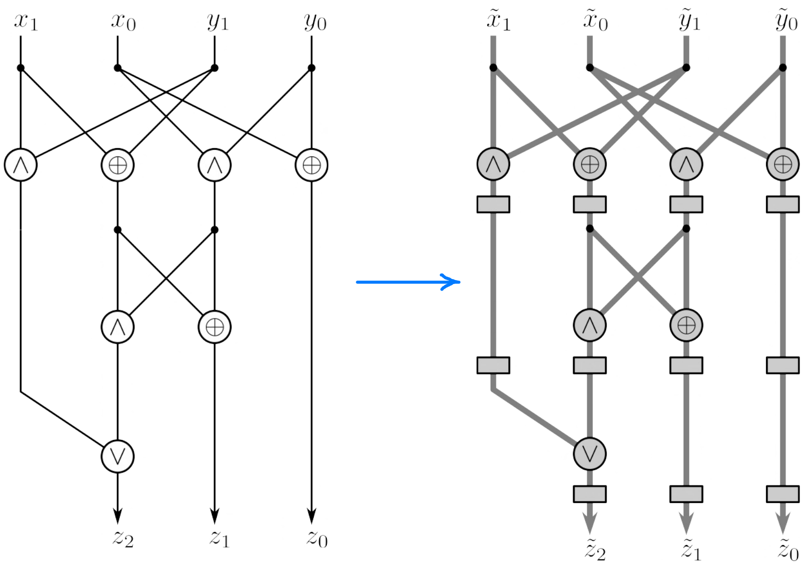
Fault-tolerant computation (introduction)

Fault-tolerant classical computation

Important requirement: Operations are done in parallel

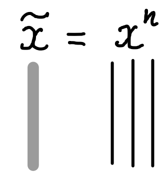
(because error correction should be performed periodically even on idle bits)

Simplifying assumption: Gates can be applied to arbitrary bit pairs (no locality constraints)

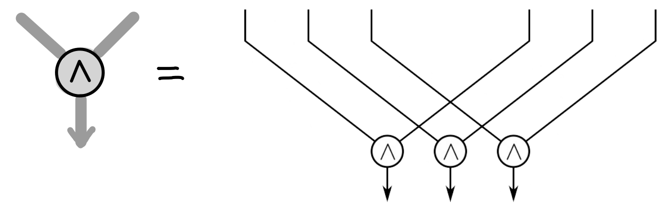


The fault-tolerant circuit consists of "gadgets": logical gates and error-correcting circuits

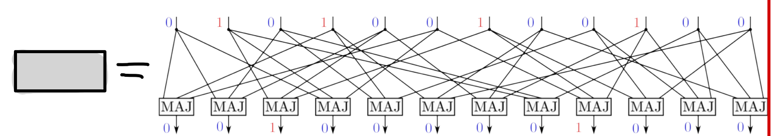
Repetition code:



Transversal implementation of logical gates



Error-correcting circuit



(may need to be applied several times in a row)

Probabilistic fault model: <u>Each physical gate produces a wrong result with probability p</u>

(independently of other gates)

Basic fact

Let $0<p<\varepsilon<1$. If faults occur with probability p,

then the probability to have more than εm faults in m gates is exponentially small in m.

$$\Pr[\#\text{ of faults} > \varepsilon m] = \sum_{s>\varepsilon m} \underbrace{\binom{m}{s}}_{2^{m H(s/m)}} p^s (1-p)^{m-s}$$

$$\sim \sum_{s>\varepsilon m} 2^{-m\, D(s/m \| p)} \sim 2^{-m\, D(\varepsilon \| p)}$$

because $D(q\|p)$ increases with q for $q>p$

We will use this to prove

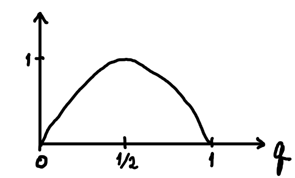
Threshold theorem

There exists some constant $\varepsilon>0$ with the following property:

If faults in physical gates or idle bits occur with probability $p<\varepsilon$,
then the overall error probability does not exceed $L\, \underbrace{e^{-\alpha n}}_{\text{logical fault rate}}$, where
L is the number of logical gates and $\alpha=\alpha(p)>0$.

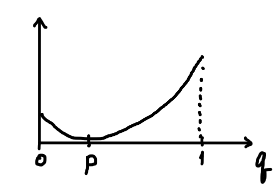
Entropy function:

$$H(q) = q \log_2 \frac{1}{q} + (1-q)\log_2 \frac{1}{1-q}$$



Relative entropy:

$$D(q\|p) = q \log_2 \frac{q}{p} + (1-q)\log_2 \frac{1-q}{1-p}$$



Combinatorial error model

Within each gadget, we classify fault patterns into "acceptable" and "bad".
(The probability of a bad pattern should be small, but this condition is not part of the formal model.)

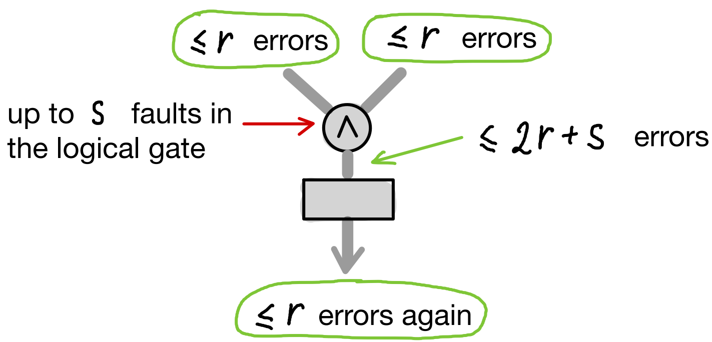
Transversal logical gate: s faults (in n simultaneously executed physical gates) are deemed acceptable

Error-correcting circuit: a fault pattern is acceptable if the action of the circuit satisfies the condition

$C(\ell, r)$: If at most l input bits are in error, then at most r output bits are in error $\left(r < \ell < \frac{n}{2}\right)$

This guarantees the correct operation, provided $2r + s \le \ell$

Specifically, there are at most r errors before each logical gate and after error correction



Reduction of the probabilistic model to the combinatorial model and a proof of the threshold theorem (outline)

0) Choose suitable constants ε (the admissible fraction of faulty gates), a, b,... that would guarantee the success of the subsequent steps for $n \to \infty$

(In practice, this is done after the analysis of those steps)

1) Construct an error-correcting circuit of size *m=O(n)* that satisfies the following conditions:

$A(a,b)$: If there are no faults and $l \le an$ input bits are in error, then at most bl output bits are in error

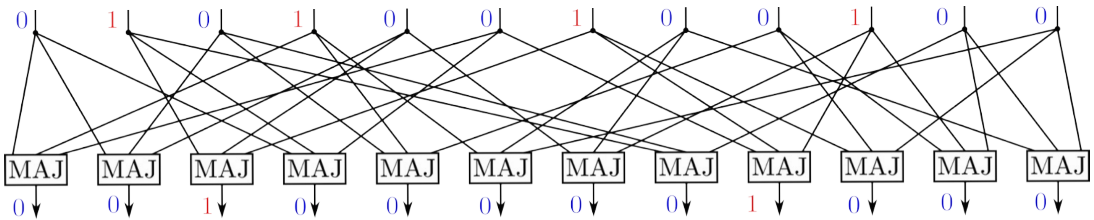
B: Each fault affects at most one output bit

2) By using the previous circuit a constant number of times and allowing εm faults in each run, obtain an error-correcting circuit satisfying the condition $C(a_+ n,\ a_- n)$

We also allow εn faults in each transversal logical gate.
According to an earlier argument, this scheme works if $\underline{2a_- + \varepsilon \le a_+}$

<u>Probability of a bad fault pattern: $\sim 2^{-m D(\varepsilon \| p)}$ or $2^{-n D(\varepsilon \| p)}$, provided $p < \varepsilon$.</u>

Constructing the error-correcting circuit



This picture shows a 3-voting circuit. We need a 5-voting circuit to satisfy the condition $A(a,b)$

A 5-voting circuit MAJ_Γ is defined by choosing for each output bit $j=1,..,n$ and index $t=1,..,5$ the corresponding input bit $\Gamma_{jt} \in \{1,..,n\}$.

Lemma

There exist some constants $0<a<1$, $0<b<1$ such that for all sufficiently large *n*, there is a Γ such that

$$\boxed{\text{If } \ |x| = \ell \le an, \ \text{ then } \ |MAJ_\Gamma(x)| \le b\,\ell} \qquad (*)$$

(equivalent to *A(a,b)*: *x* represents 0 in the repetition code with *l* errors)

$$x=\theta(X) \Leftrightarrow x_j = \begin{cases} 1, & \text{if } j\in X \\ 0, & \text{if } j\notin X \end{cases}$$

Proof

the set of input errors

subset of output errors

Γ fails to satisfy condition $(*)$ if and only if

$$\exists\ \ell \le an, \quad x=\theta(A) \text{ with } |A| = \ell, \quad B \text{ with } |B| = \underline{r := \lceil b\ell \rceil + 1}$$

such that $\forall j \in B, \quad MAJ_\Gamma(x)_j = 1$

$$\text{failure}(\Gamma) := \begin{cases} \exists\, \ell \le an, & x = \theta(A) \text{ with } |A| = \ell, \quad B \text{ with } |B| = r := \lceil b\ell \rceil + 1 \\ \text{such that} & \forall j \in B, \quad MAJ_\Gamma(x)_j = 1 \end{cases}$$

If Γ is chosen randomly (with the uniform probability), then

$$\Pr_\Gamma[\text{failure}(\Gamma)] \le \sum_{\ell=1}^{\lfloor an \rfloor} \sum_{A,B} \Pr_\Gamma[\text{failure}(\Gamma, A, B)]$$

3-voting?

$$\Pr_\Gamma[\text{failure}(\Gamma, A, B)] = \prod_{j\in B} \Pr_\Gamma[\text{ at least } 3 \text{ of } \Gamma_{j1}, \dots, \Gamma_{j5} \text{ belong to } A]$$

2

$$\le \left(\binom{5}{3}\left(\frac{\ell}{n}\right)^3\right)^r = 10^r (\ell/n)^{3r}$$

2r

$$\sum_{A,B} \Pr_\Gamma[\text{failure}(\Gamma, A, B)] \le \binom{n}{\ell}\binom{n}{r} 10^r (\ell/n)^{3r}$$

$$< \exp\Big(\ell \ln\frac{ne}{\ell} + \underbrace{r \ln \frac{ne}{r}}_{\ln\frac{n}{\ell} + \ln\frac{\ell}{r} + 1} + r\big(\ln 10 + 3 \ln \frac{\ell}{n}\big)\Big)$$

2

$$\binom{n}{m} = \frac{n\cdots(n-m+1)}{m!} \le \frac{n^m}{m!} < \left(n\frac{e}{m}\right)^m$$

1

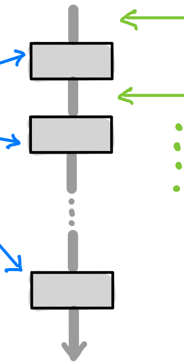
$$= \exp\left[-\ell\left[\left(-1 + 2\frac{r}{\ell}\right)\ln\frac{n}{\ell} - 1 - \left(\frac{r}{\ell}\ln\frac{\ell}{r} + 1 + \ln 10\right)\right]\right] \le \exp\left[-\ell\left[(2b-1)\ln\frac{1}{a} + f(b)\right]\right]$$

1

$\boxed{\text{choose } b = \frac{3}{4} \text{ and } a \text{ sufficiently small}}$

Using the error-correcting circuit repeatedly

up to $s = \varepsilon m$ faults in each error-correcting circuit, where $m \le cn$, $c = \text{const}$



up to $r_0 = an$ errors

up to r_1 errors

$$\boxed{r_j = \beta r_{j+1} + s}$$

This basically works if $r_0 \ge r_1 \ge \cdots$, but we will need a stronger condition

Steady state: $r_\infty = \beta r_\infty + s \quad \Rightarrow \quad r_\infty = \dfrac{s}{1-\beta} = \dfrac{c}{1-\beta}\varepsilon n$ $\qquad s = \varepsilon c n$

A finite number of repetitions will satisfy the condition $C(a_+ n, a_- n)$ if $\quad a_+ = a, \quad a_- > \dfrac{r_\infty}{n} = \dfrac{c}{1-\beta}\varepsilon$

Fixing the constants

$a > 0, \quad \beta = \frac{3}{4}, \quad c$ are determined by the 5-voting circuit

a_+, a_-, ε are constrained as follows: $\quad a_+ = a, \quad a_- > \dfrac{c}{1-\beta}\varepsilon, \quad 2a_- + \varepsilon \le a_+$

These constrains can be satisfied by choosing a sufficiently small ε : $\quad \boxed{\varepsilon < \left(1 + \frac{2c}{1-\beta}\right)^{-1} a}$

Some remarks about classical fault tolerance

-- With repeated error correction (perhaps using different Γs), 3-voting should work, but the proof will be more complex.

-- Our relatively simple proof gives a very low (i.e. too pessimistic) estimate of the threshold. For a realistic estimate, more complex arguments and/or numerical simulation are needed.

-- The exact threshold depends on the set of elementary gates used in the error-correcting circuit. Roughly, $\varepsilon \sim 0.1$

The quantum threshold is much lower, a few percent for fault-tolerant memory and something between 10^{-3} and 10^{-2} for universal computation.

Quantum fault-tolerance

Some convenient assumptions

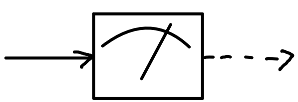
1) The set of elementary physical operations includes the initialization of a qubit in state $|0\rangle$

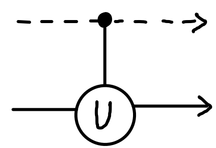
Not absolutely necessary, but error correction involves extracting (reduced) errors from the code and dumping them to the environment. This process is described by some non-unitary superoperator.

Extreme case of the amplitude damping channel: $T\rho = \mathrm{Tr}\,\rho \cdot |0\rangle\langle 0|$

2) We will assume that classical computation is reliable and fast

3) To utilize classical computation, we should use measurements and quantum gates with classical control. For example:





$U = \sigma^x, \sigma^z, H, K, \ldots$

Basic principles

-- Encode each qubit using an ECC (typically, a stabilizer code)

-- Use some logical gates that avoid exposing the logical qubit to the environment

-- Run an error-correcting circuit after each logical gate

Clifford + classical set of operations (not universal but sufficient for correcting errors in stabilizer codes)

1) Initialization of a qubit in state $|0\rangle$
2) Measurement in the $|0\rangle, |1\rangle$ basis
3) Logical operations with classical bits
4) Classically controlled Cllifford gates $H, K, CNOT$

Gottesman-Knill theorem: In the absence of other gates, the above operations can be simulated classically in polynomial time

Proof: At each step, the quantum state is a stabilizer state: $\boxed{S_j|\xi\rangle = |\xi\rangle, \quad S_j = \pm\sigma(f_j), \quad j=1,..,n}$

All elementary operations preserve this class of states. The only nontrivial operation is measurement. When we measure any operator of the form $S=\pm\sigma(g)$, there are two cases:

1) If $g \in D :=$ linear span of $f_1,..,f_n$ over $\mathbb{Z}_2$, then $S=(-1)^{\mu} S_1^{\gamma_1}\cdots S_j^{\gamma_j}$

$\Rightarrow$ the measurement outcome is μ

2) Otherwise the measurement outcome is random, and the stabilizer set is updated as follows:

Let $\gamma_j = \omega(f_j, g) \Leftrightarrow S_j S = (-1)^{\gamma_j} S S_j$. Since (1) does not hold, $\exists \ell$, $S_\ell S = -S S_\ell$

$$\boxed{\begin{aligned} &S_\ell \to S \\ &S_j \to S_j S_\ell^{\gamma_j} \quad \text{for } j \neq \ell \end{aligned}}$$

$$(S_j S_\ell^{\gamma_j}) S = S_j (-1)^{\gamma_j} S S_\ell^{\gamma_j} = S(S_j S_\ell^{\gamma_j})$$

Fundamental problems (compared to the classical case)

1) Not all logical gates can be realized transversally

-- Self-dual CSS codes allow for the transversal realization of Clifford gates

-- Some stabilizer codes allow for the transversal realization of some non-Clifford gates (at the expense of some Clifford gates)

-- Eastin-Knill theorem: A universal set of logical gated cannot be realized transversally on any code of distance greater than 1

(Will show this later)

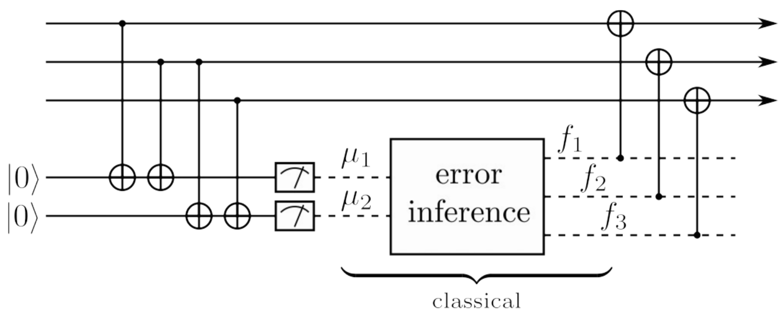
2) Error propagation: A fault in an error correcting circuit may affect multiple physical qubits

-- Mitigation is possible but not straightforward

(Will do it on the next lecture)

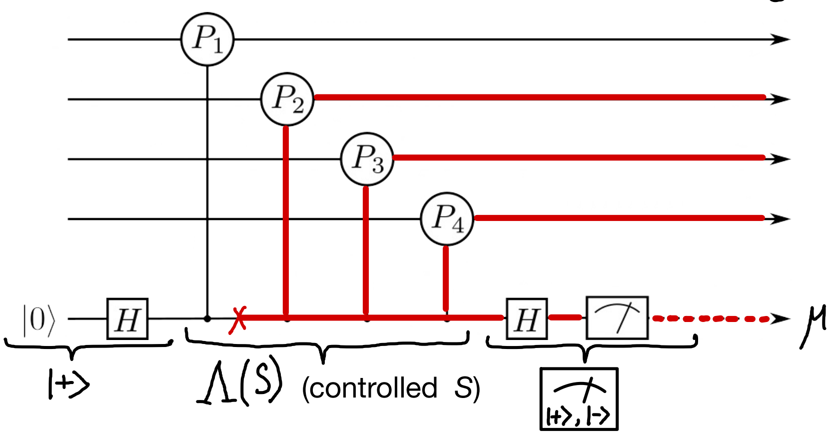
Problem with fault-tolerant syndrome measurement

Let us use the standard error correction method for a stabilizer code



This particular circuit corrects bit flips in the quantum repetition code. In general, we need more syndrome bits.

Consider the eigenvalue measurement for $S = P_1 \cdots P_m$ $\quad e.v.(S) = (-1)^{\mu}$



-- A fault may result in an incorrect measurement outcome μ . We can mitigate that by repeating the measurement 3 times.

-- A Z-error in the ancilla does not propagate

-- An X-error can propagate, affecting multiple code qubits.

$E = P_2 P_3 P_4 = P_1 S \equiv P_1.$ The worst case is when the error hits in the middle, e.g. $E = P_3 P_4 \equiv P_1 P_2$

Plan for the next few lectures

-- Quantum fault models

-- Fault-tolerant error correction (avoiding error propagation) to implement quantum memory

We will construct error-correcting circuits for stabilizer codes such that:

1) If the input is in a correctable state in relation to the logical qubit (e.g., for a distance 3 code, at most one physical input qubit is in error) and there are no faults in the operation of the circuit, then the error is actually corrected.

2) If the input is in the code subspace (i.e. no input qubits are in error) and at most one fault happens, then the output is correctable.

Consequence: Unless two faults occur in adjacent error correction cycles, the encoded information remains intact

Logical error rate: $\boxed{p_{logical} \sim O((n\,p)^2)}$

-- Threshold theorem for fault-tolerant memory and Clifford gates

-- Adding non-Clifford operations (in particular "magic ancillas", e.g. $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi/4}|1\rangle\right)$)
to achieve universality; implementing those operations fault-tolerantly