Introduction to Quantum Information Processing

QIC 710 / CS 768 / PH 767 / CO 681 / AM 871

Lecture 15 (2019)

Richard Cleve

QNC 3129

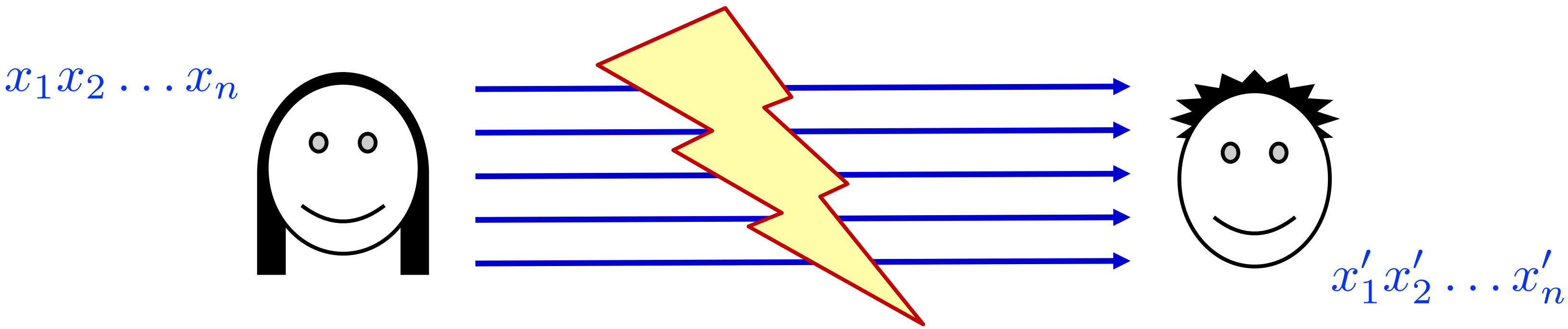
cleve@cs.uwaterloo.ca

Classical error correcting codes

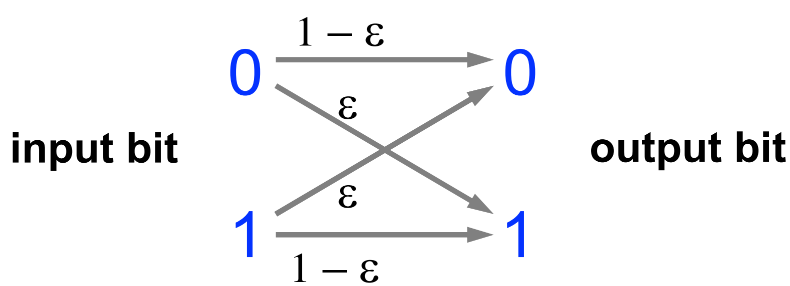
Classical error-correcting codes

Useful for:
- transmitting information through a ***noisy communication channel***
- storing information in a ***noisy storage medium***

Noisy means the states of bits can change (usually unpredictably)



One simple noise model is the ***binary symmetric channel,*** where each bit flips with probability ε (independently)



3-bit repetition code

One way of coping with this noisy channel:

- Encode each bit b as bbb
- Decode each received message $b_1b_2b_3$ as $\mathrm{majority}(b_1,b_2,b_3)$

Is this useful?

It reduces the effective error probability per data bit to $3\varepsilon^2 - 2\varepsilon^3$ why?

ε	$3\varepsilon^2 - 2\varepsilon^3$	error reduced by a factor of
0.10	0.009	11
0.01	0.0001	100
0.001	0.000001	1000

E.g., if $\varepsilon = 0.10$ and this is applied to n-bit messages then $< 1\%$ of the n bits will be in error (rather than 10%)

… but this is at a cost of tripling the message length ("rate" is $1/3$)

Repetition > 3 times: a smaller effective error probability; but worse rate

Can one do better?

For a given error rate ε, what's the "best" that can be done?

A rough "big picture" view (1)

An ***error-correcting code*** can be viewed as two mappings:

- Encoding function $E : \{0,1\}^n \longrightarrow \{0,1\}^m$
- Decoding function $D : \{0,1\}^m \longrightarrow \{0,1\}^n$

We assume some error model χ (including ε) is given to us by the hardware

Some considerations:

- Error probability of the code: probability that $D(\chi(E(x_1x_2\ldots x_n))) \neq x_1x_2\ldots x_n$
- Rate of the code: n/m

Amazing* fact: For any constant $\varepsilon < 1/2$, there is a constant rate sufficient to attain arbitrarily small error probability of the code

Message: 0100110101110101 any n-bit string

Encoding: 0110011010100101111101010111010 (m bits) *constant* expansion

Errors: 0100111010101101101101110110110 *constant fraction* of the bits

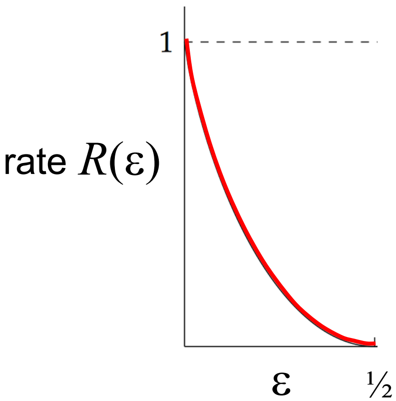
Decoding: 0100110101110101 *perfect* recovery of n-bit string with probability $\rightarrow 1$

---

* At least it's amazing the first time you think about it

A rough "big picture" view (2)

Rate as a function of noise level ε (assume binary symmetric channel)



$$
\begin{aligned}
R(\varepsilon) &= 1 - H(\varepsilon, 1-\varepsilon) \\
&= 1 - \big(-\varepsilon \log(\varepsilon) - (1-\varepsilon)\log(1-\varepsilon)\big)
\end{aligned}
$$

For noise level ε, can attain arbitrarily high recovery probability with rate arbitrarily close to $R(\varepsilon)$ (and exceeding $R(\varepsilon)$ is provably impossible)

Some further considerations:

- Block length n
  (as the recovery probability $\longrightarrow 1$, block length $\longrightarrow \infty$)
- Computational efficiency: how difficult it is to compute E and D
  (this is tricky, but polynomial-time—and practical—approaches exist)

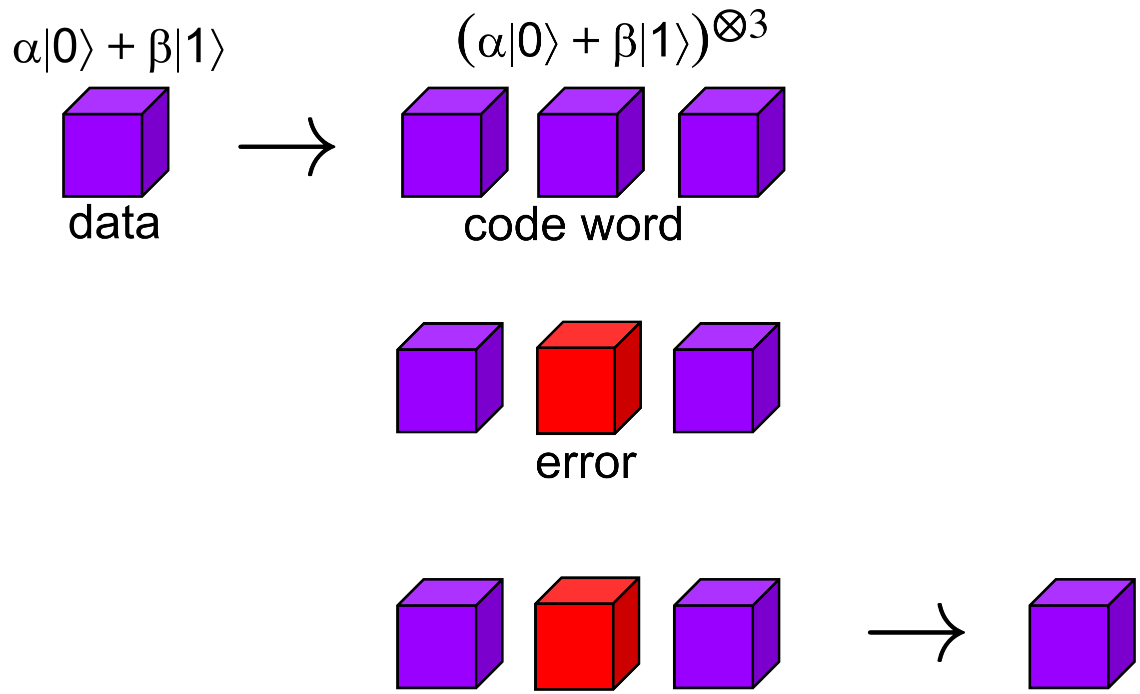
Question: What about ***quantum*** error correcting codes?

Quantum repetition code?

encoding

noise

decoding



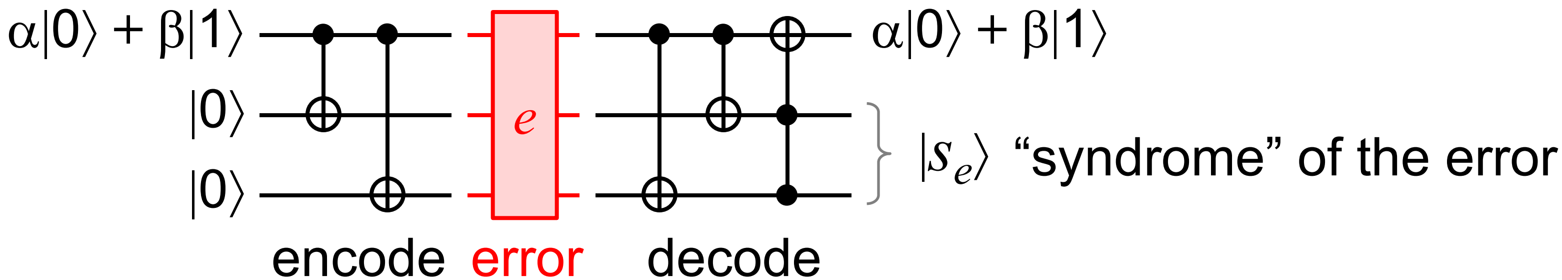
NOPE

This would violate the no-cloning theorem, for starters …

Shor’s 9-qubit code

3-qubit code for one X-error

The following 3-qubit quantum code protects against up to one error, ***if*** the error can only be a quantum bit-flip (an X operation)



Error can be any one of: $I\otimes I\otimes I$ $\quad X\otimes I\otimes I$ $\quad I\otimes X\otimes I$ $\quad I\otimes I\otimes X$

Corresponding syndrome: $|00\rangle$ $\quad |11\rangle$ $\quad |10\rangle$ $\quad |01\rangle$

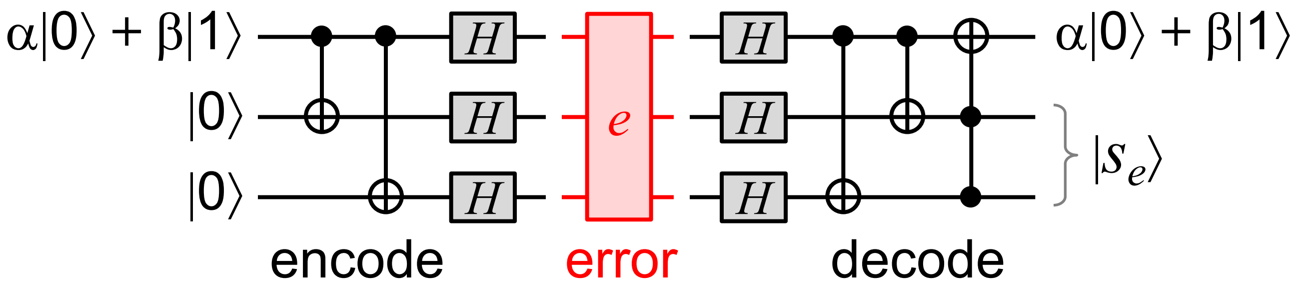
The essential property is that, in each case, the data $\alpha|0\rangle + \beta|1\rangle$ is shielded from (i.e., unaffected by) the error

What about Z errors?

This code leaves them intact: one Z error is equivalent to a Z operation on the original data

3-qubit code for one Z-error

Using the fact that $HZH = X$, one can adapt the previous code to protect against Z-errors instead of X-errors

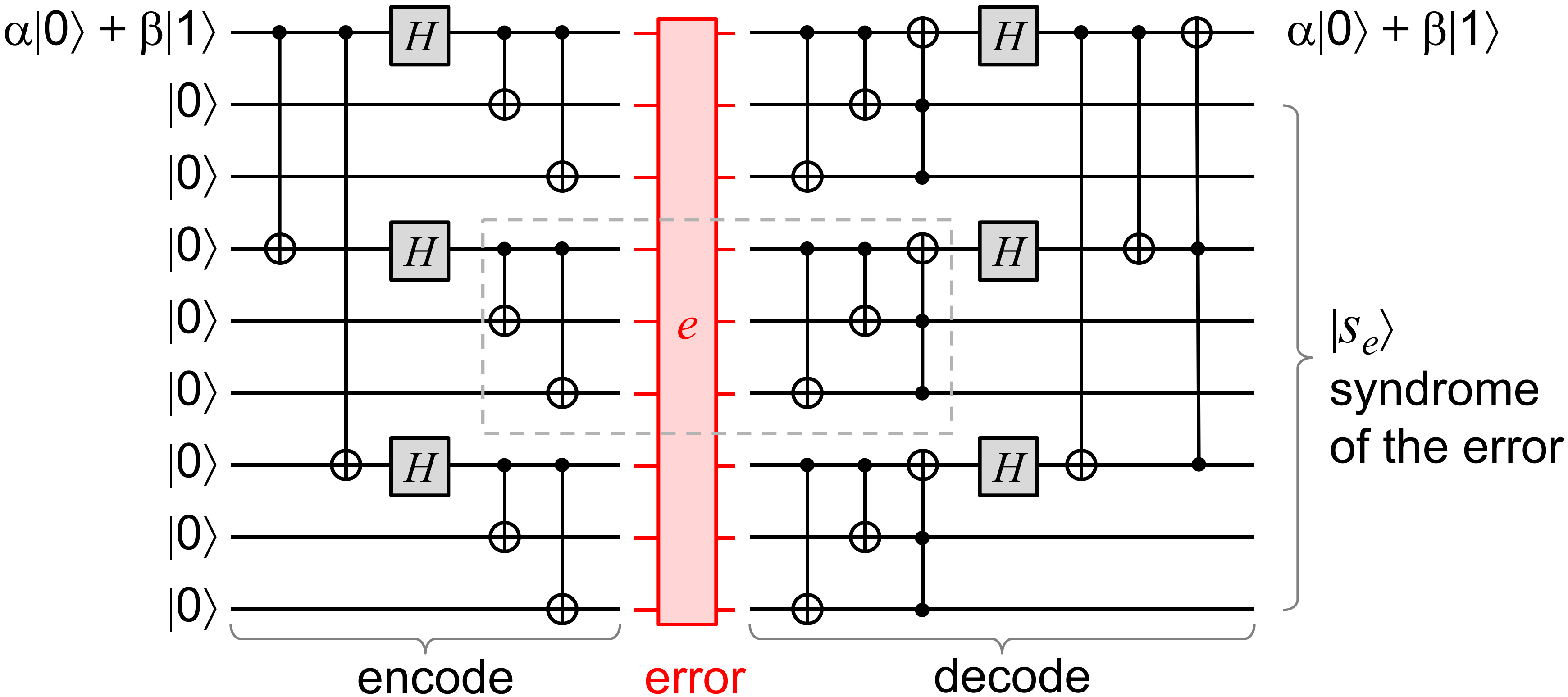


Error can be any one of: $I \otimes I \otimes I$ $\quad Z \otimes I \otimes I$ $\quad I \otimes Z \otimes I$ $\quad I \otimes I \otimes Z$

This code leaves X-errors intact

Is there a code that protects against errors that are arbitrary one-qubit unitaries?

Shor's 9-qubit quantum code



The "inner" part corrects any single-qubit X-error

The "outer" part corrects any single-qubit Z-error

Since $Y = iXZ$, single-qubit Y-errors are also corrected

Arbitrary one-qubit errors

Suppose that the error is some arbitrary one-qubit unitary operation U

Since there exist scalars $\lambda_1, \lambda_2, \lambda_3$ and λ_4, such that

$$U = \lambda_1 I + \lambda_2 X + \lambda_3 Y + \lambda_4 Z$$

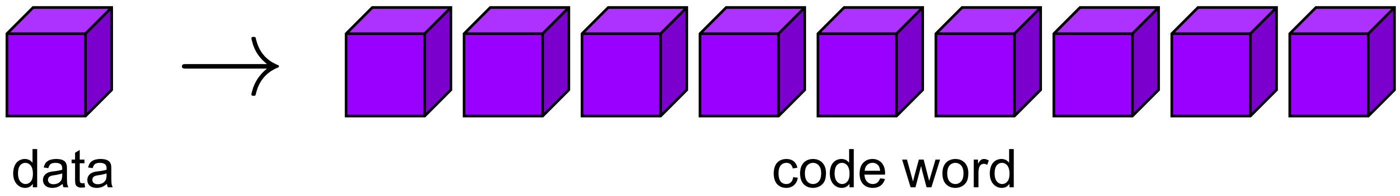
a straightforward calculation shows that, when a U-error occurs on the k^{th} qubit, the output of the decoding circuit is

$$(\alpha|0\rangle + \beta|1\rangle)(\lambda_1 |s_{e_1}\rangle + \lambda_2 |s_{e_2}\rangle + \lambda_3 |s_{e_3}\rangle + \lambda_4 |s_{e_4}\rangle)$$

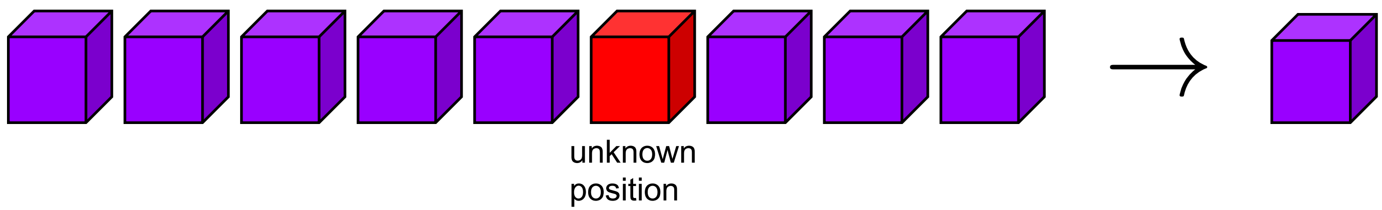
where s_{e_1}, s_{e_2}, s_{e_3} and s_{e_4} are the syndromes associated with the four errors (I, X, Y and Z) on the k^{th} qubit

Hence the code actually protects against ***any*** unitary one-qubit error (in fact the error can be any one-qubit quantum operation)

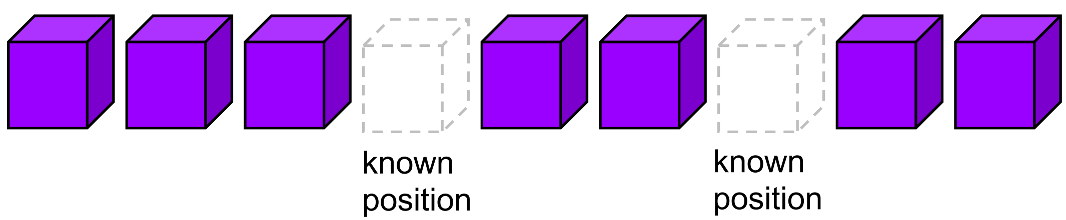
Summary of 9-qubit code



Can recover data from ***any*** 1 qubit error:



Moreover, it turns out the data can also be recovered data from ***any*** 2 qubit ***erasure*** error:



CSS Codes

Introduction to CSS codes

CSS codes (named after Calderbank, Shor, and Steane) are quantum error correcting codes that are constructed from classical error-correcting codes with certain properties

A classical ***linear*** code is one whose codewords (a subset of $\{0,1\}^m$) constitute a vector space

In other words, they are closed under linear combinations (here the underlying field is $\{0,1\}$ so the arithmetic is $\bmod 2$)

Examples of linear codes

For $m = 7$, consider these codes (which are linear):

basis for space

$C_2 = \{$0000000, 1010101, 0110011, 1100110,
0001111, 1011010, 0111100 , 1101001$\}$

$C_1 = \{$0000000, 1010101, 0110011, 1100110,
0001111, 1011010, 0111100, 1101001,
1111111, 0101010, 1001100, 0011001,
1110000, 0100101, 1000011 , 0010110$\}$

Note that the minimum Hamming distance between any pair of codewords is: 4 for C_2 and 3 for C_1

The minimum distances imply each code can correct one error

These two codes will serve as a running example of a CSS code

Orthogonal complement

For a linear code C, define its ***orthogonal complement*** as

$$C^{\perp} = \{w \in \{0,1\}^m : \text{for all } v \in C,\ w \cdot v = 0\}$$

(where $w \cdot v = \sum_{j=1}^{m} w_j v_j \bmod 2$, the "dot product")

Note that, in the previous example, $C_2^{\perp} = C_1$ and $C_1^{\perp} = C_2$

$C_2 = \{$0000000, 1010101, 0110011, 1100110,
0001111, 1011010, 0111100 , 1101001$\}$

$C_1 = \{$0000000, 1010101, 0110011, 1100110,
0001111, 1011010, 0111100, 1101001,
1111111, 0101010, 1001100, 0011001,
1110000, 0100101, 1000011 , 0010110$\}$

We will use some of these properties in the CSS construction

Encoding

Since , $|C_2| = 8$, it can encode 3 bits

To encode a 3-bit string $b = b_1 b_2 b_3$ in C_2, one multiplies [$b_1\ b_2\ b_3$] (on the right) by an appropriate 3×7 ***generator matrix***

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(generator for C_2)

Similarly, C_1 can encode 4 bits and an appropriate generator matrix for C_1 is

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(generator for C_1)

Parity check matrix

Every n-dimensional linear code can be alternately specified by its ***parity-check matrix*** M (m by m–n) such that:

$$v \in \{0,1\}^m \text{ is a codeword } v \text{ if and only if } vM = 0$$

Exercise: determine the parity check matrix for C_1 and for C_2

Error syndrome of an error vector

For any ***error-vector*** $e \in \{0,1\}^m$, the damaged data is $v+e$ (addition mod 2)

Note that $(v+e)M = eM$, and define the ***error syndrome*** of e as $s_e = eM$

Exercise: for C_1 and for C_2, work out the error syndromes for all $e \in \{0,1\}^m$, that correspond to single bit errors

Capability of a code: we are interested in sets of errors e with the property that each error e in the set can be uniquely identified (hence corrected) by s_e

For linear codes with maximum distance d, this includes the errors that are up to $\lfloor \frac{d-1}{2} \rfloor$ bit-flip errors

CSS construction

Let $C_2 \subset C_1 \subset \{0,1\}^m$ be two classical linear codes such that:

- The minimum distance of C_1 is d
- $C_2{}^{\perp} \subseteq C_1$

Let $r = \dim(C_1) - \dim(C_2) = \log(|C_1|/|C_2|)$

Then the resulting ***CSS code*** maps each r-qubit basis state $|b_1 ... b_r\rangle$ to some "coset state" of the form

$$\frac{1}{\sqrt{|C_2|}} \sum_{v \in C_2} |v + w\rangle$$

where $w = w_1 ... w_m$ is a linear function of $b_1 ... b_r$ chosen so that each value of w occurs in a unique coset in the quotient space C_1/C_2

The resulting quantum code can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors

Example of CSS construction

For $m = 7$, for the C_1 and C_2 in the previous example we obtain these basis codewords:

$$|0_L\rangle = |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle$$
$$+ |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle$$

$$|1_L\rangle = |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle$$
$$+ |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle$$

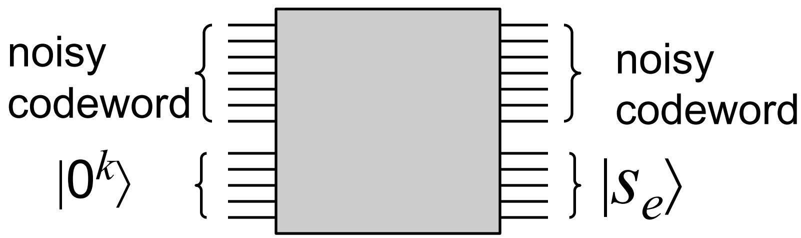
and the linear function mapping b to w can be given as $w = b \cdot G$

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{G}$$

There is a quantum circuit that transforms between $(\alpha|0\rangle + \beta|1\rangle)|0^{m-1}\rangle$ and $\alpha|0_L\rangle + \beta|1_L\rangle$

CSS error correction (1)

Using the error-correcting properties of C_1, one can construct a quantum circuit consisting of CNOT gates that computes the syndrome s for any combination of up to d X-errors in the following sense



Once the syndrome s_e, has been computed, the X-errors can be determined and undone

What about Z-errors?

The above procedure for correcting X-errors has no effect on any Z-errors that occur

CSS error correction (2)

Note that any Z-error is an X-error in the Hadamard basis

Changing to Hadamard basis is like changing from C_2 to C_1 since

$$H^{\otimes m}\left(\sum_{v\in C_2}|v\rangle\right) = \sum_{u\in C_2^{\perp}}|u\rangle \quad \text{and} \quad H^{\otimes m}\left(\sum_{v\in C_2}|v+w\rangle\right) = \sum_{u\in C_2^{\perp}}(-1)^{w\cdot u}|u\rangle$$

Applying $H^{\otimes n}$ to a superposition of basis codewords yields

$$H^{\otimes m}\left(\sum_{b\in\{0,1\}^r}\alpha_b \sum_{v\in C_2}|v+b\cdot G\rangle\right) = \sum_{b\in\{0,1\}^r}\alpha_b \sum_{u\in C_2^{\perp}}(-1)^{b\cdot G\cdot u}|u\rangle = \sum_{u\in C_2^{\perp}}\sum_{b\in\{0,1\}^r}\alpha_b(-1)^{b\cdot G\cdot u}|u\rangle$$

Note that, since $C_2^{\perp}\subseteq C_1$, this is a superposition of elements of C_1, so we can use the error-correcting properties of C_1 to correct

Then, applying Hadamards again, restores the codeword with up to d Z-errors corrected

CSS error correction (3)

The two procedures together correct up to d errors that can each be either an X-error or a Z-error — and, since $Y = iXZ$, they can also be Y-errors

From this, a simple linearity argument can be applied to show that the code corrects up to d arbitrary errors (that is, the error can be any quantum operation performed on up to d qubits)

Since there exist pretty good *classical* linear codes that satisfy the properties needed for the CSS construction, this approach can be used to construct pretty good *quantum* codes

In our running example, we obtain a 7-qubit quantum code for 1 qubit, that protects against one error (beating the Shor 9-qubit code)

Depolarizing channel

Roughly speaking, it's a quantum analogue of the binary symmetric channel

Each qubit incurs the following type of error ($0 \le \varepsilon \le \frac{3}{4}$):

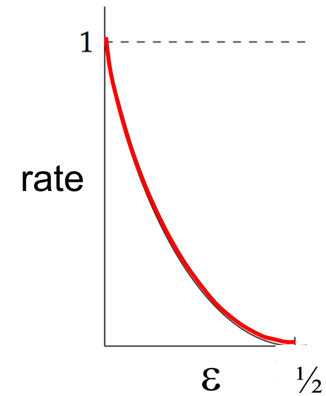
$$\begin{cases} I & \text{with probability } 1-\varepsilon & \text{(no error)} \\ X & \text{with probability } \varepsilon/3 & \text{(bit flip)} \\ Z & \text{with probability } \varepsilon/3 & \text{(phase flip)} \\ Y & \text{with probability } \varepsilon/3 & \text{(both)} \end{cases}$$

For any noise rate ε below some constant, there are codes with:

- constant rate $r = n/m$
- error probability of code $\rightarrow 0$ as $n \rightarrow \infty$

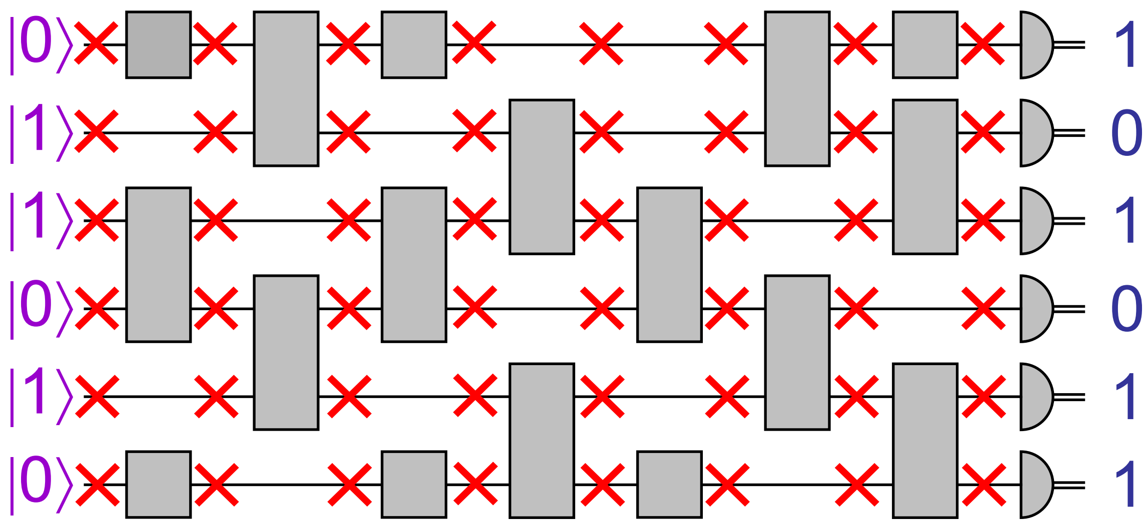
quantum depolarizing channel

classical binary symmetric channel



Brief remarks about fault-tolerant computing

A simple error model



At each qubit there is an × error per unit of time, that denotes the following noise:

$$\begin{cases} I & \text{with probability } 1-\varepsilon \\ X & \text{with probability } \varepsilon/3 \\ Y & \text{with probability } \varepsilon/3 \\ Z & \text{with probability } \varepsilon/3 \end{cases}$$

Threshold theorem

If ε is very small then this is okay — a computation of size* less than $1/(10\varepsilon)$ will still succeed most of the time

But, for every ***constant*** value of ε, the size of the maximum computation possible in this manner is constant

Threshold theorem:

There's a ***fixed*** constant $\varepsilon_0 > 0$ such that a circuit of ***any*** size T can be translated into a circuit of size $O(T \log^c(T))$ that is robust against the error model with parameter $\varepsilon \leq \varepsilon_0$

(The proof is omitted here)

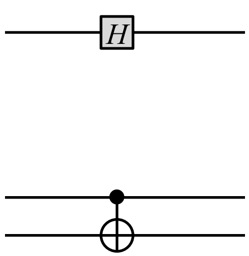
* where size = (# qubits)x(# time steps)

Comments about the threshold theorem

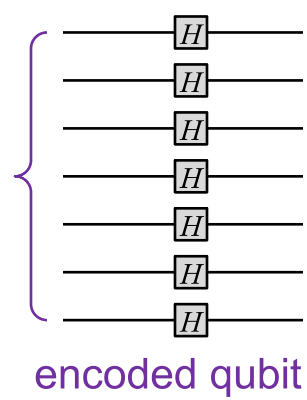
Idea is to use a quantum error-correcting code at the start and then perform all the gates ***on the encoded data***

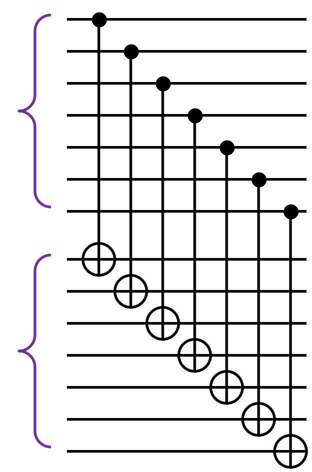
At regular intervals, an error-correction procedure is performed, very carefully, since these operations are also subject to errors!

The 7-qubit CSS code has some nice properties that enable some (not all) gates to be directly performed on the encoded data: H and $CNOT$ gates act "transversally" in the sense that:



are equivalent to





Also, codes applied recursively become stronger