Quantum Error Correction

Using Quantum Computing

Under guidance of
Prof. Jaideep Mulherkar

202001267 – Meet Bhatt
202003039 – Mukund Ladani
202003040 – Kashyap Halavadia

Why do we need QEC?

1) Each gate is slightly wrong.

1) Qubits get perturbed by small external forces

1) Measurements sometimes lie.

Remember that Quantum Computers are generally meant to perform huge computations, involving circuits which have large no of gates and qubits.

Physical and Logical Qubits

1) Textbook quantum algorithms assume that they are operating on perfect qubits(logical qubits).

1) Running these algos on physical qubits results in output that has no meaning.

1) Measurements sometimes lie.

Remember that Quantum Computers are generally meant to perform huge computations, involving circuits which have large no of gates and qubits.

Physical and Logical Qubits

1) Each of the gates that we use made out of physical qubits has some failure rate.

1) Desirable failure rate of a single or 2 qubit gate is 10^{-15}, but we have currently reached to failure rate of 1e-2 only.

1) The way out is to make a logical qubit by using many physical qubits, that’s what we do in QEC.

1) With 10,000 qubits it may be possible to have a failure rate of the formed logical qubit as low as 10^{-15}.

Error Correction

1) **Input:-** Some information to protect.

1) **Encoding:-** Transform the information to make it easier to protect

1) **Errors:--** Random perturbations of the encoded message.

1) **Decoding:--** Trying to deduce the input from the perturbed message

Repetition Codes

1) Make redundant copies of the logical qubit value on many physical qubits, this state is called encoded state.

1) The receiver will decode the message through majority voting.

1) Misunderstanding can only happen when the majority of the copies are flipped , its possible but it is unlikely.

Repetition Codes

P (probability that misunderstanding occurs decreases exponentially as we increase the no of copies i.e. d).

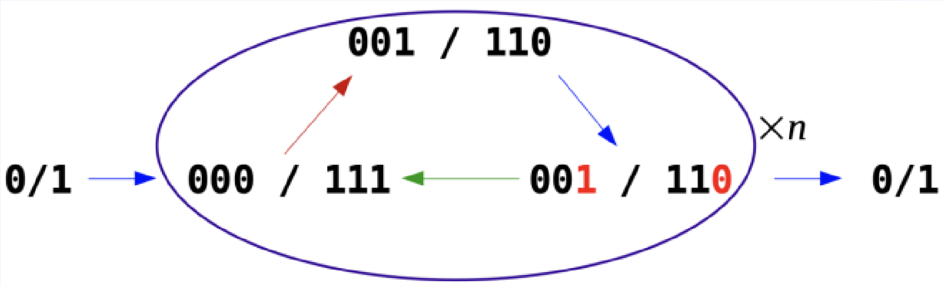
$$P = \sum_{n=d/2}^{n=d} \binom{d}{n} . p^n (1-p)^{d-n} \approx \left(\frac{p}{1-p}\right)^{d/2}$$

Error correction in Computation

1) For computation, errors are introduced whenever we perform an operation.

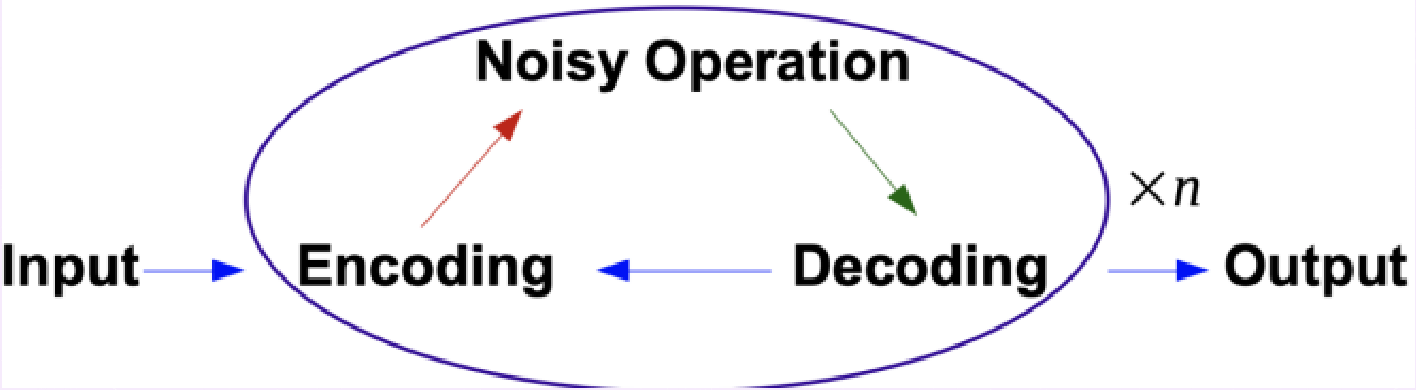
1) We need to keep on correcting errors as they are introduced.

1) Can be done by constantly decoding and re-encoding



Error correction in Computation

4) But when we consider here qubits, the decoding requires measurements and that will destroy superposition.



Syndrome Measurements

- A decoding technique which only extracts the information that is required to find out where would have errors occurred . It doesn’t measure the qubit values.

- It’s not a logical qubit measurement at all, its information about the noise that has happened.

Quantum Repetition Code

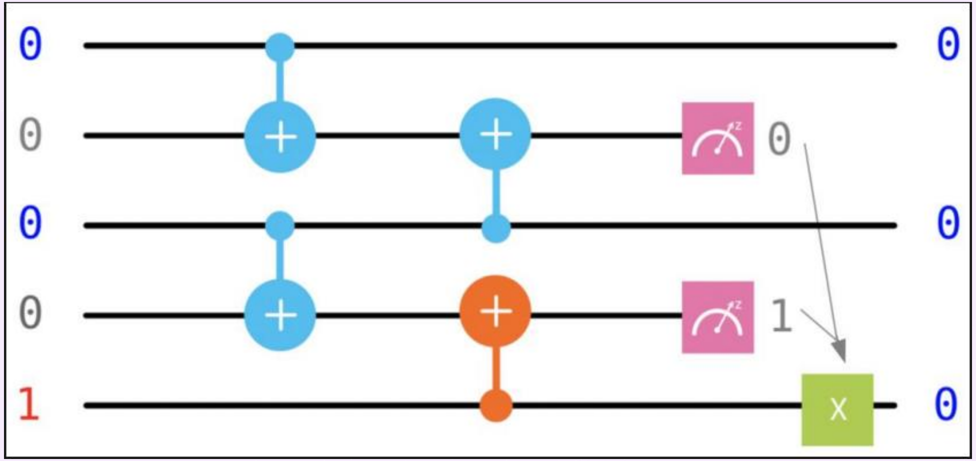
- We use an extra qubit(ancilla qubit) for each pair of code qubits.
- It’s not a logical qubit measurement at all, its information about the noise that has happened.

Quantum Repetition Code

- Can be done with some extra qubits: one for each pair of code qubits.
- They are are initialized in state $|0\rangle$, and are used as the target for two CX gates
- The net effect is to measure the observable Z_jZ_{j+1}: the Z basis parity of the two qubits
- In short: whether they are the same or different

Quantum Repetition Code

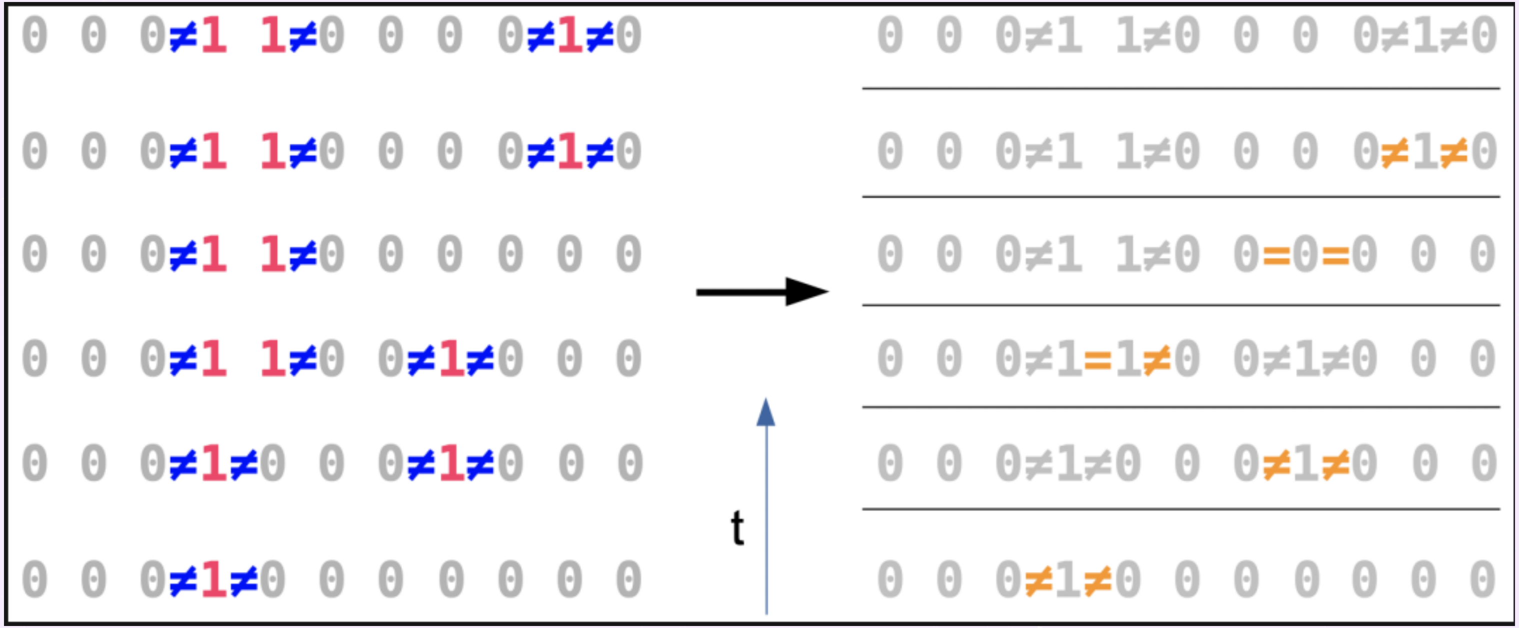
- cx $|00\rangle = |00\rangle$
- cx $|01\rangle = |00\rangle$
- cx $|10\rangle = |11\rangle$
- cx $|11\rangle = |10\rangle$



Decoding

- Start with an unrealistically simple case: errors between parity measurements only (not during)
- Focus on identifying errors for now (not correcting)
- Look for changes between rounds
- Errors create pairs of ‘defects’. Majority voting can be used to find a minimal pairing.

Decoding

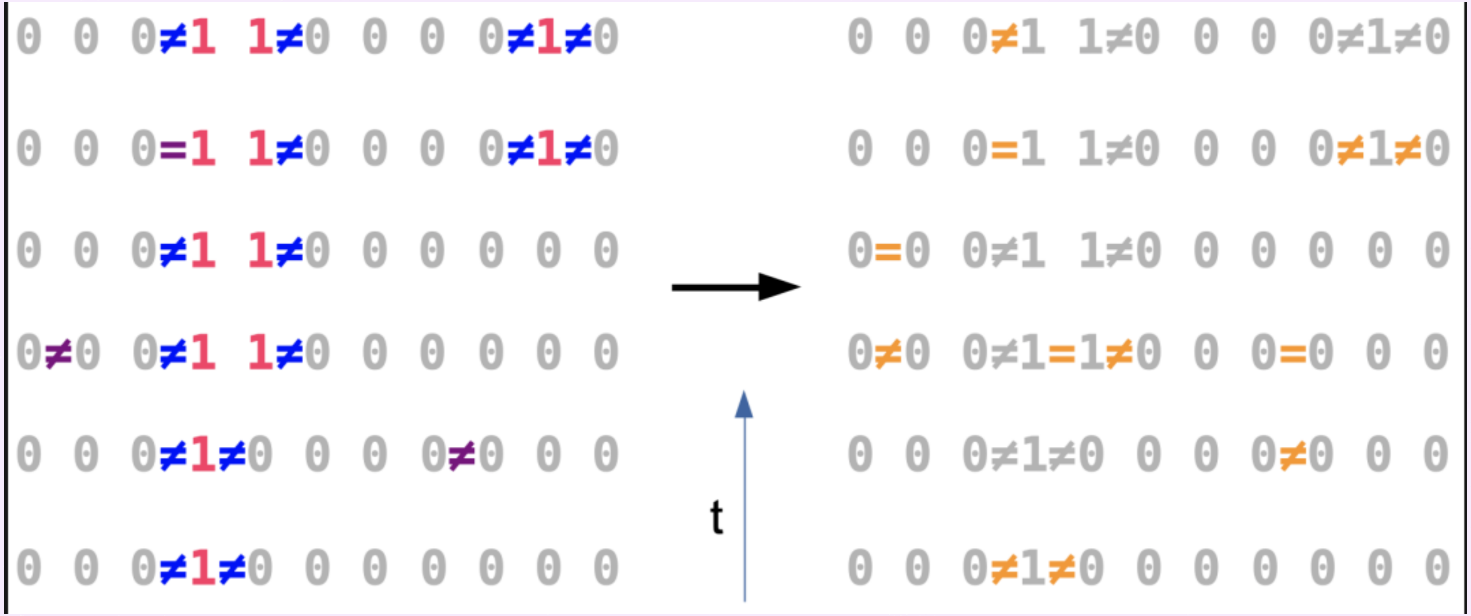


0 0 0≠1 1≠0 0 0 0≠1≠0
0 0 0≠1 1≠0 0 0 0≠1≠0
0 0 0≠1 1≠0 0 0 0 0 0
0 0 0≠1 1≠0 0≠1≠0 0 0
0 0 0≠1≠0 0 0≠1≠0 0 0
0 0 0≠1≠0 0 0 0 0 0 0
t
0 0 0≠1 1≠0 0 0 0≠1≠0
0 0 0≠1 1≠0 0 0 0≠1≠0
0 0 0≠1 1≠0 0=0=0 0 0
0 0 0≠1=1≠0 0≠1≠0 0 0
0 0 0≠1≠0 0 0≠1≠0 0 0
0 0 0≠1≠0 0 0 0 0 0 0

Decoding

- Next, a simple model of noise in the measurements: they randomly lie
- Again, look for changes between rounds
- Bit flips create defects with space-like separation, measurement errors with time-like
- Pairing is now a 2D problem, majority voting won't work

Decoding



0 0 0≠1 1≠0 0 0 0≠1≠0
0 0 0=1 1≠0 0 0 0≠1≠0
0 0 0≠1 1≠0 0 0 0 0 0
0≠0 0≠1 1≠0 0 0 0 0 0
0 0 0≠1≠0 0 0 0≠0 0 0
0 0 0≠1≠0 0 0 0 0 0 0
0 0 0≠1 1≠0 0 0 0≠1≠0
0 0 0=1 1≠0 0 0 0≠1≠0
0=0 0≠1 1≠0 0 0 0 0 0
0≠0 0≠1=1≠0 0 0=0 0 0
0 0 0≠1≠0 0 0 0≠0 0 0
0 0 0≠1≠0 0 0 0 0 0 0
t

THANKYOU