Qubit metrology for building a fault-tolerant quantum computer

npj Quantum Information (2015) 1, 15005; doi:10.1038/npjqi.2015.5; published online 27 October 2015

Recent progress in quantum information has led to the start of several large national and industrial efforts to build a quantum computer. Researchers are now working to overcome many scientific and technological challenges. The program’s biggest obstacle, a potential showstopper for the entire effort, is the need for high-fidelity qubit operations in a scalable architecture. This challenge arises from the fundamental fragility of quantum information, which can only be overcome with quantum error correction.1 In a fault-tolerant quantum computer the qubits and their logic interactions must have errors below a threshold: scaling up with more and more qubits then brings the net error probability down to appropriate levels needed for running complex algorithms. Reducing error requires solving problems in physics, control, materials and fabrication, which differ for every implementation. I explain here the common key driver for continued improvement - the metrology of qubit errors.

We must focus on errors because classical and quantum computation are fundamentally different. The classical NOT operation in CMOS electronics can have zero error, even with moderate changes of voltages or transistor thresholds. This enables digital circuits of enormous complexity to be built as long as there are reasonable tolerances on fabrication. In contrast, quantum information is inherently error prone because it has continuous amplitude and phase variables, and logic is implemented using analog signals. The corresponding quantum NOT, a bit-flip operation, is produced by applying a control signal that can vary in amplitude, duration and frequency. More fundamentally, the Heisenberg uncertainty principle states that it is impossible to directly stabilise a single qubit as any measurement of a bit-flip error will produce a random flip in phase. The key to quantum error correction is measuring qubit parities, which detects bit flips and phase flips in pairs of qubits. As explained in the text box, the parities are classical like so their outcomes can be known simultaneously.

When parity changes, one of the two qubits had an error, but which one is not known. To identify, encoding must use larger numbers of qubits. This idea can be understood with a simple classical example, the 3-bit repetition code as described in Figure 1. Logical states 0 (1) are encoded as 000 (111), and measurement of parities between adjacent bits A–B and B–C allows the identification (decoding) of errors as long as there is a change of no more than a single bit. To improve the encoding to detect both order n = 1 and n = 2 errors, the repetition code is simply increased in size to 5 bits, with four parity measurements between them. Order n errors can be decoded from 2n+1 bits and 2n parity measurements.

Quantum codes allow for the decoding of both bit- and phase-flip errors given a set of measurement outcomes. As for the above example, they decode the error properly as long as the number of errors is order n or less. The probability for a decoding error can be computed numerically using a simple depolarisation model that assumes a random bit- or phase-flip error of probability ε for each physical operation used to measure the parities. By comparing the

\[ P_l \approx \Lambda^{-n+1} \quad (1) \]

\[ \Lambda = \epsilon_t / \epsilon \quad (2) \]

where εt is the threshold error, fit from the data. The error suppresion factor is Λ, the key metrological figure of merit that quantifies how much the decoding error drops as the order n increases by one. Note that P_l scales with ε−n−1, as expected for n+1 independent errors. The key idea is that once the physical errors ε are lower than the threshold εt, then Λ > 1 and making the code larger decreases the decoding error exponentially with n. When Λ < 1 error detection fails, and even billions of bad qubits do not help.

A key focus for fault tolerance is making qubit errors less than the threshold. For Λ to be as large as possible, we wish to encode with the highest threshold εt. The best practical choice is the surface code,2,3 which can be thought of as a two-dimensional version of the repetition code that corrects for both bit and phase errors. A 4n+1 by 4n+1 array of qubits performs n-th order error correction, where about half of the qubits are used for the parity measurements. It is an ideal practical choice for a quantum computer because of other attributes: (i) only nearest neighbour interactions are needed, making it manufacturable with integrated circuits; (ii) the code is upward compatible to logical gates, where measurements are simply turned off; (iii) the code is tolerant up to a significant density (~10%) of qubit dropouts from fabrication defects; (iv) the high error threshold arises from the low complexity of the parity measurement; a code with higher threshold is unlikely; (v) the simplicity of the measurement brings more complexity to the classical decoding algorithm, which fortunately is efficiently scalable; (vi) detected errors can be tracked in software, so physical feed-forward corrections using bit- or phase-flip gates are not needed; (vii) the prediction equation (1) for P_l is strictly valid only for the operative range Λ ≳ 10, where the threshold is εt ≈ 2%. At break-even Λ = 1, the threshold is significantly smaller, 0.7%.

Typical quantum algorithms use ~1018 operations,3 so we target a logical error P_l = 10−18. Assuming an improvement Λ = 10 for each order, we need n = 17 encoding. The number of qubits for the surface code is \(4(17+1)^2 = 4,761\). For Λ = 100, this number lowers by a factor of 4. Although this seems like a large number of qubits from the perspective of present technology, we should remember that a cell phone with \(10^{12}\) transistors, now routinely owned by most people in the world, was inconceivable only several decades ago.

Hardware requirements can be further understood by separating out the entire parity operation into one- and two-qubit logic and measurement components. Assuming errors in only one of these components, break-even thresholds are, respectively, 4.3%, 1.25% and 12%: the two-qubit error is clearly the most important, whereas measurement error is the least important. For the practical case when all components have non-zero errors, I propose the threshold targets

\[ \epsilon_t \leq 0.1\% \quad (3) \]
In level I, one- and two-qubit experiments measure coherence times $T_1$ and $T_2$, and show basic functionality of qubit gates. Along with the one-qubit gate time $t_{\text{g},1}$, an initial estimate of gate error can be made. Determining the performance of a two-qubit gate is much harder as other decoherence or control errors will typically degrade performance. Swapping an excitation between two qubits is a simple method to determine whether coherence has degraded, with a magnitude that is consistent with the bit-flip threshold is reached. Note that although it is tempting to isolate qubits from the environment to lower one-qubit errors, in practice this often makes it harder to couple them together for two-qubit logic; I call such strategy 'neutrino-ised qubits'.

In the life cycle of a qubit technology, experiments start with a single qubit and then move to increasing complexity. Two-qubit demonstrations and metrology. The typical progression is illustrated in Figure 2, where the technology levels and their metrics are shown together.

In level II, more qubits are measured in a way that mimics the interactions are desired. A variety of cross-talk measurements based on RB are useful metrology tools.

In level III an error detection or correction algorithm is performed, representing a complex systems test of all components. Qubit errors have to be low enough to perform many complex qubit operations. Experiments work to extend the lifetime of an encoded logical state, typically by adding errors to the various components to show improvement from the detection protocol relative to the added errors.

At level IV, the focus is measuring $\Lambda > 1$, demonstrating how a logical qubit can have less and less error by scaling up the order $n$ of error correction. The logical qubit must be measured in first and second orders, which requires parity measurements that are repetitive in time so as to include the effect of measurement errors. Note that extending the lifetime of a qubit state in first order is not enough to determine $\Lambda$. Measuring $\Lambda > 1$ indicates that all first-order decoding errors have been properly corrected, and that further scaling up should give lower logical errors. Because 81 qubits are needed for the surface code with $n = 2$, a useful initial test is for bit-flip errors, requiring a linear array of nine qubits. These experiments are important as they connect the error metrics of the qubits, obtained in level II, to actual fault-tolerant performance $\Lambda$. As there are theoretical and experimental approximations in this connection, including the depolarisation assumption for theory and RB measurement for experiment, this checks the whole framework of computing fault tolerance. A fundamentally important test for $n \geq 2$ is whether $\Lambda$ remains constant, as correlated errors would cause $\Lambda$ to decrease. Level IV tests continue until the order $n$ is high enough to convincingly demonstrate an exponential suppression of error. A significant challenge here is to achieve all error thresholds in one device and in a scalable design.

An experiment measuring the bit-flip suppression factor $\Lambda_X$ has been done with a linear chain of nine superconducting qubits. The measurement $\Lambda_X = 3.2$ shows logical errors have been reduced, with a magnitude that is consistent with the bit-flip threshold of 3% and measured errors. This is the first demonstration that individual error measurements can be used to predict fault tolerance. For bit and phase fault tolerance, we need to improve only two-qubit errors and then scale.

---

**Figure 1.** 3-bit classical repetition code for bits A, B and C with parity measurements between A–B and B–C. Table shows all combination of inputs and the resulting parity measurements. For an initial state of all zeros, a unique decoding from the measurement to the actual error is obtained for only the top four entries, where there is no more than a single bit error (order $n = 1$).

$e_2 \leq 0.1\%$  \hspace{1cm} (4)

$e_m \leq 0.5\%$, \hspace{1cm} (5)

which gives $\Lambda \geq 17$. It is critical that all three error thresholds be met, as the worst performing error limits the logical error $P_L$. Measurement errors $e_2$ and $e_m$ can be larger because its single-component threshold 12% is high. Two-qubit error $e_2$ is the most challenging because its physical operation is much more complex than for single qubits. This makes $e_2$ the primary metric around which the hardware should be optimised. The single-qubit error $e_1$, being easier to optimise, should readily be met if the two-qubit threshold is reached. Note that although it is tempting to isolate qubits from the environment to lower one-qubit errors, in practice this often makes it harder to couple them together for two-qubit logic; I call such strategy 'neutrino-ised qubits'.

In the life cycle of a qubit technology, experiments start with a single qubit and then move to increasingly more complex multi-qubit demonstrations and metrology. The typical progression is illustrated in Figure 2, where the technology levels and their metrics are shown together.

---

**Figure 2.** Life cycle of a qubit. Illustration showing the increasing complexity of qubit experiments, built up on each other, described by technology levels I through VII. Numbers in parenthesis shows approximate qubit numbers. Key metrics are shown at bottom. Errors for one qubit, two qubit and measurement are described by $e_1$, $e_2$ and $e_m$ respectively, which leads to an error suppression factor $\Lambda$. Fault-tolerant error correction is achieved when $\Lambda > 1$. Scaling to large $n$ leads to $P_L \rightarrow 0$.
In level V, as the lifetime of a logical state has been extended, the goal is to perform logical operations with minuscule error. Similar to classical logic that can be generated from the NOT and AND gates, arbitrary quantum logic can be generated from a small set of quantum gates. Here all the Clifford gates are implemented, such as the S, Hadamard or controlled NOT. The logical error probabilities should be measured and tested for degradation during logical gates.

In level VI, the test is for the last and most difficult logic operation, the T gate, which phase shifts the logical state by $45^\circ$. Here state distillation must be demonstrated, and feed-forward from qubit errors conditionally controls a logical S gate. Because logical errors can be readily accounted for in software for all the logical Clifford gates in level V, feed-forward is only needed for this non-Clifford logical T gate.

Level VII is for the full quantum computer.

The strategy for building a fault-tolerant quantum computer is as follows. At level I, the coherence time should be at least 1,000 times greater than the gate time. At level II, all errors need to be less than threshold, with particular attention given to hardware scaling of CMOS and Moore’s law, where the main aim is to decrease transistor size.

In conclusion, progress in making a fault-tolerant quantum computer must be closely tied to error metrology, as improvements with scaling will only occur when errors are below threshold. Research should particularly focus on two-qubit gates, as they are the most difficult to operate well with low errors. As experiments are now within the fault-tolerant range, many exciting developments are possible in the next few years.

**COMPETING INTEREST**

The author declares no conflict of interest.

John M Martinis,1,2

1Department of Physics, University of California, Santa Barbara, CA, USA and
2Google, Inc., Santa Barbara, CA, USA

E-mail: martinis@physics.ucsb.edu

REFERENCES

1. Shor PW. Scheme for reducing decoherence in quantum computer memory. Phys Rev A 1995; 52: R2493.
2. Bravyi SB, Yu A. Quantum codes on a lattice with boundary. Kitaev 1998; arXiv quant-ph/9811052.
3. Fowler AG, Mariotti M, Martinis JM, Cleland AN. Surface codes: Towards practical largescale quantum computation. Phys Rev A 2012; 86: 032324.
4. Devoret MH, Schoelkopf RJ. Superconducting Circuits for Quantum Information: An Outlook. Science 2013; 339: 1169.
5. Benhelm J, Kirchmair G, Roos CF, Blatt R. Towards faulttolerant quantum computing with trapped ions. Nat Phys 2008; 4: 463.
6. Ryan CA, Laforest L, Lflammme R. Randomized benchmarking of single and multqubit control in liquidstate NMR quantum information processing. New J Phys 2009; 11: 013034.
7. Barends R, Kelly J, Megrant A, Veitia A, Sank D, Jeffrey E et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. Nature 2014; 508: 500.
8. Kelly J, Barends R, Campbell B, Chen Y, Chen Z, Chiaro B et al. Optimal Quantum Control Using Randomized Benchmarking. Phys Rev Lett 2014; 112: 240504.
9. Nigg D, Muller M, Martinez EA, Schindler P, Hennrich M, Monz T et al. Quantum computations on a topologically encoded qubit. Science 2014; 345: 302.
10. Kelly J, Barends R, Fowler AG, Megrant A, Jeffrey E, White TC et al. State preservation by repetitive error detection in a superconducting quantum circuit. Nature 2015; 519: 66.

This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/