Quantum Error Correction

Julie Butler

Method 0: Hardware and Architecture. Examples?

Method 1: Repetition Codes (Fully Hardware)

Repetition Codes to Catch Bit Flips

- Consider the following state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\alpha|^2 + |\beta|^2 = 1$$

- The first step in the repetition code is to encode this one qubit state into a three qubit state:

$$|\psi\rangle_3 = \alpha|000\rangle + |111\rangle = |\psi\rangle|\psi\rangle|\psi\rangle$$

- Caveat: You need to know $|\psi\rangle$ for this to work

Creating the $|\psi\rangle_3$ circuit

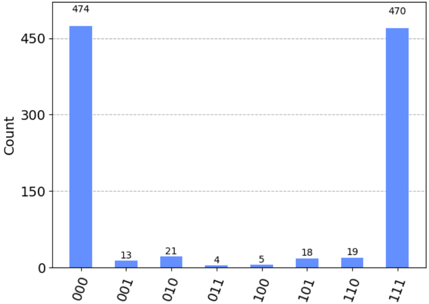


Figure 1: Repetition Code for Bit-Flip

Determining The Bit-Flip: Syndrome Qubits

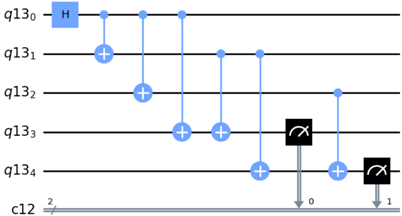


Figure 2: Syndrome Qubits Added to a Bit-Flip Repetition Code

Determining Corrections for a Single Bit-Flip

State	Syndrome	Correction
$\alpha\|000\rangle + \beta\|111\rangle$	00	$\mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I}$
$\alpha\|100\rangle + \beta\|011\rangle$	10	$X \otimes \mathbb{I} \otimes \mathbb{I}$
$\alpha\|010\rangle + \beta\|101\rangle$	11	$\mathbb{I} \otimes X \otimes \mathbb{I}$
$\alpha\|001\rangle + \beta\|110\rangle$	01	$\mathbb{I} \otimes \mathbb{I} \otimes X$

Figure 3: Bit-Flip Corrections Table

The Failure of the Bit-Flip Repetition Code

- This method will catch a single bit-flip and does not work correctly on more than one bit-flip
- Note: Unlike the other forms of quantum error, bit flips are also an error in classical computing

Repetition Codes to Catch Phase Flips

- Phase Flip Error: $|1\rangle \longrightarrow -|1\rangle$
- Previous method will not catch a phase-flip error, so we need a new method
- Instead of $|\psi\rangle_3 = \alpha|000\rangle + \beta|111\rangle$ we will use $|\psi\rangle_3 = \alpha|+++\rangle + \beta|---\rangle$
- Also need to modify syndrome qubits, **To Jupyter!**

Phase-Flip Circuit

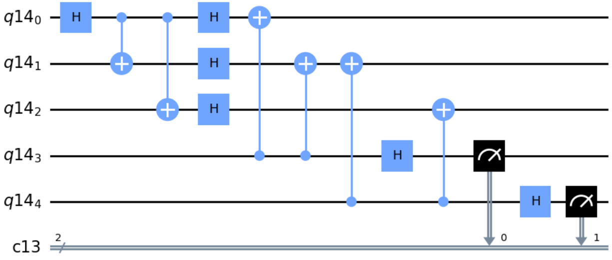


Figure 4: Phase-Flip Circuit

Phase-Flip Correction Table

State	Syndrome	Correction
$\alpha\|+++\rangle + \beta\|---\rangle$	00	$\mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I}$
$\alpha\|-++\rangle + \beta\|+--\rangle$	10	$Z \otimes \mathbb{I} \otimes \mathbb{I}$
$\alpha\|+-+\rangle + \beta\|-+-\rangle$	11	$\mathbb{I} \otimes Z \otimes \mathbb{I}$
$\alpha\|++-\rangle + \beta\|--+\rangle$	01	$\mathbb{I} \otimes \mathbb{I} \otimes Z$

Figure 5: Phase-Flip Correction Table

The Failure of the Phase-Flip Repetition Code

- This method will catch a single phase-flip and does not work correctly on more than one phase-flip

The Solution: Shor's 9 Qubit Code

Shor's 9 Qubit Code

- Essentially combines the previous two examples **BUT** each of the three qubits are represented by three qubits

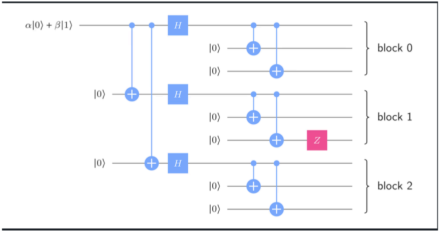


Figure 6: Shor's 9 Qubit Code

Shor's 9 Qubit Code

- We won't code this in class, see Qiskit's *Correcting Quantum Errors* tutorial for more information
- Shor's 9 Qubit code can catch and correct multiple bit and phase flips...on a single qubit?
- Most number of qubits which can be used on modern computers?
- Examples of codes we have used in this class with more qubits?

Method 2: Error Mitigation Matrix (Part Hardware, Part Post-Processing)

Consider a Two-Qubit Circuit...

- We can write the ideal results of a two qubit simulation as follows:

$$C_{ideal} = \begin{bmatrix} N_{00,i} \\ N_{01,i} \\ N_{10,i} \\ N_{11,i} \end{bmatrix}$$

- For example, for the Bell State we made Monday:

$$C_{ideal}(|\Psi^+\rangle) = \begin{bmatrix} 512 \\ 0 \\ 0 \\ 512 \end{bmatrix}$$

However Simulating on a Real Circuit Does Not Lead to Ideal Results

- Let the following vector represent the real result of a simulation with noise:

$$C_{real} = \begin{bmatrix} N_{00,r} \\ N_{01,r} \\ N_{10,r} \\ N_{11,r} \end{bmatrix}$$

- **To Jupyter!**

Moving from Noisy to Ideal

- Assume that we can construct a matrix, M, such that:

$$C_{real} = MC_{ideal}$$

- Then:

$$C_{ideal} = M^{-1}C_{real}$$

- But what is M?

Error or Mitigation Matrix

- To continue the example of our two-qubit system:

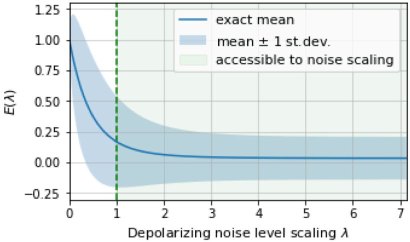
$$M = \begin{bmatrix} \frac{C_{00,r}}{shots} & \frac{C_{01,r}}{shots} & \frac{C_{10,r}}{shots} & \frac{C_{11,r}}{shots} \end{bmatrix}$$

- Can compute this with our previous circuit, but more complex for larger numbers of qubits.
- Not often used due to the complexity of creating the matrix (and the assumption that the noise is constant).

Method 3: Zero Noise Extrapolation (ZNE)

The Idea Zero Noise Extrapolation

- Noise is intentionally added to the circuit at a controlled level.
- Measure the results at each noise level and then extrapolate back



to zero noise

Methods of Adding Noise

- Unitary Folding (Definition of Unitary?)
  - Global Folding
  - Local Folding
- Identity insertion scaling
- **NOTE:** There are other ways to increase the quantum noise in a system, this is the method performed by the library we will be using. Some of the methods are specific to the type of quantum computer you are using (simulated, simulated with real noise, real)

Depth of a Circuit

- The longest path (in terms of gates) through the circuit.
- Examples: **To the board!**
- **NOTE** If running a circuit on a real quantum computer (or a simulated real quantum computer) the depth of the transpiled circuit may be different than the depth of the schematic due to the connectivity of the qubits.

ZNE Benefits and Drawbacks

- Benefits
  - Simple and Popular
  - Does not require much additional sampling
  - Does not require an understanding of the underlying noise of the quantum computer
- Drawbacks
  - Inaccuracies due to extrapolation
  - Sensitivity to noise scaling method
  - Resources used to run noisy circutis

Example 1: Test Circuit

** To Jupyter!**

Variational Quantum Eigensolver: A New Way to Find Expectation Values

$$E = Tr(\rho H)$$

$$\rho = |\psi\rangle\rangle\psi|$$

$$\langle H\rangle = E = \langle\psi|H|\psi\rangle = tr(\rho H)$$

Example 2: VQE

** To Jupyter!**