are left in a joint state that will fail to be properly corrected by the error-correction circuits). Calculating p_{enc} is a complex task, since we must take into account the correlations formed between the code bits or qubits during the encoding, as well as the action of our error-correction procedure. We obtain, for $n=3$, a strict overestimate of the encoding error as a 9th-order polynomial in p , the full expression for which is given in the Supplemental Material [49]. The most

important property of p_{enc} is p_{crit} , defined as the value of p for which $p_{\text{enc}} = p$, and for which $p_{\text{enc}} < p$ whenever $p < p_{\text{crit}}$. The encoding circuit for $n=3$ has $p_{\text{crit}} = 2.8\%$, and when $p \ll 2.8\%$ the relationship is $p_{\text{enc}} \approx (32/63)p \approx 0.51p$. This encoding error is precisely the measurement error of the black box being simulated.

Once the classical information in the qubits input to our unitary circuit has been encoded, it can be processed essentially error-free using the error-correction and computation methods present above. Thus it remains to apply an operation to m qubits that is conditional on the processed information. The bits containing this information are stored in our repetition code. To ensure that the feedback correctly mimics the operation of feedback applied by a classical controller we must take account of the following: (i) a classical controller is assumed to be error-free, and so does not introduce errors that are correlated between the m target bits; (ii) the feedback operation must be implemented with a single control qubit for each target qubit because we are restricted to mesoscopic circuits. Fortunately we can satisfy both demands. Transversal CNOT operations can copy logical bits fault tolerantly. This can provide an ensemble of m logical bits that are essentially correct (to the basic logical failure rate). Applying a series of MAJ1 gates to each logical bit (using $n+1$ iterations for a code at level n), we can provide m qubits with independent errors which approach (to first order in p) the individual failure rate of the MAJ1 [55]. Using these qubits as the controls for the feedback operations gives an error of $(32/63)p \approx 0.51p$ over that of the classical feedback operation. Such an additional error in the feedback appears to be a necessary consequence of the use of mesoscopic circuits for this purpose.

Here we have presented a scheme for fault-tolerant classical computation that significantly outperforms all previous schemes. In doing so we have shown that multiplexing neither requires randomization nor large code sizes as have previously been thought. We have used this new scheme to show that unitary mesoscopic circuits can perform all functions of measurement with errors that remain very close to p .

This research was supported in part by the NSF Project No. PHY-1212413 and by an appointment to the Student Research Participation Program at the U.S. Army Research Laboratory administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and USARL.

-
- [1] J. von Neumann, in *Automata Studies*, edited by C. E. Shannon and J. McCarthy (Princeton University Press, Princeton, NJ, 1956), pp. 43–98.
 - [2] N. Pippenger, in *Proceedings of Symposia in Pure Mathematics*, Vol. 50, The Legacy of John von Neumann, edited by J. Glimm, J. Impagliazzo, and I. Singer (American Mathematical Society, Providence, RI, 1990), p. 311.

- [3] K. Nikolic, A. Sadek, and M. Forshaw, *Nanotechnology* **13**, 357 (2002).
- [4] S. Roy and V. Beiu, *IEEE Trans. Nanotechnol.* **4**, 441 (2005).
- [5] D. Bhaduri, S. Shukla, P. Graham, and M. Gokhale, *IEEE Trans. Nanotechnol.* **6**, 265 (2007).
- [6] V. Beiu, W. Ibrahim, and S. Lazarova-Molnar, in *International Work-Conference on Artificial Neural Networks* (Springer, New York, 2007), pp. 487–496.
- [7] X. Ma and F. Lombardi, in *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems* (IEEE, New York, 2008), pp. 236–244.
- [8] J. Han, E. R. Boykin, H. Chen, J. Liang, and J. A. Fortes, *IEEE Trans. Nanotechnol.* **10**, 1099 (2011).
- [9] B. Sen, Y. Sahu, R. Mukherjee, R. K. Nath, and B. K. Sikdar, *Microelectron. J.* **47**, 7 (2016).
- [10] A. L. Toom, *Adv. Probab.* **6**, 549 (1980).
- [11] P. Gács, in *Proceedings of the fifteenth annual ACM symposium on Theory of computing* (ACM, New York, NY, 1983), pp. 32–41.
- [12] P. Gács and J. Reif, in *Proceedings of the seventeenth annual ACM symposium on Theory of computing* (ACM, New York, NY, 1985), pp. 388–395.
- [13] J. Han and P. Jonker, *Nanotechnology* **14**, 224 (2003).
- [14] F. Peper, J. Lee, F. Abo, T. Isokawa, S. Adachi, N. Matsui, and S. Mashiko, *IEEE Trans. Nanotechnol.* **3**, 187 (2004).
- [15] J. Di and P. K. Lala, in *Emerging Nanotechnologies* (Springer, New York, 2008), pp. 203–226.
- [16] L. Žaloudek and L. Sekanina, in *International Conference on Unconventional Computation* (Springer, New York, 2011), pp. 234–245.
- [17] A. de Maere and L. Ponselet, *J. Stat. Phys.* **147**, 634 (2012).
- [18] L. Ponselet, Ph.D. thesis, Université catholique de Louvain Faculté des Science, 2013.
- [19] J. Mairesse and I. Marcovici, *Theor. Comput. Sci.* **559**, 42 (2014).
- [20] P. Słowiński and R. S. MacKay, *J. Stat. Phys.* **159**, 43 (2015).
- [21] J. Lee, F. Peper, K. Leibnitz, and P. Gu, *Information Sciences (NY)* **352–353**, 150 (2016).
- [22] P. W. Shor, in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS '96* (IEEE Computer Society, Washington, DC, USA, 1996), p. 56.
- [23] A. Y. Kitaev, *Russ. Math. Surv.* **52**, 1191 (1997).
- [24] D. Aharonov and M. Ben-Or, in *Proceedings of the 29th Annual ACM Symposium on the Theory of Computation* (Association for Computing Machinery, New York, 1998), p. 176.
- [25] E. Knill, R. Laflamme, and W. H. Zurek, *Science* **279**, 342 (1998).
- [26] J. Preskill, *Proc. R. Soc. A* **454**, 385 (1998).
- [27] D. Gottesman, *Phys. Rev. A* **57**, 127 (1998).
- [28] A. M. Steane, *Nature (London)* **399**, 124 (1999).
- [29] M. Knill, *Nature (London)* **434**, 39 (2005).
- [30] B. M. Terhal and G. Burkard, *Phys. Rev. A* **71**, 012336 (2005).
- [31] M. A. Nielsen and C. M. Dawson, *Phys. Rev. A* **71**, 042323 (2005).
- [32] P. Aliferis and D. W. Leung, *Phys. Rev. A* **73**, 032308 (2006).
- [33] T. Szkopek, P. Boykin, H. Fan, V. Roychowdhury, E. Yablonovitch, G. Simms, M. Gyure, and B. Fong, *IEEE Trans. Nanotechnol.* **5**, 42 (2006).
- [34] P. Aliferis, D. Gottesman, and J. Preskill, *Quantum Inf. Comput.* **6**, 97 (2006).
- [35] C. M. Dawson, H. L. Haselgrove, and M. A. Nielsen, *Phys. Rev. Lett.* **96**, 020501 (2006).
- [36] K. M. Svore, D. P. Divincenzo, and B. M. Terhal, *Quantum Inf. Comput.* **7**, 297 (2007).
- [37] D. Aharonov and M. Ben-Or, *SIAM J. Comput.* **38**, 1207 (2008).
- [38] P. Aliferis, D. Gottesman, and J. Preskill, *Quantum Inf. Comput.* **8**, 181 (2008).
- [39] K. Fujii and Y. Tokunaga, *Phys. Rev. Lett.* **105**, 250503 (2010).
- [40] R. Koenig, G. Kuperberg, and B. W. Reichardt, *Ann. Phys. (Amsterdam)* **325**, 2707 (2010).
- [41] A. Paetznic and B. W. Reichardt, *Phys. Rev. Lett.* **111**, 090505 (2013).
- [42] C. Jones, *Phys. Rev. A* **87**, 042305 (2013).
- [43] T. Jochym-O'Connor and R. Laflamme, *Phys. Rev. Lett.* **112**, 010505 (2014).
- [44] A. M. Stephens, *Phys. Rev. A* **89**, 022321 (2014).
- [45] E. T. Campbell, *Phys. Rev. Lett.* **113**, 230501 (2014).
- [46] H. Bombín, *Phys. Rev. X* **5**, 031043 (2015).
- [47] V. P. Roychowdhury and P. O. Boykin, in *2005 International Conference on Dependable Systems and Networks (DSN'05)* (IEEE Computer Society, Los Alamitos, CA, 2005), pp. 444–453.
- [48] B. Antonio, J. Randall, W. K. Hensinger, G. W. Morley, and S. Bose, [arXiv:1509.03420](https://arxiv.org/abs/1509.03420).
- [49] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.119.030503> for calculations of the error rates for the gates, error-correction circuits, and universal computation, as well as a brief discussion of postselection, which includes Refs. [50,51].
- [50] J. Lee and F. Peper, in *Automata-2008: Theory and Applications of Cellular Automata*, edited by A. Adamatzky, R. Alonso-Sanz, A. Lawniczak, J. Martinez, K. Morita, and T. Worsch (Luniver Press, Frome, UK, 2008), p. 278.
- [51] D. H. Hoe, in *2010 42nd Southeastern Symposium on System Theory (SSST)* (IEEE, New York, 2010), pp. 258–262.
- [52] K. Jacobs, *Quantum measurement theory and its applications* (Cambridge University Press, Cambridge, England, 2014).
- [53] A. G. Fowler, A. M. Stephens, and P. Groszkowski, *Phys. Rev. A* **80**, 052312 (2009).
- [54] D. P. DiVincenzo and P. Aliferis, *Phys. Rev. Lett.* **98**, 020501 (2007).
- [55] G. A. Paz-Silva, G. K. Brennen, and J. Twamley, *Phys. Rev. Lett.* **105**, 100501 (2010).
- [56] J. Fitzsimons and J. Twamley, *Electronic Notes in Theoretical Computer Science* **258**, 35 (2009).
- [57] K. Fujii, M. Negoro, N. Imoto, and M. Kitagawa, *Phys. Rev. X* **4**, 041039 (2014).
- [58] M. Herold, E. T. Campbell, J. Eisert, and M. J. Kastoryano, *New Phys. J. Quantum Inf.* **1**, 15010 (2015).
- [59] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).
- [60] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, *Phys. Rev. A* **86**, 032324 (2012).
- [61] It should be noted that ϵ (along with its threshold) is more fundamental to the MAJ3 network than p . The former is an

- error rate for any module which acts as a MAJ3. The latter depends on our unitary implementation of the MAJ3 and our quantum error model [49].
- [62] Decoding the logical state (reading it out) is not required for our application here, but we treat it in the Supplemental Material [49].
- [63] For $n = 3$ with the qubits arranged in a square, the maximum number of qubits that lie between any two that must interact is 5, always along straight lines in the array. If three dimensions are utilized, this same maximum distance still applies for universal computation at $n = 4$, which can be accomplished with a $9 \times 9 \times 9$ array.