ARTICLE   OPEN

Check for updates

Real-time processing of stabilizer measurements in a bit-flip code

Diego Ristè [1][✉], Luke C. G. Govia [1], Brian Donovan[1], Spencer D. Fallek[1], William D. Kalfus [1], Markus Brink[2], Nicholas T. Bronn [2] and Thomas A. Ohki [1]

Although qubit coherence times and gate fidelities are continuously improving, logical encoding is essential to achieve fault tolerance in quantum computing. In most encoding schemes, correcting or tracking errors throughout the computation is necessary to implement a universal gate set without adding significant delays in the processor. Here, we realize a classical control architecture for the fast extraction of errors based on multiple cycles of stabilizer measurements and subsequent correction. We demonstrate its application on a minimal bit-flip code with five transmon qubits, showing that real-time decoding and correction based on multiple stabilizers is superior in both speed and fidelity to repeated correction based on individual cycles. Furthermore, the encoded qubit can be rapidly measured, thus enabling conditional operations that rely on feed forward, such as logical gates. This co-processing of classical and quantum information will be crucial in running a logical circuit at its full speed to outpace error accumulation.

INTRODUCTION

Fault-tolerant quantum computation offers the potential for vast computational advantages over classical computing for a variety of problems[1]. The implementation of quantum error correction (QEC) is the first step toward practical realization of any of these applications. This typically requires detecting the occurrence of an error by performing a stabilizer measurement, followed by either a corrective action on the physical device (active QEC), or a frame update in software (passive QEC)[2–14]. In either case, these checkpoints must occur regularly to protect a quantum state throughout the computation. To do that, one needs to rapidly measure the stabilizers with high fidelity and without disrupting the encoded qubit. In spite of these challenges, recent progress has been made in repetitive stabilizer measurements across a diverse range of physical architectures, including trapped ions[15,16], superconducting qubits[17–21], and defects in diamond[22].

Active feed-forward control is useful not only for active error correction[23–26], but also for other QEC schemes employing state injection and magic-state distillation[3,27–38], both of which may be used in implementations of a universal logical gate set. In all cases, determining the appropriate control and implementing it in real time, with minimal latency is a particularly attractive capability for efficient error correction techniques[10].

Furthermore, when the stabilizer measurements cannot be trusted as they themselves are error prone, one can introduce a decoder that uses information from multiple rounds of stabilizer measurements[39,40] or from the spatial connectivity of the device[41] to determine the appropriate correction. Unfortunately, performing the decoding calculation in software at a high level of the hardware stack hinders low-latency correction and fast feed-forward control due to the communication and computation overhead[17].

In this work, we overcome this bottleneck by performing both QEC decoding and control with custom low-latency hardware, which acts as a classical co-processor to our quantum processor. We demonstrate repeated active correction, as well as real-time decoding of multi-round stabilizer measurements. We show that the decoding strategy successfully mitigates stabilizer errors, and identifies the encoded state with a latency far below the qubit coherence times, while matching the results obtained by post processing on a conventional computer.

RESULTS

Repeated stabilizer measurements on a five-qubit device

For our demonstration, we implement a three-qubit code that corrects bit-flip errors ($\hat{X}$), and is sufficient to encode one logical bit of classical memory. We use an IBM five-transmon device similar to *ibmqx2* (refs [42,43]; Fig. 1), of which three transmons (D_1, D_2, D_3) are used as data qubits, and two (A_t, A_b) as ancilla qubits to measure the stabilizers. Each qubit is coupled to a dedicated resonator for readout and control. Additional resonators dispersively couple D_1, D_2 with A_t and D_2, D_3 with A_b.

We perform CNOT gates between data and ancilla qubits by using a sequence of single-qubit gates, and a ZX_{90} rotation driven by the cross-resonance interaction[44]. By applying two CNOT gates in succession controlled by two different data qubits with a single ancilla as the target, the parity of the data qubit pair is mapped onto the ancilla state. The same protocol is applied simultaneously to both data qubit pairs, with the shared qubit D_2 interacting first with A_t, then with A_b (Fig. 1b). The ancilla measurement result $a_{t,b} = 0(1)$ ideally corresponds to even (odd) parity for the corresponding pair. We refer to the complete sequence comprised of four CNOT gates and ancilla measurement as a single error correction cycle. The result of each cycle (the measurements a_t, a_b) is a syndrome that identifies which data qubit (if any) has most likely been subjected to an $\hat{X}$ error.

Key to preserving a logical state is the capability of repeating such stabilizer measurements[16,17,20,21], which has two technical requirements. First, the two ancilla qubits must be reused at every cycle, either by resetting them to the ground state[45,46] or by tracking their state. For either reset or state tracking by

---
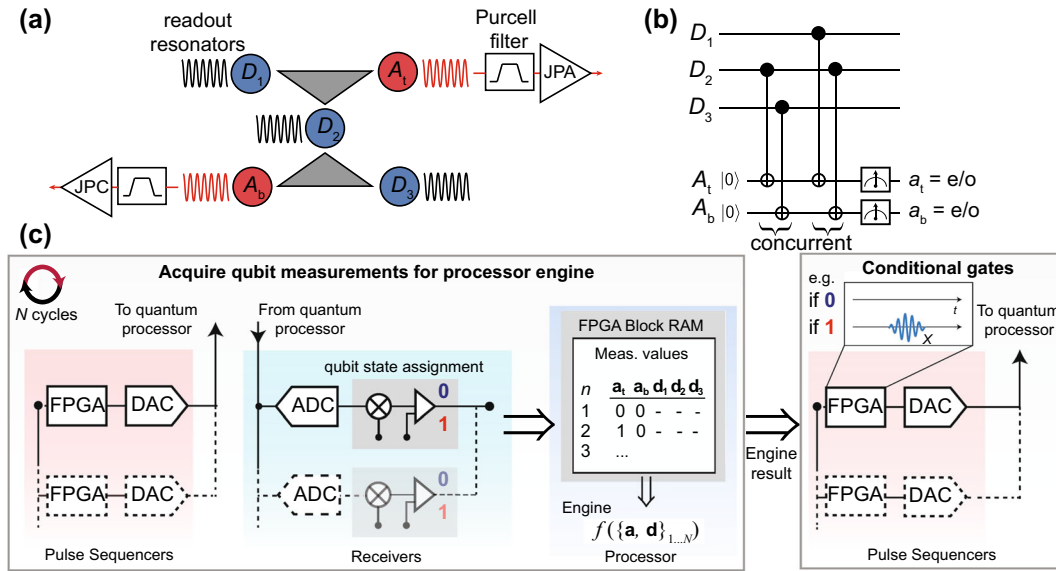
Fig. 1 Stabilizer measurements on a five-transmon device. a Schematics of the device implementing the bit-flip code with the data qubits D_1, D_2, D_3 and the ancilla qubits A_t, A_b. Triangles represent the bus resonators coupling the qubits at their vertices. A Josephson Parametric Amplifier[52] (Converter[53]) enhances the readout of A_t (A_b). **b** Gate and measurement sequence for one round of stabilizer measurements. CNOT gates map the parity of D_1–D_2 (D_2–D_3) onto A_t (A_b) and are applied concurrently two at a time. **c** Simplified setup diagram highlighting the closed loop central to active error correction and decoding. For QEC cycles $n \leq N$, the processor stores the stabilizer results $\{a_t, a_b\}_n$ acquired by the receivers. When $n = N$, it executes a custom function (here decoder) and broadcasts the result back to the pulse sequencers for conditional gates (here $\hat{X}$). The same framework is used to execute a logical data measurement, where the majority function is applied on a single acquisition $\{d_1, d_2, d_3\}$.

measurement, we need to ensure that the readout process is nondestructive, i.e., the result is consistent with the qubit state at the end of the measurement. This sets an upper limit to the allowable photon number, and therefore to the readout fidelity (Supplementary Methods). Second, the readout cavities must be depleted of photons before starting the new cycle to prevent gate errors. To accelerate the cavity relaxation to its steady state (near vacuum), we employ the CLEAR technique[47] for the slower resonator coupled to A_b (Supplementary Table 2), reducing its average photon population to <0.1 in 600 ns. Altogether, we measure a single-round joint stabilizer readout fidelity of 0.61, averaged over the eight computational states of the data qubits (Supplementary Fig. 3).

Real-time processing of measurement results
An integral part of our experiment is the interdependence of qubit readout and control, mediated by fast processing of the measurement results. For this purpose, we use a combination of custom-made and off-the-shelf hardware, consisting of pulse sequencers, receivers, and a processor (Fig. 1c), all based on field-programmable gate arrays (FPGAs). In particular, the storage capability of the processor FPGA enables expansion beyond one-time feedback protocols[18,48,49], where conditional actions rely on single, or joint but simultaneous measurements. During each QEC cycle, digital-to-analog converters in the pulse sequencers produce a pre-programmed series of gate and measurement pulse envelopes. Each returning readout signal is captured by a receiver channel via an analog-to-digital converter, where it is integrated and compared against a calibrated threshold to determine the qubit state[48]. The processor collects all the digitized results and stores them in memory. After a preset number N of cycles have been executed, the processor feeds the stored values to an internal custom calculation engine. The engine function result is broadcast to the pulse sequencers, which conditionally apply a corresponding set of gates. The overall latency to store and process the classical data and to issue a conditional pulse is

590 ns, a small fraction of coherence times (Supplementary Methods).

Preserving an encoded qubit state by active error correction
We will explore three distinct approaches to the bit-flip code. In all cases, we first prepare the logical excited state $|111\rangle$. Next, we apply one of the following schemes: (i) uncorrected, in which the cycle is performed but no correction is applied to the data qubits based on the syndrome measurements, and the ancilla qubits are not reset; (ii) repeated error correction (REC), in which, at the end of each cycle, a correction gate is conditionally applied to the data qubits based on the last syndrome results only, and the ancillas are reset; (iii) decoder error correction (DEC), in which N cycles are performed without ancilla reset or corrective gates, and the set of syndromes from all N cycles are used to determine and apply the optimal correction via a decoder[40]. To assess how well each code has protected the prepared state after a desired number of cycles, we perform a logical data measurement. This involves measuring the constituent physical data qubits and computing the majority function over the digitized results $\{d_1, d_2, d_3\}$. In cases (i) and (ii), the majority function is calculated offline. In case (iii), the processor computes both the decoding and majority functions sequentially, making the result available for further conditional operations.

We begin by comparing the REC protocol to the uncorrected case (Fig. 2). For REC, there is a one-to-one relation between the two-bit value syndrome $\{a_t, a_b\}$ and one of the three possible corrective $\hat{X}$ gates (in blue in Fig. 2a), or no gate at all. The same syndrome value is used to actively reset the ancilla qubits for use in subsequent cycles (Fig. 2b).

When error detection is based on a single round of stabilizer measurements, it is impossible to distinguish between the targeted data errors and stabilizer measurement errors, largely caused by imperfect CNOT gates and ancilla readout. Thus, the resulting active correction directly propagates syndrome errors to the data qubits. These errors dominate in the case of d_1 and d_3,
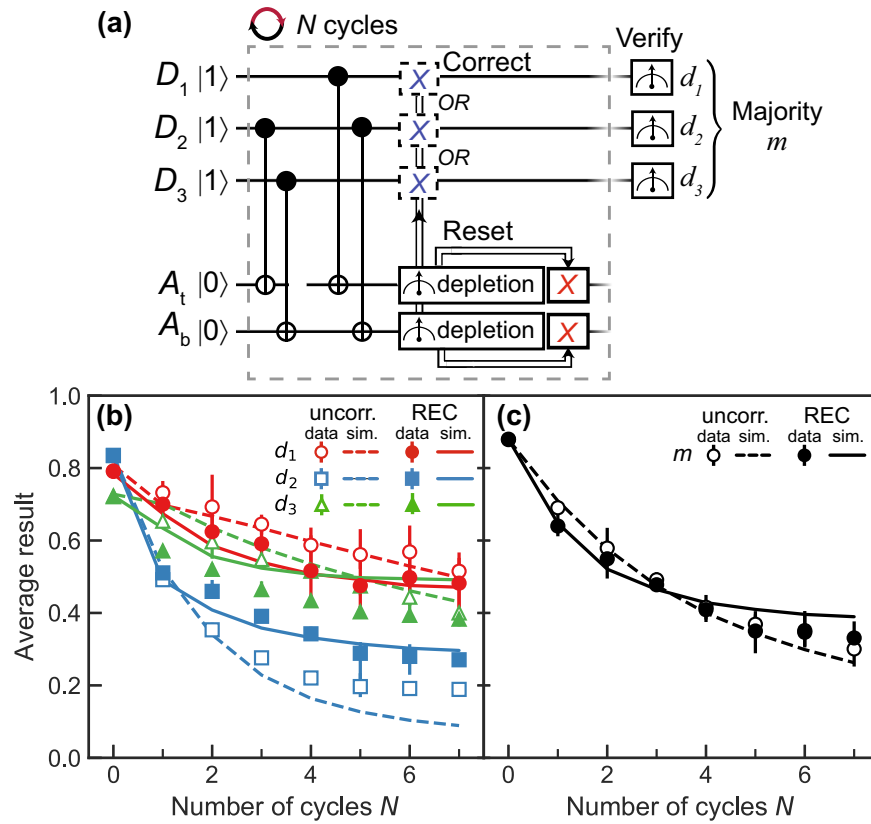
Fig. 2 Real-time repeated error correction with ancilla reset. a Gate sequence for independent cycles of active error correction. **b** Average digitized result for each of D_1 (red), D_2 (blue), and D_3 (green) measurements after initialization in $|111\rangle$ and N cycles of **a** (full symbols). The results are compared to those obtained with the same circuit, but where data correct and ancilla reset operations are omitted (empty symbols). **c** Average majority vote m for the data in **b**. The closed-loop circuit (full symbols) does not improve over open loop (empty). In both **b** and **c**, solid (dashed) curves are obtained from numerical simulation for the corrected (uncorrected) case (Supplementary Methods). Error bars: range over five repetitions of the experiment (3000 shots each).

whose average values decay faster with the number of cycles than without active correction (Fig. 2b). Conversely, the larger intrinsic error per cycle for d_2 (due to its shorter T_1) is partially compensated by the protocol. Overall, this gain nearly balances out the errors introduced by the active error correction, as shown by comparing the results of the majority function (Fig. 2b). REC can be thought of as repeated one-time feedback, where the processor storage and calculation engine are unused. The added latency is considerable: for each cycle, the stabilizer results are aggregated by the processor and forwarded to the pulse sequencers (400 ns), followed by correction and reset operations (160 ns).

Improvements in logical state protection are achieved by correlating multiple stabilizer measurements using the DEC protocol. In ref. [17], a simplified minimum-weight perfect-matching decoder[40] was used to post-process the syndrome results, and differentiate between true data bit flips and false positives. We apply the same method (Fig. 3), but with the crucial difference that the results are processed in real time. Specifically, the processor acquires stabilizer measurement results for N cycles, and uses the engine to decode them into the appropriate set of $\hat{X}$ gates using a precomputed lookup table. These corrections are then applied by the pulse sequencers on the data qubits. Finally, the data qubits are measured as in Fig. 2, with the majority function also computed on the processor. Whereas for $N \leq 2$ the decoder is ineffective—as there are not enough records to identify ancilla readout errors—a gap emerges at larger N (Fig. 3c) in favor of the decoder. Furthermore, this scheme eliminates the per-cycle latency cost; the latest ancilla results can be processed while the next cycle is executing. The total additional latency becomes fixed

at 1300 ns (590 ns for each processor engine call, 120 ns for corrective gates), approximately equal to that accrued over two REC cycles (Table 1).

In an effort to minimize errors by avoiding unnecessary quantum gates, it is important to move as many operations as possible to the classical hardware. In this case, we note that measuring the data qubits immediately after conditional $\hat{X}$ gates is equivalent to inverting the classical measurement result. Therefore, in a final experiment (Fig. 4, triangles), we dispense with the active correction and instead filter the d_i results based on the decoder output. This corresponds to a Pauli frame update (PFU)[2] applied just before the measurement. The slight reduction in latency (120 ns) and error rate due to the absence of these pulses consistently achieves 1–2% improvement for all N over the actively corrected case. The results match those obtained by post-processing all the data in software (diamonds), confirming that the fast classical loop works as expected.

Finally, we evaluate the decoder against the majority result we would obtain by replacing the QEC gates and measurements, with an idle time of equal duration. The result shows that, for all $N > 1$, DEC has a higher success probability of determining the initial state compared to free decay (Fig. 4, gray curve).

DISCUSSION

Although the experiment ends with the measurement of the data qubits, the readily available majority result may be used to condition additional operations on a second encoded qubit. This would be the case, for instance, to teleport an S gate using a logical ancilla[27]. More generally, the ability to update the Pauli
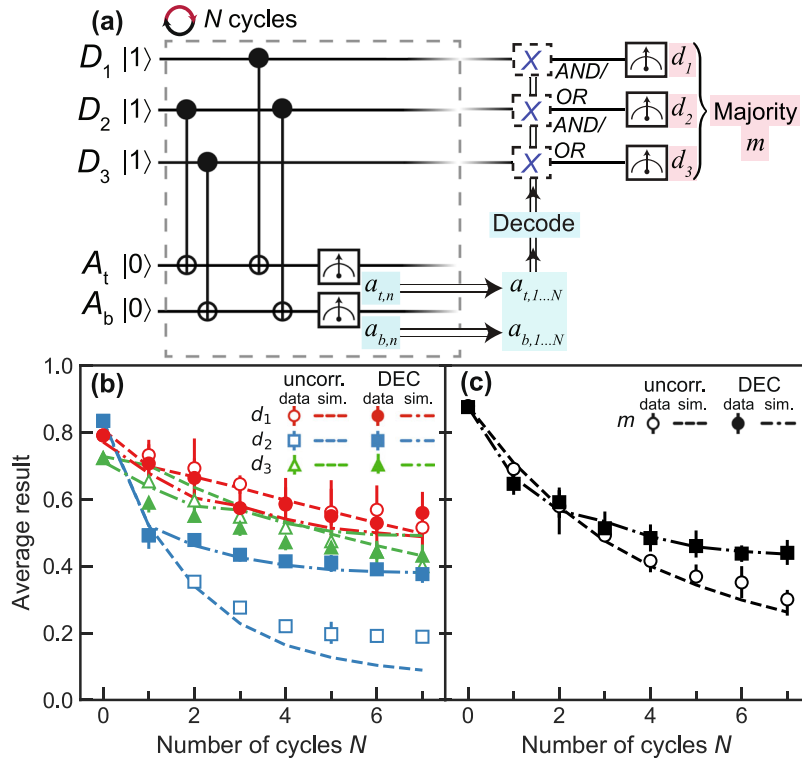
Fig. 3 Real-time decoder-based error correction and logical measurement. a Gate sequence with active correction, based on N rounds of stabilized measurements, followed by simultaneous data qubit measurements and majority function. The sequence in each cycle is identical to the uncorrected case in Fig. 2 (including the cavity depletion, not shown). The shaded areas highlight the real-time processing. **b** Average results d_1, d_2, d_3 following the conditional $\hat{X}$ gates in **a** (solid symbols), compared to the uncorrected case (same as Fig. 2b; empty). Dash-dotted (dashed) curves are obtained from numerical simulation for the real-time corrected (uncorrected) case. **c** Average majority vote m for the data in **b**. Error bars: range over five repetitions of the experiment (3000 shots each).

Table 1. The latency includes the time for classical processing, as well as active correction, where applicable.

Protocol	Latency/cycle (ns)	Fixed latency (ns)
REC	560	>1e6
DEC	0	1300
DEC + PFU	0	1180

frame in real time will be essential to implement quantum algorithms at the logical level. Since not all gates can be transversal in any given code[50], conditional operations based on the current frame can be used to complete the universal gate set (e.g., T gates in the surface code[30]).

In conclusion, we have demonstrated the repeated measurement and real-time processing of stabilizers for a minimal bit-flip code. An intertwined readout and control system provides a real-time interface to the quantum processor, converting a series of stabilizer results to the current Pauli frame without interrupting the execution of a potential algorithm. This approach is not limited to superconducting qubits, but is applicable to any quantum computing platform that faces coherence-limited operation.

Finally, we touch upon the applicability of the presented control architecture to larger circuits. Clearly, the use of a look-up-table as a decoder is a limiting factor, with the number of entries scaling exponentially with both circuit depth and width. However, we predict that 3–4 cycles of a small surface code[40] are within reach of this approach, provided that the processor is upgraded with commercially available, albeit significantly larger, memory.
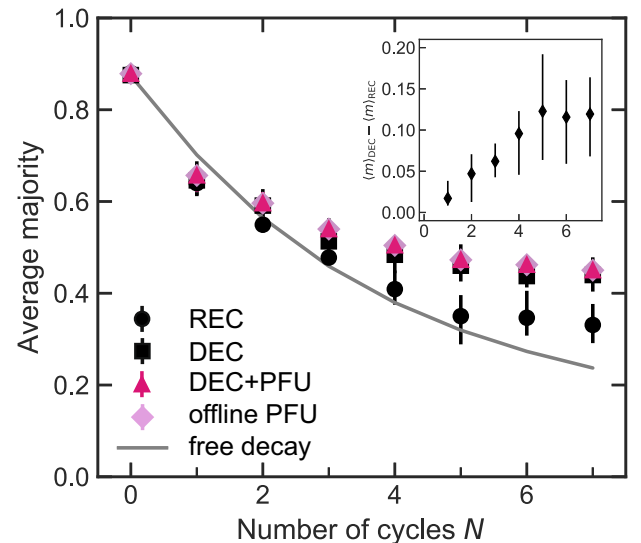
Fig. 4 Comparison between different QEC schemes. Measured average majority result m for REC (circles, from Fig. 2) and DEC with final active correction (squares, Fig. 3). The results obtained by replacing the correction in DEC with a PFU (triangles) match those obtained by post-processing (diamonds) the uncorrected data, using the same decoder. In all DEC cases and for $N > 1$, m stays above the result expected from an equivalent idle time (gray curve). Inset: difference between the average majority result for DEC with PFU and REC. Error bars: range over five repetitions of the experiment (3000 shots each).

METHODS

Classical control hardware

The real-time protocols presented above rely on the interconnection between the receivers (two Innovative Integration X6-1000M digitizers), the processor (BBN Trigger Distribution Module—TDM in ref. [48]), and the pulse sequencers (BBN Arbitrary Pulse Sequencers—APS2). The event sequence and the communication between those instruments, as well as their interface with the qubit device (hosted in a Bluefors BF-LD400), are detailed in the Supplementary Methods.

Numerical simulations

In this section, we describe the model and methods used to obtain the numerical simulation results shown in the main text. We chose not to do a full time-dependent master equation simulation of the error correction, as such an open system simulation is numerically intensive. Further, for the real-time error correction with ancilla reset, interspersing strong measurement and conditional operations within a time-dependent evolution is a nontrivial task. Instead, we use a simulation model that is approximate, but with well-controlled error that does not significantly reduce the accuracy of our results. We aim for qualitative agreement with the experimental results, using a model with no fit parameters (i.e., we do not search among models for a best fit to the data), with all free parameters in our model determined by independent characterization of the device.

Each round of the error correction can be thought of as an entangling operation (CNOT gates), followed by a measurement operation, followed by an optional correction and ancilla reset operation. This can be represented by the following composition of linear maps

$$\rho_{i+1} = \mathcal{R}_{\mathbf{m}} \circ \mathcal{M} \circ \mathcal{E}(\rho_i), \tag{1}$$

where ρ_i is the state after the ith round of error correction, and $\mathcal{E}$, $\mathcal{M}$, and $\mathcal{R}_{\mathbf{m}}$ are the entangling, measurement, and correction/reset operations, respectively, with the correction/reset operation being conditional on the vector of ancilla measurement outcomes $\mathbf{m}$. The modeling of these operations is described in detail in the Supplementary Methods.

DATA AVAILABILITY

The data that support the findings of this study and the code used to analyze the data are available from the corresponding author upon request.

REFERENCES

1. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
2. Knill, E. Quantum computing with realistically noisy devices. *Nature* **434**, 39–44 (2005).
3. DiVincenzo, D. P. & Aliferis, P. Effective fault-tolerant quantum computation with slow measurements. *Phys. Rev. Lett.* **98**, 020501 (2007).
4. Bombín, H. Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes. *N. J. Phys.* **17**, 083002 (2015).
5. Bravyi, S. & Cross, A. Doubled color codes. Preprint at https://arxiv.org/abs/1509.03239 (2015).
6. Jochym-O'Connor, T. & Bartlett, S. D. Stacked codes: universal fault-tolerant quantum computation in a two-dimensional layout. *Phys. Rev. A* **93**, 022323 (2016).
7. Chamberland, C., Jochym-O'Connor, T. & Laflamme, R. Thresholds for universal concatenated quantum codes. *Phys. Rev. Lett.* **117**, 010501 (2016).
8. Chamberland, C., Jochym-O'Connor, T. & Laflamme, R. Overhead analysis of universal concatenated quantum codes. *Phys. Rev. A* **95**, 022313 (2017).
9. Chamberland, C. & Jochym-O'Connor, T. Error suppression via complementary gauge choices in Reed-Muller codes. *Quantum Sci. Technol.* **2**, 035008 (2017).
10. Chamberland, C., Iyer, P. & Poulin, D. Fault-tolerant quantum computing in the Pauli or Clifford frame with slow error diagnostics. *Quantum* **2**, 43 (2018).
11. Chamberland, C. & Cross, A. W. Fault-tolerant magic state preparation with flag qubits. *Quantum* **3**, 143 (2019).
12. Jochym-O'Connor, T. Fault-tolerant gates via homological product codes. *Quantum* **3**, 120 (2019).
13. Lavasani, A., Zhu, G. & Barkeshli, M. Universal logical gates with constant overhead: Instantaneous Dehn twists for hyperbolic quantum codes. *Quantum* **3**, 180 (2019).
14. Chamberland, C. & Noh, K. Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits. Preprint at https://arxiv.org/abs/2003.03049 (2020).
15. Schindler, P. et al. Experimental repetitive quantum error correction. *Science* **332**, 1059–1061 (2011).
16. Negnevitsky, V. et al. Repeated multi-qubit readout and feedback with a mixed-species trapped-ion register. *Nature* **563**, 527 (2018).
17. Kelly, J. et al. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature* **519**, 66–69 (2015).
18. Ofek, N. et al. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **536**, 441–445 (2016).
19. Hu, L. et al. Quantum error correction and universal gate set operation on a binomial bosonic logical qubit. *Nat. Phys.* **15**, 503 (2019).
20. Andersen, C. K. et al. Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits. *npj Quantum Inf.* **5**, 1–7 (2019).
21. Bultink, C. C. et al. Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements. *Sci. Adv.* **6**, eaay3050 (2020).
22. Cramer, J. et al. Repeated quantum error correction on a continuously encoded qubit by real-time feedback. *Nat. Commun.* **7**, 11526 (2016).
23. Paetznick, A. & Reichardt, B. W. Universal fault-tolerant quantum computation with only transversal gates and error correction. *Phys. Rev. Lett.* **111**, 090505 (2013).
24. Jochym-O'Connor, T. & Laflamme, R. Using concatenated quantum codes for universal fault-tolerant quantum gates. *Phys. Rev. Lett.* **112**, 010505 (2014).
25. Anderson, J. T., Duclos-Cianci, G. & Poulin, D. Fault-tolerant conversion between the Steane and Reed-Muller quantum codes. *Phys. Rev. Lett.* **113**, 080501 (2014).
26. Yoder, T. J., Takagi, R. & Chuang, I. L. Universal fault-tolerant gates on concatenated stabilizer codes. *Phys. Rev. X* **6**, 031039 (2016).
27. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A* **71**, 022316 (2005).
28. Bravyi, S. & Haah, J. Magic-state distillation with low overhead. *Phys. Rev. A* **86**, 052329 (2012).
29. Reichardt, B. W. Quantum universality from magic states distillation applied to CSS codes. *Quantum Inf. Process.* **4**, 251–264 (2005).
30. Fowler, A. G., Mariantoni, M., Martinis, J. M. & Cleland, A. N. Surface codes: towards practical large-scale quantum computation. *Phys. Rev. A* **86**, 032324 (2012).
31. Fowler, A. G., Devitt, S. J. & Jones, C. Surface code implementation of block code state distillation. *Sci. Rep.* **3**, 1–6 (2013).
32. Meier, A. M., Eastin, B. & Knill, E. Magic-state distillation with the four-qubit code. *Quantum Info. Comput.* **13**, 195–209 (2013).
33. O'Gorman, J. & Campbell, E. T. Quantum computation with realistic magic-state factories. *Phys. Rev. A* **95**, 032338 (2017).
34. Haah, J., Hastings, M. B., Poulin, D. & Wecker, D. Magic state distillation with low space overhead and optimal asymptotic input count. *Quantum* **1**, 31 (2017).
35. Haah, J. & Hastings, M. B. Codes and protocols for distilling T, controlled-S, and Toffoli gates. *Quantum* **2**, 71 (2018).
36. Haah, J., Hastings, M. B., Poulin, D. & Wecker, D. Magic state distillation at intermediate size. *Quantum Info. Comput.* **18**, 114–140 (2018).
37. Hastings, M. B. & Haah, J. Distillation with sublogarithmic overhead. *Phys. Rev. Lett.* **120**, 050504 (2018).
38. Litinski, D. Magic state distillation: not as costly as you think. *Quantum* **3**, 205 (2019).
39. Dennis, E., Kitaev, A., Landahl, A. & Preskill, J. Topological quantum memory. *J. Math. Phys.* **43**, 4452–4505 (2002).
40. Tomita, Y. & Svore, K. M. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A* **90**, 062320 (2014).
41. Heim, B., Svore, K. M. & Hastings, M. B. Optimal circuit-level decoding for surface codes. Preprint at https://arxiv.org/abs/1609.06373 (2016).
42. Abdo, B. et al. *Qiskit/ibmq-device-information*. https://github.com/Qiskit/ibmq-device-information/tree/master/backends/yorktown/V1 (2019).
43. Ristè, D. et al. Demonstration of quantum advantage in machine learning. *npj Quantum Inf.* **3**, 16 (2017).
44. Chow, J. M. et al. Implementing a strand of a scalable fault-tolerant quantum computing fabric. *Nat. Commun.* **5**, 4015 (2014).
45. Ristè, D., Bultink, C. C., Lehnert, K. W. & DiCarlo, L. Feedback control of a solid-state qubit using high-fidelity projective measurement. *Phys. Rev. Lett.* **109**, 240502 (2012).
46. Egger, D. et al. Pulsed reset protocol for fixed-frequency superconducting qubits. *Phys. Rev. Appl.* **10**, 044030 (2018).

47. McClure, D. T. et al. Rapid driven reset of a qubit readout resonator. *Phys. Rev. Appl.* **5**, 011001 (2016).
48. Ryan, C. A., Johnson, B. R., Ristè, D., Donovan, B. & Ohki, T. A. Hardware for dynamic quantum computing. *Rev. Sci. Instrum.* **88**, 104703 (2017).
49. Salathé, Y. et al. Low-latency digital signal processing for feedback and feed-forward in quantum computing and communication. *Phys. Rev. Appl.* **9**, 034011 (2018).
50. Eastin, B. & Knill, E. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.* **102**, 110502 (2009).
51. Das, P. et al. A Scalable decoder micro-architecture for fault-tolerant quantum computing. Preprint at https://arxiv.org/abs/2001.06598 (2020).
52. Hatridge, M., Vijay, R., Slichter, D. H., Clarke, J. & Siddiqi, I. Dispersive magneto-metry with a quantum limited SQUID parametric amplifier. *Phys. Rev. B* **83**, 134501 (2011).
53. Abdo, B., Schackert, F., Hatridge, M., Rigetti, C. & Devoret, M. Josephson amplifier for qubit readout. *Appl. Phys. Lett.* **99**, 162506 (2011).

AUTHOR CONTRIBUTIONS

D.R. designed and performed the experiment. L.C.G.G. implemented the numerical simulations. B.D. developed the classical control architecture with input from S.D.F. and W.D.K. M.B. and N.T.B. designed and fabricated the qubit device and packaging. D.R. and L.C.G.G. interpreted the data with advice from T.A.O. D.R., L.C.G.G. and S.D.F. wrote the manuscript with input from W.D.K. and T.A.O.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information is available for this paper at https://doi.org/10.1038/s41534-020-00304-y.

Correspondence and requests for materials should be addressed to D.R.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.