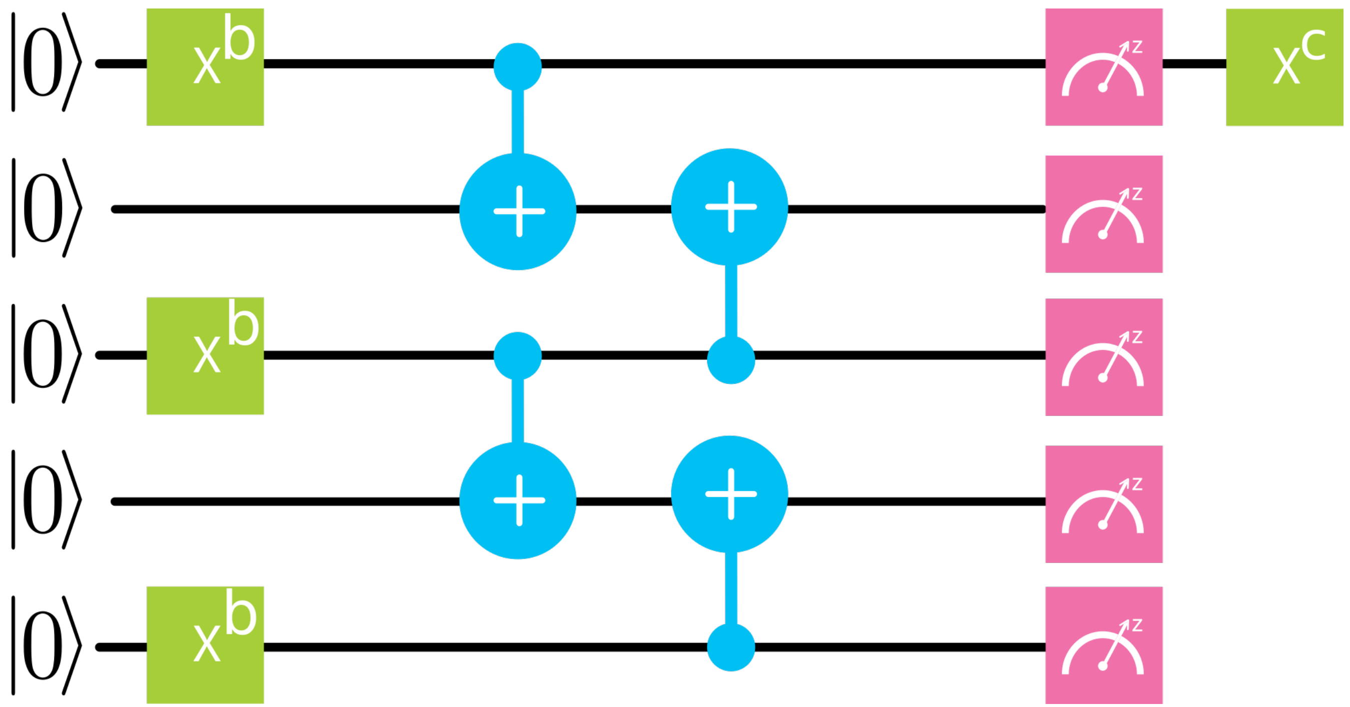
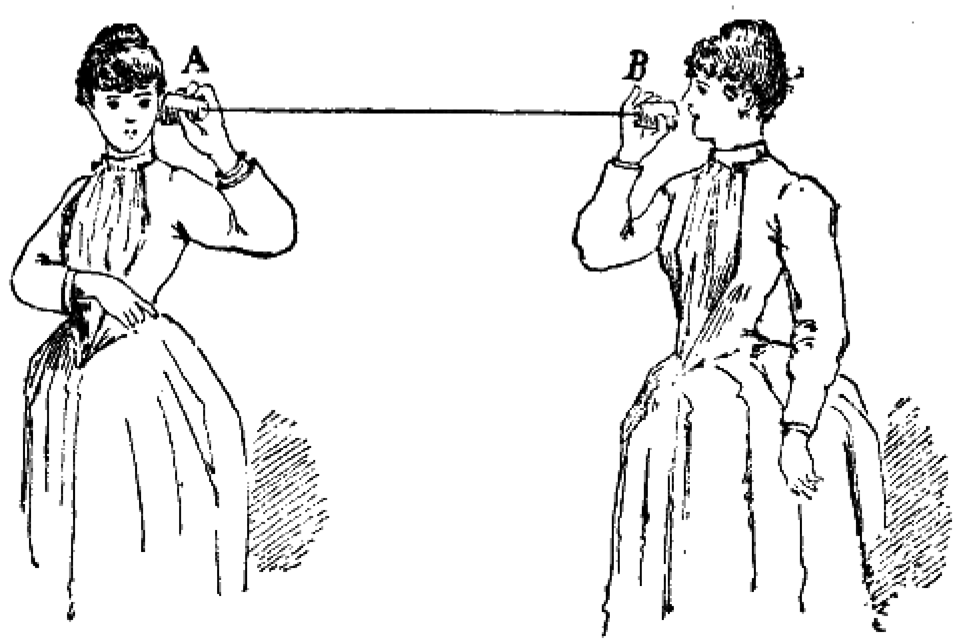
Introduction to the Repetition Code

Dr James R. Wootton
University of Basel



UNI BASEL
QSIT Quantum Science and Technology
National Centre of Competence in Research

What is error correction?



- Suppose you are talking on the phone
- You need to answer a question with ‘yes’ or ‘no’

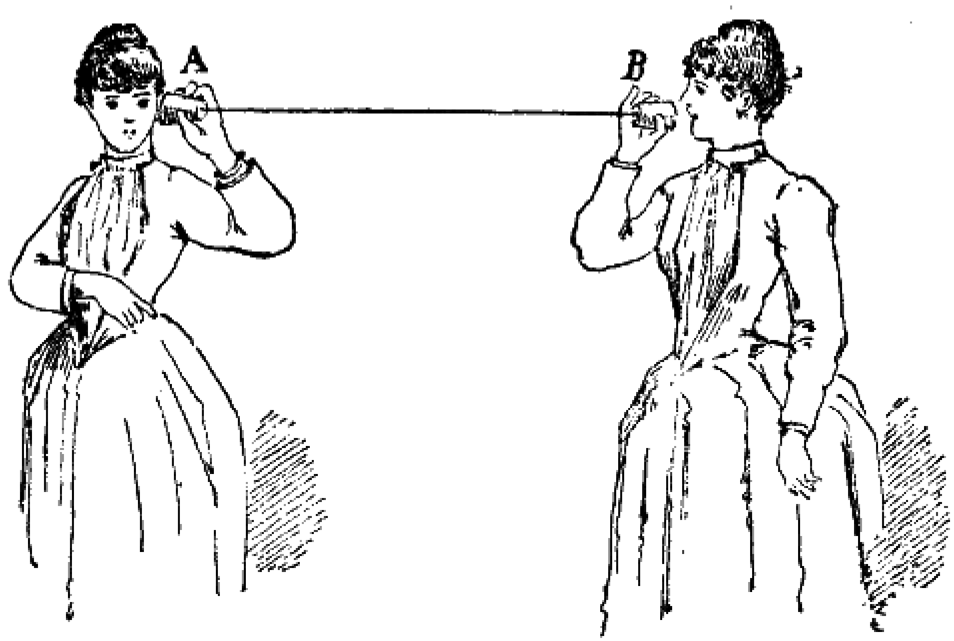
- How likely are you to be misunderstood? Is it a noisy line?

  p = probability that ‘no’ sounds like ‘yes’, and vice-versa

- How much do you care about being misunderstood?

  P_a = maximum acceptable error probability

What is error correction?



- Usually $p \ll P_a$, so we don't need to worry
- What if we are being asked life-or-death questions over a noisy line?
- How can we make ourselves understood?

The Repetition Code

- We could repeat ourselves
- A torrent of ‘no’s will sound like you mean ‘no’
- So would lots of ‘no’s with a few apparent ‘yes’s thrown in
- Message becomes tolerant to small faults

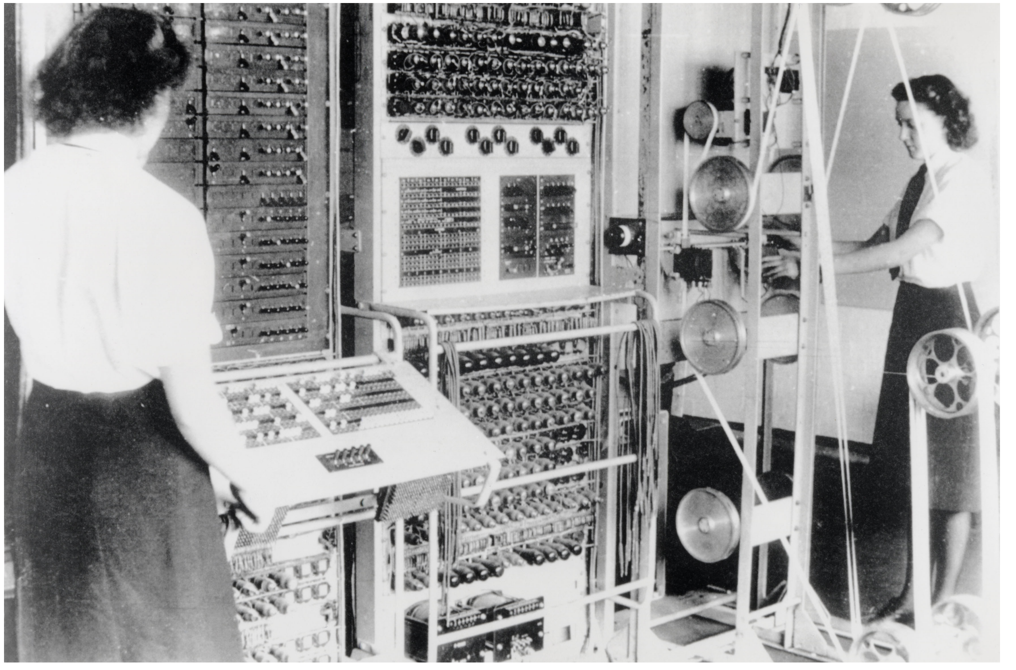
The Repetition Code

- Receiver will interpret message using *majority voting*
- If they hear mostly ‘no’, they’ll think you are saying ‘no’
- If they hear mostly ‘yes’, they’ll think you are saying ‘yes’
- You will only be misunderstood if noise causes the majority to flip
- For *d* repetitions

$$P = \sum_{n=d/2}^{d} \binom{d}{n} p^n (1-p)^{d-n} \sim \left(\frac{p}{1-p}\right)^{d/2}$$

- Probably decays exponentially with *d*
- We can ensure that $P \ll P_a$ for any *p*, just by using enough repetitions

Encoding and decoding



- This simple example contains the basics error correction
  - **Input:** Some information to protect from errors
  - **Encoding:** Input is altered to make it fault tolerant
  - **Transmission:** Noise affects the encoded message, altering it
  - **Decoding:** Most likely input is deduced, given the message received

Storage

- So far we've been focussing on *sending* information. What about storing it?

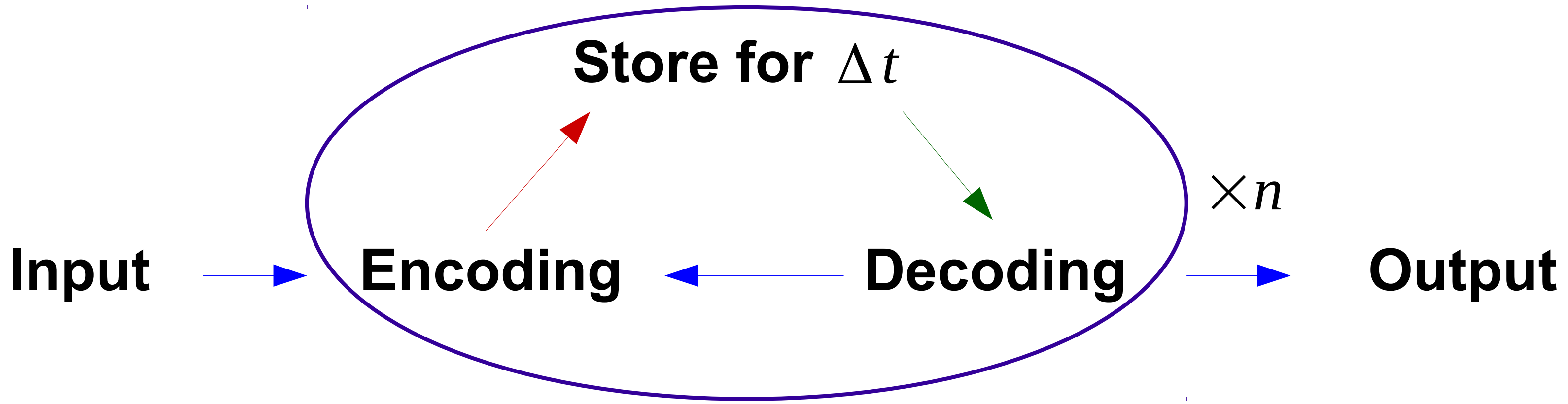
- The probability for errors increases with time

$$p(t) \rightarrow 0.5, \text{ as } t \rightarrow \infty \qquad \therefore \quad \left(\frac{p}{1-p}\right)^{d/2} = O(1)$$

- How can we store information for indefinitely long times?

Storage

- Just keep decoding and encoding



- To store for a time *T*, use $n = T/\Delta t$ rounds

$$P(T) < n\ P(\Delta t) \sim \frac{T}{\Delta t}\left(\frac{p(\Delta t)}{1-p(\Delta t)}\right)^{d/2}$$

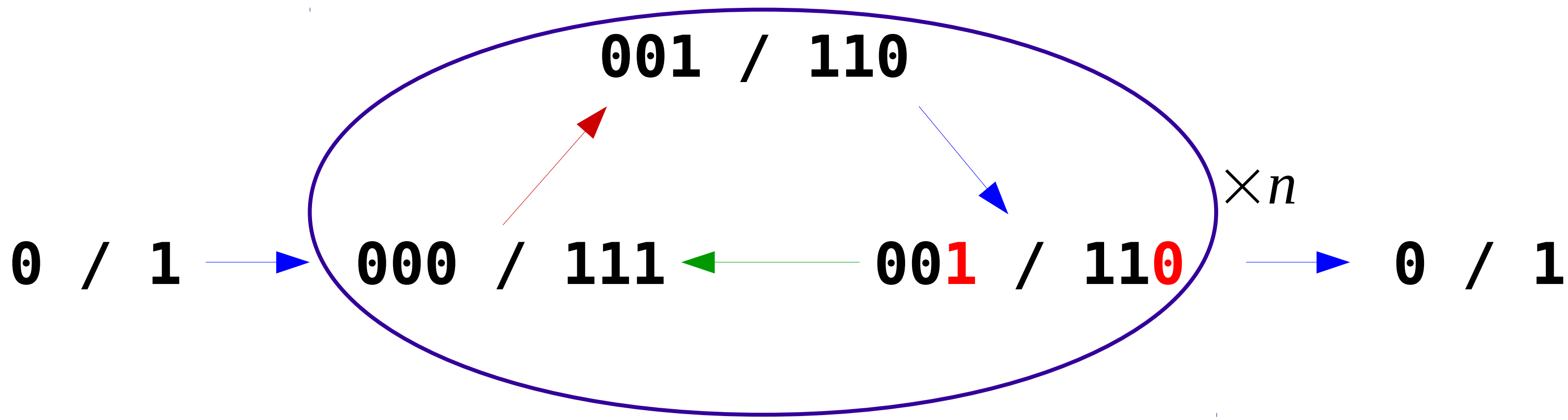
- Exponential decay depends on error probability for each round

- Lifetime increases exponentially with *d*

$$T_{\max} > P_a\ \Delta t\left(\frac{1-p(\Delta t)}{p(\Delta t)}\right)^{d/2}$$

What about qubits?

- This process works fine with bits



- But for qubits we might store a superposition state

$$\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|000\rangle+|111\rangle\right)$$

- Decoding requires measurement, which collapses superposition

- How do we extract the information we want (effects of noise) without getting information we don't (measurement of stored qubit)?

What about qubits?

- Even with bits, we don’t actually need the values to decode
- Just need domains walls between errors and non-errors

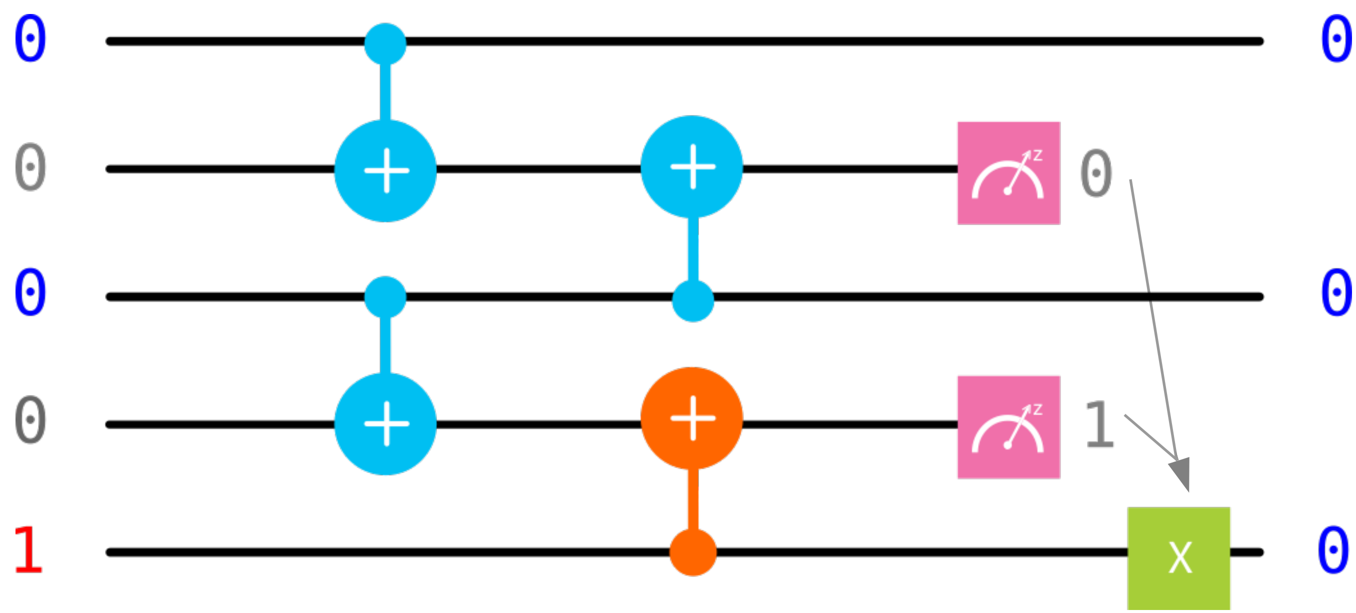
```
0 0 0 0 0 0 0 0 0 0 0        1 1 1 1 1 1 1 1 1 1 1

0 1 1 1 0 0 0 0 1 0 0        1 0 0 0 1 1 1 1 0 1 1

0≠1 1 1≠0 0 0 0≠1≠0 0        1≠0 0 0≠1 1 1 1≠0≠1 1
```

- NOT gate can be applied to minority domain to correct
- So how to measure the domain walls?

Quantum repetition code



- Can be done with the controlled-NOT gate

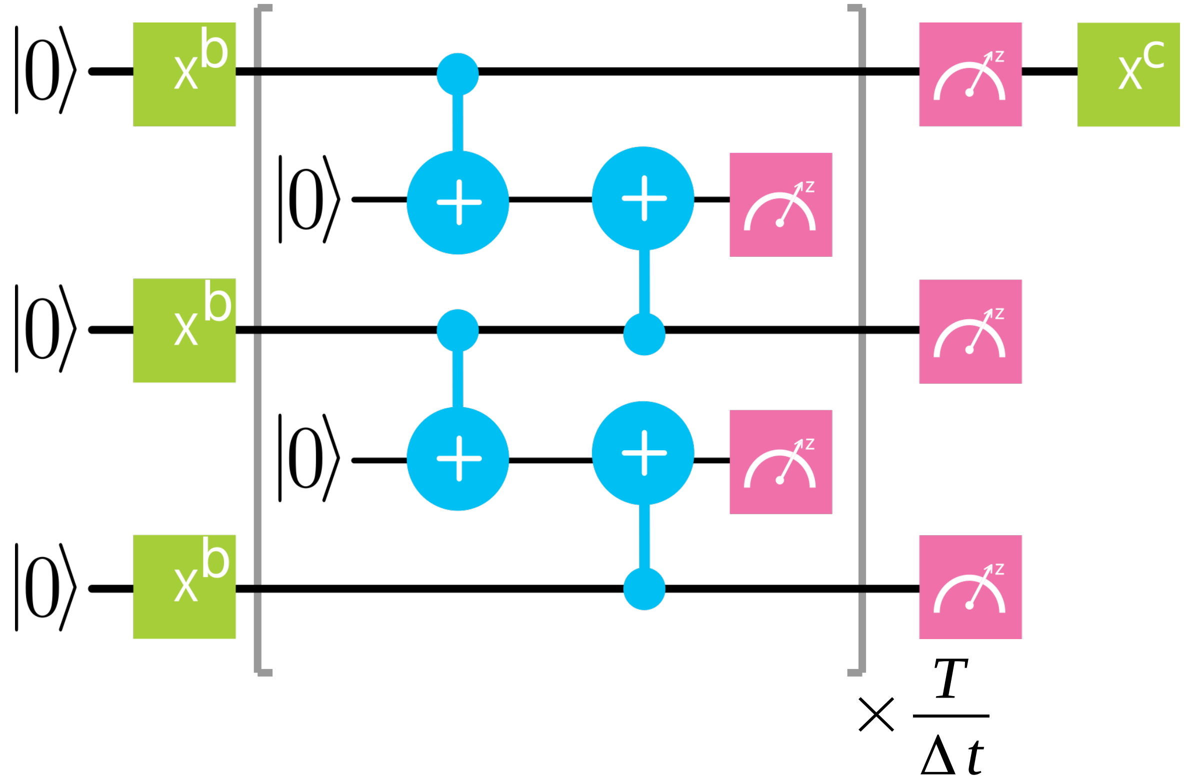
$$\mathrm{cx}|00\rangle = |00\rangle$$
$$\mathrm{cx}|01\rangle = |01\rangle$$
$$\mathrm{cx}|10\rangle = |11\rangle$$
$$\mathrm{cx}|11\rangle = |10\rangle$$

- Does nothing when *control qubit* is in state 0,
  Applies an X to *target qubit* when control is in state 1

$$\mathrm{cx}(1,2)\ \mathrm{cx}(3,2)\ |x,\ 0,\ y\rangle = |x,\ x\oplus y,\ y\rangle$$

- Corresponds to measuring the observable $\sigma_z^j \sigma_z^{j+1}$

Quantum repetition code



- By repeating this process, arbitary quantum states can be protected from bit flip noise (random application of σ_x)

$$a|0\rangle + b|1\rangle \rightarrow a|000\rangle + b|111\rangle \rightarrow a|0\rangle + b|1\rangle$$

- But they become even more susceptible to dephasing

$$a|000\rangle - b|111\rangle \quad \begin{aligned} &= \sigma_z^1 \left(a|000\rangle + b|111\rangle \right) \\ &= \sigma_z^2 \left(a|000\rangle + b|111\rangle \right) \\ &= \sigma_z^3 \left(a|000\rangle + b|111\rangle \right) \end{aligned}$$

$$P_x \sim \left(\frac{p_x}{1 - p_x} \right)^{d/2}$$

$$P_z \sim d \ \ p_z$$

Towards a better quantum code

- How does the repetition code protect against bit flip noise (σ_x) ?
  - An isolated σ^x creates a pair of *defects* — 0 0 0≠1≠0 0 0 0 0 0 0
  - Further σ^xs can move the defects — 0 0 0≠1 1≠0 0 0 0 0 0
  - Or create new pairs of defects — 0 0 0≠1 1≠0≠1≠0 0 0 0
  - Or annihilate pairs of defects — 0 0 0≠1 1 1 1≠0 0 0 0
  - A distance of >d/2 is needed for a logical error — 0 0 0≠1 1 1 1 1 1≠0 0
  - Then decoding will complete the job, pulling the defects off the ends — ≠1 1 1 1 1 1 1 1 1 1 1≠
- The code is like a ‘universe’ in which the defects are its particles
- Bit flips create and manipulate these particles, but only large scale effects can cause a logical error

Towards a better quantum code

- Why doesn't the repetition code protect against phase flip noise (σ_z) ?
- Measurement is too easy, even when the information is encoded

0 0 0 0 0 0 0 0 0 0 0 **1 1 1 1 1 1 1 1 1 1 1**

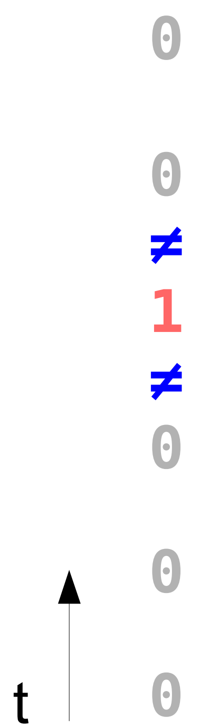
- Once errors are removed, a quick peek at any qubit reveals the stored information
- If it is easy for us to see, it is easy for the environment to dephase
- Consider measuring in the X basis instead

$$|\pm\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \pm |1\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|000\rangle \pm |111\rangle\right)$$

- Requires multi qubit process for the encoded states

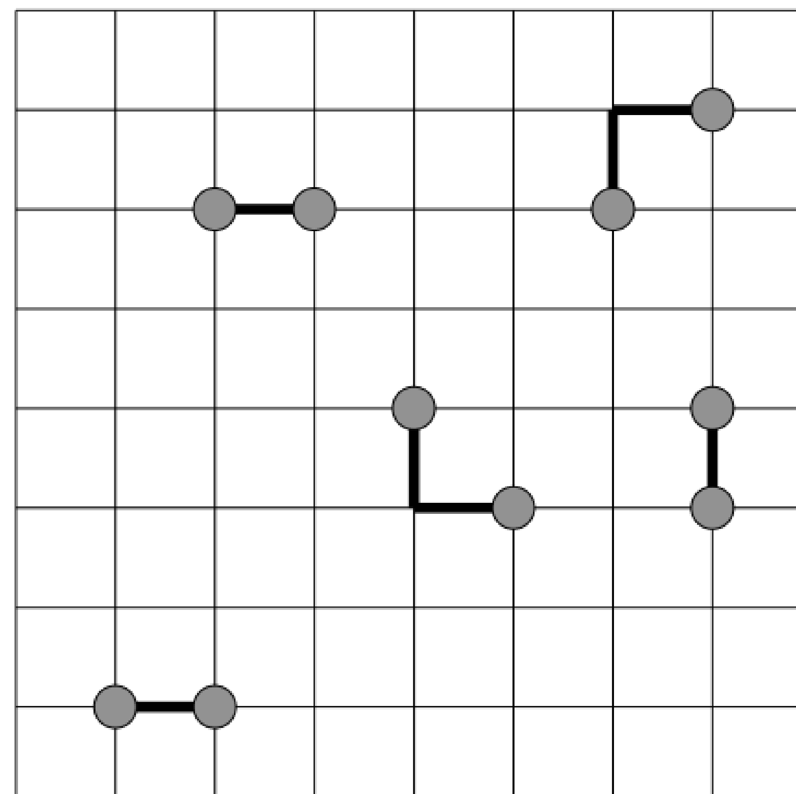
Imperfect measurement

- What about imperfect measurements throughout?
- Consider a measurement of a single qubit that lies with prob. *P*, but doesn't disturb the measured qubit (beyond projection)
- How do we extract information correctly? Repetition!
- Lies create pairs of defects in the time direction



Imperfect measurement

- Combine this with the repetition code:
  - Defects = changes in ancilla measurement result
  - Bit flips create space-like separated defect pairs
  - Lies create time-like separate defect pairs
  - Combinations create combinations

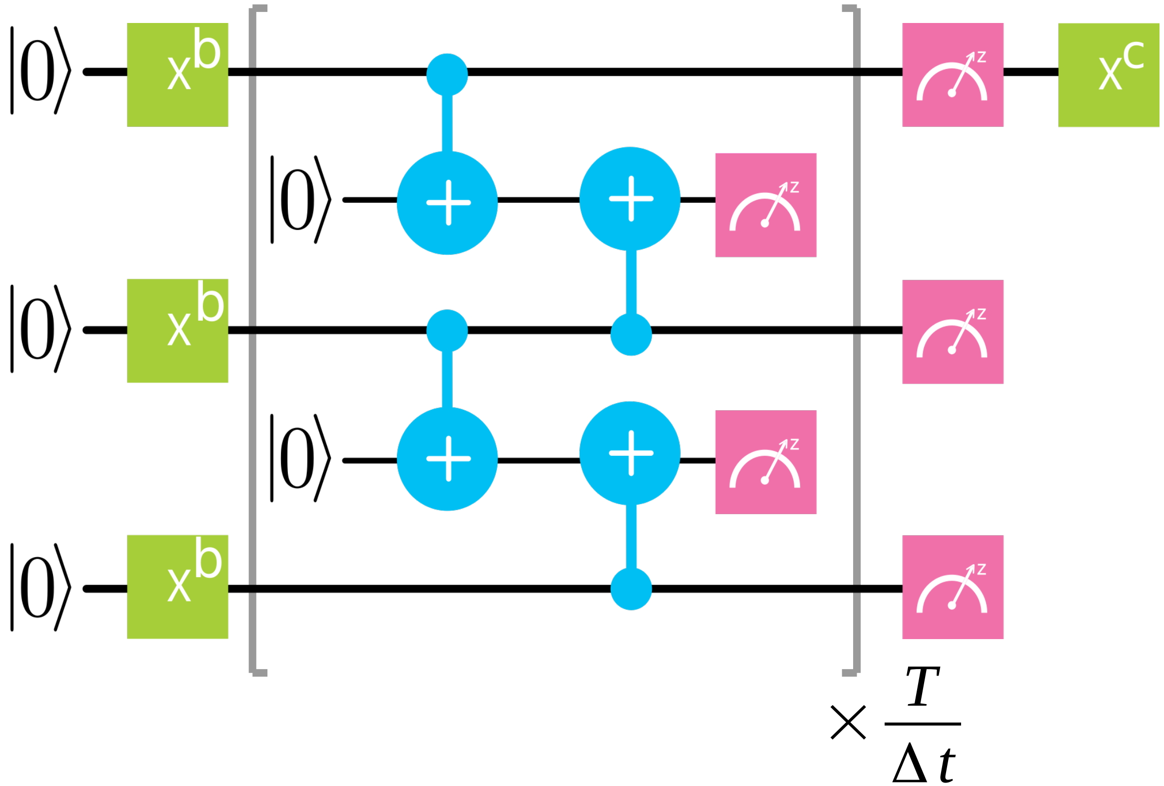


arXiv:quant-ph/0110143

- Noisy measurements just increase the space of the ‘universe’ by 1 dimension

Repetition code experiments

- We can run repetition codes with current devices

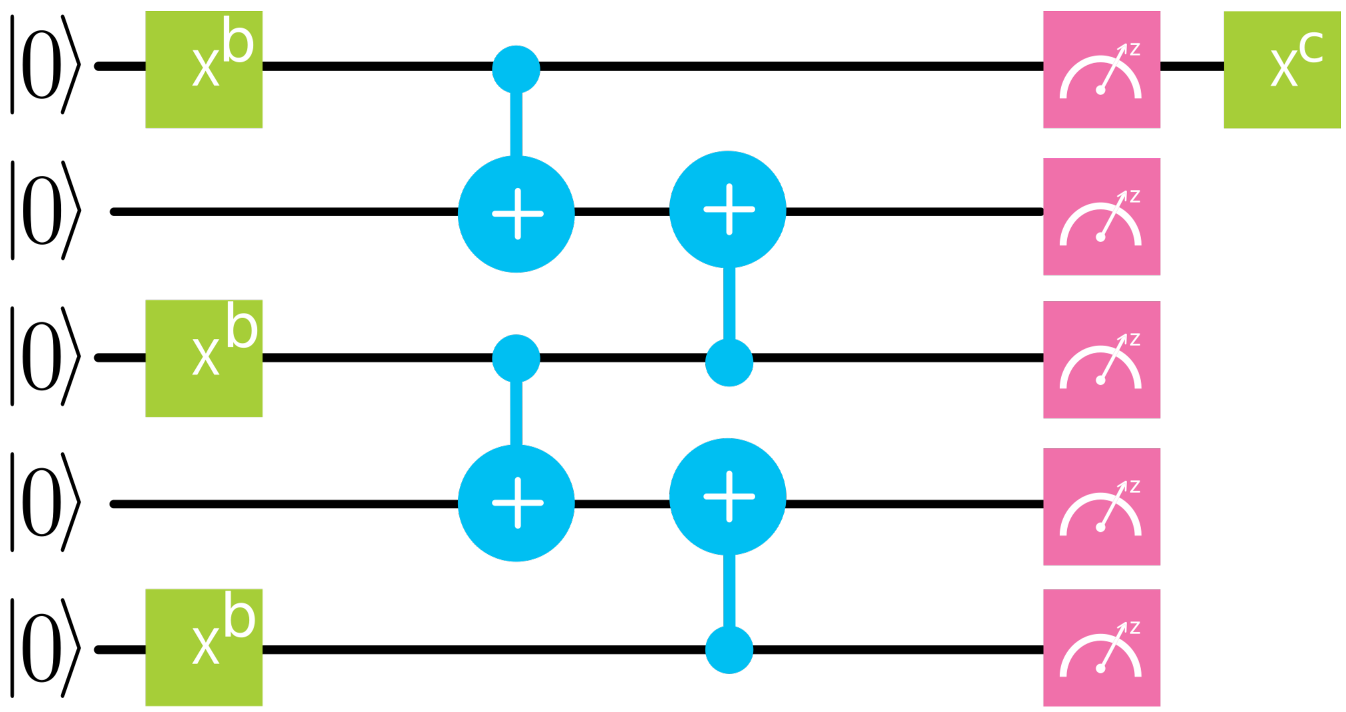


- An experiment has been done with limited size, but many rounds
  J. Kelly, et al., Nature 519, 66–69 (2015)

- Let's look at the other extreme: large size but a single round

Repetition code experiments

- This means we simply
  - Prepare a bit state *b*
  - Perform syndrome measurements, moving error info to ancillas
  - Measure everything, and try to work out what was encoded



- From many samples, and different encoded states, we can calculate logical error probabilities (*P*)

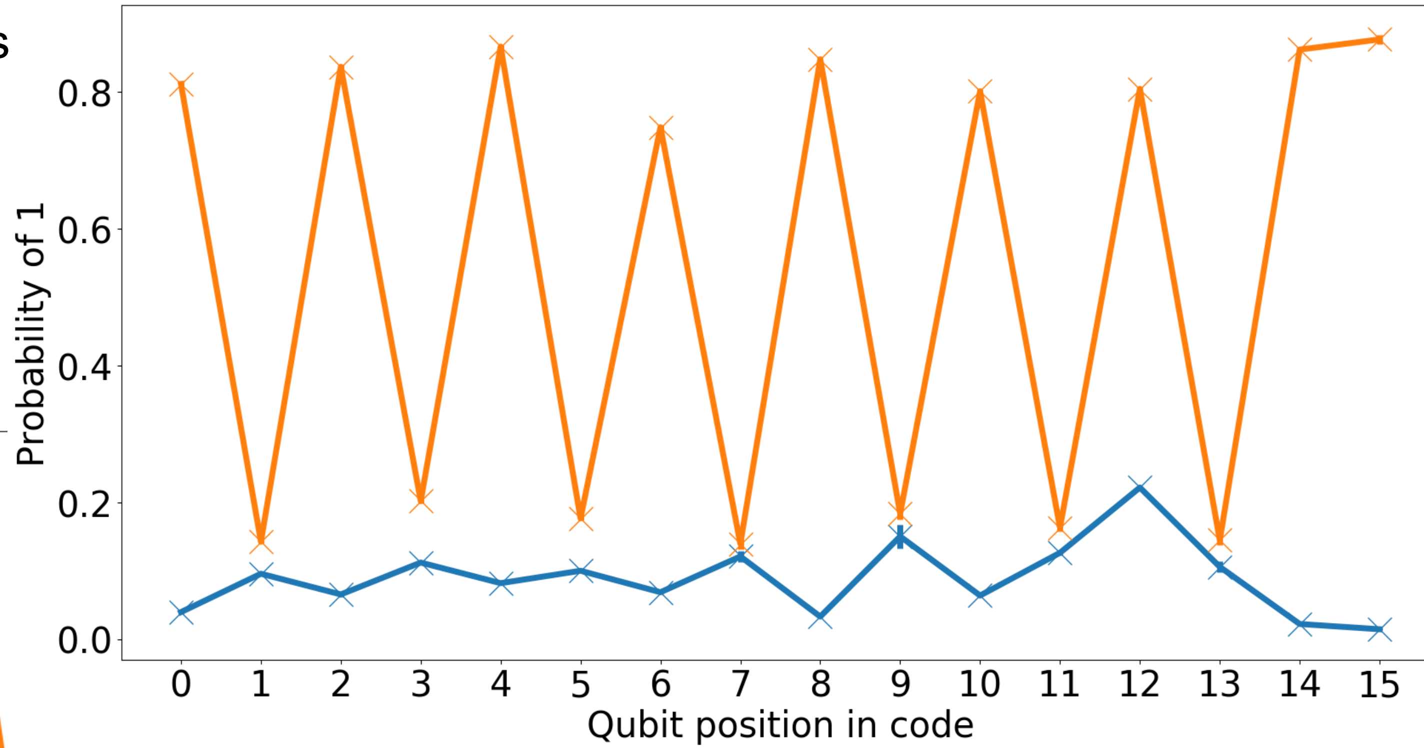
- We can compare with using just a single qubit (*p*)

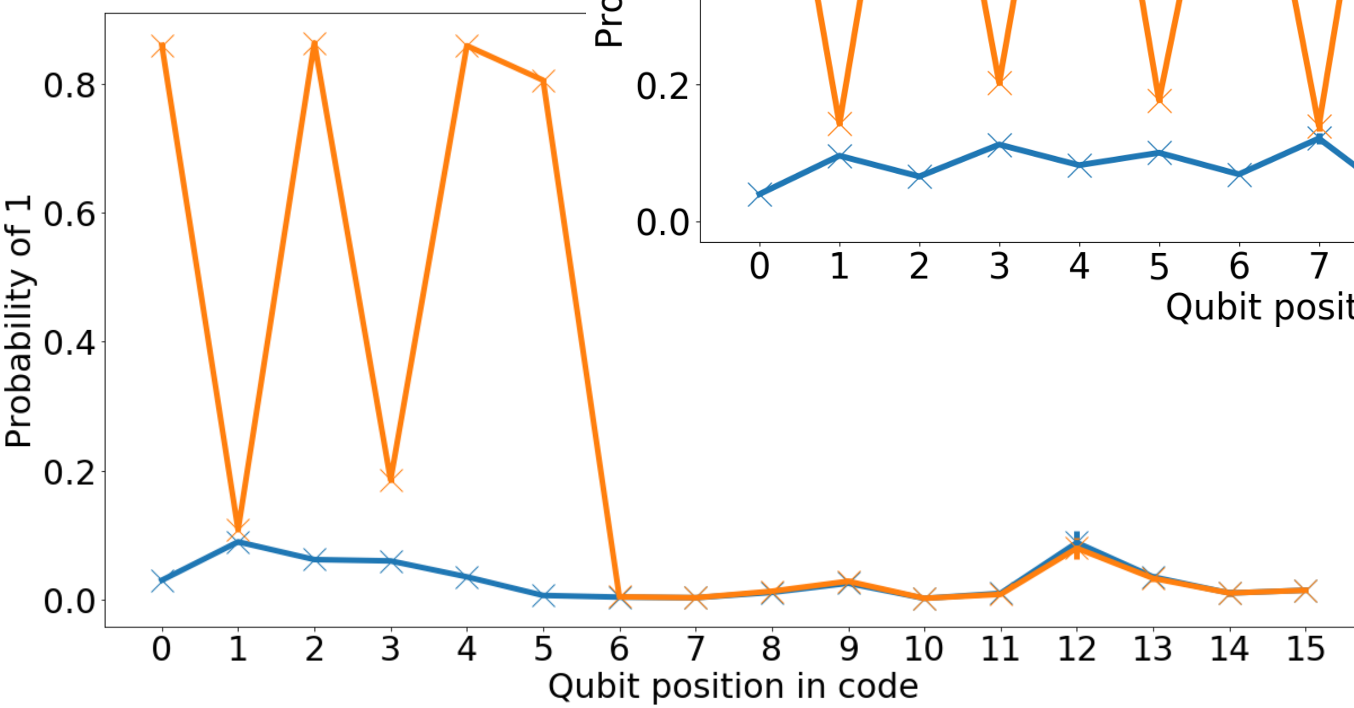
Repetition code experiments

- I did this using IBM's cloud based 16 qubit processor

github.com/decodoku/repetition_code

8 code qubits

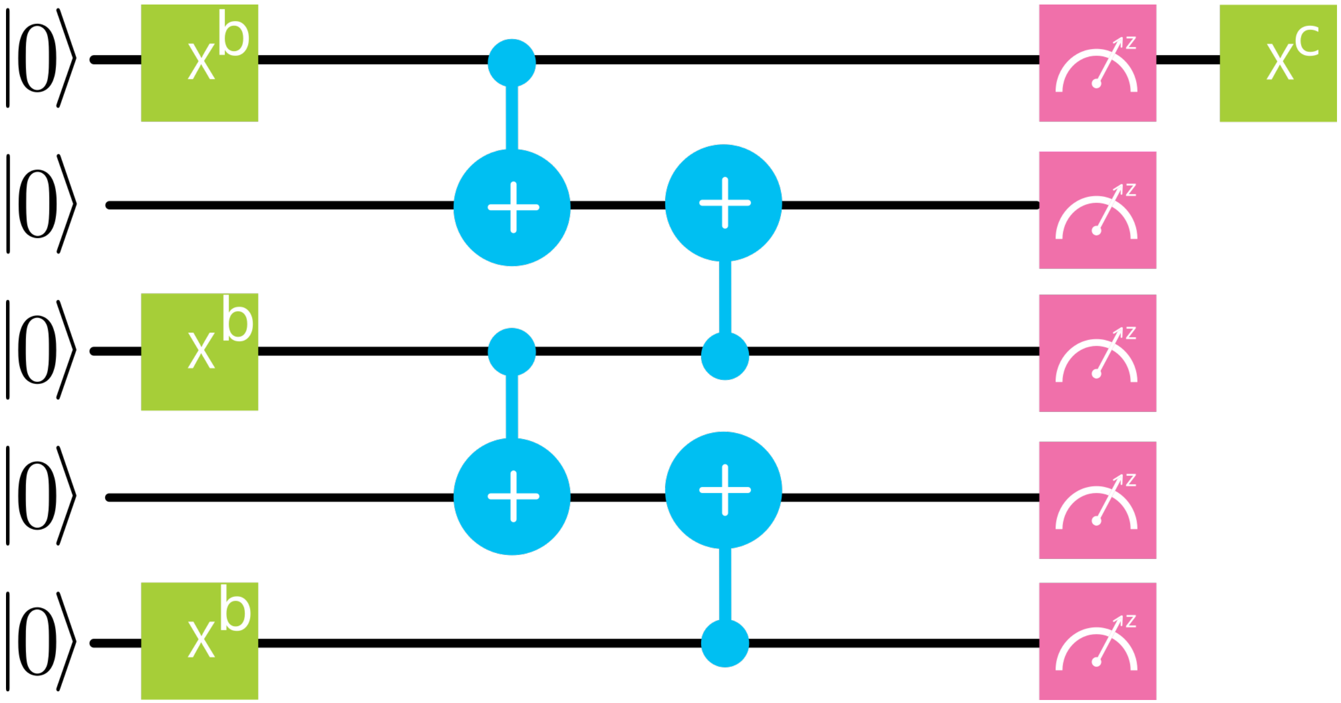




3 code qubits

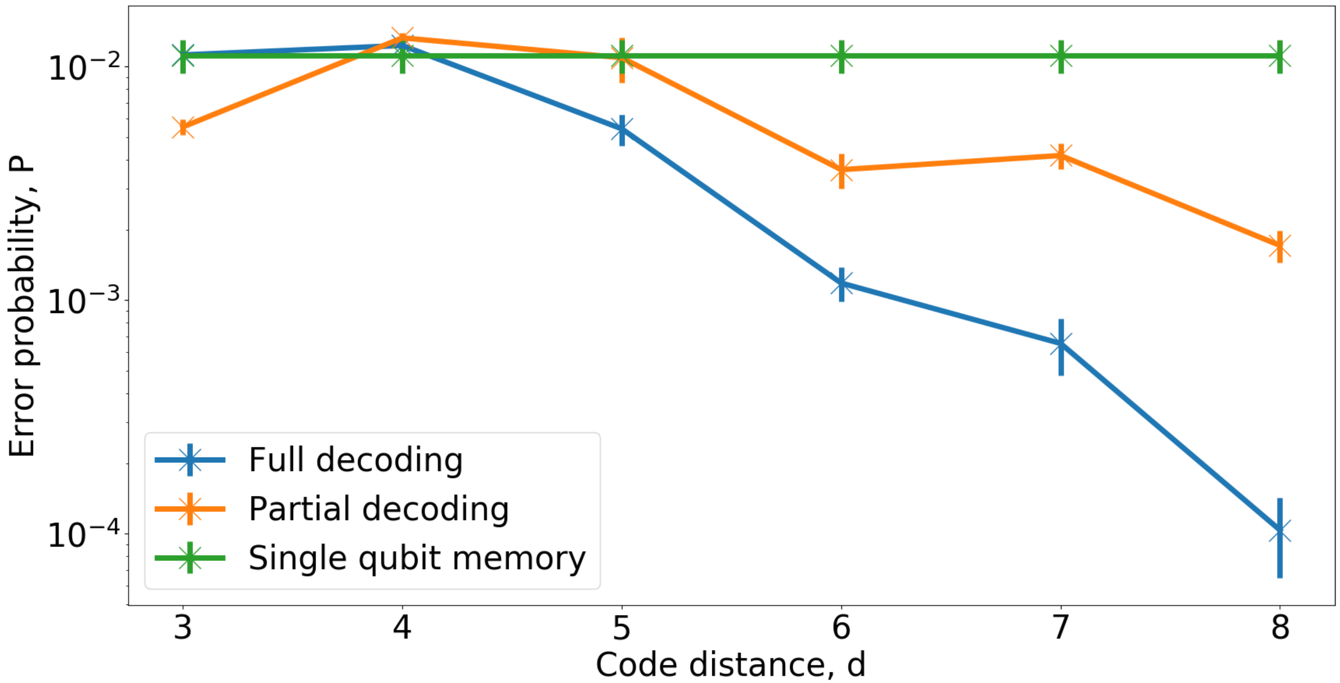
Full and partial decoding

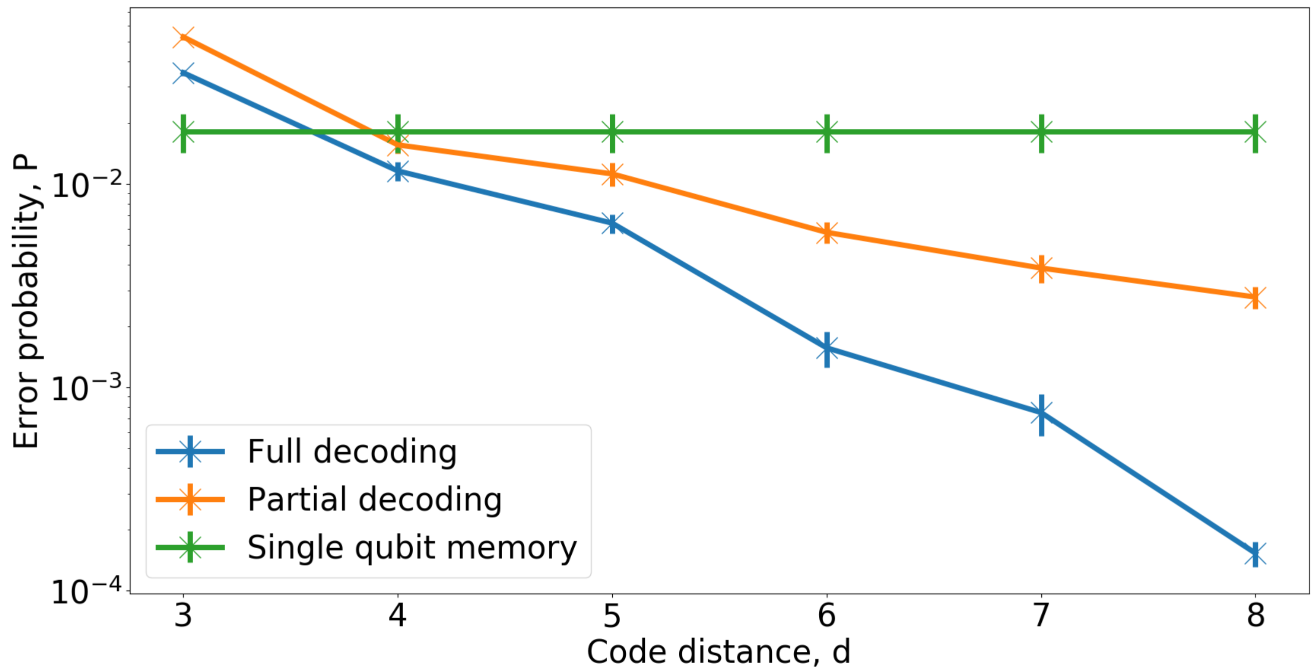
- Note that we could just ignore the ancillas
- The syndrome measurement is then useless: just a source of noise



- We'll compare decoding with the ancillas (full decoding) to that just with code qubits (partial decoding), to see how effective the cx-assisted measurements really are

Full and partial decoding

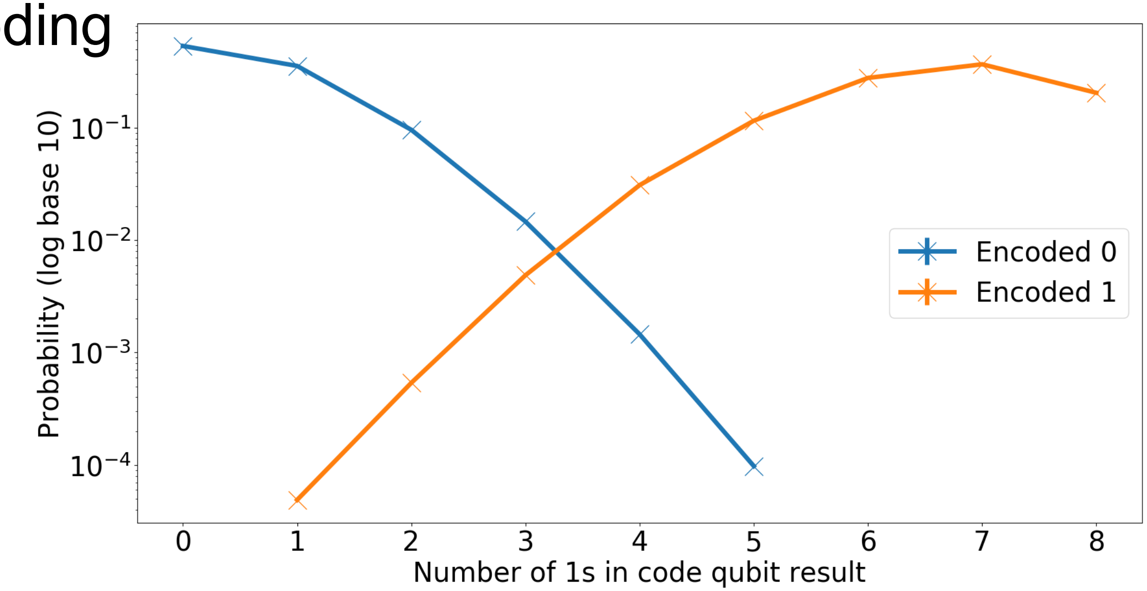




- Code > Single qubit
- Full > Partial

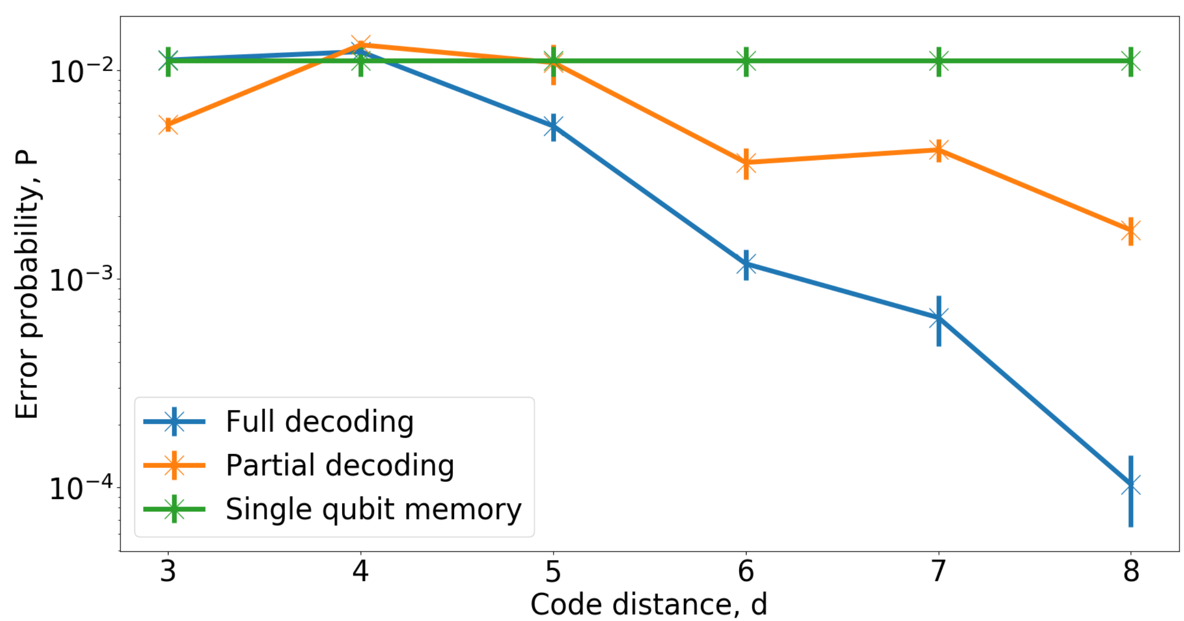
Look up table decoder

- We can do better than just majority voting
- We can use experimental data to determine the most likely encoded bit
- For example, with partial decoding



- Accounts for true nature of noise (bias, correlations, …)

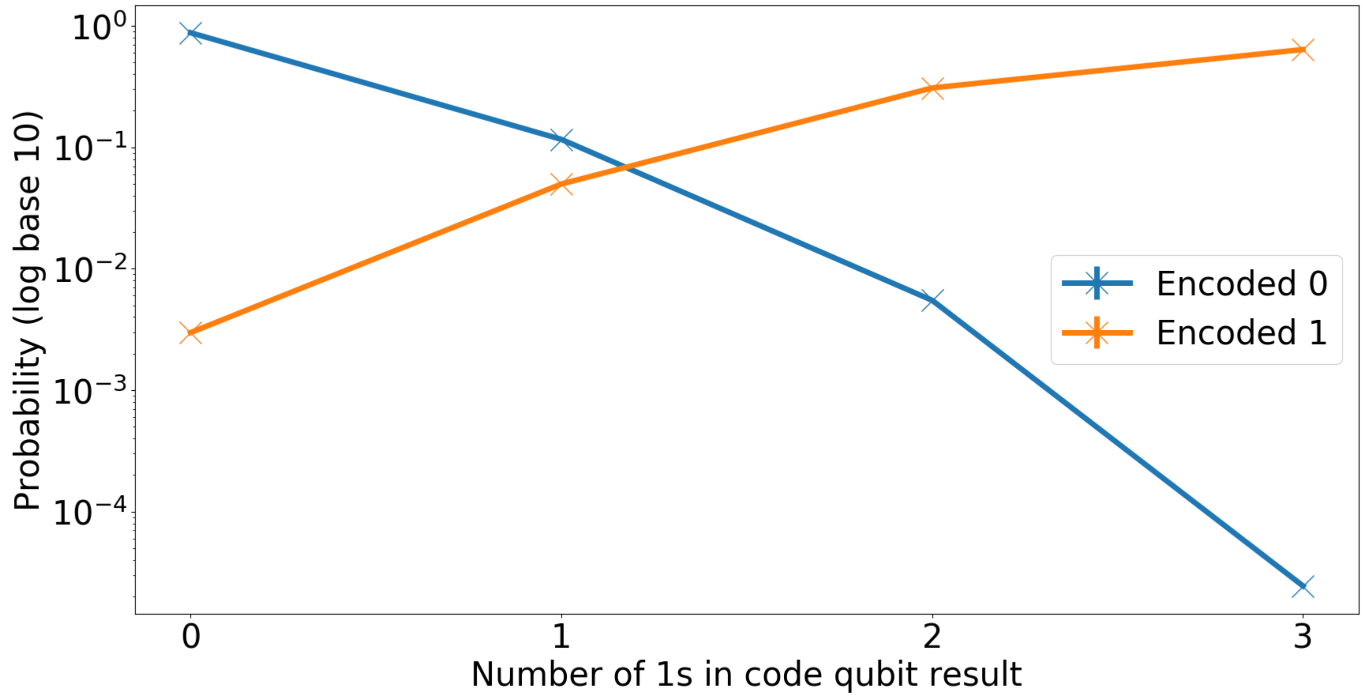
- Can explain counterintuitive finite size effects

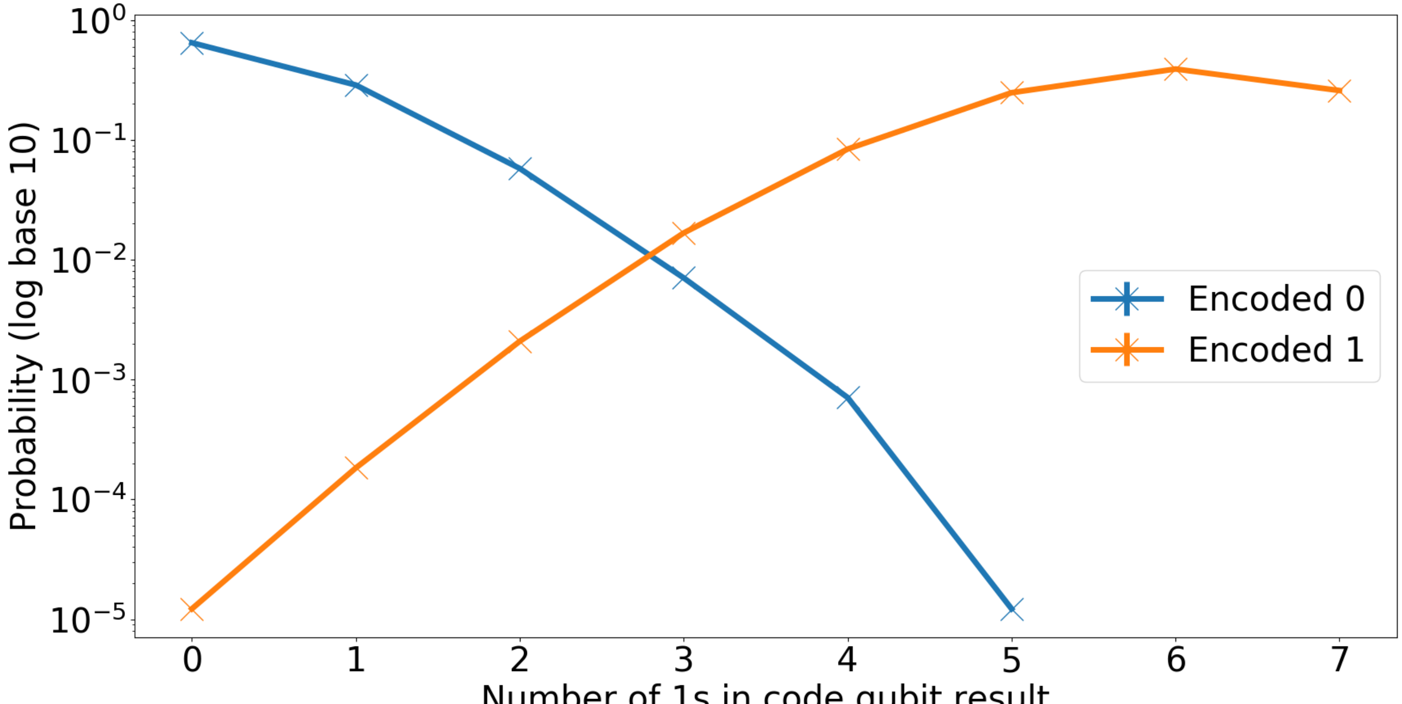


How can partial be better than full?

- Biased noise shifts crossover point

- Smaller codes are less able to adapt





Thanks for listening!