BATTLING DECOHERENCE: THE FAULT-TOLERANT QUANTUM COMPUTER

Quantum computers have the potential to do certain calculations faster than any foreseeable classical computers, but their success will depend on preserving complex coherent quantum states. Recent discoveries have shown us how to do that.

John Preskill

Information carried by a quantum system has notoriously weird properties. Physicists and engineers are now learning how to put that weirdness to work. Quantum computers, which manipulate quantum states rather than classical bits, may someday be able to perform tasks that would be inconceivable with conventional digital technology. (See the article by Charles H. Bennett, PHYSICS TODAY, October 1995, page 24, and the "Search and Discovery" report in PHYSICS TODAY, March 1996, page 21.)

Formidable obstacles must be overcome before large-scale quantum computers can become a reality (see the article by Serge Haroche and Jean-Michel Raimond, PHYSICS TODAY, August 1996, page 51). A particularly daunting difficulty is that quantum computers are highly susceptible to making errors. The magical power of the quantum computer comes from its ability to process coherent quantum states; but such states are very easily damaged by uncontrolled interactions with the environment—a process called decoherence. In response to the challenge posed by decoherence, the new discipline of quantum error correction has arisen at the interface of physics and computer science. We have learned that quantum states can be cleverly encoded so that the debilitating effects of decoherence, if not too severe, can be resisted.

The power of the quantum computer

The indivisible unit of classical information is the bit, which takes one of the two possible values, 0 or 1. Any amount of classical information can be expressed as a sequence of bits. A classical computer executes a series of simple operations (often called "gates"), each of which acts on a single bit or pair of bits. By executing many gates in succession, the computer can evaluate any Boolean function of a set of input bits.

Quantum information, too, can be reduced to elementary units, called quantum bits or qubits. A qubit is a two-level quantum system (like the spin of an electron). A quantum computer executes a series of elementary quantum gates, each of which is a unitary transformation that acts on a single qubit or pair of qubits. By executing many such gates in succession, the quantum computer can apply a complicated unitary transformation to a particular initial state of a set of qubits. Finally, the qubits can be measured; the measurement outcome is the final result of a quantum computation.

JOHN PRESKILL *is a professor of theoretical physics at Caltech.*

A classical computer can faithfully simulate a quantum computer, so that anything the quantum computer could do, the classical computer could also do. Still, there is a sense in which the quantum computer appears to be a more powerful device: Its simulation by the classical computer is very inefficient. The quantum state of even a modest number of qubits (let's say 100) lives in a Hilbert space of unimaginably large dimension: $2^{100} \sim 10^{30}$. To simulate a typical quantum computation, a classical computer would need to work with matrices of exponentially large size, which would take a very long time. In more physical terms, running a classical simulation of a quantum computer is hard because (as exemplified by John Bell's famous inequalities) correlations among quantum bits are qualitatively different from correlations among classical bits. The exponential explosion in the size of Hilbert space as we increase the number of qubits arises because the correlations among qubits are too weird to be expressed easily in classical language.

That simulating a quantum computer with a classical computer takes an unmanageably long time suggested to Richard Feynman[1] that using a quantum computer might enormously speed up finding solutions to certain hard computational problems. David Deutsch,[2] developing the idea further, observed that a quantum computer can invoke a kind of massive parallelism, by operating on a coherent superposition of a vast number of classical states. In fact, a single computation acting on just 300 qubits can achieve the same effect as 2^{300} simultaneous computations acting on classical bits, more than the number of atoms in the visible universe. We could never build a conventional computer with that many processors!

Peter Shor[3] discovered how, in principle, to apply quantum parallelism to the problem of finding the prime factors of a large integer. The difficulty of factoring an integer escalates very rapidly as the number of its digits increases. For example, suppose that we want to find the 65-digit prime factors of a 130-digit composite number. A network of hundreds of powerful workstations, collaborating and communicating over the Internet and running the best algorithms known, might solve the problem in a few months. To factor a 400-digit number, the same network of workstations running the same algorithms would need about 10 billion years (the age of the universe). Even with vast improvements in technology, no one will be factoring 400-digit numbers using conventional computers anytime soon, unless there is an unexpected algo-

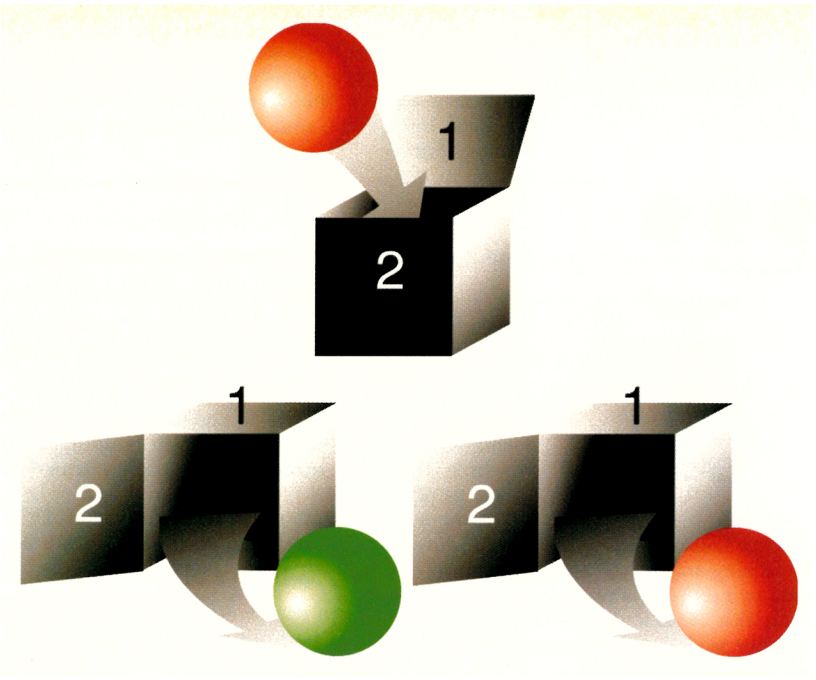


FIGURE 1. DOOR NUMBER 1 or door number 2? To read quantum information reliably, we need to know how it was stored. We can represent an unknown quantum bit (qubit) as a colored ball placed in a box through one of two doors. The doors represent two ways of measuring the qubit (such as the axis along which to measure spin), and the two colors represent the possible outcomes of the measurement. If the ball is placed in the box through door 1, and then it is observed through door 2, the color of the ball that comes out of the box is random.

rithmic breakthrough.

But now suppose we have a quantum computer that runs just as fast as that network of workstations—that is, it can perform the same number per second of elementary operations on pairs of qubits as the classical computer can perform elementary logic gates on pairs of bits. That quantum computer could factor the 130-digit number in a few seconds, and the 400-digit number in just minutes. Thanks to quantum parallelism, the difficulty scales in a much more reasonable way with the size of the input to the problem. For very large numbers, the advantage enjoyed by the quantum computer is truly stupendous.

The challenge of error correction

If quantum computers would be so marvelous, why don't we just build one? There are technological challenges, to be sure. But are there any obstacles that might be fundamental matters of principle, that would prevent us from ever constructing a quantum computer?

In fact, there is a problem of principle that is potentially very serious: decoherence. Unavoidable interactions with the environment will cause the quantum information stored in a quantum computer to decay, thus inducing errors in the computation. Decoherence occurs very rapidly in complex quantum systems, which is why we never observe macroscopic superpositions (such as a coherent superposition of a live cat and a dead cat). If quantum computers are ever to be capable of solving hard problems, a means must be found to control decoherence and other potential sources of error.

Errors can be a problem even for classical information. We all have bits that we cherish, while everywhere there are dragons lurking who delight in tampering with our bits. But we have learned some ways to protect classical information from the dragons. If I have a bit with the value 0 that I want to preserve, then I can store two backup copies of the bit. Eventually, a dragon could come along and flip one of my three bits from 0 to 1. But I can employ a busy beaver to check the three bits frequently; when he finds that one has a different value than the others, he flips that bit so that all three match again. That way, as long as the dragon has not had a chance to flip two bits, the error can be corrected and the information will be protected.

We would like to apply the same principle of redundant storage to quantum information, but, because qubits are different from classical bits, there are complications. We might visualize a qubit as a colored ball, either red or green, concealed in a locked box, that can be opened through either of two doors. The doors represent two ways of measuring the qubit, just as we could measure the spin of an electron along either the z or the x axis; the two possible colors represent the possible outcomes of the measurement. If we store a ball in the box through door 1 or door 2 and we later open the same door, we can recover our bit and read it, just as we would read classical information. But if we store the ball through door 1 and then open door 2, what comes out will be completely random (has equal probability of being red or green); the outcome tells us nothing about what we put inside the box (see figure 1). To read quantum information reliably we need to know how it was stored; otherwise we are bound to damage it irrevocably.

The first problem we encounter in the battle against decoherence is that an unknown quantum state cannot be perfectly duplicated;[4] hence we cannot safeguard a quantum computer against errors by storing backup copies of its state. Roughly speaking, the trouble is that to duplicate the information in a quantum box, a copier must open a door to see what is inside. If it just happens to open the same door that was used to store the information, it can make an accurate copy. But if it guesses wrong, it will irrevocably damage the information instead. We can clone a sheep, but not a qubit!

A second problem is that there are more things that can go wrong with quantum information than with classical information. The dragon might open door 1, change the color of the ball, and reclose the box—that would be a *bit-flip* error analogous to the errors that can afflict classical information. Or he might open door 2, change the color, and reclose the box—that would be a *phase* error, for which classical information has no analog. The beaver

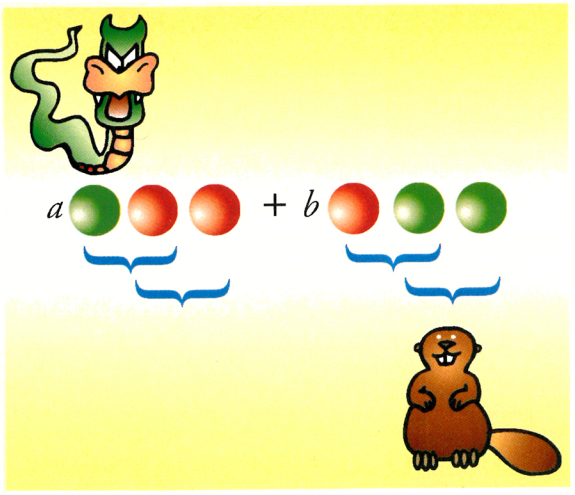


FIGURE 2. ERROR CORRECTION by collective measurement preserves a coherent quantum state. The lurking dragon has flipped one of the three qubits. Measuring two qubits at a time (blue brackets), the busy beaver determines that the first and second qubits are different colors and that the second and third qubits are the same color. He then infers that the first bit has flipped, and repairs the damage.

needs to be able to fix the error without knowing ahead of time whether the dragon is going to use door 1 or door 2.

Third, whereas errors in classical information are discrete, errors in quantum information form a continuum. Rather than simply flipping a bit, the dragon might introduce a more subtle kind of error by performing the bit flip with some (small) probability amplitude ε. The beaver must be able to recover from that kind of small error; otherwise small errors will accumulate over time, eventually building up to become large errors.

Finally, to diagnose whether errors have occurred, the beaver must look at some qubits—and therefore must open some boxes. But quantum measurement necessarily disturbs the state that is being measured, so we worry that the beaver cannot check for errors without introducing further errors.

Quantum error-correcting codes

As recently as four years ago, the difficulties described above seemed highly discouraging. But in 1995, Shor and Andrew Steane discovered[5,6] that the obstacles were illusory—that quantum error correction really is possible. Theirs is one of the most important discoveries about quantum information in recent years, and it can be expected to have far-reaching implications.

To appreciate the insights of Shor and Steane, let's first consider how to defend quantum information against a dragon who performs only bit flips (we'll return to the issue of phase errors shortly). We are to protect the state

$$a|0\rangle + b|1\rangle, \tag{1}$$

a coherent superposition of the red ($|0\rangle$) and green ($|1\rangle$) states of a single qubit, where the complex coefficients a and b are unknown. Were the dragon to attack, the bit flip would transform the state to

$$a|1\rangle + b|0\rangle, \tag{2}$$

and damage would be inflicted unless $a = \pm b$. The beaver's assignment is to diagnose and reverse bit flips, but without disturbing the delicate superposition state, that is without modifying a and b.

Well schooled in classical error correction, the beaver applies the principle of redundant storage by encoding the qubit in a state of three qubits. The red state is encoded as three red qubits, and the green state as three green qubits; that is,

$$\begin{aligned} |0\rangle &\rightarrow |\bar{0}\rangle \equiv |000\rangle, \\ |1\rangle &\rightarrow |\bar{1}\rangle \equiv |111\rangle. \end{aligned} \tag{3}$$

Thus the unknown superposition state becomes

$$a|0\rangle + b|1\rangle \rightarrow a|\bar{0}\rangle + b|\bar{1}\rangle = a|000\rangle + b|111\rangle. \tag{4}$$

This redundant state is *not* the same as three identical copies of the original unknown state, which would be

$$(a|0\rangle + b|1\rangle)\,(a|0\rangle + b|1\rangle)\,(a|0\rangle + b|1\rangle). \tag{5}$$

Although it is impossible to copy unknown quantum information, nothing prevents us from building a (unitary) machine that will execute the encoding transformation given as equation 4.

Now suppose that the dragon flips one of the three qubits, let's say the first one, so that the state becomes

$$a|100\rangle + b|011\rangle, \tag{6}$$

and the beaver is to detect and reverse the damage. His first impulse would be to open the boxes and look to see if one ball was a different color from the others, just as he would to diagnose errors in classical information, but he must resist that temptation. If he were to open door 1 of all three boxes, he would find either $|100\rangle$ (with probability $|a|^2$), or $|011\rangle$ (with probability $|b|^2$); either way, the coherent quantum information (the values of a and b) would be irrevocably lost.

But he is a clever beaver who knows he need not restrict his attention to single-qubit measurements. Instead, he performs collective measurements on two qubits at once (see figure 2). The beaver asks whether the first two qubits have the same color or different colors, without trying to ascertain the color of either one. He finds that the colors are different. Then he asks whether the second and third qubits have the same color or different colors. He finds that the colors are the same. From the two measurement outcomes, the beaver infers that the first qubit has flipped relative to the other two and should be flipped back to repair the damage. In executing this protocol, the beaver has not learned anything about the encoded state (the values of a and b), hence the recovery procedure itself has inflicted no damage.

The beaver won that round, but now the dragon tries a more subtle approach. Rather than flipping the first qubit, he rotates it only slightly, so that the three-qubit state becomes

$$\begin{aligned} &a|000\rangle + b|111\rangle \rightarrow \\ &a|000\rangle + b|111\rangle + \varepsilon\,(a|100\rangle + b|011\rangle) + O(\varepsilon^2), \end{aligned} \tag{7}$$

where $|\varepsilon| \ll 1$. What should the beaver do now? In fact, he can do the same thing as before. If he performs a collective measurement on the first two qubits, then most of the time (with probability $1 - |\varepsilon|^2$), the measurement will project the damaged state (equation 7) back to the completely undamaged state (equation 4). Only much more rarely (with probability $|\varepsilon|^2$) will the measurement project onto the state given as equation 6 with a bit-flip error. But then the measurement outcome tells the beaver what action to take to repair the damage, just as in the previous case.

Of the four difficulties for quantum error correction cited above, then, we have already seen how three can be overcome. We can encode a quantum state redundantly without violating the no-cloning principle. We can perform

collective measurements that let us acquire information about the nature of the errors without revealing anything about the state, and so without damaging the state. We can control the accumulation of small errors by repeatedly making measurements that either reverse the damage or introduce large errors that we know how to correct. It remains only to resolve one more issue: the problem of phase errors.

Fixing phases

The code we have devised so far provides no protection against a dragon who flips the relative phase of $|0\rangle$ and $|1\rangle$. If such a dragon attacks any one of our three qubits, then our encoded state $a|\bar{0}\rangle + b|\bar{1}\rangle$ is transformed to $a|\bar{0}\rangle - b|\bar{1}\rangle$, and the encoded quantum information is damaged if a and b are both nonzero. But the method we developed to conquer the bit-flip errors can be extended to deal with phase errors as well—just as we protected against bit-flip errors by encoding bits redundantly, we can protect against phase-flip errors by encoding phases redundantly.

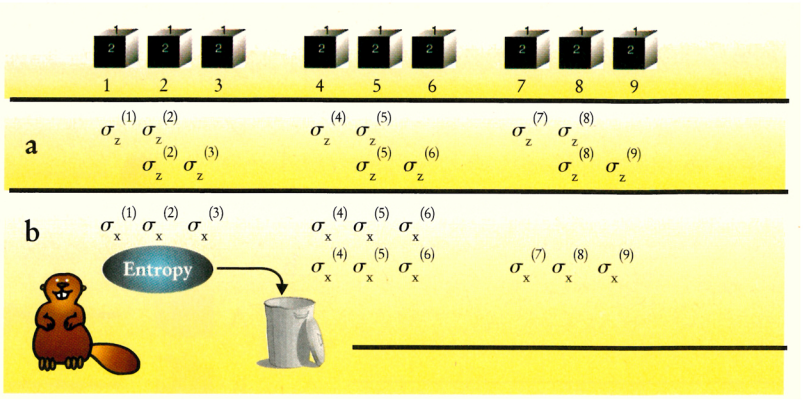


FIGURE 3. A QUANTUM CODE. It is possible to correct both bit-flip and phase-flip errors by encoding one qubit of quantum information in a block of nine qubits. Collective measurements preserve unknown individual qubit states (represented by closed boxes). **(a)** Six two-qubit observables (such as the tensor product of Pauli matrices $\sigma_z^{(1)} \otimes \sigma_z^{(2)}$) are measured to diagnose bit flips. **(b)** Two six-qubit observables (such as the tensor product of Pauli matrices $\sigma_x^{(1)} \otimes \sigma_x^{(2)} \otimes \sigma_x^{(3)} \otimes \sigma_x^{(4)} \otimes \sigma_x^{(5)} \otimes \sigma_x^{(6)}$) are measured to diagnose phase flips. Entropy introduced by errors is extracted in the form of a random measurement record, which can be discarded.

Following Shor,[5] we may encode a single qubit using a block of nine qubits (see figure 3), according to

$$\begin{aligned} |0\rangle &\to |\bar{0}\rangle \\ &\equiv \frac{1}{2^{3/2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle), \\ |1\rangle &\to |\bar{1}\rangle \\ &\equiv \frac{1}{2^{3/2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle). \end{aligned} \tag{8}$$

Both $|\bar{0}\rangle$ and $|\bar{1}\rangle$ consist of three clusters of three qubits each, with each cluster prepared in the same quantum state. Each of the clusters has triple-bit redundancy, so we can correct a single bit flip in any cluster by the method already discussed above.

Now suppose that a phase flip occurs in one of the clusters. The error changes the relative sign of $|000\rangle$ and $|111\rangle$ in that cluster so that

$$\begin{aligned} |000\rangle + |111\rangle &\to |000\rangle - |111\rangle, \\ |000\rangle - |111\rangle &\to |000\rangle + |111\rangle. \end{aligned} \tag{9}$$

The relative phase of the damaged cluster will now differ from the phases of the other two clusters. Thus, we can identify the damaged cluster, not by *measuring* the relative phase in each cluster (which would disturb the encoded information) but by *comparing* the phases of pairs of clusters—a six-qubit collective measurement. The measurement outcomes allow us to infer which cluster has a sign different from the others, and we may then apply a unitary phase transformation to one of the qubits in that cluster to reverse the sign and correct the error.

Error recovery will fail if there are two bit-flip errors in a single cluster (which would induce a phase error in the encoded data) or if phase errors occur in two clusters (which would induce a bit-flip error in the encoded data). But if the qubits interact only weakly with the environment and with one another, a double error will be relatively unlikely. Loosely speaking, if each qubit decoheres with a probability p and the decohering qubits are not

Box 1. Fault Tolerance and Topology

Topological ideas arise naturally in the theory of fault tolerance. The topological properties of an object remain invariant when we smoothly deform the object. Similarly, how a fault-tolerant gate acts on encoded information should remain unchanged when we deform the gate by introducing a small amount of noise. In seeking fault-tolerant implementations of quantum logic, we are led to contemplate physical interactions with a topological character.

What comes quickly to mind is the Aharonov–Bohm effect. When an electron is transported around a magnetic flux tube, its wave function acquires a phase that depends only on the winding number of the electron about the solenoid; it is unmodified if the electron's trajectory is slightly deformed. A device that processes quantum information by means of Aharonov–Bohm interactions would be intrinsically fault tolerant; accordingly, we would not need to implement a quantum gate with great precision for it to act as we desire.

Unfortunately, the Aharonov–Bohm effect is abelian, and we need noncommuting gates to build up a complex quantum computation. But it is possible in principle to devise two-dimensional spin systems that exhibit more intricate Aharonov–Bohm phenomena; long-range quantum correlations in the ground state of such a system can induce topological interactions among the localized quasiparticle excitations.[12] In a suitable spin system, the Aharonov–Bohm interactions are adequate for executing interesting computations like the quantum factoring algorithm.

Such an implementation of quantum computation seems futuristic from the perspective of current technology, but it is conceptually important. If we could perform quantum logic by means of topological interactions, then we would be able to give the beaver a rest! We could protect encoded information not by vigilantly checking for errors and reversing them, but rather by weaving fault tolerance into the design of our hardware.

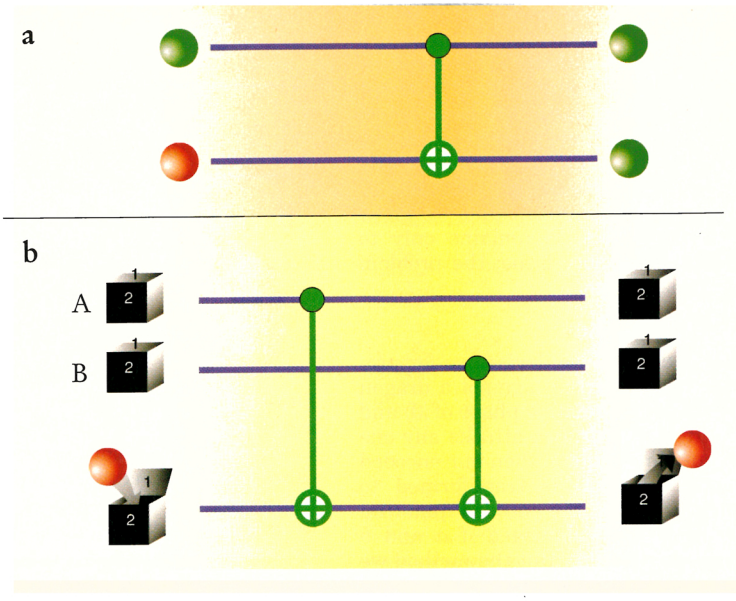


FIGURE 4. QUANTUM LOGIC of a collective measurement. (a) A controlled-NOT gate flips the target qubit if the control qubit (on top) is green. Otherwise, it acts trivially. (b) A collective observable of two data qubits (marked A and B) is measured by preparing an ancilla qubit, executing two controlled-NOT gates, and then measuring the ancilla.

strongly correlated, then the encoded information will decohere with a probability of order p^2. For p sufficiently small, coding will improve the reliability of the quantum information.

The nine-qubit code is conceptually simple, but it is not the most efficient quantum code that can protect against an arbitrary error afflicting any one of the qubits in the code block. It turns out that a five-qubit code can be devised to accomplish the same thing.[7] More sophisticated codes can be constructed that can protect against many damaged qubits in the code block.[8]

Collective measurement and fault tolerance

Collective measurements, which can diagnose errors without damaging the coherence of the data, are crucial to quantum error correction. Let's consider more closely how collective measurements can be carried out. The beaver would like to learn, for example, whether boxes A and B (both opened through door 1) contain balls of the same color or different color, but he doesn't want to find out the color of either ball.

To measure such collective observables, he will need a rudimentary quantum computer that can perform quantum logic gates in which two qubits come together and interact (see figure 4). A two-qubit gate that is particularly useful for this purpose is the controlled-NOT gate that acts according to this rule: If the first (control) qubit is $|0\rangle$, then the gate acts trivially, but if the first qubit is $|1\rangle$, the gate flips the value of the second (target) qubit.

When the beaver wants to measure the collective observable, he first prepares a third ("ancilla") qubit in the red state $|0\rangle$. Then a quantum circuit is executed in which two successive controlled-NOT gates are performed, each with the ancilla as the target and with the successive qubits A and B as the controls. If qubits A and B have the same color, the color of the ancilla qubit is flipped either zero times or twice, so it is still red when measured; but if qubits A and B have different colors, there is only one flip, and the ancilla becomes green. Measuring the ancilla reveals only the collective property, not the colors of the two individual qubits.

The ancilla is an essential part of the quantum error correction procedure, because it serves as a repository for the entropy that is introduced into the code block by the errors—it "heats" as the protected quantum system "cools." To protect quantum information for a long time, we need a continual supply of fresh ancilla qubits. Alternatively, if the ancilla is to be recycled, it must be erased. The erasure is a dissipative process; that is why quantum (or classical) error correction requires the expenditure of power.

Since our quantum computer will not be flawless, errors might occur during the collective measurement. Therefore, we must

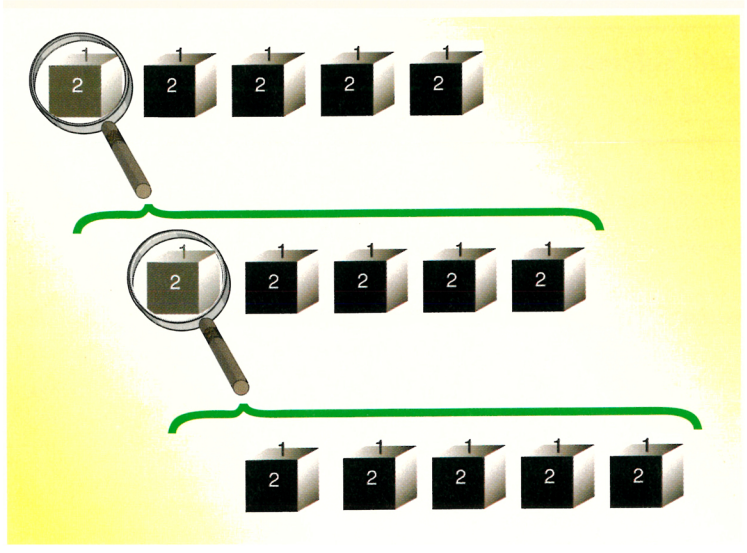
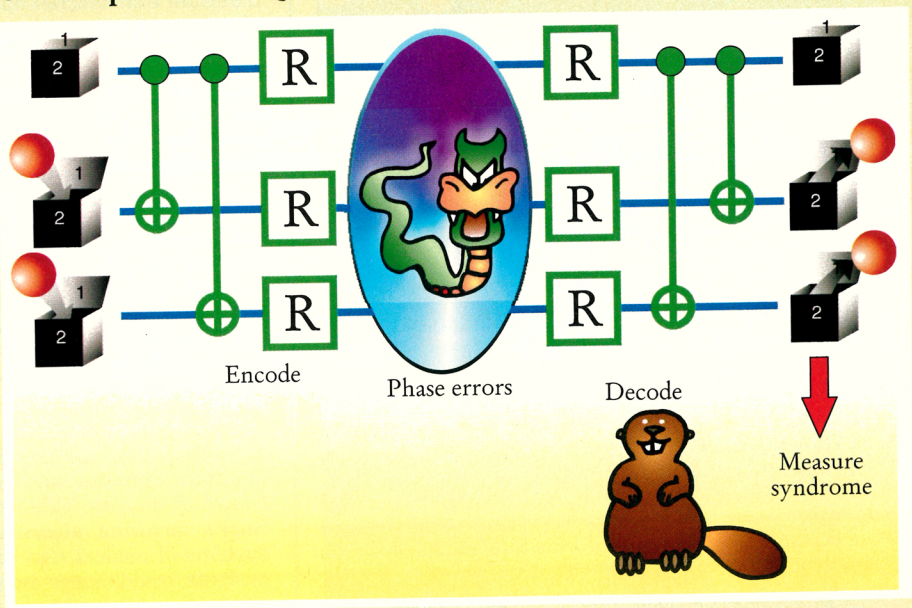


FIGURE 5. CODES WITHIN CODES. A single logical qubit is encoded in a block of five qubits. Each of the five qubits in that block, when inspected at higher resolution, is itself really a block of five qubits. And so on.

Box 2. Experimental Quantum Error Correction

The first experimental demonstrations of quantum error correction, using the methods of nuclear magnetic resonance (NMR), were reported in the past year. In those experiments, qubits were carried by nuclear spins that were manipulated by radio-frequency pulses, and quantum coding was used to protect a spin from dephasing. In an experiment by a group from Los Alamos National Laboratory and MIT,[13] (schematically illustrated in the figure), two ancilla spins were provided, and the qubit to be protected was encoded in correlations among the three by means of a simple quantum circuit. The three spins were exposed to the dephasing dragon for a while, and then the qubit was decoded. The ancilla spins were measured to reveal whether a phase error had been sustained; if it had, the damage could be repaired.



PROTECTING A NUCLEAR SPIN from phase errors. First, some controlled-NOTs and some single-qubit quantum gates are executed to encode the spin to be protected (top left) in correlations with the two ancilla spins (shown below it). Then the three spins, now in an entangled state, are subjected to weak dephasing. Finally, the spins are decoded, and two are measured to extract a syndrome that diagnoses whether a phase error has occurred.

In an experiment conducted by a group from IBM/Almaden and Stanford University,[14] a two-qubit code that could detect a phase error in either qubit was used, and the output was rejected when an error was detected. In the cases in which no error was detected, an improvement in fidelity could be verified.

Quantum error correction demonstrations that exploit the tools of quantum optics and atom trapping should be possible in the near future.[15]

be careful to design a protocol for error recovery that is fault tolerant, one that will still work effectively even if it is not executed perfectly. Indeed, fault-tolerant protocols can be constructed both for error correction and for executing quantum gates that process the encoded information.[9] Box 1 on page 27 describes a topological approach to fault tolerance.

If we wish to perform a long quantum computation reliably, we will need to use codes that can protect against many errors. One family of such codes can be envisioned as follows[10] (see figure 5): Suppose that we encode a single qubit in a block of five qubits. But each of those five qubits, when inspected more closely, is itself really another block of five, encoded as before. And so on. Such an intricate code requires substantial storage space, but in return we achieve high reliability. For an error to occur in the encoded qubit at the highest level, two qubits in the block of five would need to fail. And for either of those to fail, two would need to fail at the next level down. And so on. As we add more levels to the code, the probability of an error in the encoded qubit drops sharply.

Because of the overhead associated with processing encoded information, if our quantum hardware is highly inaccurate, then coding alone may not improve the performance of a quantum computer. But when the hardware becomes reliable enough, an encoded block will be more resistant to error than a raw qubit. Then adding another level to the code will improve the accuracy further. By using a sufficiently complex code, we can make the error rate in the encoded data as small as we please.[11]

In principle, then, an arbitrarily long quantum computation can be performed reliably, provided that the average probability of error per elementary quantum gate is less than a certain critical value, the accuracy threshold. The numerical value of the accuracy threshold depends on the model of decoherence that we adopt, and on other characteristics of our hardware. If we assume that the quantum hardware is highly parallelizable (so that we can execute many quantum gates in a single time step), and that the qubits decohere more or less independently, then an error probability per gate of 10^{-4} can be shown to be acceptable. (Roughly speaking, this error probability can be interpreted as the ratio of the time required to execute an elementary gate to the decoherence time of a single raw qubit.) Of course, to perform a longer computation, more redundancy will be needed for adequate reliability. But the required block size of the code grows at a modest rate with the length of the computation, as a power of a logarithm of the number of gates to be executed.[11]

Outlook

We may now claim to understand, in principle, how to fight off the destructive effects of decoherence. Though we may never see a real cat in a superposition of a dead state and a live state, someday we may be able to prepare an encoded cat that is half dead and half alive, and to

maintain that macroscopic superposition for as long as we please.

At present, though, quantum information technology remains in its pioneering stage. It is currently possible to do experiments involving a few qubits and a few quantum gates (box 2 on page 29). For a quantum computer to compete with a state-of-the-art classical computer, we will need machines with hundreds or thousands of qubits capable of performing millions or billions of operations. The technology clearly has far to go before quantum computers can assume their rightful place as the world's fastest machines. But now that we know how to protect quantum information from errors, there are no known insurmountable obstacles blocking the path. Quantum computers of the 21st century may well unleash the vast computational power woven into the fundamental laws of physics.

Apart from enabling a new technology, the discovery of fault-tolerant methods for quantum error correction and quantum computation may have deep implications for the future of physics. Efficient quantum algorithms (such as Shor's factoring algorithm) demonstrate that quantum systems of modest size can behave in ways that classical systems could never imitate. What else might coherent quantum systems be capable of? In what ways will they surprise, baffle, and delight us? Armed with new tools for maintaining and controlling intricate quantum states, physicists of the next century will seek the answers.

References

1. R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).
2. D. Deutsch, Proc. Roy. Soc. London, Ser. A **400**, 96 (1985).
3. P. Shor, in *Proc. of the 35th Annual Symp. on Foundations of Computer Science*, Los Alamitos, Calif., IEEE Press (1994), p. 124.
4. D. Dieks, Phys. Lett. A **92**, 271 (1982). W. K. Wootters, W. H. Zurek, Nature **299**, 802 (1982).
5. P. Shor, Phys. Rev. A **52**, 2493 (1995).
6. A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
7. R. Laflamme, C. Miquel, J. P. Paz, W. Zurek, Phys. Rev. Lett. **77**, 198 (1996). C. Bennett, D. DiVincenzo, J. Smolin, W. Wootters, Phys. Rev. A **54**, 3824 (1996).
8. A. R. Calderbank, P. W. Shor, Phys. Rev. A **54**, 1098 (1996). A. M. Steane, Proc. Roy. Soc. London, Ser. A **452**, 2551 (1996).
9. P. Shor, in *Proc. of the 37th Annual Symp. on Foundations of Computer Science*, Los Alamitos, Calif., IEEE Press (1996), p. 56.
10. E. Knill, R. Laflamme, preprint, http://xxx.lanl.gov/abs/quant-ph/9608012.
11. E. Knill, R. Laflamme, W. Zurek, Proc. Roy. Soc. London, Ser. A **454**, 365 (1998). A. Yu. Kitaev, Russ. Math. Surveys **52**, 1191 (1997). D. Aharonov, M. Ben-Or, in *Proc. of the 29th Annual ACM Symp. on the Theory of Computing*, New York, ACM (1997), p. 176. J. Preskill, Proc. Roy. Soc. London, Ser. A **454**, 385 (1998).
12. A. Yu. Kitaev, preprint, http://xxx.lanl.gov/abs/quant-ph/9707021. J. Preskill, preprint, http://xxx.lanl.gov/abs/quant-ph/9712048.
13. D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, S. S. Samaroo, Phys. Rev. Lett. **81**, 2152 (1998).
14. D. Leung, L. Vandersypen, X. Zhou, M. Sherwood, C. Yannoni, M. Kubinec, I. Chuang, preprint, http://xxx.lanl.gov/abs/quant-ph/9811068.
15. C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, D. J. Wineland, Phys. Rev. Lett. **75**, 4714 (1995). Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi, H. J. Kimble, Phys. Rev. Lett. **75**, 4710 (1995). ■