Week 8

Lecture 15

Simple classical error correction codes, Quantum error correction codes, stabilizer formalism, 5-qubit code, 7-qubit Steane code

Lecture 16

The more advanced quantum error correction codes, Fault Tolerance, surface code.

Lab 8

Quantum error correction

Introduction to Quantum Error Correction

Lecture 15

Overveiw

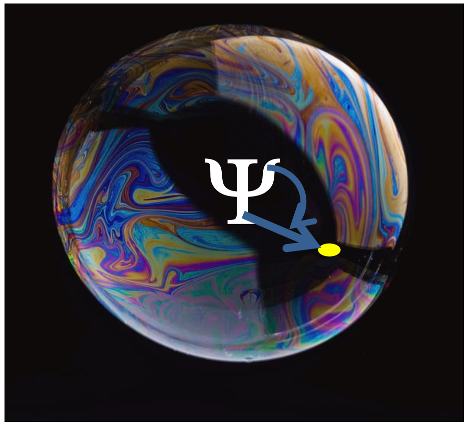
This lecture we will introduce error correction for quantum computers:

- Overview of need for quantum error correction
- Simple <u>classical</u> error correction codes
- <u>Quantum</u> error correction codes
- The stabilizer formalism
- The five qubit code and seven qubit Steane code

Reiffel, Chapter 11
Kaye, Chapter 10
Nielsen and Chuang, Chapter 10

Decoherence and control errors

Qubit: Bloch sphere (or fragile “quantum bubble”)

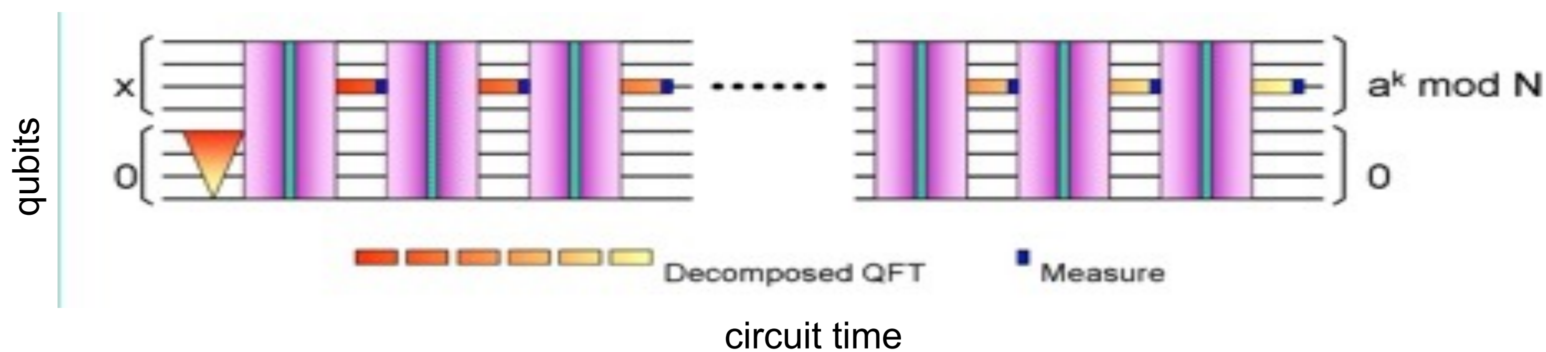


Decoherence: interaction with environment affects quantum state/operation

Even if you get decoherence under control…

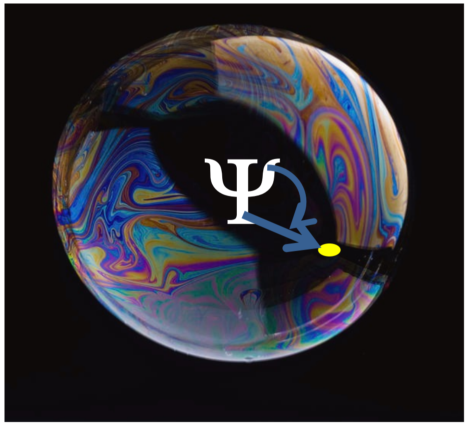
Control: imprecise control leads to error in quantum state/operation

Impact of errors on fidelity of Shor’s quantum factoring algorithm at logic level:



Decoherence and control errors

Qubit: Bloch sphere (or fragile "quantum bubble")

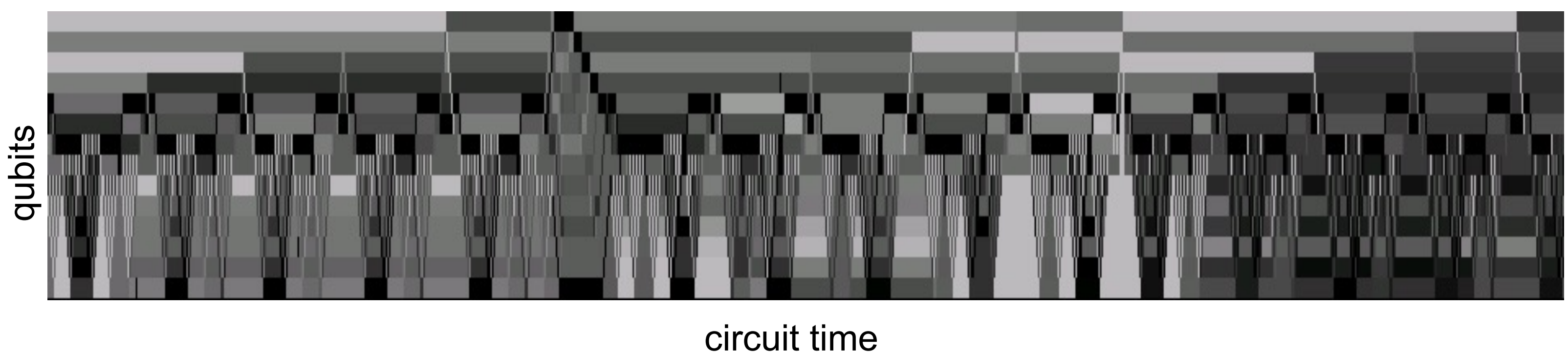


Decoherence: interaction with environment affects quantum state/operation

Even if you get decoherence under control…

Control: imprecise control leads to error in quantum state/operation

Impact of errors on fidelity of Shor's quantum factoring algorithm at logic level:



Some error locations are very sensitive…it doesn't take much to rattle an algorithm…

Even after reducing physical errors, a quantum computer needs error correction…

Classical Error Correction

The simplest example of a classical error correction code is a repetition code:

$$0 \rightarrow 000 \qquad \text{Logical "0"}$$

$$1 \rightarrow 111 \qquad \text{Logical "1"}$$

"Codewords"

If an error occurs (ie. bit flip) then using the *redundant information* we can still correct by simply taking the majority:

$$0 \begin{cases} 000 \\ 001 \\ 010 \\ 100 \end{cases} \qquad 1 \begin{cases} 111 \\ 110 \\ 101 \\ 011 \end{cases}$$

With one error, we can correct the error and continue the computation.

Code distance

The **distance** of the code is the (minimum) number of logical errors between codewords.

3 bit-flips takes 000 to 111, so the distance of the 3-bit repetition code is 3.

For classical codes, the distance is simply the minimum Hamming distance between any two codewords.

Often (for linear codes) you will see the notation:

$$[n, k, d]$$

n: number of bits

k: number of encoded bits

d: distance of code

The three-bit repetition code is a [3, 1, 3] code.

Code failure

Too many errors can overwhelm an error correction code. For example if we have two distinct errors on the codeword, 000:

$$000 \rightarrow 101$$

Which we would (wrongly) decode as "1".

A distance d code can correct

$$\left\lfloor \frac{d-1}{2} \right\rfloor$$

errors.

Quantum Error Correction

Similar to classical error correction codes, we can have a quantum repetition code:

$$|0\rangle \rightarrow |000\rangle \qquad \text{“Logical 0”}$$

$$|1\rangle \rightarrow |111\rangle \qquad \text{“Logical 1”}$$

In particular, a quantum superposition would be encoded as:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle$$

Two key differences between quantum and classical error correction codes:

1. Cannot measure the codewords directly; would collapse the state
2. Phase errors

Syndrome Measurements

If we measured our qubits, we would collapse the state. For example, if we had the three qubit error correction code, and measured the first qubit as "0" then we would collapse:

$$\alpha |000\rangle + \beta |111\rangle \rightarrow |000\rangle$$

We do not measure the qubits individually, but instead measure correlations between qubits. The measurements are known as **syndrome** measurements.

[Recall: $Z|0\rangle = +1|0\rangle \quad Z|1\rangle = -1|1\rangle \quad \rightarrow Z_1 Z_2 |01\rangle = (+1)\times(-1)|01\rangle = -|01\rangle$]

We measure: $Z_1 Z_2$ $\qquad Z_2 Z_3$

"Are the first two qubits the same?" and "Are the second two qubits the same?" If an X- error has occurred, we can tell that an error has happened, and where it is, but we have not measured any information about the encoded state.

Syndrome Measurement example

We have an encoded (logical) qubit:

$$\alpha |000\rangle + \beta |111\rangle$$

An X-error occurs on the first physical qubit:

$$\alpha |100\rangle + \beta |011\rangle$$

We measure:

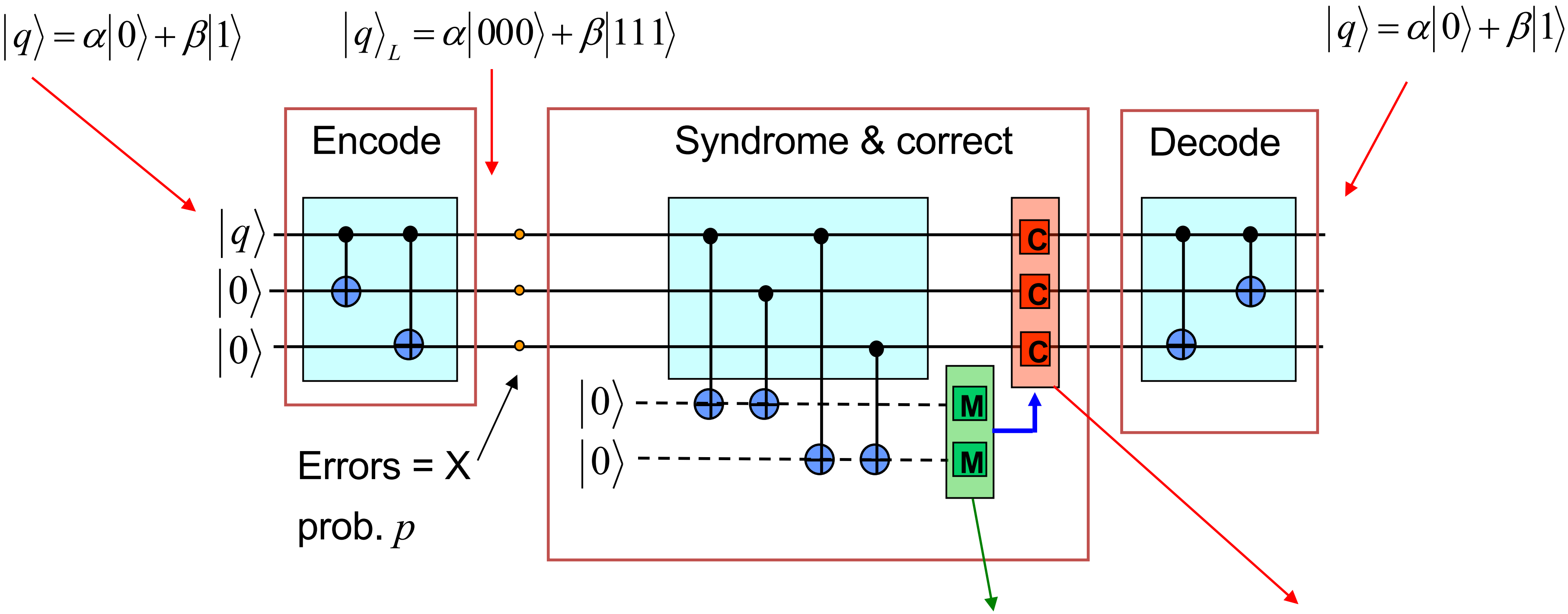
$Z_1 Z_2 = -1$ First two qubits different

$Z_2 Z_3 = +1$ Second two qubits same

From this we can deduce that an error has occurred on the first qubit, and correct (with an X gate we apply):

$$X_1(\alpha |100\rangle + \beta |011\rangle) = \alpha |000\rangle + \beta |111\rangle$$

3-bit code circuit example



Scenario	Syndrome	Correction
No error	$\|00\rangle$	No action
X error on qubit-1	$\|11\rangle$	X on qubit-1
X error on qubit-2	$\|10\rangle$	X on qubit-2
X error on qubit-3	$\|01\rangle$	X on qubit-3

Phase errors

In QM, bit flips are not the only type of errors which can occur. We can also have phase errors (and in practice these are more common).

$$Z_1 \left(\alpha \left|000\right\rangle + \beta \left|111\right\rangle\right) = \alpha \left|000\right\rangle - \beta \left|111\right\rangle$$

We have seen in the labs these errors are just as detrimental as bit flip errors!

We can make a phase-flip repetition code:

$$\left|0\right\rangle \rightarrow \left|+++\right\rangle \qquad \left|1\right\rangle \rightarrow \left|---\right\rangle$$

where

$$\left|+\right\rangle = \frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}} \qquad \left|-\right\rangle = \frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}$$

$$\begin{aligned} X\left|\pm\right\rangle &= \frac{1}{\sqrt{2}}\left(X\left|0\right\rangle \pm X\left|1\right\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(\left|1\right\rangle \pm \left|0\right\rangle\right) \\ &= \pm\left|\pm\right\rangle \end{aligned}$$

$$\rightarrow X_1 X_2 \left|+-\right\rangle = -\left|+-\right\rangle$$

The syndrome measurements we make are:

$$X_1 X_2 \qquad\qquad X_2 X_3$$

This code detects and corrects phase flip errors, but does not detect bit flip errors. Quantum error correction codes need to do both!

Phase flip code example

We have an encoded (logical) qubit:

$$\alpha\,|{+}{+}{+}\rangle + \beta\,|{-}{-}{-}\rangle$$

An Z-error occurs on the third physical qubit:

$$\alpha\,|{+}{+}{-}\rangle + \beta\,|{-}{-}{+}\rangle$$

We measure:

$$X_1X_2 = +1$$ First two qubits same

$$X_2X_3 = -1$$ Second two qubits different

$$X\,|\pm\rangle = \pm\,|\pm\rangle$$
$$\rightarrow X_1X_2\,|{-}{-}\rangle = +\,|{-}{-}\rangle$$

From this we can deduce that a phase error has occurred on the third qubit, and correct (with an Z gate we apply):

$$Z_3\left(\alpha\,|{+}{+}{-}\rangle + \beta\,|{-}{-}{+}\rangle\right) = \alpha\,|{+}{+}{+}\rangle + \beta\,|{-}{-}{-}\rangle$$

The Bacon-Shor Code

THE UNIVERSITY OF MELBOURNE

Codes exist which correct **both** phase flips, and bit flips, such as the Bacon-Shor 9-qubit code:

$$|0_L\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)$$

Syndrome measurements are a combination the bit-flip and phase-flip codes. First as if this is three bit flip codes:

$$\boxed{Z_1Z_2, Z_2Z_3}, \boxed{Z_4Z_5, Z_5Z_6}, \boxed{Z_7Z_8, Z_8Z_9}$$

Then treating it as thee **logical** qubits of three qubits each, and checking for a bit flip on any of these:

$$X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9$$

Stabilizer Formalism

THE UNIVERSITY OF MELBOURNE

Instead of specifying the codewords, we will specify the syndrome measurements which should give a "+1" result. From this we can derive the codewords/codespace.

An operator, S, is a stabilizer of the state $|\psi\rangle$ if

$$S|\psi\rangle = |\psi\rangle$$

Similarly, an operator S is a stabilizer of a subspace, if it stabilizes every basis state of that subspace.

For example:

$$Z|0\rangle = |0\rangle \qquad X\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

For our purposes, the stabilizers will all be tensor products of Pauli operators and the identity.

Aside: The Stabilizer Group

Mathematically, the stabilizers of a state (or a subspace) form an group, known as the stabilizer group, S. Verifying the four group axioms:

$$I\left|\psi\right\rangle = \left|\psi\right\rangle$$

If S_1, S_2 and S_3 stabilize $\left|\psi\right\rangle$ then:

$$S_1 S_2 \left|\psi\right\rangle = S_1 \left|\psi\right\rangle = \left|\psi\right\rangle$$

Associativity:

$$(S_1 S_2) S_3 = S_1 (S_2 S_3)$$

If S stabilizes $\left|\psi\right\rangle$ then

$$S^{-1}\left|\psi\right\rangle = S^{-1} S \left|\psi\right\rangle = \left|\psi\right\rangle$$

Typically (and for all of these lectures) we will choose the stabilizer group to be a subset of the Pauli group, and it is **Abelian** (ie. AB=BA).

We can specify the stabilizer group by writing its generators (S_1, S_2, S_3,…S_k).

Stabilizers and QEC

For the bit-flip code, the "stabilizers" (generators of the stabilizer group) of the code are:

$$\begin{cases} Z_1 Z_2 \\ Z_2 Z_3 \end{cases}$$

The codewords are stabilized by these operators:

$$Z_1 Z_2 |000\rangle = |000\rangle \qquad Z_1 Z_2 |111\rangle = |111\rangle$$
$$Z_2 Z_3 |000\rangle = |000\rangle \qquad Z_2 Z_3 |111\rangle = |111\rangle$$

Any linear combination is also stabilized by these operators:

$$\alpha |000\rangle + \beta |111\rangle$$

Commutation of Pauli operators

Commutation properties of the Pauli operators X, Y and Z are very useful at this point. We get the relations by considering actions on an arbitrary state.

For an arbitrary state we have different Pauli operators anti-commute (a negative sign when they are switched in order):

$$XZ|\psi\rangle = -ZX|\psi\rangle \rightarrow XZ = -ZX$$

$$XY|\psi\rangle = -YX|\psi\rangle \rightarrow XY = -YX$$

$$ZY|\psi\rangle = -YZ|\psi\rangle \rightarrow ZY = -YZ$$

Operators on different qubits commute (self evident):

$$X_1Z_2|\psi\rangle = Z_2X_1|\psi\rangle \rightarrow X_1Z_2 = Z_2X_1$$

But even products of operators commute:

$$X_1X_2Z_1Z_2|\psi\rangle = Z_1Z_2X_1X_2|\psi\rangle \rightarrow X_1X_2Z_1Z_2 = Z_1Z_2X_1X_2$$

Error And Stabilizers

If an error **anti-commutes** with a syndrome measurement operator (ie. stabilizer generator) then the measurement result changes sign.

No Error

For example, consider the three qubit code, for which

$$Z_1 Z_2 |\psi\rangle = +1 |\psi\rangle$$

The syndrome measurement outcome is +1 (since the system is in the +1 eigenstate)

X Error

After an X-error on the first qubit:

$$Z_1 Z_2 |\psi'\rangle = Z_1 Z_2 X_1 |\psi\rangle = -X_1 Z_1 Z_2 |\psi\rangle = -X_1 |\psi\rangle = -|\psi'\rangle$$

The syndrome measurement outcome is -1 (since the system is in the -1 eigenstate)

Error and Stabilizers

Error	State	$Z_1 Z_2$	$Z_2 Z_3$
I	$\alpha\|000\rangle + \beta\|111\rangle$	+1	+1
X_1	$\alpha\|100\rangle + \beta\|011\rangle$	-1	+1
X_2	$\alpha\|010\rangle + \beta\|101\rangle$	-1	-1
X_3	$\alpha\|001\rangle + \beta\|110\rangle$	+1	-1

Unique syndromes means that we can identify which error has occurred.

The Five Qubit Code

The smallest d=3 code to identify both bit and phase flips has five qubits.

Optimal
5 (qubits) x 3 (possible {X, Y or Z} errors on each qubit) + 1 (no error) = 16 syndromes
2^4 = 16 possible syndromes from four measurements.

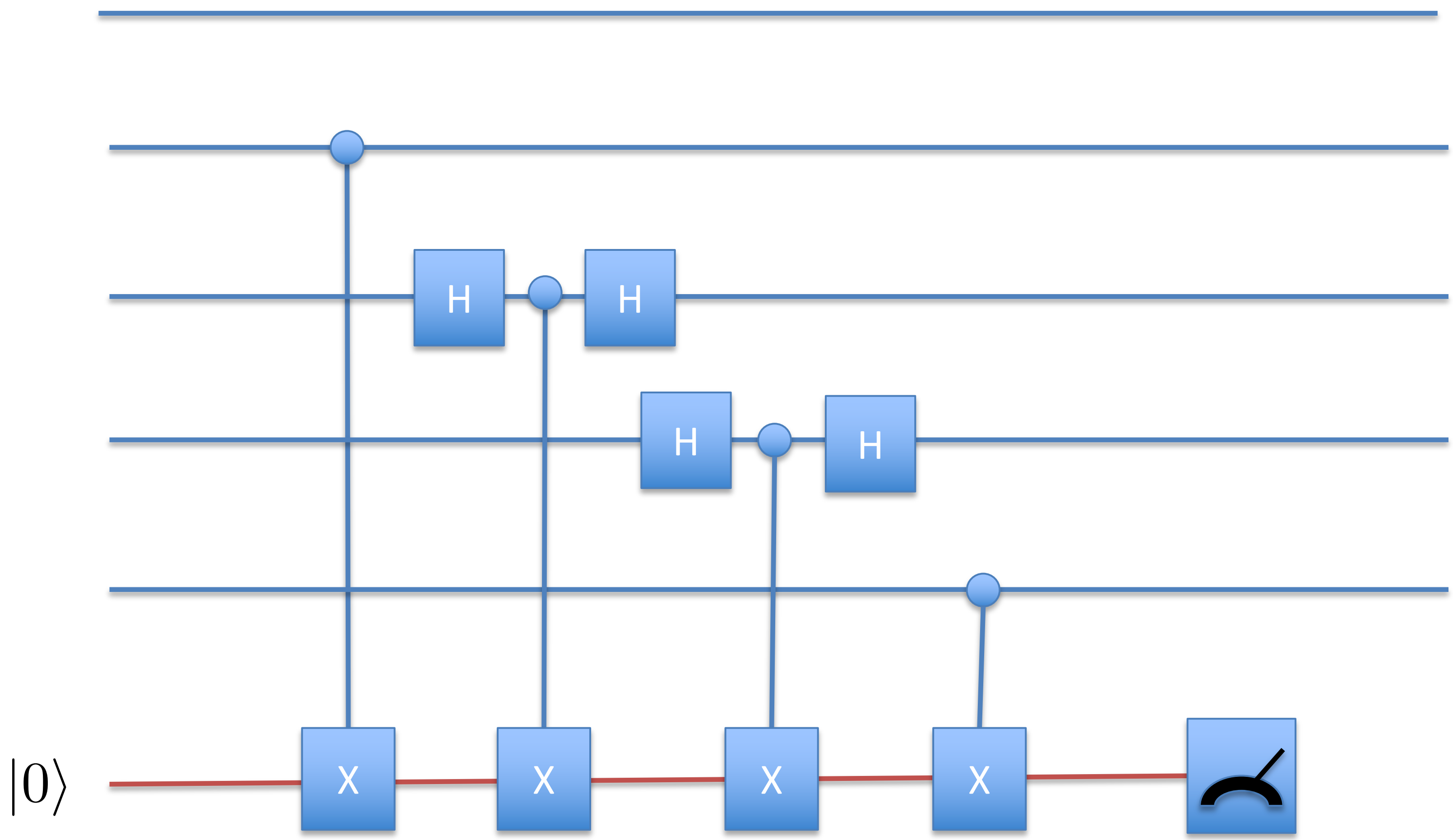
The stabilizers of this code are:

$$\begin{cases} IXZZX \\ XIXZZ \\ ZXIXZ \\ ZZXIX \end{cases}$$

Exercise: Write out all 15 single qubit errors and check that their syndromes are unique

Syndromes of the five qubit code

Construction of the syndrome circuits is easier to see. Eg. Measure IZXXZ:



Four different measurements required for the five qubit code.

Seven Qubit Steane Code

7 qubit "Steane" code. It is also known as the seven qubit "colour" code (which is a topological code – more next lecture). Stabilizers of this code are:

$$\left\{ \begin{array}{ccccccc} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \\ I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & I & Z & Z \\ Z & I & Z & I & Z & I & Z \end{array} \right.$$

Exercise: Check that every single qubit error produces a unique syndrome!

Logical States of the Steane code

Logical States

$$|0_L\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle$$
$$+ |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle$$
$$+ |111000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$$

We want to operate on these states while remaining protected Ie. without decoding.

Logical X Operator

$$X_L = XXXXXXX$$

(see this by operating directly on logical states above)

Logical Operators Commute with Stabilizers

Example Stabilizers:

$$\begin{cases} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \\ I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & I & Z & Z \\ Z & I & Z & I & Z & I & Z \end{cases}$$

Logical X:

$$X_L = XXXXXXX$$

-> X operators all commute with themselves, and even number of Z commute, so logical operator commutes with the stabilisers and so code states are stabilised by the logical X operator.

Other logical operators

$$|0_L\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle$$
$$+ |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle$$
$$+ |111000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$$

Logical 0 has zero or four 1's. Logical 1 has three or seven ones. So

$$Z_L = ZZZZZZZ$$

$$S_L = S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger$$

$$i^4 = 1, i^3 = -i$$

Other logical operators

$$|0_L\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle$$
$$+ |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle$$
$$+ |111000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$$

Logical 0 has zero or four 1's. Logical 1 has three or seven. So

$$Z_L = ZZZZZZZ$$

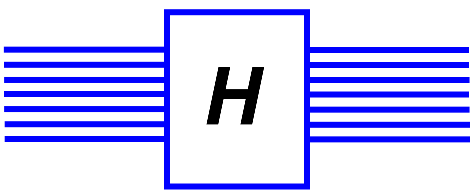
$$S_L = S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger$$

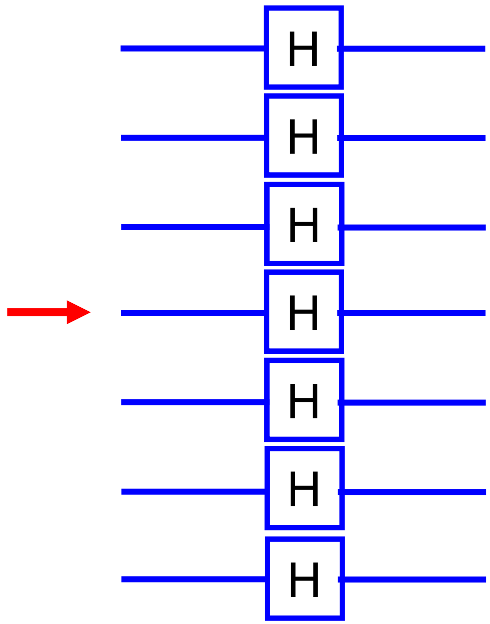
$$i^4 = 1, i^3 = -i$$

Transversal Gates

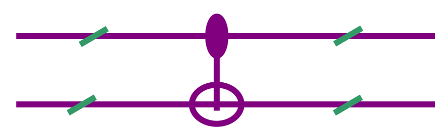
Other Steane code 7-qubit code gates include H and CNOT.

Transversal gates:

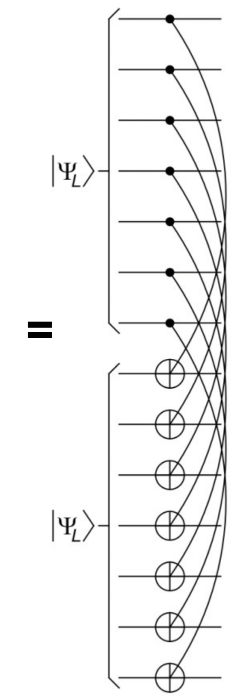




Hadamard on a single logical qubit



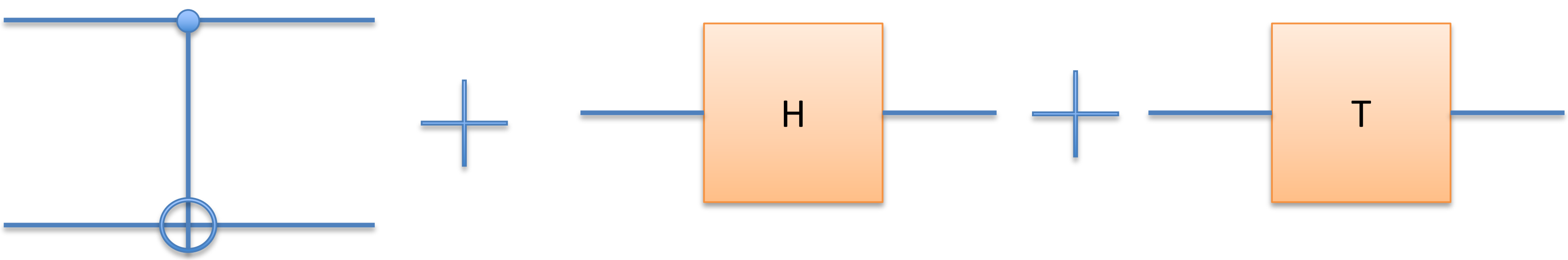
transversal CNOT



Can also implement the T gate (but this is not transversal)

Fault Tolerant Universal Gate Set

In quantum computing every quantum circuit can be expressed as a sequence of:



CNOT

Hadamard

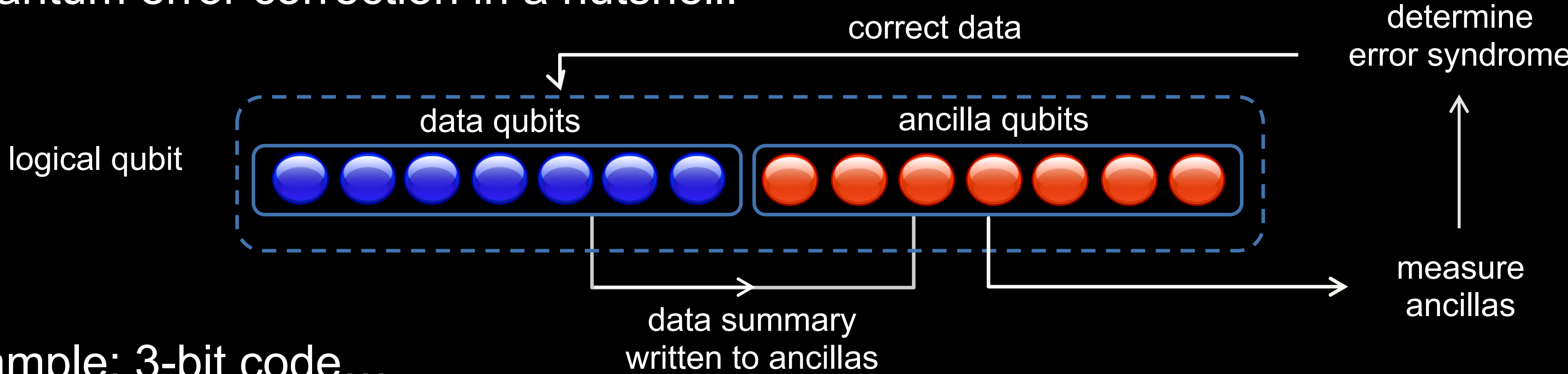
T Gate (pi/8)

These gates can be implemented "fault tolerantly" using quantum error codes.

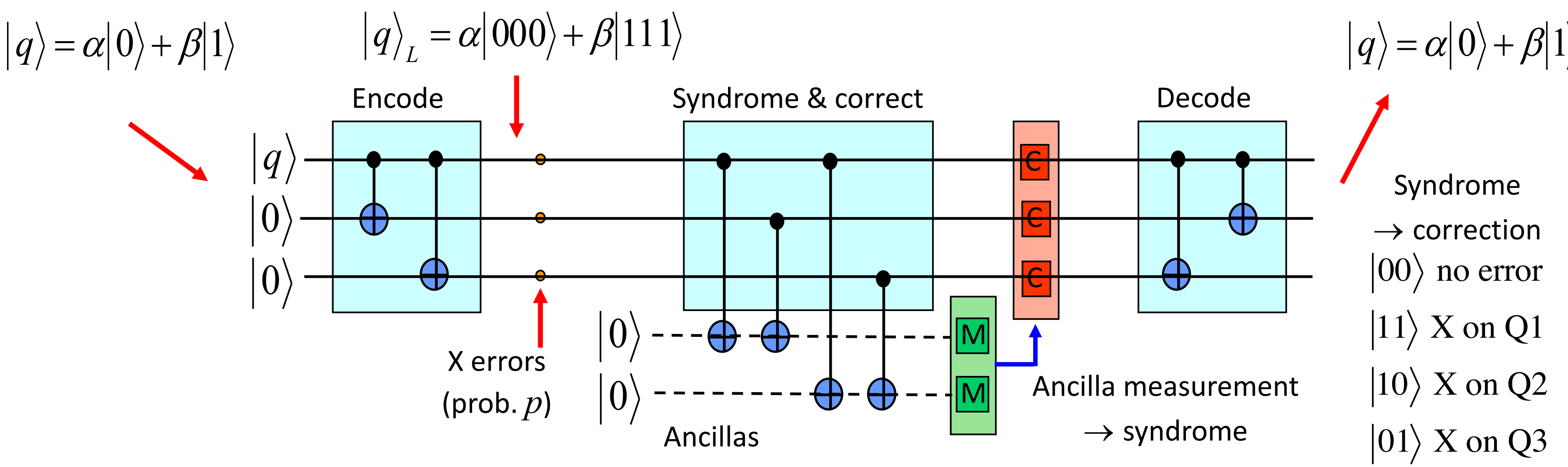
QEC Summary

THE UNIVERSITY OF MELBOURNE

Quantum error correction in a nutshell:



Example: 3-bit code…



Gottesman-Knill Theorem

We saw how generators from the Pauli group can be used to specified states. We can use this to track quantum states, so long as the operations we apply only take Paulis to other Paulis (ie. Clifford gates).

This method of simulating is **efficient.**

1) We only prepare computational basis states
2) Only apply CNOT, X, Y, Z, H, S gates (or things which can be generated from them)
3) Make measurements in the computational basis

This can be simulated efficiently on a classical computer

Put another way, T-gates make things hard!

Week 8

Lecture 15

Simple classical error correction codes, Quantum error correction codes, stabilizer formalism, 5-qubit code, 7-qubit Steane code

Lecture 16

The more advanced quantum error correction codes, Fault Tolerance, surface code.

Lab 8

Quantum error correction