CS 410/510 Introduction to Quantum Computing

Lecture 14

Portland State U, Spring 2017 — *May 18, 2017*

Lecturer: Fang Song — *Scribe: Asher Toback*

Version: October 9, 2017

So far, we've assumed that the channels in our quantum circuits have been noiseless. Namely, that the information sent over the channel is identical to the information that is received. However, there will always be noise in real-life channels that may cause errors during transmition. In the classical setting we see these errors as bit flips in our original message. However, when we try to communicate quantum information over a quantum channel it becomes less clear what an error is, how to detect it, and how to correct it. In this paper we'll introduce the basic concepts of QECC (Quantum Error Correcting Code). We'll start by defining our Error Model, cover X-Encoding, Z-Encoding, Shor's 9-bit QECC, and arbitrary unitary QECC, and finish with a brief description of CSS Codes.

1 Error Models

Let's first describe how we think of communicating over a noisy channel in a classical setting. We'll be trying to communicate a bit $b \in \{0,1\}$. We assume that an error occurs on the channel with probability $p \in (0, 1/2)$. Thus, we can think of our channel as:

$$b \xrightarrow{\text{Message Over a Binary Symmetric Channel}} \begin{cases} b & \text{with probability } 1-p \\ 1-b & \text{with probability } p \end{cases}$$

In the quantum setting we similarly wish to communicate the state of our system over a channel. We know that we can represent our system state as a density matrix ρ. We can then view an error on the channel as some operation Φ occuring on our state ρ with some probability $p \in (0, 1/2)$. As this is an introduction to QECC, we'll assume that the only errors on our channel can be unitary, meaning $\Phi = U\rho U^\dagger$. Our quantum error model is thus:

$$\rho \xrightarrow{\text{Message Over a Quantum Binary Symmetric Channel}} \begin{cases} \rho & \text{with probability } 1-p \\ \Phi\rho & \text{with probability } p \end{cases}$$

2 Classical Solutions

The standard way for dealing with errors in the classical setting is simply to introduce redundancies in our message. Here is basic 3-bit encoding scheme.

$$0 \to 000$$

$$1 \to 111$$

To decode a message we partition our received message into three bit chunks, find what the majority of each three bit section is, and assign that majority to that section. If no errors occur it's clear that we'll receive the same message we sent after decoding. However, it's also the case that if only one error occurs on a 3 bit section, then we'll also recover the original message as if no error had occured. Thus, by a simple counting argument, we see that:

$$Pr[\text{Correct Transmission}] = (1-p)^3 + 3p(1-p)^2 = 1 - (3p^2 - 2p^3)$$

And most importantly, when $p < 1/2$:

$$1 - (3p^2 - 2p^3) \leq 1 - p$$

Thus, using this encoding scheme will help the receiver receive the correct message. Now that we've demonstrated this, it's clear that we could get a better result if we simply introduce more and more redundancies into our message. Thus, we could communicate our message correctly with as high a probability as we'd like.

It's also worth nothing that, if $p > 1/2$, we can simply do the same exact process as above, but at the end of the protocol we manually flip all the bits ourselves. This is due to the fact that the channel is reliable, in that we can expect it to always apply an error. Thus, we can correct for that by simply flipping all the bits at the end. However, if $p = 1/2$, then we're out of luck. The channel is just as likely to preserve our message as it is to cause on error, so we can't rely on it at all. This is clearly the worst case and represents white noise.

3 Quantum Bit Flip Error

It's not immediately obvious how to even approach a quantum error. Applying a unitary operation onto a quantum state can be viewed as rotating that state in the complex plane. This means there are uncountable many different types of errors that we need to correct. And, even if we knew which error were to occur, it's not immediately obvious how we could introduce redundacies like in the classical setting. Namely, we can't clone our input.

To begin tackling these issues let's first assume that the only errors that occur on our channel are X, the Pauli Matrix that represents a bit flip in the quantum setting. As a reminder, for some qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$|\psi\rangle \xrightarrow{X} \alpha|1\rangle + \beta|0\rangle$$

This is a start, but how will we deal with no cloning? It turns out there does exist a 3-bit encoding scheme which resembles the classical one which doesn't involve cloning. Namely:

$$|0\rangle \xrightarrow{X-Encoding} |000\rangle$$
$$|1\rangle \xrightarrow{X-Encoding} |111\rangle$$

This encoding scheme is called X-Encoding, as it will be able to correct for X errors, which we'll show shortly. First, note that, although they look similar, this encoding scheme is not:

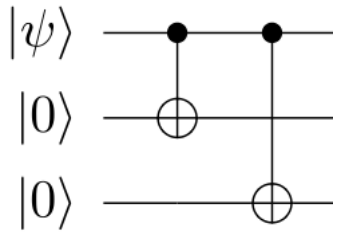
$$|\psi\rangle \xrightarrow{X-Encoding} |\psi\psi\psi\rangle$$

It instead works like the following. If the qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then the encoding works as follows:

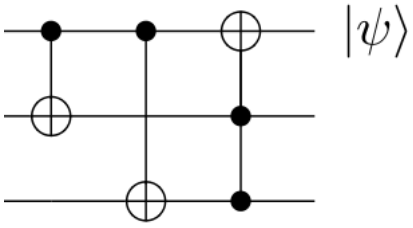
$$|\psi\rangle \xrightarrow{X-Encoding} \alpha|000\rangle + \beta|111\rangle$$

To see why these aren't the same, note that $(|00\rangle + |11\rangle) \neq (|0\rangle + |1\rangle)^{\otimes 2}$. Clearly these must be different states as the first state is maximally entangled and the other is not.

Now, to achieve X-Encoding we use the following circuit.



where the hollow circles are CNOT gates. It's left to the reader to verify that this behaves as intended. We now want to decode in the same way that we did in the classical setting. That is, we want to find the majority of the three qubits and assign these three qubits to be this majority instead. It turns out that this can be done, and it's done by the following circuit.



The last gate is the Toffoli gate, and can be thought of as a CNOT gate which depends on two control wires. Specifically, it require that both are in the state $|1\rangle$ in order to apply a NOT.

With these encoding and decoding schemes let's observe that they correct an X error as claimed. We'll assume that the X error occurs on the second bit of the encoding.

$$\begin{aligned}\alpha|0\rangle + \beta|1\rangle &\xrightarrow{X-Encoding} \alpha|000\rangle + \beta|111\rangle \\ &\xrightarrow{\mathbb{I}\otimes X\otimes\mathbb{I}} \alpha|010\rangle + \beta|101\rangle \\ &\xrightarrow{X-Decoding} (\alpha|0\rangle + \beta|1\rangle) \otimes |10\rangle\end{aligned}$$

It's left to the reader to verify that the above mappings are indeed correct. We see then that the top wire does indeed recover the original qubit, as desired! It's left to the reader to verify that this code does correct every possible X error, as long as only one error occurs in total.

It's also worth noting that the bottom two wires hold the information of the error. That is, if you measured them, you could discover which bit that the X error occurred. These are called the *syndrome* and will be useful for future QECC.

We've thus accomplished what we set out to do, namely fix a bit flip error in the quantum setting.

4 Quantum Phase Flip Error

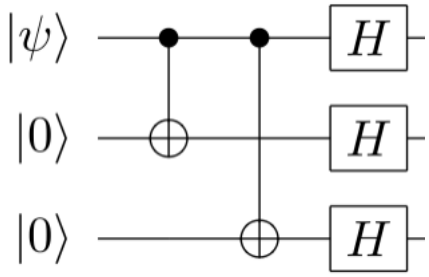
What if an error other than a bit flip were to occur? We'll now look at the case of a Z error, or a phase flip. As a reminder, for some qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$|\psi\rangle \xrightarrow{Z} \alpha|0\rangle - \beta|1\rangle$$

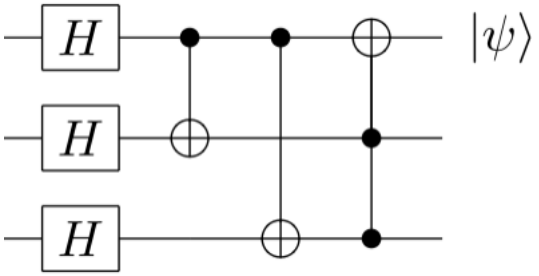
To correct this error, we make an important observation. Namely: $Z|+\rangle = |-\rangle$, $Z|-\rangle = |+\rangle$. Thus, we see that a phase flip can be thought of as a bit flip in the Hadamard basis $\{|+\rangle, |-\rangle\}$. In other words, $HZH = X$, where H is the Hadamard gate. And, since we've already corrected bit flip errors, we're essentially finished. Thus, our encoding scheme should behave as follows:

$$|0\rangle \xrightarrow{Z-Encoding} |+++\rangle$$
$$|1\rangle \xrightarrow{Z-Encoding} |---\rangle$$

To acheive this encoding we simply add Hadamard gates after our X-Encoding scheme:



Our decoding scheme similarly just adds a Hadamard gate before the X-Decoding scheme:



Just as before, let's say a Z-Error occurs on the second bit of our encoding. We then get:

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{Z-Encoding} \alpha|+++\rangle + \beta|---\rangle$$

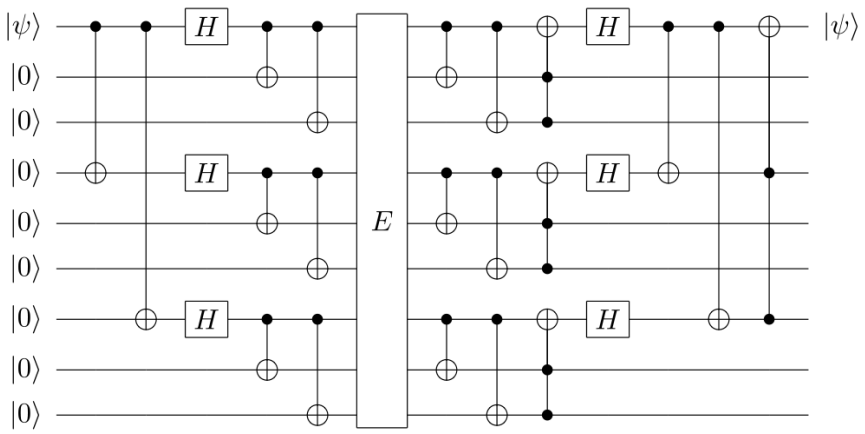


Figure 1: Shor's 9-bit QECC. All pictures from Wikipedia

$$\xrightarrow{\mathbb{I}\otimes Z\otimes\mathbb{I}} \alpha|+-+\rangle + \beta|-++\rangle$$
$$\xrightarrow{Z-Decoding} (\alpha|0\rangle + \beta|1\rangle) \otimes |10\rangle$$

Which is just what we wanted.

5 Shor's 9-qubit QECC

We can correct an X error and a Z error, but what if our channel's noise has the potential to apply both an X or a Z gate? Shor came up with a solution to this which is intuitive. Essentially, you feed the Z and X encodings into each other. Namely:

$$Shor's: |0\rangle \xrightarrow{Z-Encoding} |+++\rangle \xrightarrow{X-Encoding} (1/\sqrt{2}(|000\rangle + |111\rangle))^{\otimes 3}$$
$$Shor's: |1\rangle \xrightarrow{Z-Encoding} |---\rangle \xrightarrow{X-Encoding} (1/\sqrt{2}(|000\rangle - |111\rangle))^{\otimes 3}$$

You then X-decode this state, then Z-decode it. It turns out that this can correct 1 error on the 9-qubit encoding, which is left to the reader to verify. However, it's clear that it works due to linearity. The circuit that accomplishes these encoding and decoding schemes is as follows:

Where E represents the X or Z error.

6 Arbitrary Unitary QECC

It's natural to wonder if we can correct an error of the remaining Pauli Gate, the Y gate. Well, we know that $Y = -iXZ$. And, because we can correct X and Z errors, we know we can also correct Y errors.

However, we also know that $\forall U$ where U is a unitary matrix, that $U = \alpha I + \beta X + \gamma Y + \delta Z$. Thus, as it turns out, we know there exists encoding and decoding schemes such that we can correct 1 arbitrary Unitary error.

7 CSS Codes

In this paper, we assumed that our errors were all Unitary transformations. However, we can't always make this assumption. To fix this, we use a type of QECC called CSS Codes. These general constructions are based off of classical linear ECC. The basic ideas rellies on group theory and the idea of cosets. The encoding maps certain messages into different cosets based off what the message is. We then have QECC that corrects errors just on each coset. As it turns out, given some $t \in \mathbb{N}$, we can correct up to t errors using CSS Codes.