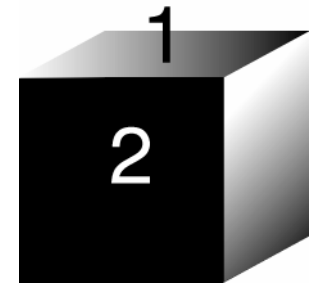
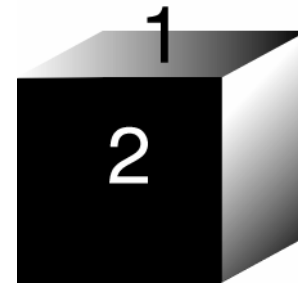
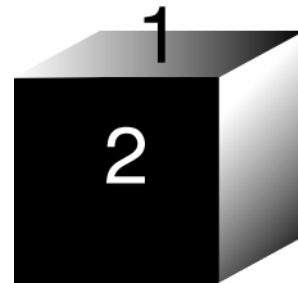
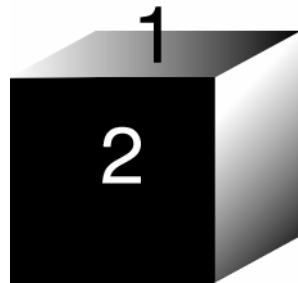
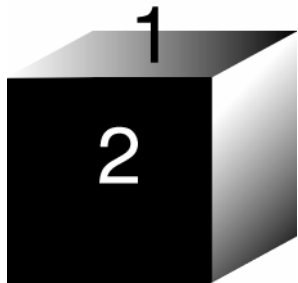


Battling decoherence:

The fault-tolerant quantum computer



John Preskill, Caltech
KITP

30 March 2006

<http://www.iqi.caltech.edu/>



Quantum fault tolerance: Topological vs. “Brute force”

- Error correction and fault tolerance will be essential in the operation of large-scale quantum computers.
- Topological quantum computing is the elegant approach, in which the “hardware” is intrinsically robust due to principles of local quantum physics (if operated at a temperature well below the mass gap). We hope it will work, but a physical realization of a topological quantum computer may be hard to achieve.
- There is also a “standard” approach to fault-tolerant quantum computing, which uses clever circuit design to overcome the deficiencies of quantum hardware. It, too, works in principle, if the hardware is not too noisy.
- Either approach (or perhaps a combination of the two) might eventually lead to quantum computers capable of solving hard problems. Which path we eventually follow will depend on which turns out to be more feasible technologically, and we don’t know that yet.

Quantum error correction

Can large-scale quantum computers really be built and operated? Surely there are daunting technical challenges to be overcome. But are there obstacles *in principle* that might prevent us from ever attacking hard computational problems with quantum computers?

What comes to mind is the problem of *errors*. Quantum computers will be far more susceptible to error than conventional digital computers. A particular challenge is to prevent *decoherence* due to interactions of the computer with the environment. Even aside from decoherence, the unitary quantum gates will not be perfect, and small imperfections will accumulate over time...

Quantum factoring

For example, suppose we would like to factor a 200 digit number (which hasn't yet been done with today's classical computers). The quantum factoring algorithm requires a few thousand qubits and a few billion (nontrivial) quantum gates. Suppose that in each gate (including "identity gates"), there is a probability ε of a serious error due to an interaction with the environment. Then for the algorithm to have a good probability of success, we require (ignoring parallelization, just to get a crude idea)

$$\varepsilon \cdot 10^3 \cdot 10^9 \leq O(1) \quad \Rightarrow \quad \varepsilon \leq 10^{-12}.$$

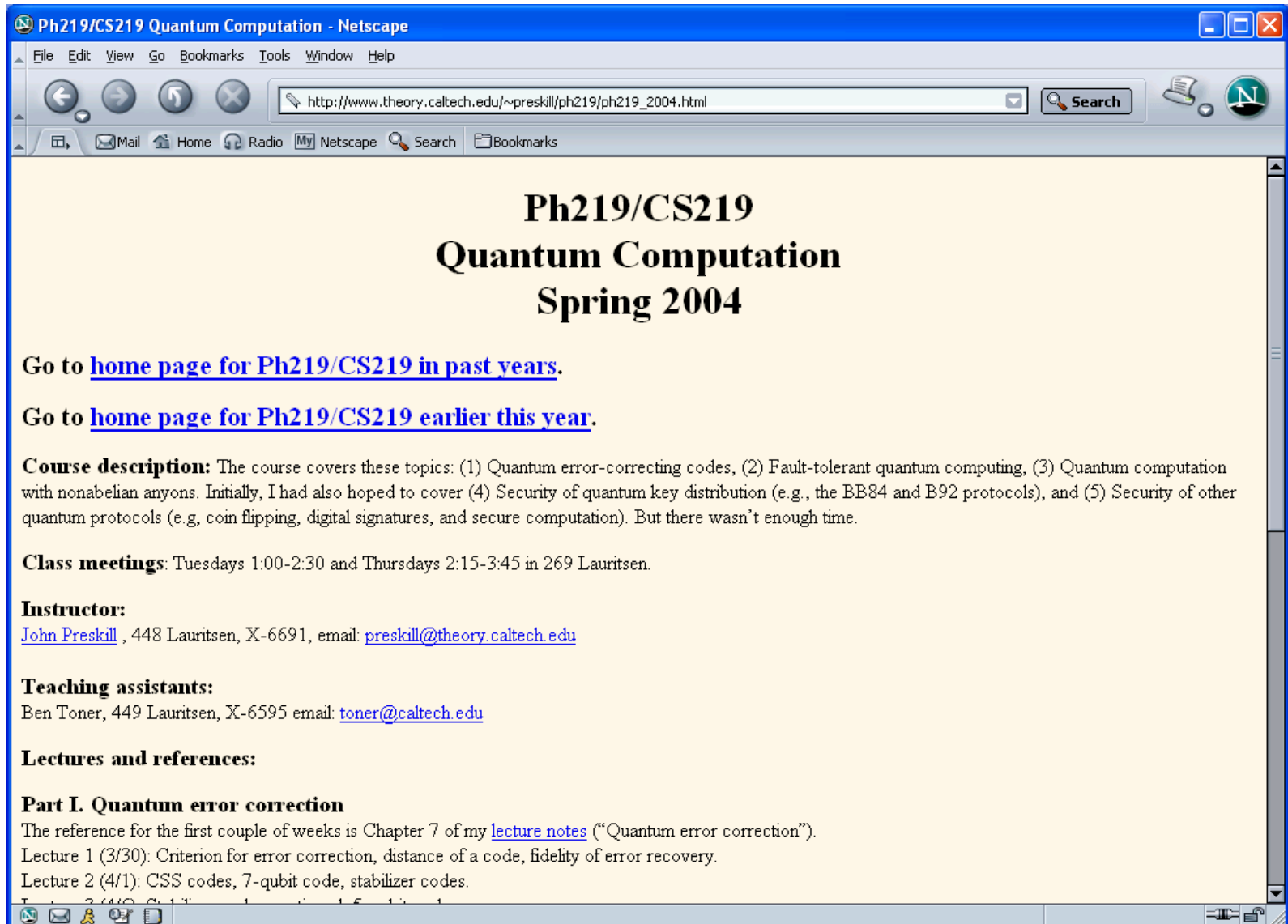
This is a very severe limitation!

Our confidence that large-scale quantum computations will someday be possible has been bolstered by the development of the theory of quantum error correction --- much larger error probabilities can be tolerated.

Quantum error correction

1. Error models and error correction
2. Quantum error-correcting codes
3. Stabilizer codes
4. 5-qubit code and 7-qubit code
5. Fault-tolerant quantum computation
6. Accuracy threshold

<http://www.theory.caltech.edu/~preskill/ph219/>



Ph219/CS219 Quantum Computation - Netscape

File Edit View Go Bookmarks Tools Window Help

http://www.theory.caltech.edu/~preskill/ph219/ph219_2004.html Search

Go Home Radio My Netscape Search Bookmarks

Ph219/CS219 Quantum Computation Spring 2004

Go to [home page for Ph219/CS219 in past years](#).

Go to [home page for Ph219/CS219 earlier this year](#).

Course description: The course covers these topics: (1) Quantum error-correcting codes, (2) Fault-tolerant quantum computing, (3) Quantum computation with nonabelian anyons. Initially, I had also hoped to cover (4) Security of quantum key distribution (e.g., the BB84 and B92 protocols), and (5) Security of other quantum protocols (e.g. coin flipping, digital signatures, and secure computation). But there wasn't enough time.

Class meetings: Tuesdays 1:00-2:30 and Thursdays 2:15-3:45 in 269 Lauritsen.

Instructor:
[John Preskill](#), 448 Lauritsen, X-6691, email: preskill@theory.caltech.edu

Teaching assistants:
Ben Toner, 449 Lauritsen, X-6595 email: toner@caltech.edu

Lectures and references:

Part I. Quantum error correction

The reference for the first couple of weeks is Chapter 7 of my [lecture notes](#) ("Quantum error correction").

Lecture 1 (3/30): Criterion for error correction, distance of a code, fidelity of error recovery.

Lecture 2 (4/1): CSS codes, 7-qubit code, stabilizer codes.



Chapter 7

Quantum Error Correction

7.1 A Quantum Error-Correcting Code

In our study of quantum algorithms, we have found persuasive evidence that a quantum computer would have extraordinary power. But will quantum computers really work? Will we ever be able to build and operate them?

To do so, we must rise to the challenge of protecting quantum information from errors. As we have already noted in Chapter 1, there are several aspects to this challenge. A quantum computer will inevitably interact with its surroundings, resulting in decoherence and hence in the decay of the quantum information stored in the device. Unless we can successfully combat decoherence, our computer is sure to fail. And even if we were able to prevent decoherence by perfectly isolating the computer from the environment, errors would still pose grave difficulties. Quantum gates (in contrast to classical gates) are unitary transformations chosen from a continuum of possible values. Thus quantum gates cannot be implemented with perfect accuracy; the effects of small imperfections in the gates will accumulate, eventually leading to a serious failure in the computation. Any effective strategem to prevent errors in a quantum computer must protect against small unitary

QUANTUM ACCURACY THRESHOLD FOR CONCATENATED DISTANCE-3 CODES

PANOS ALIFERIS

*Institute for Quantum Information, California Institute of Technology
Pasadena, CA 91125, USA*

DANIEL GOTTESMAN

*Perimeter Institute
Waterloo ON N2V 1Z3, Canada*

JOHN PRESKILL

*Institute for Quantum Information, California Institute of Technology
Pasadena, CA 91125, USA*

We prove a new version of the quantum threshold theorem that applies to concatenation of a quantum code that corrects only one error, and we use this theorem to derive a rigorous lower bound on the quantum accuracy threshold ϵ_0 . Our proof also applies to concatenation of higher-distance codes, and to noise models that allow faults to be correlated in space and in time. The proof uses new criteria for assessing the accuracy of fault-tolerant circuits, which are particularly conducive to the inductive analysis of recursive simulations. Our lower bound on the threshold, $\epsilon_0 \geq 2.73 \times 10^{-5}$ for an adversarial independent stochastic noise model, is derived from a computer-assisted combinatorial analysis; it is the best lower bound that has been rigorously proven so far.

1 Introduction

Our hopes that large-scale quantum computers will be built and operated someday are founded on the theory of quantum fault tolerance [1]. A centerpiece of this theory is the *quantum threshold theorem*, which asserts that an arbitrarily long quantum computation can be executed with high reliability, provided that the noise afflicting the computer's hardware is weaker than a certain critical value, the *accuracy threshold* [2, 3, 4, 5, 6]. In this paper, we will present new proofs of this fundamental theorem, and new rigorous lower bounds on the accuracy threshold.

Quantum computer: the standard model

- (1) Hilbert space of n qubits: $\mathfrak{H} = \mathbb{C}^{2^n}$
- (2) prepare initial state: $|0\rangle^{\otimes n} = |000\dots 0\rangle$
- (3) execute circuit built from set of universal quantum gates: $\{U_1, U_2, U_3, \dots, U_{n_G}\}$
- (4) measure in basis $\{|0\rangle, |1\rangle\}$

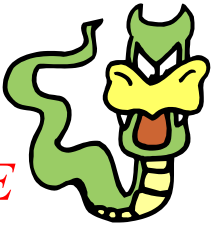
The model can be simulated by a classical computer with access to a random number generator. But there is an exponential slowdown, since the simulation involves matrices of exponential size... Thus we believe that quantum model is intrinsically more powerful than the corresponding classical model.

Our goal is to simulate accurately the ideal quantum circuit model using the imperfect noisy gates that can be executed by an actual device (assuming the noise is not too strong).

Errors

The most general type of error acting on n qubits can be expressed as a unitary transformation acting on the qubits and their environment:

$$U : |\psi\rangle \otimes |0\rangle_E \rightarrow \sum_a E_a |\psi\rangle \otimes |a\rangle_E$$



The states $|a\rangle_E$ of the environment are neither normalized nor mutually orthogonal. The operators $\{E_a\}$ are a basis for operators acting on n qubits, conveniently chosen to be “Pauli operators”:

$$\{I, X, Y, Z\}^{\otimes n},$$

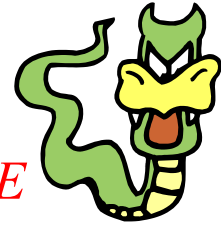
where

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The errors could be “unitary errors” if $|a\rangle_E = C_a |0\rangle_E$ or decoherence errors if the states of the environment are mutually orthogonal.

Errors

$$U : |\psi\rangle \otimes |0\rangle_E \rightarrow \sum_a E_a |\psi\rangle \otimes |a\rangle_E$$



Our objective is to recover the (unknown) state $|\psi\rangle$ of the quantum computer. We can't expect to succeed for arbitrary errors, but we might succeed if the errors are of a restricted type. In fact, since the interactions with the environment are *local*, it is reasonable to expect that the errors are not too strongly correlated.

Define the “weight” w of a Pauli operator to be the number of qubits on which it acts nontrivially; that is X, Y , or Z is applied to w of the qubits, and I is applied to $n-w$ qubits. If errors are weakly correlated (and rare), then Pauli operators E_a with large weight have small amplitude $\| |a\rangle_E \|$.

Error recovery

We would like to devise a recovery procedure that acts on the data and an *ancilla*:

$$V : E_a |\psi\rangle \otimes |0\rangle_A \rightarrow |\psi\rangle \otimes |a\rangle_A$$



which works for any $E_a \in \{\text{Pauli operators of weight } \leq t\}$.

Then we say that we can “correct t errors” in the block of n qubits. Information about the error that occurred gets transferred to the ancilla and can be discarded:

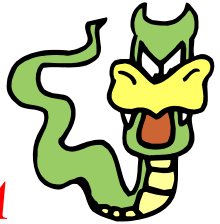
$$|\psi\rangle \otimes |0\rangle_E \otimes |0\rangle_A \xrightarrow{\text{error}} \sum_a E_a |\psi\rangle \otimes |a\rangle_E \otimes |0\rangle_A$$

recover

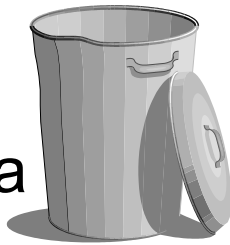
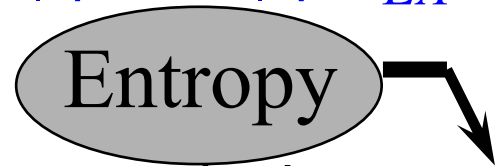
$$\rightarrow \sum_a |\psi\rangle \otimes |a\rangle_A \otimes |a\rangle_E = |\psi\rangle \otimes |\varphi\rangle_{EA}$$

Error recovery

$$|\psi\rangle \otimes |0\rangle_E \otimes |0\rangle_A \xrightarrow{\text{error}} \sum_a E_a |\psi\rangle \otimes |a\rangle_E \otimes |0\rangle_A$$



$$\xrightarrow{\text{recover}} \sum_a |\psi\rangle \otimes |a\rangle_A \otimes |a\rangle_E = |\psi\rangle \otimes |\varphi\rangle_{EA}$$



Errors entangle the data with the environment, producing *decoherence*. Recovery transforms entanglement of the data with the environment into entanglement of the ancilla with the environment, “purifying” the data. Decoherence is thus reversed. Entropy introduced in the data is transferred to the ancilla and can be discarded --- we “refrigerate” the data at the expense of “heating” the ancilla. If we wish to erase the ancilla (cool it to $T \approx 0$, so that we can use it again) we need to pay a power bill.

Quantum error-correcting code

We won't be able to correct all errors of weight up to t for arbitrary states $|\psi\rangle \in \mathfrak{H}_{n \text{ qubits}}$. But perhaps we can succeed for states contained in a *code subspace* of the full Hilbert space,

$$\mathfrak{H}_{\text{code}} \in \mathfrak{H}_{n \text{ qubits}}.$$

If the code subspace has dimension 2^k , then we say that k **encoded qubits are embedded in the block of n qubits.**

How can such a code be constructed? It will *suffice* if

$$\{E_a \mathfrak{H}_{\text{code}}, E_a \in \{\text{Pauli operators of weight } \leq t\}\}$$

are mutually orthogonal.

If so, then it is possible in principle to perform an (incomplete) orthogonal measurement that determines the error E_a (without revealing any information about the encoded state). We recover by applying the unitary transformation E_a^{-1} .

2-qubit “code”

The key concept in quantum coding theory is the *stabilizer* of a state. E.g., $|\phi^+\rangle = (|00\rangle_{AB} + |11\rangle_{AB})/\sqrt{2}$ is the simultaneous eigenstate with eigenvalue 1 of two commuting Pauli operators:

$$M_X = X \otimes X, \quad M_Z = Z \otimes Z.$$

These two conditions on two qubits determine a one-dimensional subspace, i.e., a unique state.

Suppose that Bob’s qubit is protected, but Alice’s qubit might have been damaged. Can we diagnose the damage?

The space of possible errors is spanned by $\mathcal{E} = \{I, X, Y, Z\} \otimes I$

By measuring the 2 generators of the code’s *stabilizer group*, we can distinguish all possible errors acting on Alice’s qubit.

2-qubit “code”

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle_{AB} + |11\rangle_{AB})$$
$$M_X = X \otimes X$$
$$M_Z = Z \otimes Z$$

All of the errors in $\mathcal{E} = \{I, X, Y, Z\} \otimes I$ anticommute with the stabilizer generators in distinguishable ways.

| | M_X | M_Z |
|---------------|-------|-------|
| $I \otimes I$ | + | + |
| $X \otimes I$ | + | - |
| $Y \otimes I$ | - | - |
| $Z \otimes I$ | - | + |

By measuring the 2 generators of the code’s *stabilizer group*, we can distinguish all possible errors acting on Alice’s qubit.

4-qubit code

$$M_X = XXXX$$

$$M_Z = ZZZZ$$

There are $n - k = 4 - 2 = 2$ encoded qubits.

The 4-dimensional code space is spanned by:

$$\begin{aligned} &|0000\rangle + |1111\rangle \\ &|0011\rangle + |1100\rangle \\ &|0101\rangle + |1010\rangle \\ &|0110\rangle + |1001\rangle \end{aligned}$$

| | M_X | M_Z |
|--------|-------|-------|
| $IIII$ | + | + |
| $XIII$ | + | - |
| $YIII$ | - | - |
| $ZIII$ | - | + |

(An X changes the parity, a Z changes the relative phase, a Y does both..)

Suppose that one qubit out of the four is damaged, and we know which one, but we don't know the nature of the damage. By measuring the two stabilizer generators, we can distinguish among I, X, Y, Z acting on (e.g.) the first qubit.

General stabilizer codes

Operators M_1, M_2, \dots, M_{n-k} are independent mutually commuting Pauli operators, $M_i^2 = I$, which generate an abelian subgroup S of the “Pauli group.” S is the code’s stabilizer group.

A vector $|\psi\rangle$ in the n -qubit Hilbert space is in the code subspace iff $M|\psi\rangle = |\psi\rangle$ for all M in S .

The dimension of the code space is: $2^n \left(\frac{1}{2^{n-k}} \right) = 2^k$.
(There are k encoded qubits.)

The code can correct t errors if each (nontrivial) Pauli operator of weight up to $2t$ anticommutes with at least one of the stabilizer generators:

$$\begin{aligned} \langle \psi | E_a^\dagger E_b | \psi \rangle &= \langle \psi | E_a^\dagger E_b M_i | \psi \rangle = -\langle \psi | M_i E_a^\dagger E_b | \psi \rangle \\ &= -\langle \psi | E_a^\dagger E_b | \psi \rangle \Rightarrow E_a \mathcal{H}_{\text{code}} \perp E_b \mathcal{H}_{\text{code}} \text{ for } \text{weight}(E_a) \leq t \end{aligned}$$

All errors up to weight t are distinguishable (and correctable).

5-qubit code

Suppose we would like to encode $k=1$ protected qubits in a block of n qubits, and be able to correct all weight-1 errors. How large must n be?

There are two mutually orthogonal “codewords” $|\bar{0}\rangle, |\bar{1}\rangle$ that span the code subspace. Furthermore all $E_a |\bar{0}\rangle, E_b |\bar{1}\rangle$ should be mutually orthogonal.

There are $3 \times 5 + 1 = 16$ Pauli operators of weight ≤ 1 , and the Hilbert space of 5 qubits has dimension $2^5 = 32$. Therefore, for $n=5$, there is just barely enough room: $16 \times 2 \leq 2^5 = 32$.

To see that the code really exists, we can construct it explicitly.

5-qubit code

The code is the simultaneous eigenspace with eigenvalue 1 of 4 commuting *check operators* (*stabilizer generators*):

All of these stabilizer generators square to I ; they are mutually commuting because there are two collisions between X and Z .

$$M_1 = XZ ZX I = +1$$

$$M_2 = IX ZZ X = +1$$

$$M_3 = XI XZZ = +1$$

$$M_4 = ZX IXZ = +1$$

The other three generators are obtained from the first by cyclic permutations. (Note that $M_5 = ZZ XIX = M_1 M_2 M_3 M_4$ is not independent.) Therefore, the code is cyclic (cyclic permutations of the qubits preserve the code space).

Claim: no Pauli operator E of weight 1 or 2 commutes with all of the check operators. Weight 1: each column contains an X and a Z . Weight 2: Because the code is cyclic, it suffices to consider $??III$ and $?I?II \dots$

5-qubit code

- $k=1$ protected qubit
- corrects $t=1$ error

The code is the simultaneous eigenspace with eigenvalue 1 of 4 commuting *check operators*:

$$M_1 = XZZXI = +1$$

$$M_2 = IXZZX = +1$$

$$M_3 = XIXZZ = +1$$

$$M_4 = ZXIXZ = +1$$

By these operators, we can distinguish all possible weight-one errors. Each “syndrome” points to a unique Pauli operator of weight 0 or 1.

| | M_1 | M_2 | M_3 | M_4 |
|-------|-------|-------|-------|-------|
| X_1 | + | + | + | - |
| Y_1 | - | + | - | - |
| Z_1 | - | + | - | + |
| X_2 | - | + | + | + |
| Y_2 | - | - | + | - |
| Z_2 | + | - | + | - |
| X_3 | - | - | + | + |
| Y_3 | - | - | - | + |
| Z_3 | + | + | - | + |
| X_4 | + | - | - | + |
| Y_4 | - | - | - | - |
| Z_4 | - | + | + | - |
| X_5 | + | + | - | - |
| Y_5 | + | - | - | - |
| Z_5 | + | - | + | + |
| I | + | + | + | + |

5-qubit code

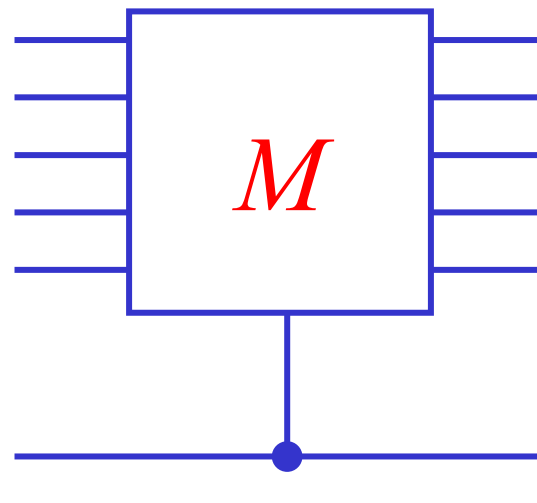
$$M_1 = XZZXI$$

$$M_2 = IXZZX$$

$$M_3 = XIXZZ$$

$$M_4 = ZXIXZ$$

How do we measure the stabilizer generators without destroying the encoded state?

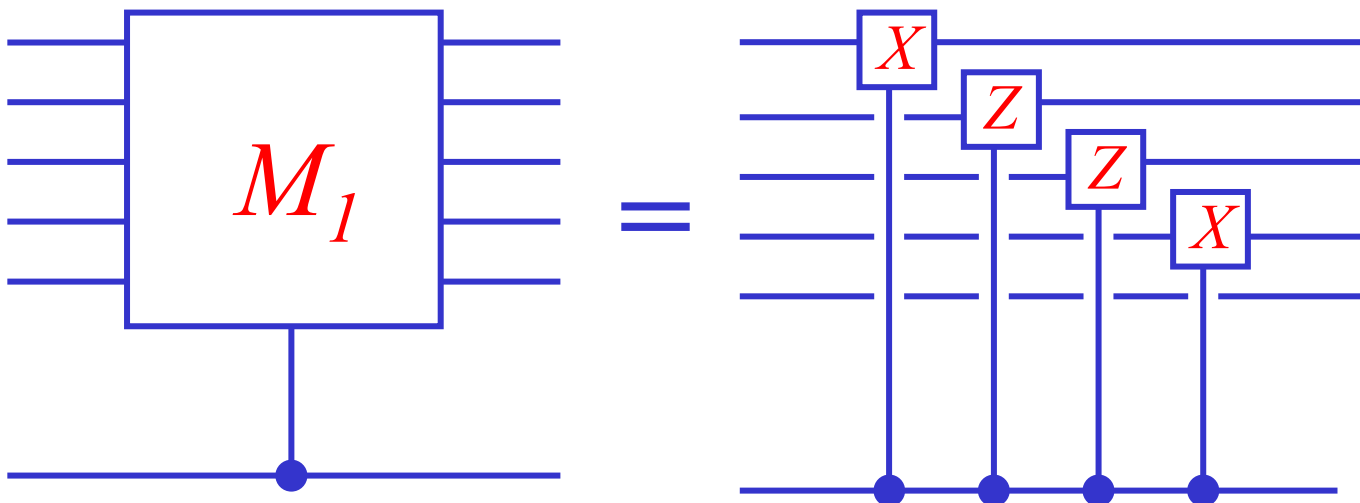


Apply M conditioned on value of an ancilla qubit.

$X=I$ Eigenstate: $|0\rangle_A + |1\rangle_A$

$|0\rangle_A + M|1\rangle_A$

Measure X



5-qubit code

What are the *encoded operations* that preserve the code space and act nontrivially within it?

$$\begin{aligned}M_1 &= XZZXI \\M_2 &= IXZZX \\M_3 &= XIXZZ \\M_4 &= ZXIXZ\end{aligned}$$

We may choose them to be:

$$\begin{aligned}\bar{Z} &= ZZZZZ \\ \bar{X} &= XXXXX\end{aligned}$$

(which anticommute with one another and commute with the stabilizer).

General stabilizer code

The code stabilizer S is an abelian subgroup of order 2^{n-k} of the the n -qubit Pauli group. Its *dual* or *normalizer* S^\perp is the subgroup of the Pauli group containing all Pauli operators that commute with S (thus S^\perp contains S). The encoded operations are the coset space S^\perp / S . The minimum weight of $S^\perp \setminus S$ is the *distance* of the code. A code with distance $d=2t+1$ can correct t errors.

7-qubit code

$$\begin{aligned}M_{Z,1} &= Z I Z I Z I Z \\M_{Z,2} &= Z Z I I Z Z I \\M_{Z,3} &= Z Z Z Z I I I\end{aligned}$$

Corrects the bit-flip (X) errors. The three-bit string $(M_{Z,1}, M_{Z,2}, M_{Z,3})$ (if nonzero) points to the position of the error.

$$\begin{aligned}M_{X,1} &= X I X I X I X \\M_{X,2} &= X X I I X X I \\M_{X,3} &= X X X X I I I\end{aligned}$$

Corrects the phase (Z) errors. The three-bit string $(M_{X,1}, M_{X,2}, M_{X,3})$ (if nonzero) points to the position of the error.

The M_Z 's commute with the M_X 's, because each row of the M_Z matrix has an even number of "collisions" with each row of the M_X matrix; i.e., the rows are orthogonal in the sense of linear algebra over the field \mathbb{Z}_2 . Any two matrices with this property define a quantum code, which is said to be of the "CSS" (Calderbank-Shor-Steane) type. With CSS codes, the bit-flip and phase error correction can be executed separately. The encoded operations can be chosen to be

$$\bar{Z} = I I I I Z Z Z, \quad \bar{X} = I I I I X X X$$

which commute with the code stabilizer and are not contained in it.

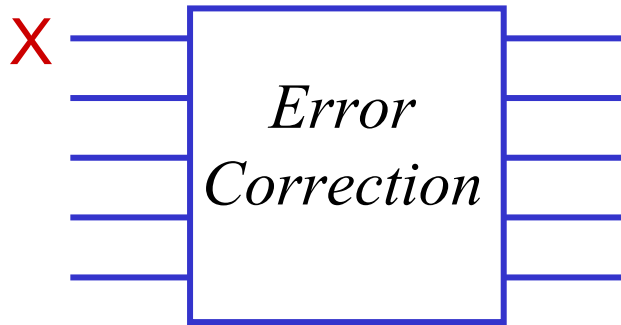
Fault tolerance

- The measured error syndrome (*i.e.*, the eigenvalues of the check operators) might be inaccurate.
- Errors might propagate during syndrome measurement.
- We need to implement a universal set of quantum gates that act on encoded quantum states, without unacceptable error propagation.
- We need codes that can correct many errors in the code block.

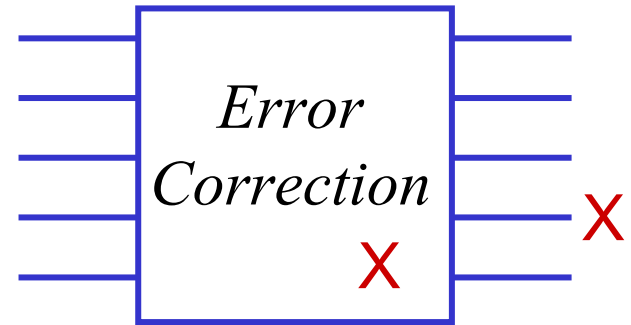
Fault-tolerant error correction

Fault: a location in a circuit where a gate or storage error occurs.

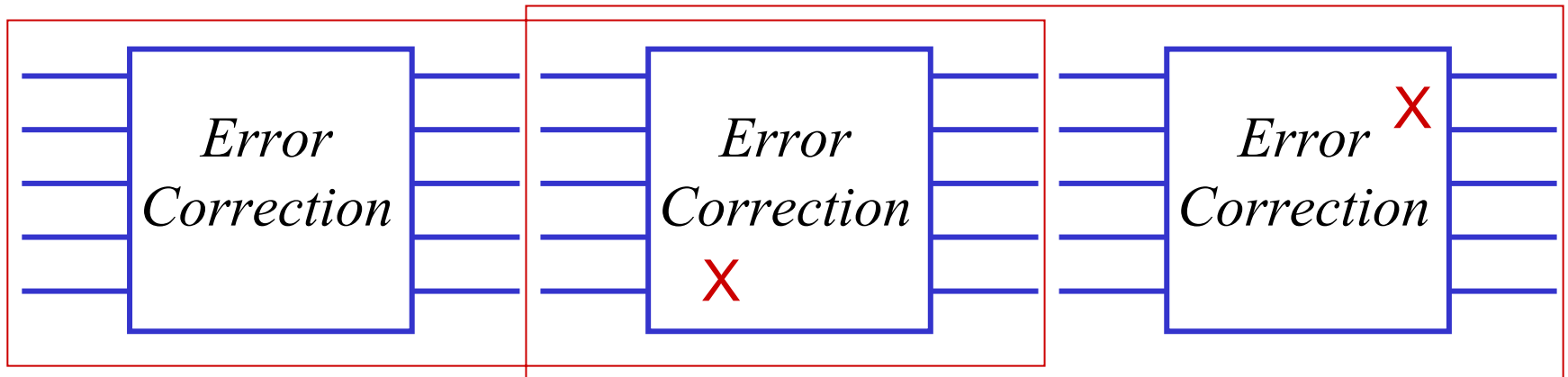
Error: a qubit in a block that deviates from the ideal state.



If input has at most one error, and circuit has no faults, output has no errors.

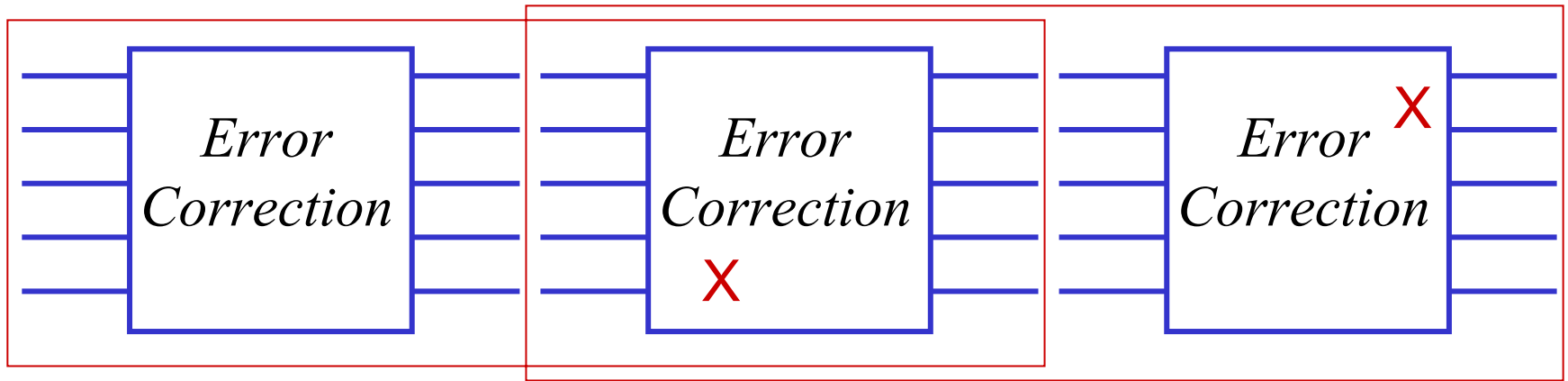


If input has no errors, and circuit has at most one fault, output has at most one error.



A quantum memory fails only if two faults occur in some “extended rectangle.”

Fault-tolerant error correction



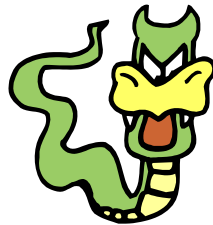
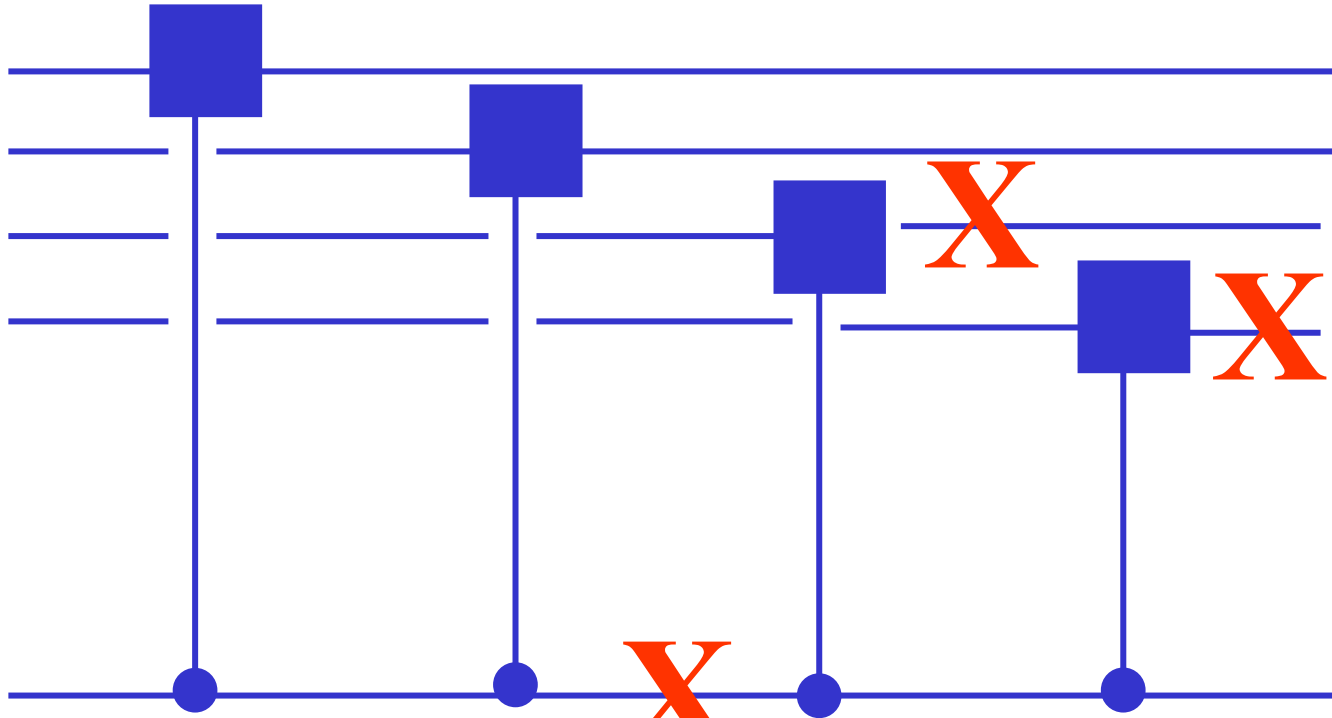
A quantum memory fails only if two faults occur in some “extended rectangle.”

If we perform T error correction steps in succession, and faults occur independently with probability ε at each circuit location, then the probability of a memory failure is

$$P_{\text{fail}} \leq TA\varepsilon^2$$

where A is the number of pairs of circuit locations in an extended rectangle. Therefore, by using a quantum code that corrects one error and a fault-tolerant error correction procedure, we can improve the “storage time” for quantum information (the time we can store a state with some specified constant fidelity) to $T=O(\varepsilon^{-2})$, compared to $T=O(\varepsilon^{-1})$ for an unprotected quantum state.

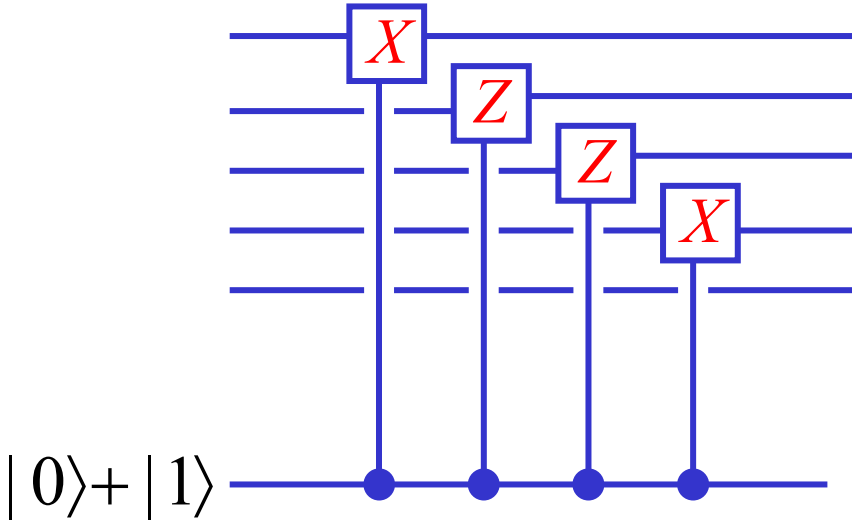
Error propagation in error correction



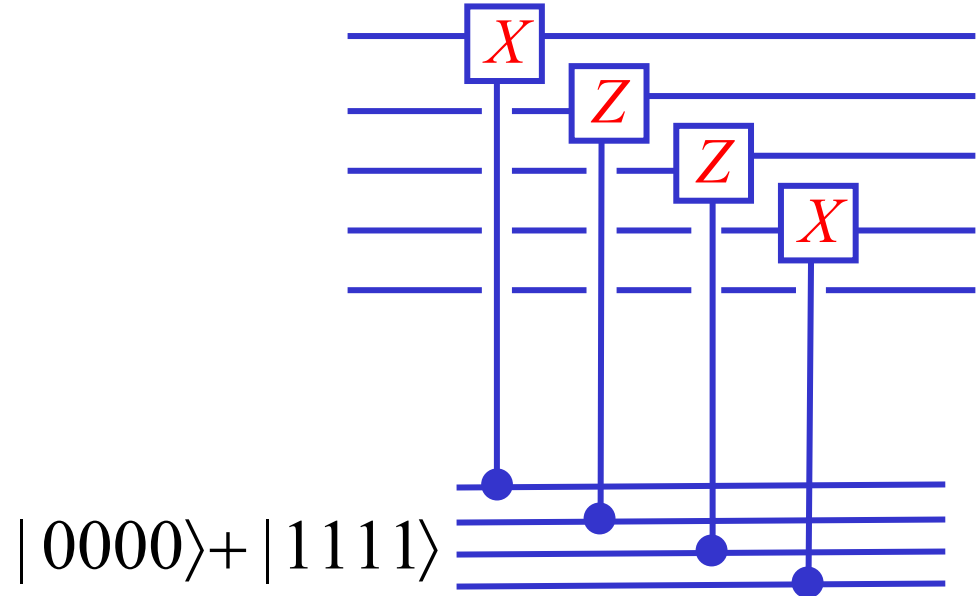
A single error due to the environment causes multiple errors in the data.

Fault-tolerant measurement

This is *bad*:



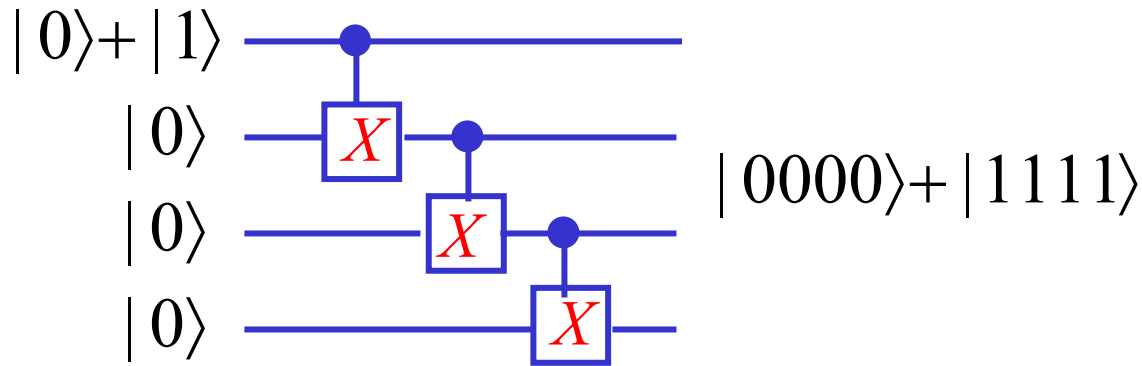
This is *better*:



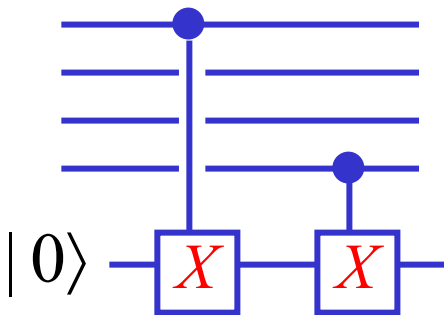
To make the procedure fault tolerant, we replace the single ancilla qubit by an encoded qubit, using a suitable code (in this case the quantum repetition code). That way, we avoid using any ancilla qubit more than once, and so control the error propagation. The state $|0000\rangle + |1111\rangle$, the $X=1$ eigenstate in the repetition code, is sometimes called a *cat state*. The phase of the cat state (the value of the *encoded X*, that is $XXXX$) is determined by measuring X of each qubit and computing the parity of the outcomes.

Preparation and verification

Included in our procedure is the preparation of the state $|0000\rangle + |1111\rangle$

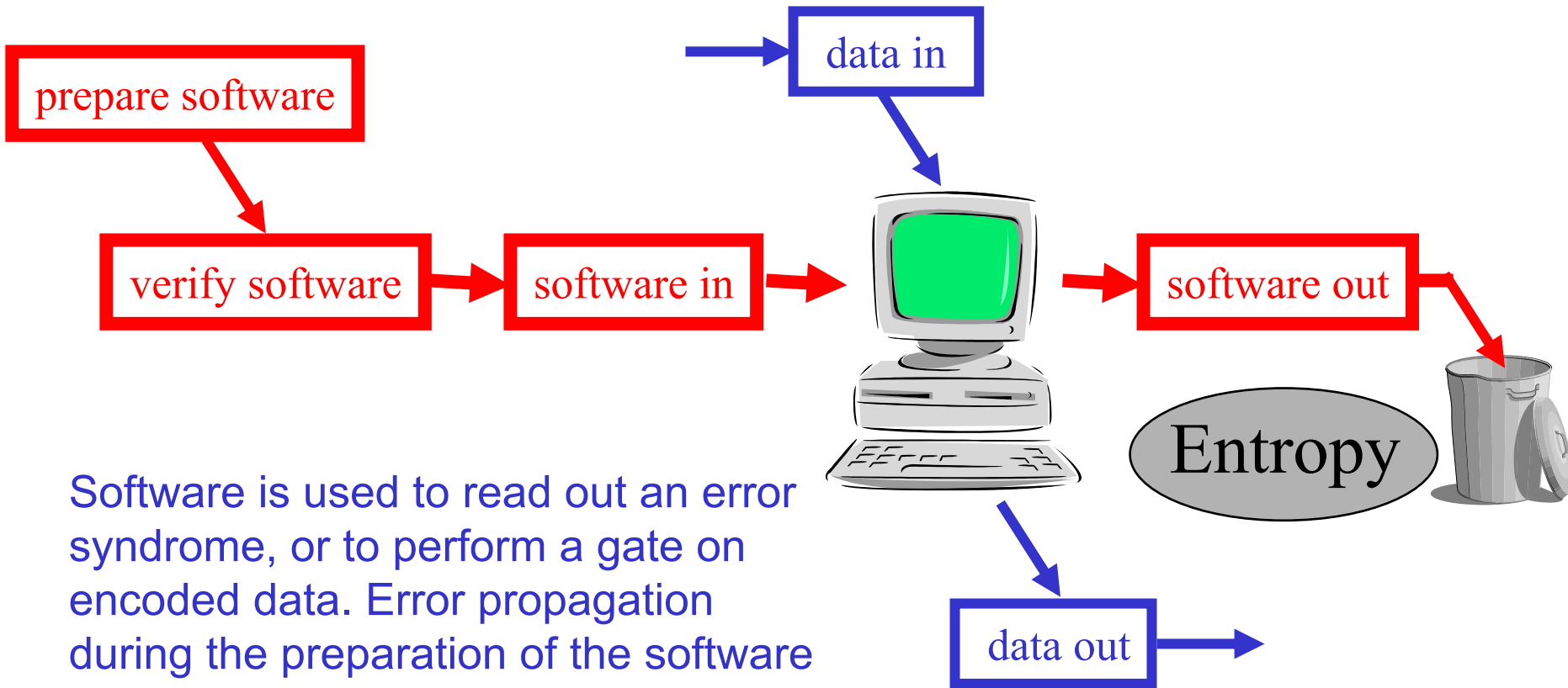


But this encoding circuit is not fault tolerant --- a single X error could propagate to two qubits, so that $|0011\rangle + |1100\rangle$ is prepared instead. Using this faulty ancilla would introduce two errors into the data. So we need to check the ancilla (after it is prepared, and before we let it touch the data) by verifying that the first and last bits agree.



If verification fails, the ancilla is discarded. Though the gates and measurements used in verification could be faulty, there would need to be two independent faults in the syndrome measurement procedure for the encoded data to be damaged.

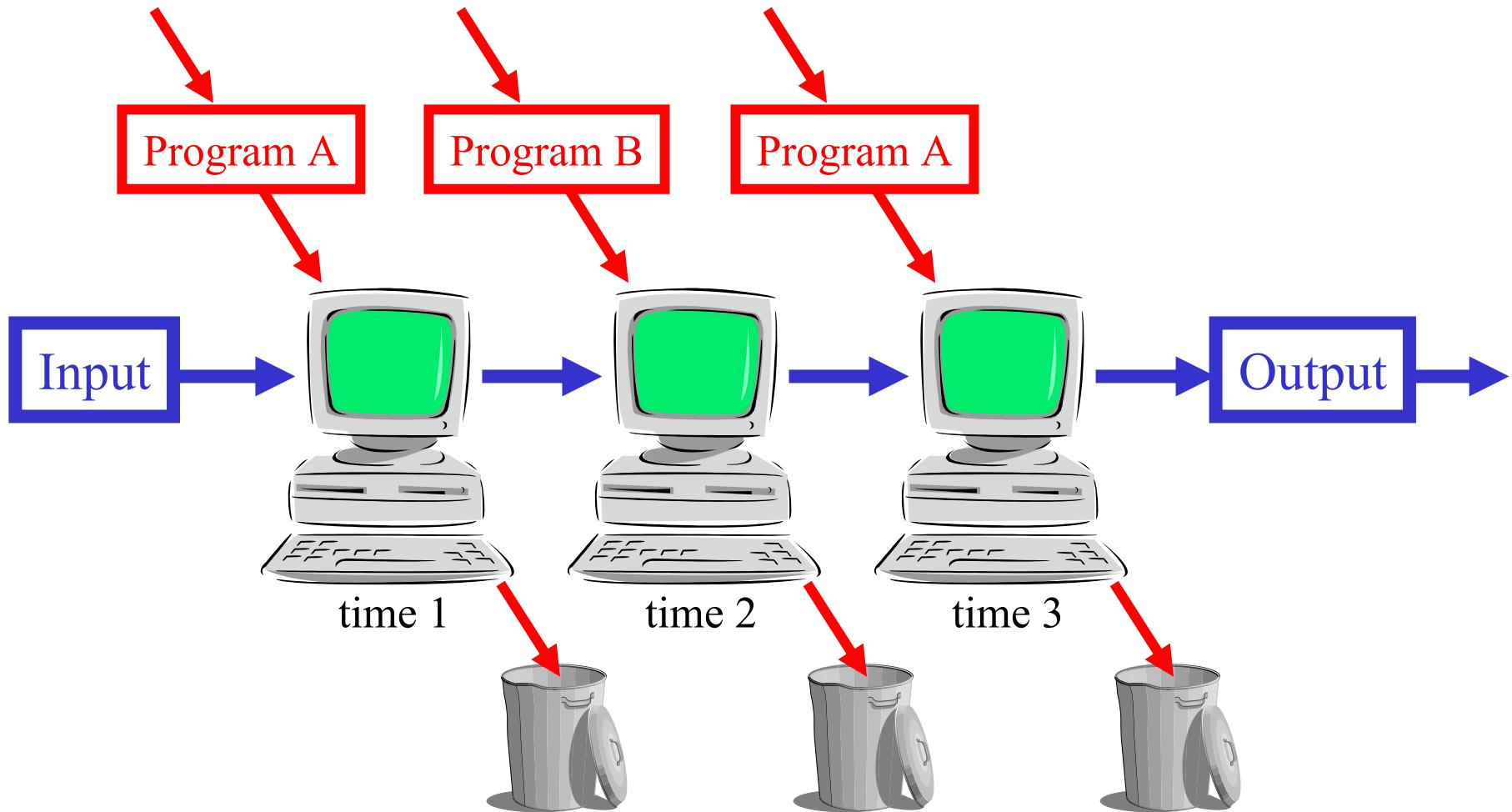
Preparation and consumption of *quantum software*



Software is used to read out an error syndrome, or to perform a gate on encoded data. Error propagation during the preparation of the software can cause *bugs* that might damage the data when the software is used.

Therefore the software must be checked and purified. After a single use, the software is irreparably damaged and must be discarded.

Consumption of *quantum software*

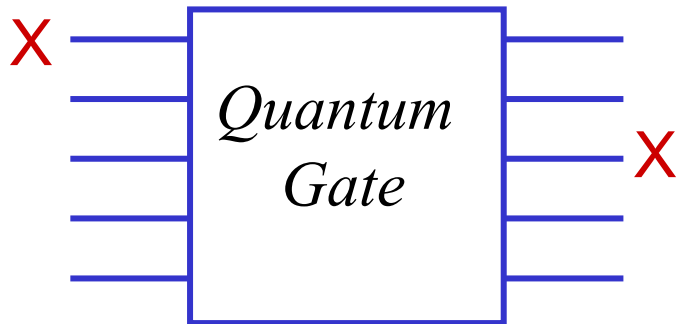


After a single use, the software is irreparably damaged and must be discarded. To execute a quantum algorithm, the user downloads and consumes a particular program many times.

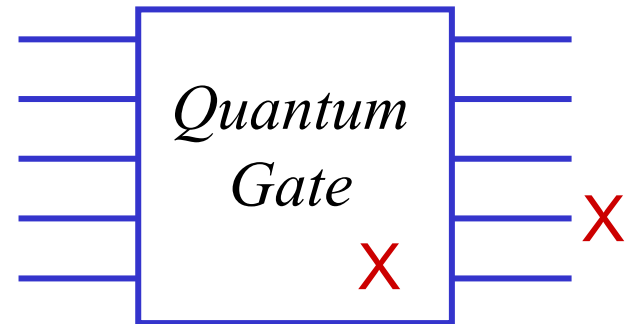
Fault-tolerant quantum gates

Fault: a location in a circuit where a gate or storage error occurs.

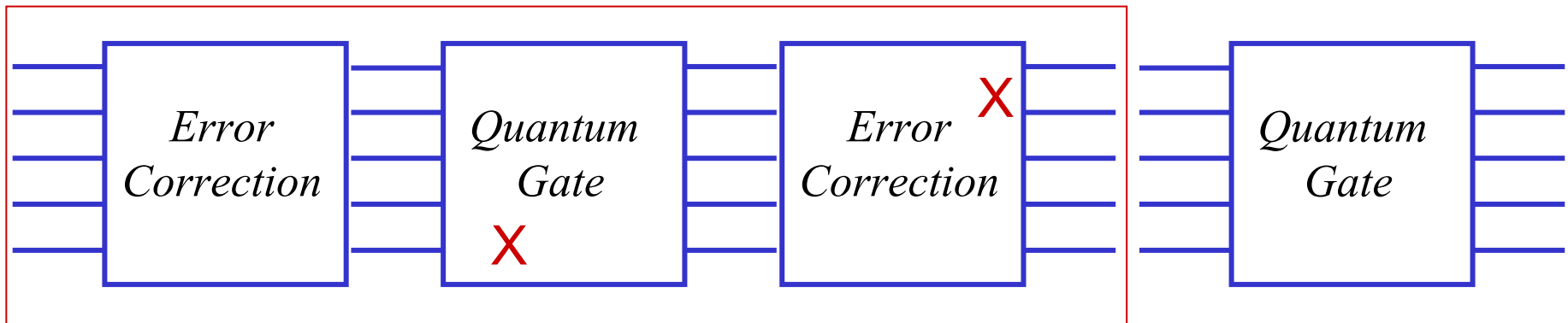
Error: a qubit in a block that deviates from the ideal state.



If input has at most one error, and circuit has no faults, output has at most one error in each block.

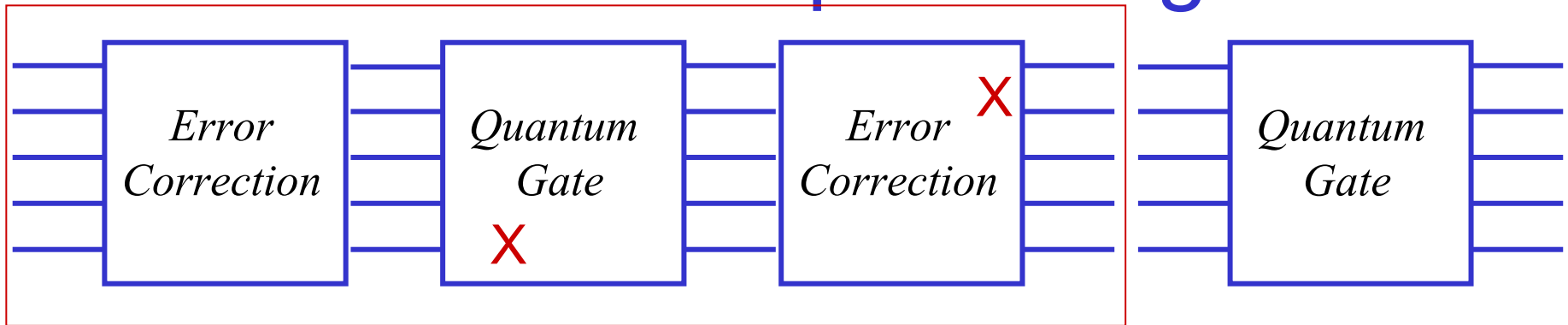


If input has no errors, and circuit has at most one fault, output has at most one error in each block.



Each gate is preceded by an error correction step. The circuit simulation fails only if two faults occur in some “extended rectangle.”

Fault-tolerant quantum gates



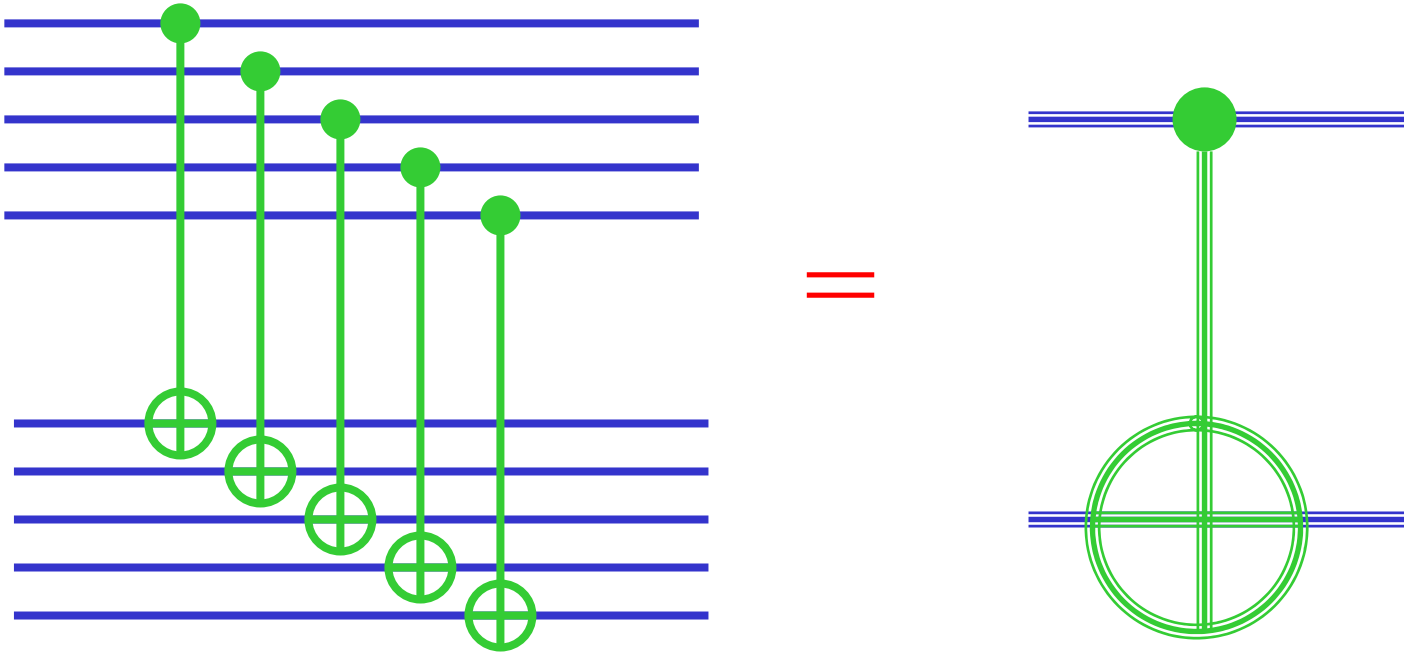
Each gate is followed by an error correction step. The circuit simulation fails only if two faults occur in some “extended rectangle.”

If we simulate an ideal circuit with L quantum gates, and faults occur independently with probability ε at each circuit location, then the probability of failure is

$$P_{\text{fail}} \leq LA_{\text{max}} \varepsilon^2$$

where A_{max} is an upper bound on the number of pairs of circuit locations in each extended rectangle. Therefore, by using a quantum code that corrects one error and fault-tolerant quantum gates, we can improve the circuit size that can be simulated reliably to $L=O(\varepsilon^{-2})$, compared to $L=O(\varepsilon^{-1})$ for an unprotected quantum circuit.

Transversal gates are fault tolerant:

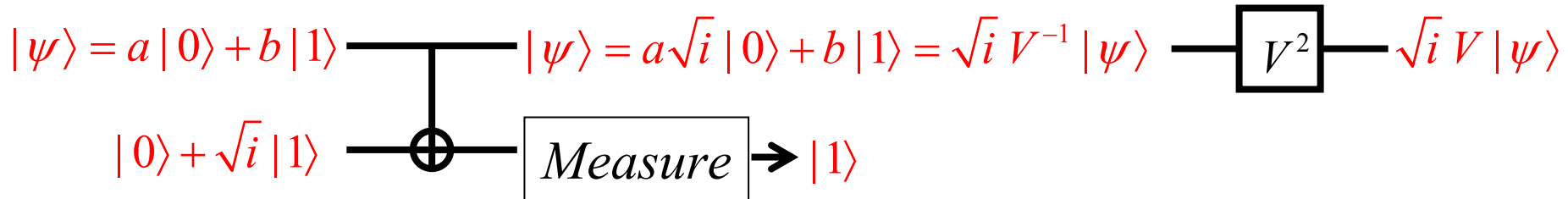
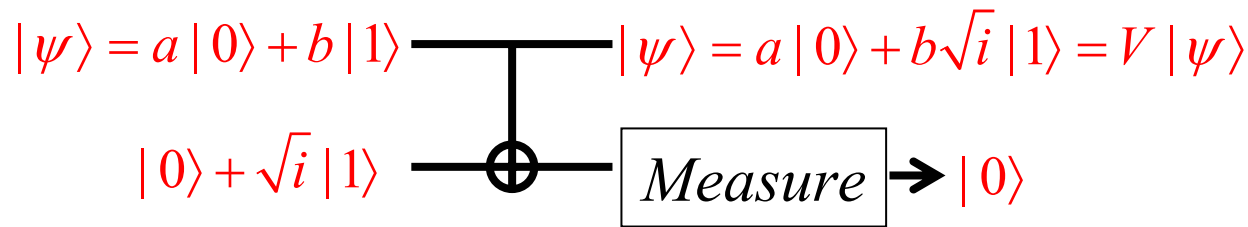


For codes of the CSS type, an encoded CNOT can be implemented by executing CNOTs between the corresponding qubits in the encoded blocks. This gate may propagate errors from one block to another, but not from one qubit to another in the same block.

Not all gates in a universal set can be realized transversally and fault tolerantly. To complete a universal fault-tolerant gate set, we must add to the transversal gates additional gates that are implemented by consuming *quantum software*.

Quantum software for the “ $\pi/8$ gate”

$P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ is easy to implement.
 $V = \sqrt{P} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{i} \end{pmatrix}$ is hard and needed for universality.

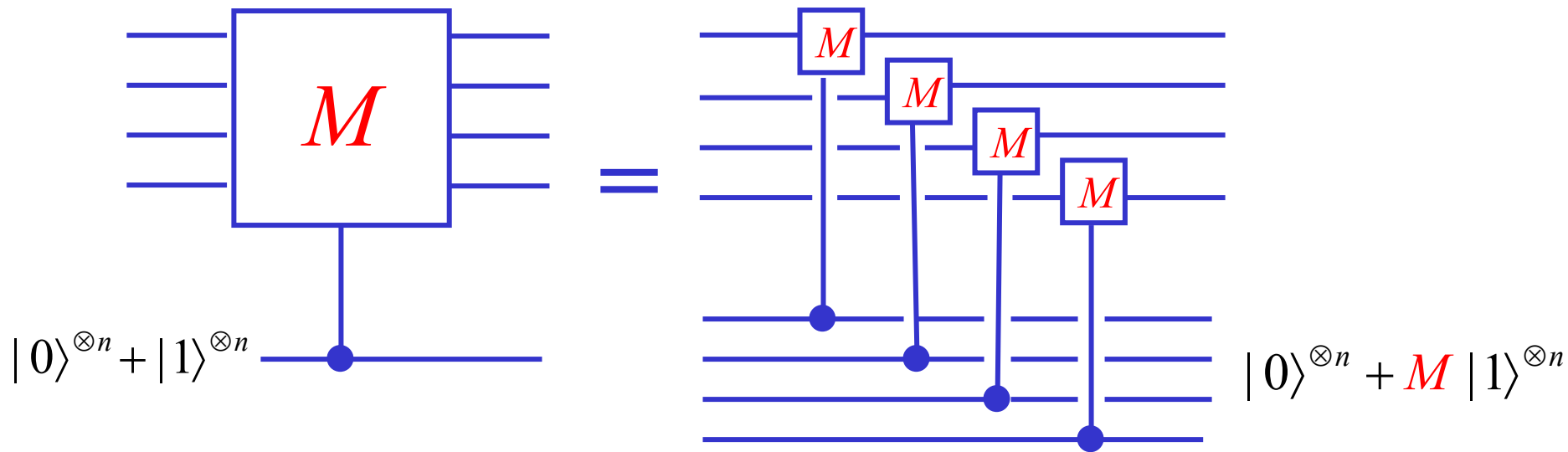


Performing the gate V is reduced to the task of preparing the software, which is achieved by measuring:

$$VXV^{-1} = \sqrt{-i}PX = \begin{pmatrix} 0 & \sqrt{-i} \\ \sqrt{i} & 0 \end{pmatrix}$$

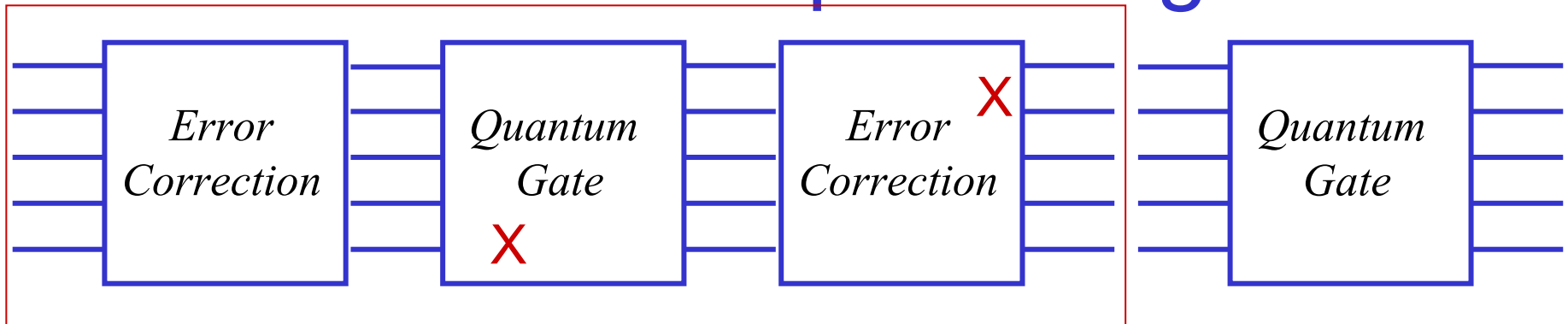
Preparing the software:

If a gate M can be *applied* transversally, then it can *measured* using a *cat state*:



The phase of the cat is determined by measuring X of each qubit and computing the parity of the outcomes. The cat state should be verified before use, and the measurement should be repeated to improve reliability.

Fault-tolerant quantum gates



Each gate is faulted by an error correction step. **The circuit simulation fails only if two faults occur in some “extended rectangle.”**

Now we have assembled all the ingredients of a fault-tolerant quantum circuit simulation.

Syndrome measurement and a universal gate set are realized through the offline preparation and verification of quantum software.

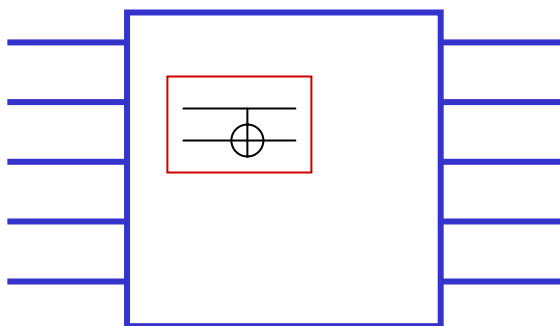
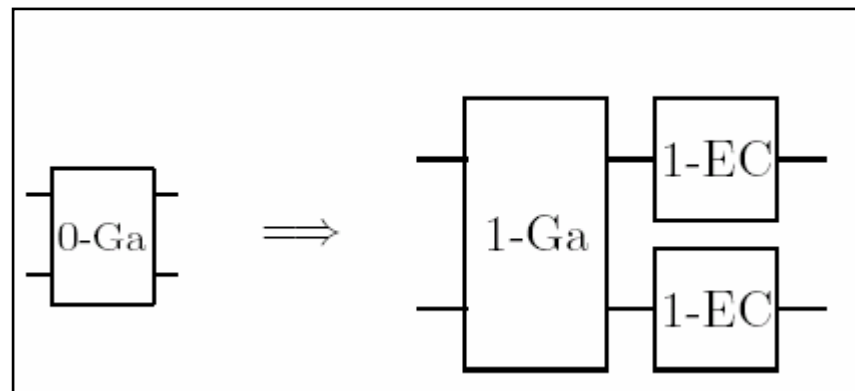
A single fault during syndrome measurement may cause an error in the syndrome. The syndrome measurement can be repeated to ensure its reliability.

Therefore if faults occur independently with probability ε at each circuit location:

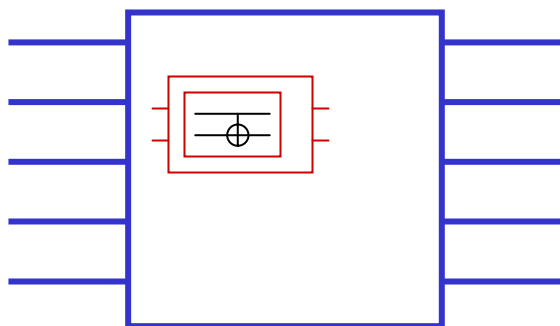
$$P_{\text{fail}} = O(\varepsilon^2)$$

Recursive simulation

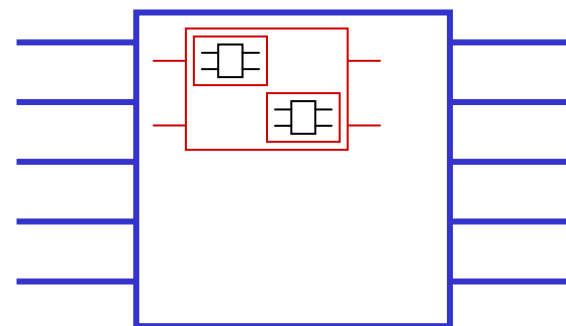
In a fault-tolerant simulation, each (level-0) ideal gate is replaced by a *1-Rectangle*: a (level-1) gate gadget followed by (level-1) error correction on each output block. In a level- k simulation, this replacement is repeated k times --- the ideal gate is replaced by a *k-Rectangle*.



A *1-rectangle* is built from quantum gates.



A *2-rectangle* is built from 1-rectangles.



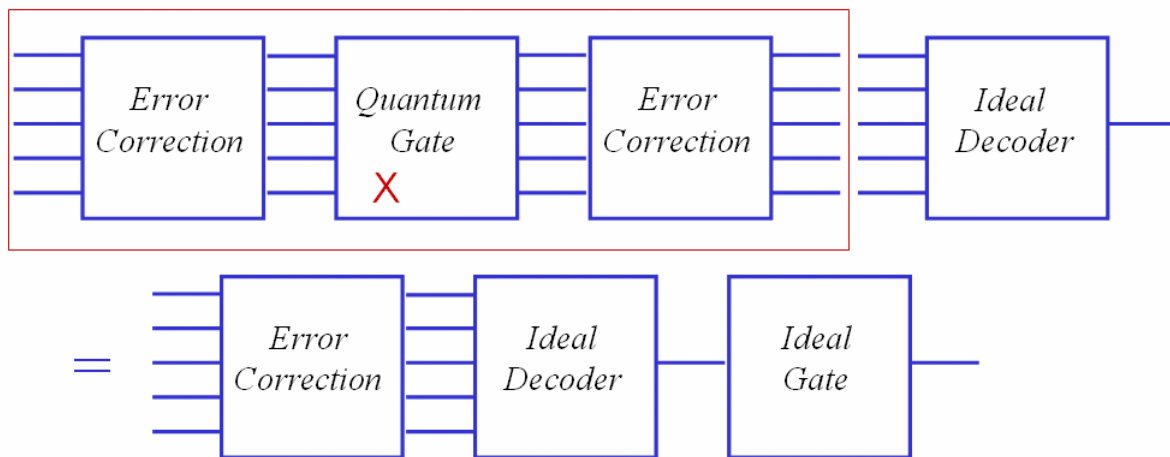
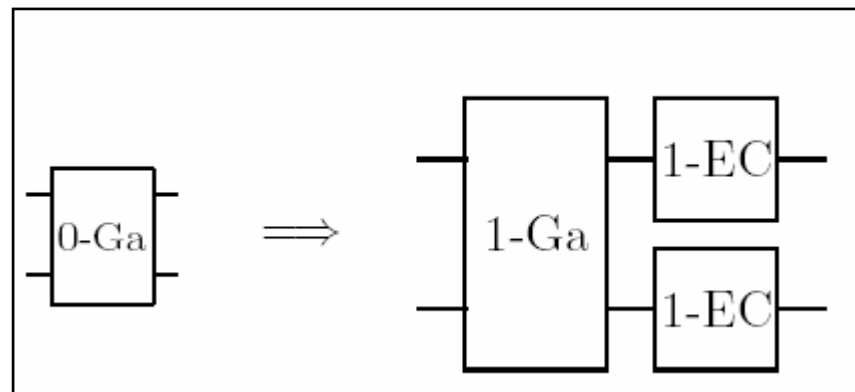
A *3-rectangle* is built from 2-rectangles.

- (1) The computation is accurate if the faults in a level- k simulation are *sparse*.
- (2) A non-sparse distribution of faults is *very unlikely* if the noise is *weak*.

There is *threshold of accuracy*. If the fault rate is below the threshold, then an arbitrarily long quantum computation can be executed with good reliability.

Recursive simulation

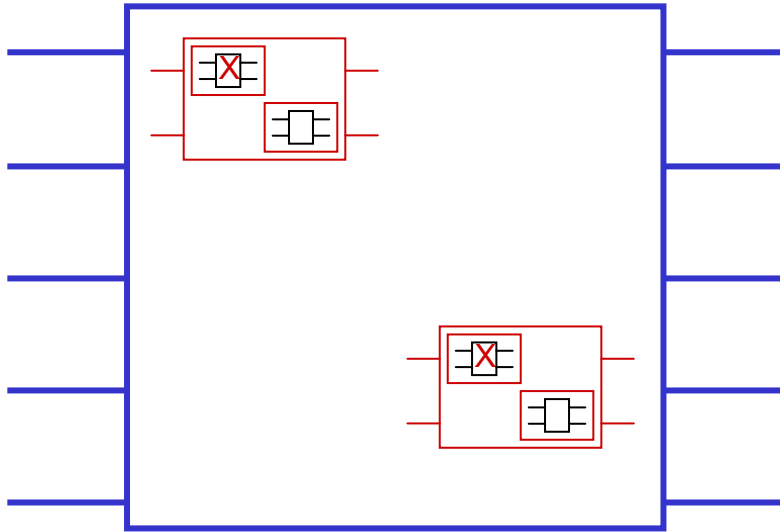
In a fault-tolerant simulation, each (level-0) ideal gate is replaced by a *1-Rectangle*: a (level-1) gate gadget followed by (level-1) error correction on each output block. In a level- k simulation, this replacement is repeated k times --- the ideal gate is replaced by a *k -Rectangle*.



A level-1 simulation based on a distance-3 code is reliable if each *extended 1-Rectangle* (the 1-Rec combined with its leading error corrections) contains no more than one fault (or faults at a benign set of locations).

If a 1-exRec is *good* (contains at most one fault, or faults at a benign set of locations), then the 1-Rec is *correct* (an ideal decoding of its output gives the same result as an ideal decoding of its input followed by an ideal execution of the simulated gate).

Recursive simulation



Definition:

A level-1 extended rectangle is *good* if it contains no more than one fault; otherwise it is *bad*.

An level- k extended rectangle is *good* if it contains no more than one bad level- $(k-1)$ extended rectangle; otherwise it is *bad*.

A level- k recursive simulation will be successful if every level- k extended rectangle is good.

Goodness implies correctness not just at level 1, but also at higher levels of the recursive hierarchy, as can be seen by an inductive argument.

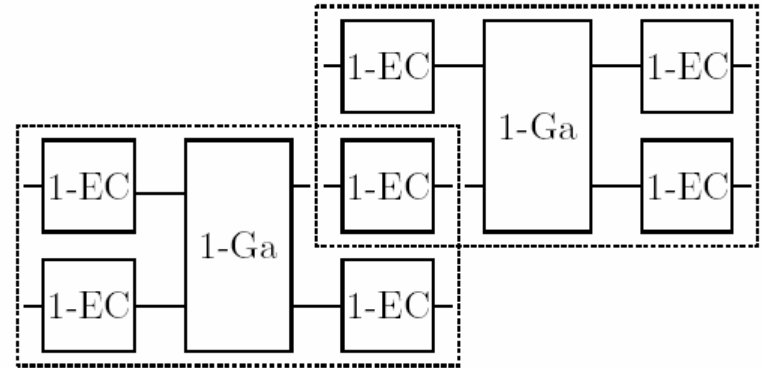
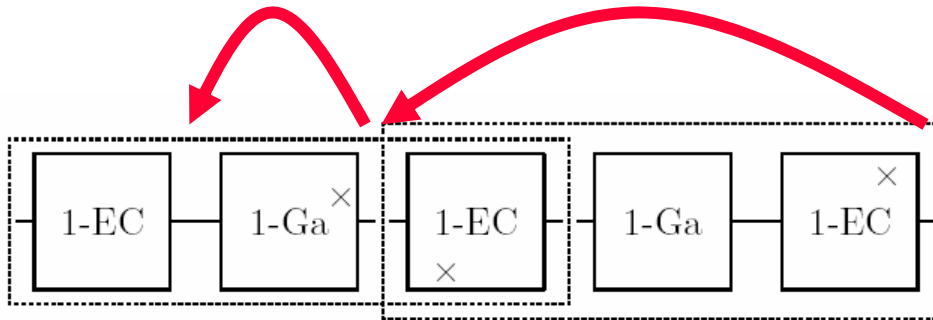
We can move ideal 1-encoders from the back of a noisy level- $(k+1)$ circuit toward the front of the circuit, transforming the 1-rectangles in good level-1 extended rectangles to ideal level-0 gates, and transforming the 1-rectangles in bad level-1 extended rectangles to faulty level 0-gates.

Thus a good level- $(k+1)$ simulation is seen to be equivalent to a good level- k simulation, which is correct by the induction hypothesis.

Extends to codes that can correct t errors, and to non-Markovian noise.

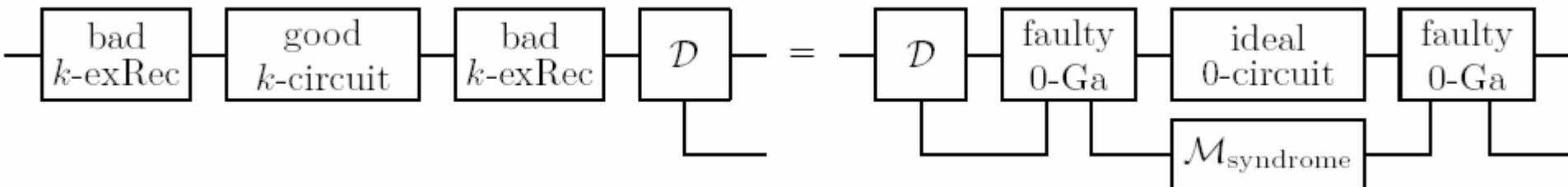
Overlapping rectangles

Successive 1-exRecs are not independent of one another because they *overlap*: a trailing error correction of the earlier exRec is also a leading error correction of the later exRec.



In the “threshold dance,” decoders moving left make a “long hop” over the later bad exRec, followed by a “short hop” over the Ga contained in the earlier *truncated* exRec.

That way, moving the decoder left transforms the two successive exRecs to two faults only if both fail independently (both the later exRec and the earlier truncated exRec both must contain faults at a malignant set of locations).



Recursive simulation

Lemma: The k -rectangle contained in a good extended k -rectangle is correct.

The level- k circuit simulation succeeds if all extended k -rectangles are good. How likely is a bad extended k -rectangle?

$$P_{\text{bad}}^{(1)} \leq A\varepsilon^2 ; \quad P_{\text{bad}}^{(k)} \leq A \left(P_{\text{bad}}^{(k-1)} \right)^2$$

$$\Rightarrow P_{\text{bad}}^{(k)} \leq A^{-1} \left(A\varepsilon \right)^{2^k} \quad (\text{double exponential scaling in } k)$$

Therefore, for a circuit with L locations (including “identity gates”)

$$P_{\text{fail}}^{(k)} \leq LA^{-1} \left(A\varepsilon \right)^{2^k} < \delta$$

is satisfied provided that $\varepsilon < \varepsilon_0 = A^{-1}$ and

$$\left(\varepsilon / \varepsilon_0 \right)^{2^k} < \delta / \varepsilon_0 L \quad \text{or} \quad 7^k = 2^{k \log_2 7} > \left[\frac{\log(\varepsilon_0 L / \delta)}{\log(\varepsilon_0 / \varepsilon)} \right]^{\log_2 7}$$

Recursive simulation

Quantum Accuracy Threshold Theorem: Suppose that faults occur independently at the locations within a quantum circuit, where the probability of a fault at each location is no larger than ε . Then there exists $\varepsilon_0 > 0$ such that for a fixed $\varepsilon < \varepsilon_0$ and fixed $\delta > 0$, any circuit of size L can be simulated by a circuit of size L^* with accuracy greater than $1 - \delta$, where, for some constant c ,

$$L^* = O\left[L(\log L)^c\right]$$

The numerical value of the *accuracy threshold* ε_0 is of practical interest!

Essential assumptions:

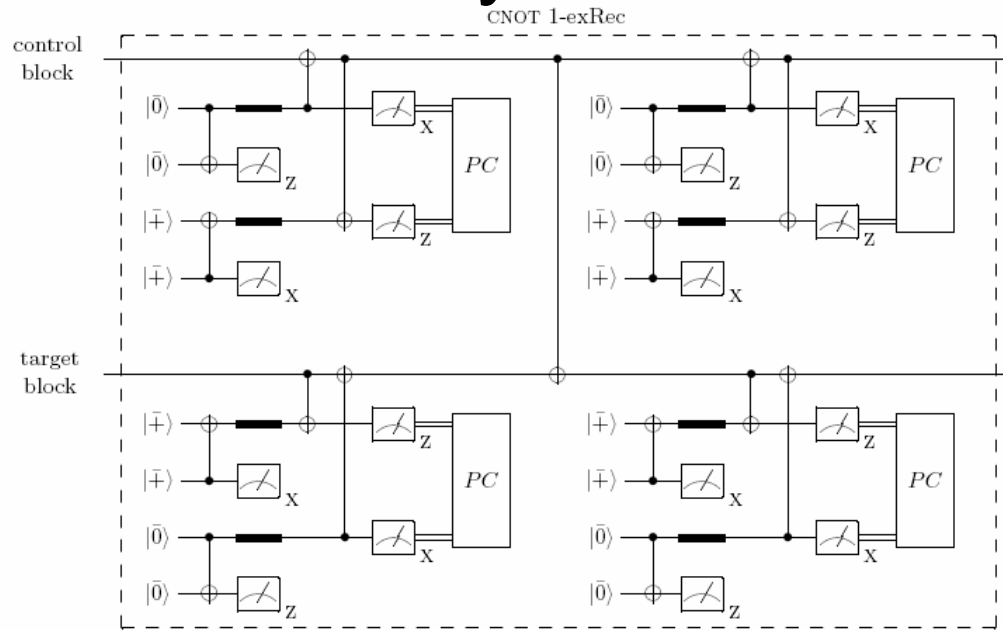
- *Constant fault rate* (independent of number of qubits).
- *Weakly correlated faults* (both in space and in time).
- *Parallelism* (to correct errors in all blocks simultaneously.)
- *Reusable memory* (to refresh ancillas that carry away entropy introduced by errors).

Helpful assumptions (used in threshold estimates):

- *Fast measurements* (to read out error syndromes -- without measurement, threshold is more demanding).
- *Fast classical processing* (to interpret error syndromes).
- *Nonlocal gates* (with local gates, threshold is more demanding).
- *No “leakage”* (e.g., loss of qubits).

Lower bound on the accuracy threshold

A **good** gadget (one with sparse faults) is **correct** (simulates the ideal gate accurately).



For each of the level-1 extended Rectangles in a universal set, e.g. for the $[[7,1,3]]$ (Steane) code, we can count the number of pairs of malignant locations; the CNOT 1-exRec dominates the threshold estimate. We find a rigorous lower bound on the accuracy threshold for *adversarial independent stochastic noise*:

$$\varepsilon_0 > 2.73 \times 10^{-5}$$

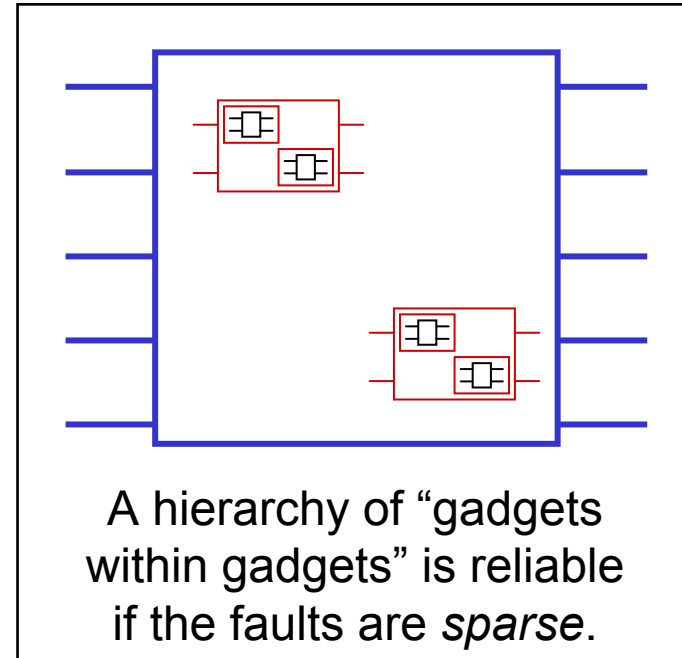
(assuming parallelism, fresh ancillas, nonlocal gates, fast measurements, fast and accurate classical processing, no leakage).

Fault-tolerant recursive simulation

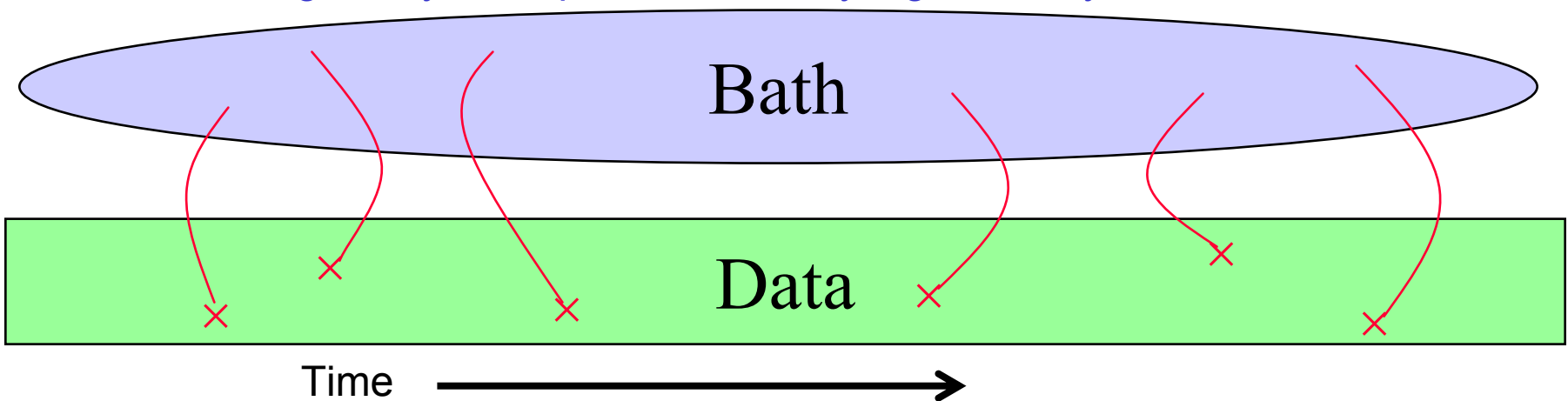
Non-Markovian noise with a *nonlocal* bath.

$$H = H_{System} + H_{Bath} + H_{System-Bath}$$

Quantum error correction works as long as the coupling of the system to the bath is *local* (only a few system qubits are jointly coupled to the bath) and *weak* (sum of terms, each with a small norm). Arbitrary (nonlocal) couplings among the bath degrees of freedom are allowed.



We find a rigorous upper bound on the norm of the sum of all “bad” diagrams (such that the faults are *not* sparsely distributed in spacetime). Actually, this works even for interactions among the system qubits that decay algebraically with distance...



Numerical value of the accuracy threshold

The value is of great practical importance, as it provides a target for the quantum hardware designer. The lower bound (for *adversarial independent stochastic noise*)

$$\varepsilon_0 > 7.55 \times 10^{-5}$$

(Aliferis, Cross, Svore) is the best that has so far been rigorously established.

However, Knill has estimated that, by using *error-detecting codes* for the the preparation of suitable encoded ancillas, the noise threshold can be much improved, **perhaps to better than 1%** (though at a higher overhead cost). We are working on making his estimate rigorous.

What if we (more realistically) assume that all gates are *local* in space, let's say in a **two-dimensional array**? Various estimates (e.g., by Svore, Terhal, and DiVincenzo) indicate that **the threshold worsens by less than a factor of 10**.

In the Hamiltonian model, faults cannot be treated probabilistically (different “fault paths” might interfere). What needs to be small is $\| H_{SB} \| t_0$ (the norm of the system-bath coupling responsible for the noise in a particular gate, times the time needed to execute the gate).

A realization of quantum error correction

J. Chiaverini *et al.*, [*Nature* **432**, 602-605 (2004)] implemented a three-qubit quantum repetition code using trapped ions. They prepared the encoded $|\bar{\psi}\rangle = a|\bar{0}\rangle + b|\bar{1}\rangle$ state, simulated noise that flips each qubit with probability ε , measured the error syndrome, and corrected the error.

The probability P of an encoded error was found to be

$$P = c + 2.6 |ab|^2 \varepsilon^2$$

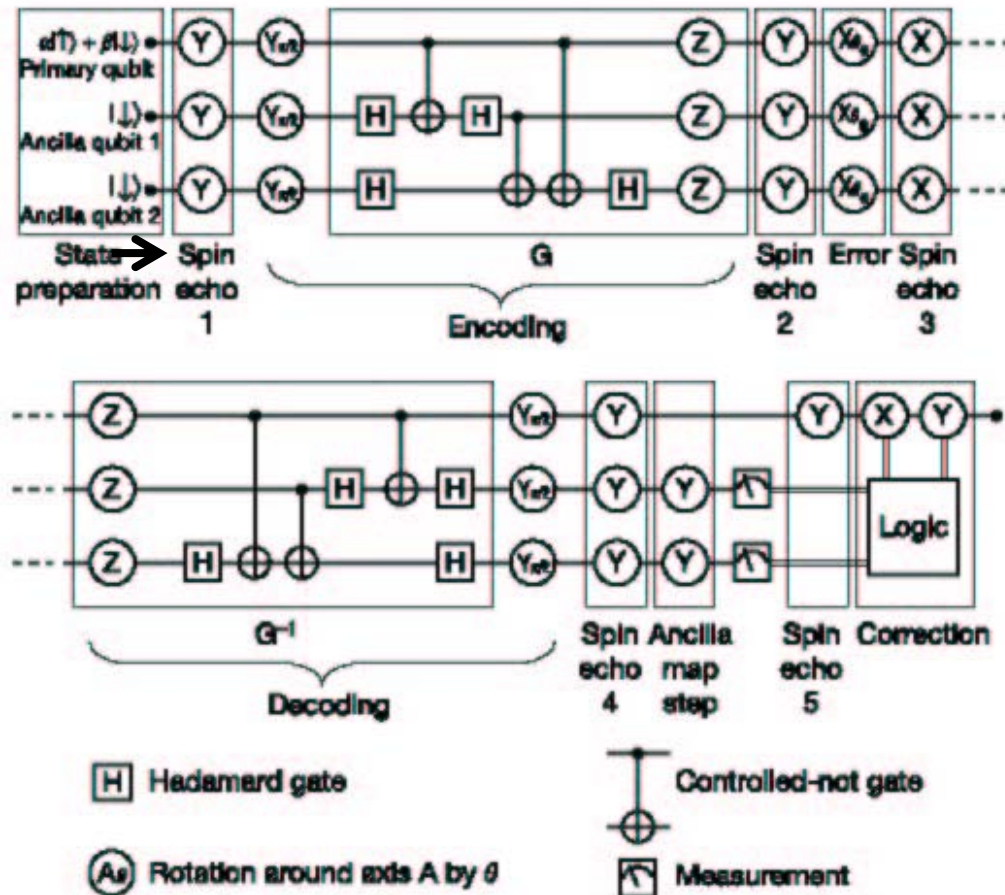
... i.e., quadratic in ε .

Realization of quantum error correction

J. Chiaverini¹, D. Leibfried¹, T. Schaetz^{1*}, M. D. Barrett^{1*}, R. B. Blakestad¹, J. Britton¹, W. M. Itano¹, J. D. Jost¹, E. Knill², C. Langer¹, R. Ozeri¹ & D. J. Wineland¹

¹Time and Frequency Division, ²Mathematical and Computational Sciences Division, NIST, Boulder, Colorado 80305, USA

* Present addresses: Max Planck Institut für Quantenoptik, Garching, Germany (T.S.); Physics Department, University of Otago, Dunedin, New Zealand (M.D.B.)



Quantum Hardware

Two-level ions in a Paul trap, coupled to “phonons.”

Two-level atoms in a high-finesse microcavity, strongly coupled to cavity modes of the electromagnetic field.

Charge in a Cooper-pair box; fluxons through a superconducting loop.

Electron spin (or charge) in quantum dots.

Cold atoms in optical lattices.

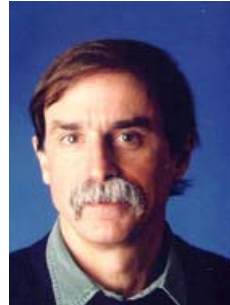
Nuclear spins in semiconductors, and in liquid state NMR.

Linear optics with efficient single-photon sources and detectors.

Electrons floating on liquid He, etc.



Kimble



Wineland



Devoret



Blatt

Quantum error correction

1. Error models and error correction
2. Quantum error-correcting codes
3. Stabilizer codes
4. 5-qubit code and 7-qubit code
5. Fault-tolerant quantum computation
6. Accuracy threshold

Quantum fault tolerance: Topological vs. “Brute force”

- Error correction and fault tolerance will be essential in the operation of large-scale quantum computers.
- The “brute force” approach to fault-tolerant quantum computing uses clever circuit design to overcome the deficiencies of quantum hardware. It works in principle, if the noise is weak enough, but achieving it in practice will be challenging.
- Topological quantum computing is a more elegant approach, in which the “hardware” is intrinsically robust due to principles of local quantum physics (if operated at a temperature well below the mass gap).
- The topological approach also looks daunting from the perspective of current technology (and might need assistance from “brute force” fault-tolerant constructions). But it is an attractive long-term path toward realistic quantum computing.
- The “standard” theory is pretty, but the theory of topological quantum computing seems especially rich, making connections with deep problems in mathematics and in condensed matter physics. And, it’s fun!