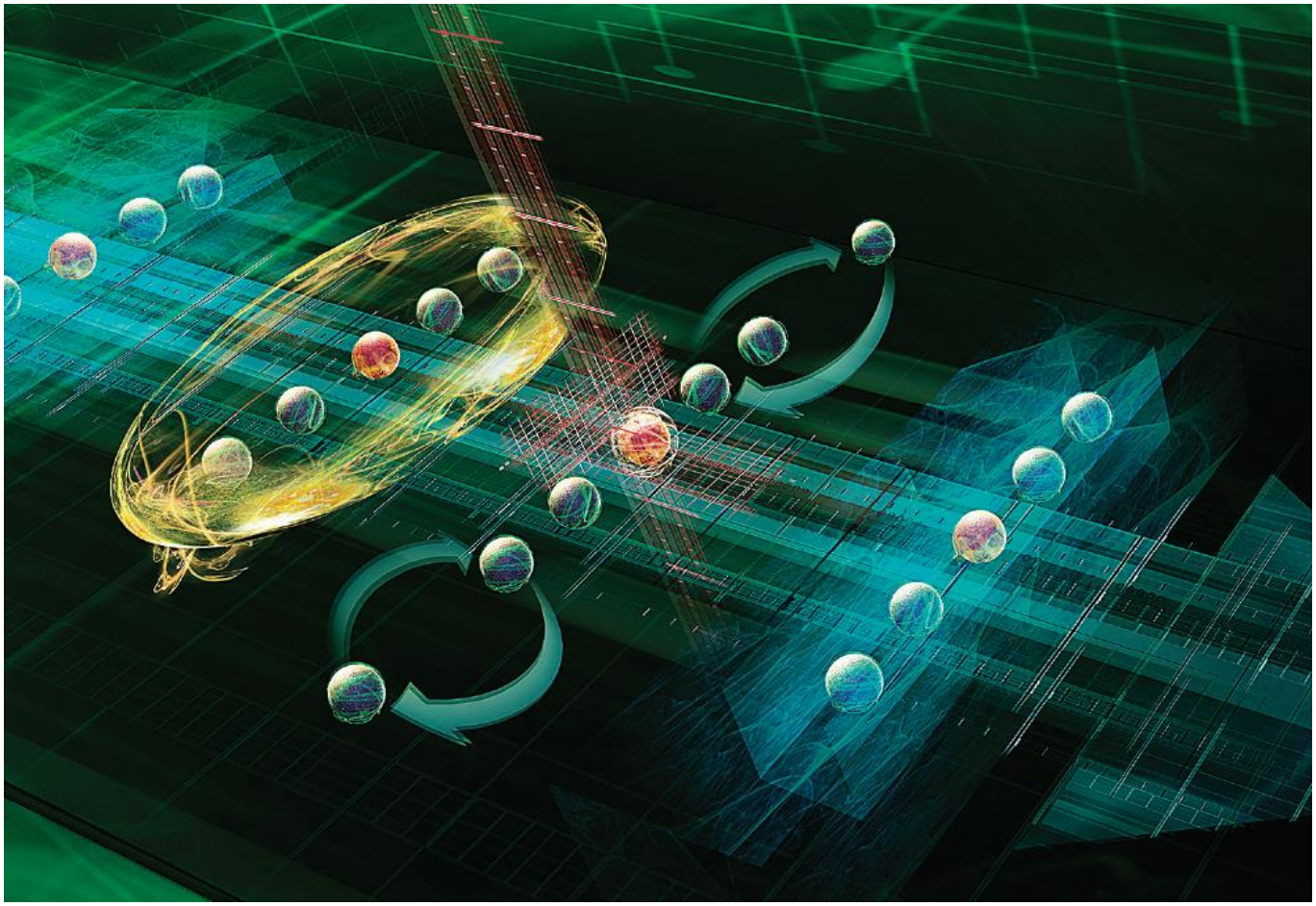


Source: [34]

Mistakes Were Made: Error Correction in Quantum Computing

Quantum computing has been heralded as a tool allowing us to solve problems unlike ever before. However, as the technology makes it way from research to realisation, the practical complications of making it work seems to only increase as these systems scale. Fortunately, quantum error correction is also a growing field. The question is: will this be enough to allows this vision to become a reality?

Word Count: 2964

Quantum Computing: The Basics

Where it all began...

Technological innovation is the lifeblood of scientific progression. Therefore, implementing the latest scientific developments into the way technical tools enhance our analytical ability is key to advancing our understanding of the universe. Quantum computing is a quintessential example of this.

Early in the 20th century, we had perhaps the greatest paradigm shift in science, as our fundamental perception of physics at the smallest scales went from a classical to quantum view. It was realized in the 80s that what, in a pre-quantum world, had been conceived as a 'universal' computer could not be truly universal if it remained using classical technologies as it could not simulate quantum effects.[1] Therefore, computing using quantum systems was postulated, and algorithms based on this idea were developed. It was quickly realized that quantum computing had the potential to have computational power superior to classical computers, allowing great reductions in running time for mathematical operations.[2] This was seen with fundamental algorithms being conceived in the 90s, such as Shor's algorithm, for integer factorization, and Grover's search algorithm, for search of a database whose organization is unknown (unstructured).[3]

Quantum Supremacy?

But why is quantum computing so special? Quantum computers can solve problems that no classic computer can due to unreasonable timescales or the inability to truly simulate quantum behaviour. What makes them different is naturally the use of fundamental quantum phenomena, such as superposition and entanglement. So, what are the implications of this? Unlike a classical system using bits, a quantum computer can use quantum bits (qubits), which use quantum superposition to represent an arbitrary linear combination of both parities, 0 and 1, as in:

$$\text{Qubit State} = \alpha\,|0\rangle + \beta\,|1\rangle\,.$$

These coefficients α and β are complex numbers, with polar form:

$$\alpha, \beta = r_{\alpha,\beta} e^{i\phi_{\alpha,\beta}}.$$

This accounts for the dynamics (i.e., time evolution) of the qubit, based on the phase $\phi_{\alpha,\beta}$. Now, in general, a quantum system is described by a wavefunction, $|\Psi\rangle$, whose square represents the probability amplitudes of different states the system could be in.

Then, fundamental to the quantum world, if a system is measured, it assumes a particular basis state. This is coined 'wavefunction collapse'. However, if we avoid measuring the system it behaves deterministically. Still, if the 2-level system of a qubit is measured, it will collapse into either of the 2 'classical' states, with random probability of ending up in $|0\rangle$ or $|1\rangle$ given by $|\alpha|^2$ and $|\beta|^2$ respectively.[4] As there are 2 possible basis states, we also write the state of qubit as a 2-dimensional complex vector, of length 1, i.e.,

$$|0\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) \;\&\; |1\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right).$$

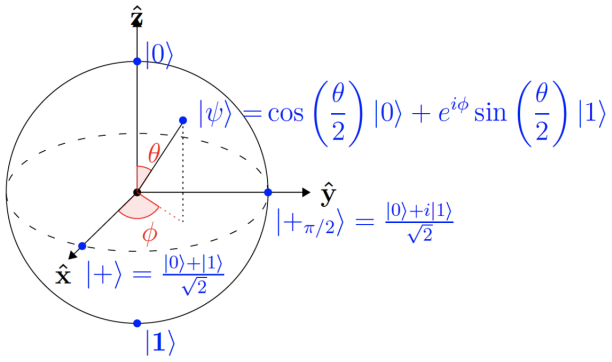
Here, each component in the vector represents the complex amplitude for the corresponding basis state in the superposition.[5]

Additionally, we visualize qubit states as points on a unit sphere called Bloch sphere, using spherical polar coordinates.[6] An arbitrary qubit state is defined

uniquely on the sphere as,

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle,$$

with superposition coefficients parametrised by only θ & ϕ.[6] It is convention to take the $|0\rangle$ term to have an 'absolute' phase of 0 such that we can then have the phase of $|1\rangle$ to be relative to this, defined simply as ϕ. This is because, to do anything useful when using multiple qubits to encode information, only the relative phase between the two basis states has any practical meaning rather than a global ('absolute') phase which is not measurable.[7]



The Bloch Sphere, with the parameters θ & ϕ and showing states corresponding to each of the x, y, z axes for a single qubit.[8]

To manipulate our qubits we apply logic gates as in classical computers, which mathematically are matrices acting on the qubits.[4] Gates transform our states around the Bloch sphere as rotations around and onto the $\hat{x}, \hat{y}, \hat{z}$ axes. For example, bit flip ($|0\rangle \rightarrow |1\rangle$) is caused by the Pauli X gate and phase flip with a Pauli Z gate ($|1\rangle \rightarrow -|1\rangle$), rotating around the x and z axes respectively.[9] Additionally, gates involving multiple qubits can entangle qubits.[9] Gates rely on quantum interference and interfere with the wavefunction, changing the qubit's probability density function. Each axes actually has a corresponding basis, which any non-basis state is a superposition of, as in the table below.[10]

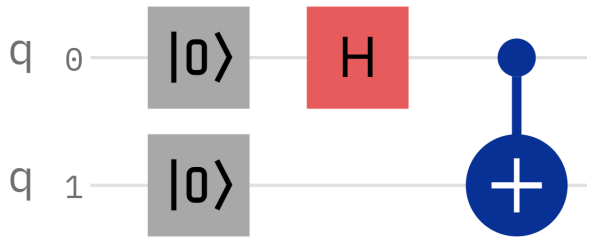
So, we've covered the basics of how a qubit functions but as measurement causes collapse to a single state, an individual qubit effectively only behaves like a classic bit. The real power of quantum computing is unlocked with multiple qubits. The Kronecker product ($\otimes$) can be used to obtain the combined state of multiple qubits.[11] For 2 qubits, a combined state of $|a\rangle$ & $|b\rangle$ is $|c\rangle = |ab\rangle$, with complex superposition coefficients c_{ij}, a_i, b_j:

Axis	Basis States	
$\hat{z}$	$\|0\rangle$	$\|1\rangle$
$\hat{x}$	$\|+\rangle = \frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$	$\|-\rangle = \frac{\|0\rangle-\|1\rangle}{\sqrt{2}}$
$\hat{y}$	$\|R\rangle = \frac{\|0\rangle+i\|1\rangle}{\sqrt{2}}$	$\|L\rangle = \frac{\|0\rangle-i\|1\rangle}{\sqrt{2}}$

$$|c\rangle = |a\rangle \otimes |b\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{bmatrix} a_0 \times \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \\ a_1 \times \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix} = \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix}$$

.[11] As each qubit can be superposition of 0 and 1, for n qubits, the combined state can be a superposition of 2^n states. Because of this, a quantum computer could process all possible combinations of 0's and 1's simultaneously, so are much faster than a classical one, which would run each combination one-by-one.[12]

We've seen that quantum mechanics is intrinsically probabilistic, but probabilities can often be dependent, so when qubits interact their wavefunctions can also become dependent on each other creating 'entangled' states. To achieve qubit entanglement we act on multiple qubits with controlled gates which have certain qubits acting as 'control', whose states determines whether action is done on 'target' qubits [Kronecker]. For example, the controlled NOT gate (CNOT) acts on 2 qubits and completes a bit flip of the target qubit, if the control qubit (first qubit) is $|1\rangle$. So, for 2 qubits, the CNOT gate maps: $|00\rangle \rightarrow |00\rangle, |01\rangle \rightarrow |01\rangle, |10\rangle \rightarrow |11\rangle, |11\rangle \rightarrow |10\rangle$. Now, consider one of the qubit's being in the $|+\rangle$ state, obtained by operating a logic gate called a Hadamard gate on $|0\rangle$.



A quantum circuit diagram for a CNOT gate acting on $|+0\rangle$. The $|+\rangle$ state is achieved by applying the red Hadamard gate. The control qubit is indicated by ● with the + on the target qubit. (Diagram by author)

As a matrix equation, acting a CNOT gate on the combined state $|+0\rangle$ is:

$$\text{CNOT}\,|+0\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

(To get the above cf. with general 2-qubit state vector from before). Through this manipulation, we create an entangled state called a Bell state.[1]3 These entangled states are used to speed up processing given that the 2 qubits, say q_A and q_B are correlated. If a measurement of q_A was made then with equal probability it collapses to either $|0\rangle$ or $|1\rangle$, causing the combined system to collapse to $|00\rangle$ or $|11\rangle$ correspondingly. Therefore, q_B will be in the same state as q_A. So, if we know the state of a single qubit in a Bell entangled pair, then we actually know the state of both qubits. This allows for what's known as superdense coding, which uses the fact that information about one qubit tells us about the state of a corresponding entangled qubit.[14] This speeds up transmission and calculation speeds beyond what a classical computer can do for equivalent processes more bits would be required.[14]

Real quantum computers come with real issues

Achieving quantum computers

So far, we have covered the underlying theory of the quantum world that would allow quantum computers to revolutionise the world of computing but this means nothing if these supposed 'wonder' machines cannot be practically brought and applied. Therefore, there has been a complementary development of physical analogs to qubits, effectively, 2-level systems in isolation. Various methods, such as trapped ion, superconducting circuit, and photonic quantum computing, have seen investment from major players like IBM, Google, D-Wave, Honeywell, IonQ and Rigetti.[15] However, whilst there is much variation in how to physically achieve quantum computing, one unifying issue is that all physical qubits are not perfect but are subject to fluctuation.

The breakdown between theory and reality

Naturally, the end goal of obtaining working quantum computers is to obtain real-world benefits through practical application. However, all theory behind quantum algorithms assumes that qubits indefinitely hold their states. In reality, their quantum nature make quantum computers face a significant challenge in terms of the prevalence of errors. Fundamental to the quantum world, wavefunctions are changed and collapsed by interaction, making them extremely delicate. Therefore, maintaining quantum coherence, which is the ability of a system's wavefunctions to remain 'uncollapsed' is limited. This is characterised by the system having a definite phase relationship

between the wavefunctions of different states.[16] For example, Google's qubits have a coherence time of only about 15 microseconds before loss of information due to noise.[17] This means quantum computers have to walk a continual tightrope. On the one hand, we want to allow qubits to have strong interactions with one another, be controllable, and be detected by us to obtain outputs.[18] However, to avoid errors we cannot have them interact with anything else. In the real world, we lose information due to what is effectively 'quantum noise' from environmental interference such as mechanical, temperature, electromagnetic and atomic disturbances which can collapse or change a qubit's state.[19]

Due to these short timescales and likelihood for decoherence to occur in quantum computers, a key element to making them a practical reality is mitigating and adapting to the resulting errors. This forms the field of quantum error correction (QEC).

Quantum Error Correction

1110001111...what's your emergency?

Before looking at how we tackle errors, we need to understand the limitations that QEC is restricted by.

We know that qubit errors occur when they are interfered with. Therefore, qubits must have the ability to remain connected but still isolated from the outside world. So, to maintain long enough coherence for useful processing to occur we must aptly 'shielding' them from each other and the environment. This requires developments in engineering and we have seen different compromises such as much longer coherence times but slower operation times in ion trap versus superconducting qubit quantum computers.[20]

Of course, we still need to be able to manipulate qubits avoiding or correcting for induced errors. To do this we must employ QEC for faults at all parts of the process: state preparation and calibration, manipulation with quantum gates and measurement.[18]

In classical computing, to avoid errors redundant copies of bits were made and their parities measured, and a majority can be taken as correct, after which we rectify errors.[21] This is fundamentally impossible in a quantum world though. Firstly, we can't measure the state of a qubit midway through computation as this would collapse its state, removing the benefit of superposition. Additionally, it is impossible to create two copies of a qubit in an unknown state, as described by the no-cloning theorem, given that this would disturb and effectively measure the state of the initial qubit.[23] Also, we have an obvious desire to increase the number of qubits available for use but this increases the likelihood of interference between them.

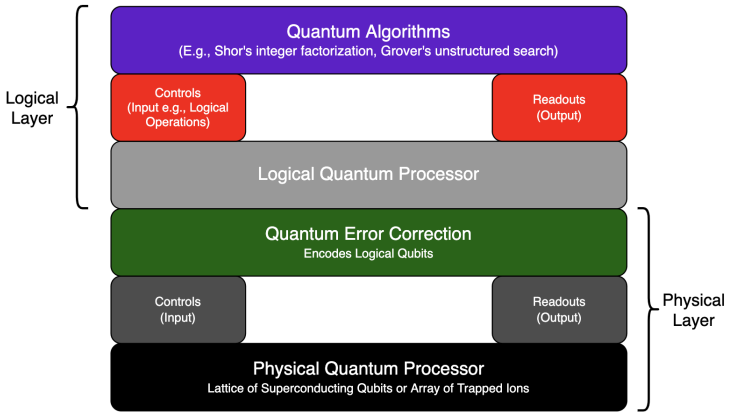
Fortunately, all errors are in fact manipulations (superposition or products) of the elementary errors of bit (X), $|0\rangle \rightarrow |1\rangle$, or phase (Z), $|0\rangle + |1\rangle \rightarrow |0\rangle - |1\rangle$, flips.[22] Therefore, any QEC method just needs to be able to identify and correct only these 2 errors. From this, foundational to any QEC method, arises the creation of logical qubits.

Let's be logical about this

A logical qubit is the name for a qubit which can be implemented in theoretical quantum algorithms but are composed of multiple, entangled physical qubits. As we know, physical qubits are imperfect but entangling them in a particular architecture allows QEC to use redundancy and 'non-local measurements' to encode information such that it is fault-tolerant (i.e., even if their is a fault within execution of error correction we can account for this).[24]

This means, if we wanted to represent a certain superposition of $|0\rangle$ and $|1\rangle$, then we entangle chains of 'data' qubits into a combined state of the same superposition but of states $|00...0\rangle$ and$|11...1\rangle$ (with length

based on the number of entangled qubits).



Systems overview of a fault-tolerant quantum computer with quantum error correction based on the creation of logical qubits from arrangements of physical qubits. Adapted from [18]

We know that when entangled, involved qubits are not in a well-defined state, but, are perfectly correlated. So, if one collapses then so do all others, to the same state as well. However, they can still individually face bit and phase flip errors but remain entangled.

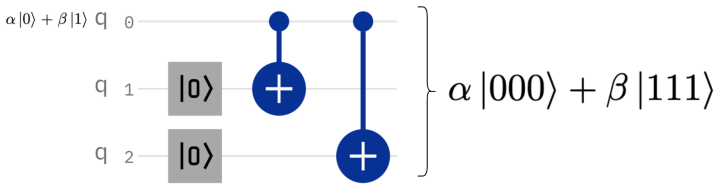
This larger entangled state is used to watch for errors based on discrepancies from the expected state, as in, if any of the entangled qubits have faced X or Z errors. However, this must be done indirectly to not cause collapse. For this, we additionally couple 'ancilla' qubits and take 'stabilizer' measurements which give information about the consistency of data qubits, meaning we can identify and correct errors in a non-disturbative way.[22] Different architectures of physical qubits forming logical qubits exist (which vary the number of physical qubits and the way they are entangled) based on the required error reduction as well as the cause of errors faced.

Error diagnosis using ancilla qubits - the basics

To begin understanding how making logical qubits is possible, it is best to start considering just 3 entangled qubits with at most a single qubit facing a bit flip (X) error. Whilst this ignores having multiple bit or phase errors which is unrealistic, this method is the basis for all error identification in the realistic cases.[22]

Encoding into physical qubits

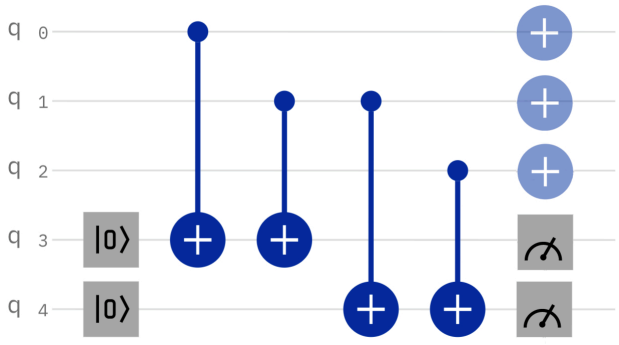
Say we want to encode the initial arbitrary state of a qubit q_0, $\alpha|0\rangle + \beta|1\rangle$, into multiple, physical qubits To do this we will use this initial qubit as a control qubit with CNOT gates acting on 2 other qubits, q_1&q_2 initialised as $|0\rangle$. This creates an encoded qubit, $q_L = \alpha|000\rangle + \beta|111\rangle$ which can be operated on by logic gates through: $X_L = X_0X_1X_2$ and $Z_L = Z_0Z_1Z_2$.



Quantum circuit to encode a qubit in an arbitrary state into 3 qubits of combined state $\alpha|000\rangle + \beta|111\rangle$. (Diagram made by author)

Identifying and correcting an error

This requires 2 ancillary qubits, q_3&q_4 which we couple to the 3 qubits and then measure. The key here is the subset of these 3 qubits that each ancillary is coupled to, and their initialised states. After encoding, this is done as in the circuit below. Coupling occurs through CNOT gates, and the ancillary qubits are measured. The response of the ancillary qubits identifies any errors.



Firstly, if no error occurs then the combined state of the data qubits is a superposition of only $|000\rangle$ & $|111\rangle$. The $|000\rangle$ term does not flip the ancillary qubits whereas the $|111\rangle$ term causes both to be flipped twice. So, for both terms the ancillary qubits remain in the state $|00\rangle$. If any single qubit flips however, then the ancillary qubits will be in a different, unique state, based on which qubit faced the error. Using this, we apply a NOT gate to the incorrect qubit, as described by the table below:

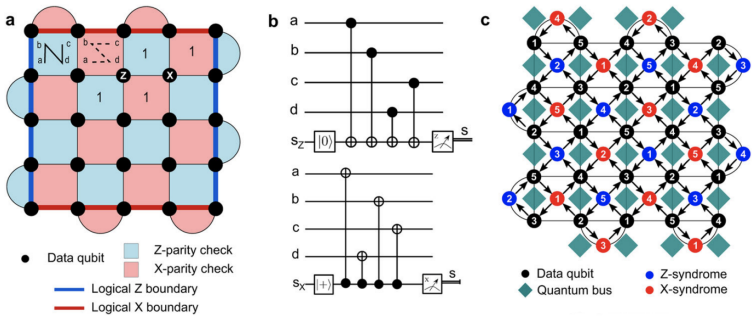
Combined Qubit State	q_3	q_4	Where to apply NOT gate?
$\alpha\|000\rangle + \beta\|111\rangle$	0	0	No errors occurred
$\alpha\|100\rangle + \beta\|011\rangle$	1	0	q_0
$\alpha\|010\rangle + \beta\|101\rangle$	1	1	q_1
$\alpha\|001\rangle + \beta\|110\rangle$	0	1	q_2

With this method we correct any single qubit bit-flip error. A similar process occurs to correct phase-flip (Z) errors. These processes can be concatenated to tackle compound errors, such as the Shor and Steane codes, developed in 1995 and 1996 respectively, founding the field of QEC.[25,26]

Scratching the surface...

The simplified methods seen so far can be generalised to create logical qubits through what is known as the surface code, a preferred method for high degrees of error correction in a scalable and fault-tolerant way. Obviously, fault-tolerance is required to make the logical error rate smaller than than the error rates of the constituent physical qubits.

The surface code is a 2D lattice arrangement of data qubits, with ancilla qubits between them for Z and X parity checks, together forming logical qubits. Being in 2D allows treatment of Z and X errors in separate dimensions. The square lattices consist of d^2 data qubits and hence, d is the number of qubits on one side. An outline of what's known as the rotated surface code, which uses fewer qubits than the surface code of the same distance, is shown below:

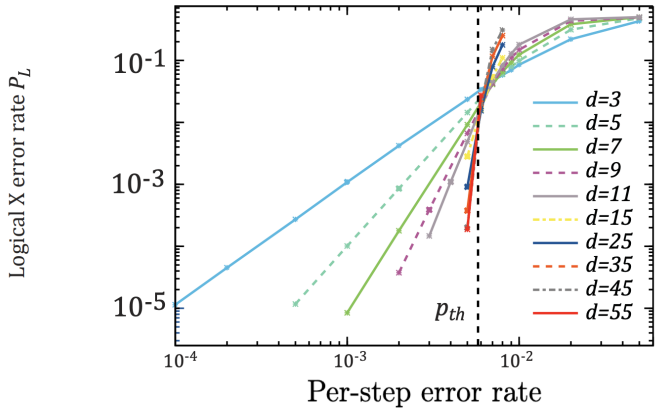


18

In **a**, there are data qubits at each lattice point, with ancillary qubits in the coloured squares shown in **c**, denoted here as syndrome. Ancilla are coupled with neighbouring data qubits through CNOT gates and then measured in the Z-basis to correct for bit-flip errors (shown in blue) and in the X-basis for phase-flip errors (shown in red).[18] Errors can be identified through the circuits in **b** based on whether the ancillary qubit remains in its initialised state. Additionally, theoretically, logical qubits can be formed from any lattice size if $d > 3$.[27] However, the best achieved was a physical-logical qubit ratio of 13:1 by IonQ.[15] To do this in surface code, fragments of a larger 2D array can be created by having some unentangled data qubits, known as defects, which are not involved in QEC or coupled to ancillary qubits. Within a logical qubit, due to the arrangement of data and ancilla qubits in surface code, and the mathematics of operators acting on them, X and Z operations are equivalent to individually operating said gate on every data qubit in any row, for X, or any column, for Z. It actually doesn't matter which row/column you choose.[28]

The surface code is a preferred method for QEC due to its 2D layout, making it practical as well as easily scalable.[29] It achieves fault-tolerant error correction of logical qubits based on repeating continuous cycles of stabilizer measurements and then correction.[27] Additionally, as long as the individual

error for a given logic gate is below a threshold value, the rate at which a logical qubit faces errors decreases exponentially with d.[27] However, above this threshold error rate,error correction actually worsens as d increases.[27] This threshold is called the noise threshold or the 'critical error rate per gate'. The surface code favourably has high thresholds of 0.5-1%.[27] So, if we achieve hardware which faces errors at a likelihood of this order, the surface code can then exponentially decrease these errors as it scales in size.



Experimental data has shown that the critical per-step error rate for the operating a Logical X is 0.57%. We see that once the per-step error rate is less than this the logical error rate decreases as we increase d.[30]

Into the quantum future

So, the strangeness of the quantum world both impedes and enhances our development. Extending computing beyond classical will lead to great leaps in our computational ability, especially as quantum computers are scaled to have more qubits. However, the instability of quantum systems, risks more errors as we scale. To remedy this, through QEC, we must embrace these quantum properties to carefully entangle qubits and make indirect inferences about them. To make this possible, again a greater quantity and stability of qubits is necessary. Therefore, the main players in the space are leading developments to achieve many more qubits.[15] IBM has plans to create a processor called the Condor by 2023, which will have over 1000 qubits.[31] The end goal of most companies, like Psi-Quantum, IBM and Google, is to reach systems with over a million qubits which will require them to ensure successful QEC implementation.[31,32,33] So, QEC must keep pace with these developments and therefore, may well be the key to allowing us to reap the benefits of quantum computing, in our day-to-day lives.

References

1. Deutsch, David. "1985 Quantum theory, the Church–Turing principle and the universal quantum computer". Proc. R. Soc. Lond. A 400 97–117

2. Highfield, Roger. "Quantum Computing: What, Who, How and When?" Science Museum Blog, 18 June 2018, https://blog.sciencemuseum.org.uk/quantum-computing-what-who-how-and-when/.

3. "TQD Exclusive: The History of Quantum Computing." The Quantum Insider, 26 May 2020, https://thequantuminsider.com/2020/05/26/tqd-exclusive-the-history-of-quantum-computing/.

4. Combarro, Elias F. "A Practical Introduction to Quantum Computing: From Qubits to Quantum Machine Learning and Beyond." CERN, https://indico.cern.ch/event/970905/attachments/2146852/3618837/PIQC%20Lecture%203.pdf. Accessed 10 Jan. 2022.

5. "The Quantum State as a Vector." Utah State University - Physics. http://www.physics.usu.edu/Wheeler/QuantumMechanics/Quantum%20states%20as%20vectors.pdf. Accessed 10 Jan. 2022.

6. Glendinning, Ian. "The Bloch Sphere." http://www.vcpc.univie.ac.at/ ian/hotlist/qc/talks/bloch-sphere.pdf. Accessed 10 Jan. 2022.

7. Dawar, Anuj. "Quantum Computing." https://www.cl.cam.ac.uk/teaching/0910/QuantComp/notes.pdf. Accessed 10 Jan. 2022.

8. Cacciapuoti, Angela Sara Van Meter, Rodney & Hanzo, L. & Van, Rodney. (2019). When Entanglement meets Classical Communications: Quantum Teleportation for the Quantum Internet. IEEE Transactions on Communications.

9. "Our Trapped Ion Technology." IonQ,

https://ionq.com/technology.

10. Voorhoede, De. "Qubit Basis States." Quantum Inspire, https://www.quantum-inspire.com/kbase/qubit-basis-states/.

11. Team, The Qiskit. "Multiple Qubits and Entangled States." Qiskit, Data 100 at UC Berkeley, 17 Nov. 2021, https://qiskit.org/textbook/ch-gates/multiple-qubits-entangled-states.html.

12. Rasmusson, AJ. "The Power of Quantum Computing: Parallelism." ScIU, 8 July 2019, https://blogs.iu.edu/sciu/2019/07/13/quantum-computing-parallelism/.

13. "351: Bell State Exercises." Chemistry LibreTexts, Libretexts - UC Davis, 6 Jan. 2022, https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Quantum_Tutorials_(Rioux)/Quantum_Teleportation/351%3A_Bell_State_Exercises.

14. Team, The Qiskit. "Superdense Coding." Qiskit, Data 100 at UC Berkeley, 17 Nov. 2021, https://qiskit.org/textbook/ch-algorithms/superdense-coding.html.

15. Ball, Philip. "Setting the Scene for a Quantum Marketplace." PhysicsWorld, Dec. 2021, https://philipball.co.uk/images/stories/docs/pdf/PWDec21Ball-feature.pdf. Accessed 10 Jan. 2022.

16. Mooij, Hans. "Superconducting Quantum Bits." Physics World, Institute of Physics, 28 Aug. 2018, https://physicsworld.com/a/superconducting-quantum-bits/.

17. Cho, Adrian. "Physicists Move Closer to Defeating Errors ... - Science.org." Science, 14 July 2021, https://www.science.org/content/article/physicists-move-closer-defeating-errors-quantum-computation.

18. Gambetta, J.M., Chow, J.M. Steffen, M. Building logical qubits in a superconducting quantum computing system. npj Quantum Inf 3, 2 (2017). https://doi.org/10.1038/s41534-016-0004-0

19. Giles, Martin. "Explainer: What Is a Quantum Computer?" MIT Technology Review, MIT Technology Review, 20 Oct. 2021, https://www.technologyreview.com/2019/01/29/66141/what-is-quantum-computing/.

20. "TQD Exclusive: A Detailed Review of Qubit Implementations." The Quantum Insider, 21 May 2020, https://thequantuminsider.com/2020/05/21/tqd-exclusive-a-detailed-review-of-qubit-implementations-for-quantum-computing/.

21. Cho, Adrian. "The Biggest Flipping Challenge in Quantum Computing." Science, 9 July 2020, https://www.science.org/content/article/biggest-flipping-challenge-quantum-computing.

22. Mermin, N. David. Quantum Computer Science : An Introduction, Cambridge University Press, 2007. ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/imperial/detail.action?docID=326023.

23. "The No-Cloning Theorem." Quantiki, 26 Oct. 2015, https://www.quantiki.org/wiki/no-cloning-theorem.

24. Preskill, John. "Fault-tolerant quantum computation". arXiv [quant-ph] 1997. Web.

25. Shor, Peter W. "Scheme for reducing decoherence in quantum computer memory". Phys. Rev. A 52 (1995): R2493–R2496. Web.

26. Steane Andrew. 1996 Multiple-particle interference and quantum error correction. Proc. R. Soc. Lond. A. 452 : 2551–2577

27. "Fault Tolerance." QuTech Academy, YouTube, 9 Dec. 2018, https://www.youtube.com/watch?v=wTaFd7nqbtQamp;list=PL5jmbd6SJYnPiYlM6pHAm2M3FL40D9otZamp;index=39. Accessed 10 Jan. 2022.

28. de Beaudrap, Niel. "How Is Computation Done in a 2D Surface Code Array?" Quantum Computing Stack Exchange, 18 Oct. 2018, https://quantumcomputing.stackexchange.com/questions/4395/how-is-computation-done-in-a-2d-surface-code-array.

29. Fowler, Austin G., Ashley M. Stephens, en Peter Groszkowski. "High-threshold universal quantum computation on the surface code". Phys. Rev. A 80 (2009): 052312. Web.

30. Fowler, Austin G. et al. "Surface codes: Towards practical large-scale quantum computation". Phys. Rev. A 86 (2012): 032324. Web.

31. Chow, Jerry, et al. "IBM's Roadmap for Scaling Quantum Technology." IBM Research Blog, 9 Feb. 2021, https://research.ibm.com/blog/ibm-quantum-roadmap.

32. "Building the World's First Useful Quantum Computer." PsiQuantum, https://psiquantum.com/.

33. "Our Quantum Computing Journey." Google Quantum AI, https://quantumai.google/learn/map.

34. Monroe, Don. "Closing in on Quantum Error Correction." ACM, 1 Oct. 2019, https://cacm.acm.org/magazines/2019/10/239668-closing-in-on-quantum-error-correction/fulltext.