
Tags:: [#on/Quantum-Computing](#)

Links:: [📄 INTRODUCTION TO CLASSICAL AND QUANTUM COMPUTING - THOMAS WONG](#)

In this chapter, we will learn about multi-qubit systems and how quantum gates can be used to perform computations.

Entanglion: A Quantum Computing Board Game

Mechanics

- We are introduced to the [ENTANGLION](#) game, and its mechanics here. Detailed instructions are available on the website, but a brief summary was provided in this section.

Connection to Quantum Computing

- The rules of [ENTANGLION](#) reflect how [QUANTUM COMPUTERS](#) work.
- The red and blue spaceships are [QUBITS](#).
- The planets are various states that qubits can be in.
- The engine cards *H*, *X*, *CNOT*, and *SWAP* are [QUANTUM GATE](#).
- Detection by planetary defenses corresponds to a [MEASUREMENT](#).

States and Measurement

Tensor Product

- When we have multiple qubits, we write their states as a tensor product \otimes , i.e. $|0\rangle \otimes |1\rangle$, which is pronounced as "zero tensor zero". Although, most of the times, we compress this notation and leave out the tensor product in both writing and speech: $|0\rangle|0\rangle$, or further still: $|00\rangle$.
- With 2 [QUBITS](#), the [Z-BASIS](#) is $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. A general state is a [SUPERPOSITION](#) of these basis states:

$$c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle.$$

Here, c_0, c_1, c_2, c_3 are the amplitudes of different basis states, the total probability is calculated as: $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2$ which must be equal to 1.

- With 3 **QUBITS**, there are 8 **Z-BASIS** states:
 $\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}$.
- Sometimes, these **BINARY STRINGS** are written as decimal numbers $|0\rangle, |1\rangle, |2\rangle, \dots, |7\rangle$. These can be converted using either **LITTLE ENDIAN** convention, or **BIG ENDIAN** convention. Here, we will be using the **LITTLE ENDIAN** convention.
- The general state of three qubits is a superposition of these basis vectors:

$$\sum_{j=0}^7 c_j |j\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_7|7\rangle,$$

and the probability of getting $|j\rangle$ when measuring in the Z -basis is $|c_j|^2$, so $\sum_j |c_j|^2 = 1$.

- With n qubits, there are $N = 2^n$ Z -basis states, which we can label as n -bit strings or by the decimal numbers 0 through $N - 1$.
- If we have just $n = 300$ **QUBITS**, then we must keep track of $N = 2^{300} \approx 2.04 \times 10^{90}$ amplitudes, which is more than the **NUMBER OF ATOMS IN THE VISIBLE UNIVERSE**. This is evidence, not proof, that **IT IS DIFFICULT FOR CLASSICAL COMPUTERS TO SIMULATE QUANTUM COMPUTERS**. It is evidence because classical computers cannot keep track of this many amplitudes, but it is not a proof because it is unknown whether quantum computers need all these amplitudes. That is, if quantum computers can function with much fewer amplitudes (a polynomial number instead of an exponential number in n), a classical computer would be able to keep track of all of them.
- A general multi-qubit state can't be represented in a Bloch sphere because of the many parameters.

Kronecker Product

- In **LINEAR ALGEBRA**, the **TENSOR PRODUCT** is simply the **KRONECKER PRODUCT**.
Then

$$c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}.$$

- With n qubits, the **VECTOR** has $N = 2^n$ elements:

$$|\psi\rangle = \sum_{j=0}^{N-1} c_j |j\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_{N-1}|N-1\rangle = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{pmatrix}.$$

- A general **QUANTUM STATE** of n -qubits, written as a bra, is:

$$\langle\psi| = \sum_{j=0}^{N-1} c_j^* \langle j| = c_0^* \langle 0| + c_1^* \langle 1| + \dots + c_{N-1}^* \langle N-1| = (c_0^* \quad c_1^* \quad \dots \quad c_{N-1}^*).$$

Measuring Individual Qubits

- Suppose we have two **QUBITS** in the state

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|01\rangle + \frac{\sqrt{3}}{4}|10\rangle + \frac{1}{4}|11\rangle.$$

Instead of measuring both qubits, if we only measure the left qubit, then the probability of getting $|0\rangle$ is given by the sum of the norm-squares of the amplitudes of those states that have a left qubit of $|0\rangle$, i.e. $|00\rangle$ and $|01\rangle$. So the probability of getting $|0\rangle$ as the left qubit is:

$$\left| \frac{1}{\sqrt{2}} \right|^2 + \left| \frac{1}{2} \right|^2 = \frac{3}{4}.$$

Similarly, if the outcome is $|1\rangle$, then from the $|10\rangle$ and $|11\rangle$ states, the probability is

$$\left| \frac{\sqrt{3}}{4} \right|^2 + \left| \frac{1}{4} \right|^2 = \frac{1}{4}.$$

- Now for the states after measurement, if the outcome is $|0\rangle$, then the state collapses to the parts where the left qubit is $|0\rangle$, so it becomes

$$A \left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|01\rangle \right),$$

where A is a normalization constant. Similarly, if the outcome is $|1\rangle$, then the state collapses to the terms where the left qubit is $|1\rangle$, so it becomes

$$B \left(\frac{\sqrt{3}}{4}|10\rangle + \frac{1}{4}|11\rangle \right),$$

where B is a normalization constant. Normalizing these, we get $A = \frac{2}{\sqrt{3}}$ and $B = 2$, so measuring the left qubit yields:

- $|0\rangle$ with probability $\frac{3}{4}$, and the state collapses to $\sqrt{\frac{2}{3}}|00\rangle + \frac{1}{\sqrt{3}}|01\rangle$
 - $|1\rangle$ with probability $\frac{1}{4}$, and the state collapses to $\frac{\sqrt{3}}{2}|10\rangle + \frac{1}{2}|11\rangle$.
- We can apply these ideas to any number of qubits.

Sequential Single-Qubit Measurements

- Measuring both qubits is the same as measuring one after another, assuming the state was not modified between the two measurements.
- If we take the qubit introduced in the **LAST SECTION**, the probability of getting $|00\rangle$ is the probability of first getting $|0\rangle$ for the left qubit, which was $3/4$, times the probability of getting $|0\rangle$ for the right qubit, which was $2/3$. Multiplying these, the probability of getting $|00\rangle$ is $(3/4)(2/3) = 2/4 = 1/2$, which is the same if we had measured both qubits at the same time.

Entanglement

Product States

- **PRODUCT STATES** are those states that can be factored into individual **QUBIT STATES**. For example,

$$\begin{aligned} \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= |+\rangle \otimes |-\rangle \\ &= |+\rangle|-\rangle. \end{aligned}$$

Entangled States

- **ENTANGLED STATES** are those **QUANTUM STATES** that cannot be factored into **PRODUCT STATES**. For example, with two **QUBITS**,

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),$$

cannot be written as $|\psi\rangle_1|\psi\rangle_0$.

Quantum Gates

One-Qubit Quantum Gates

- If we want to apply one **QUANTUM GATE** to the left-**QUBIT**, and another one to the right, we can do so by using a tensor product. e.g. to apply H gate on the left qubit and X gate on the right qubit, we would write:

$$(H \otimes X)|0\rangle|0\rangle = |+\rangle|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle).$$

- **ONE-QUBIT QUANTUM GATES** are unable to create **ENTANGLED STATES** because each qubit evolves independently of the others.

Two-Qubit Quantum Gates

- **CNOT GATE** inverts the right **QUBIT** (*target qubit*) if the left qubit (*control qubit*) is 1.
- The **CONTROLLED-U GATE** applies some **QUANTUM GATE** U to the right **QUBIT** if the left qubit is 1.
- The **SWAP GATE** simply swaps the two qubits.

Toffoli Gate

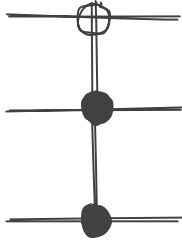
- The **TOFFOLI GATE** is a three-qubit **QUANTUM GATE** that flips the right **QUBIT** if the left and middle qubits are 1 :

$$\begin{aligned}\text{Toffoli}|000\rangle &= |000\rangle, \\ \text{Toffoli}|001\rangle &= |001\rangle, \\ \text{Toffoli}|010\rangle &= |010\rangle, \\ \text{Toffoli}|011\rangle &= |011\rangle, \\ \text{Toffoli}|100\rangle &= |100\rangle, \\ \text{Toffoli}|101\rangle &= |101\rangle, \\ \text{Toffoli}|110\rangle &= |111\rangle, \\ \text{Toffoli}|111\rangle &= |110\rangle.\end{aligned}$$

- **AS WE KNOW** that the Toffoli gate is universal for **CLASSICAL COMPUTING**, and since it is a **QUANTUM GATE**, quantum computers can efficiently do everything a classical computer can efficiently do.
- Toffoli gate, represented as a **MATRIX**:

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

- Its circuit diagram is similar to the **CNOT GATE**:



No-Cloning Theorem

- We can copy a known **QUANTUM STATE** easily, but the issue arises when we want to copy/clone an unknown quantum state.
- The No-Cloning theorem simply states/proves that the quantum information can not generally be cloned.

Quantum Adders

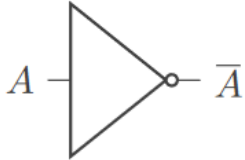
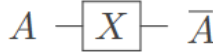
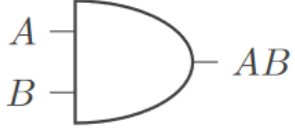
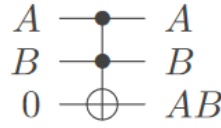
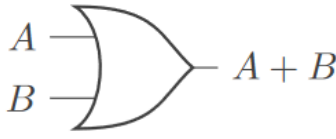
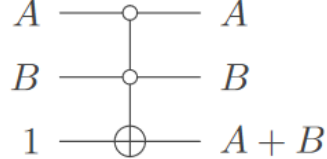

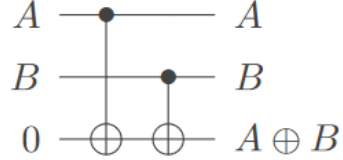
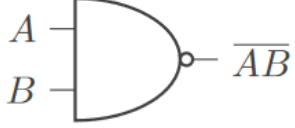
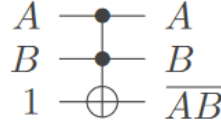
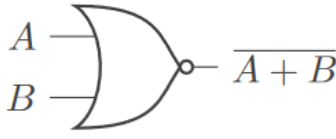
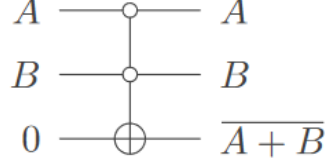
Classical Adder

- Classically, we can add **BINARY NUMBERS** using the **RIPPLY-CARRY ADDER**.

Making the Classical Adder a Quantum Gate

- We can convert a full-adder to a quantum gate in several ways.
- We can turn the full adder into a reversible circuit by taking the **XOR** of each of its outputs with an extra bit.
- We can replace all of the gates (two XOR, two AND, and one OR) in full-adder by **NAND** gates, implement each NAND gate using a **TOFFOLI GATE**.
- Or instead of first converting each gate to NAND gate, we can directly convert those classical logic gates to their **QUANTUM** counterparts, as shown in the table

below:

	Classical		Reversible/Quantum
NOT		X-Gate	
AND		Toffoli	
OR		anti-Toffoli	
XOR		CNOTs	
NAND		Toffoli	
NOR		anti-Toffoli	

© Chapter 4 - Multiple Quantum Bits#Making the Classical Adder a Quantum Gate

- The extra bits that are used during the calculation are called *ancilla bits* or **ANCILLARY BITS**, and in quantum circuits, they should be cleaned up by turning them back into zeros, so they can be reused later on and don't cause unintended **ENTANGLEMENTS**.
- One method for cleaning up **ANCILLARY BITS** is called *uncomputation*, where we apply in reverse order the inverses of the **GATES** that were used to calculate the ancillas.

Quantum Setup

- We can use two quantum registers, $|a\rangle$ and $|b\rangle$, to encode the binary numbers. One way to add them reversibly is to replace $|b\rangle$ with the sum:

$$|a\rangle|b\rangle \rightarrow |a\rangle|s\rangle,$$

where $s = a + b$.

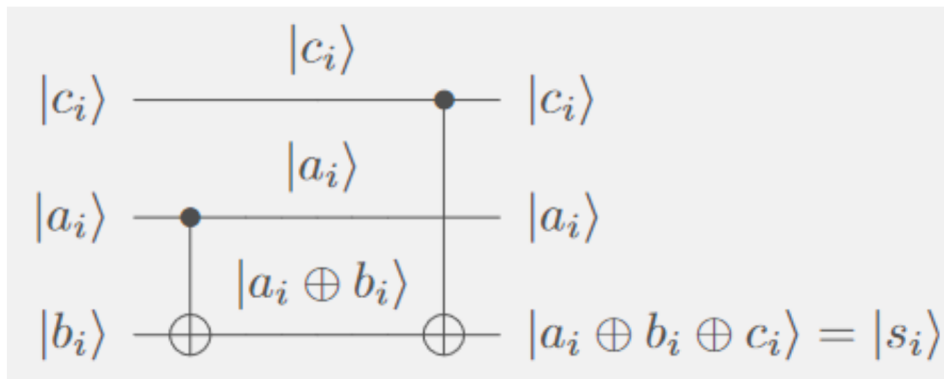
– In the intermediate steps of the computation, the quantum adder also needs to keep track of carry bits.

– Our quantum adder should map $|a\rangle|b\rangle|c\rangle \rightarrow |a\rangle|s\rangle|c\rangle$ where :

$$\begin{aligned} |a\rangle &= |a_{n-1}\rangle \dots |a_1\rangle|a_0\rangle, \\ |b\rangle &= |b_n = 0\rangle|b_{n-1}\rangle \dots |b_1\rangle|b_0\rangle, \\ |c\rangle &= |c_{n-1}\rangle \dots |c_1\rangle|c_0\rangle. \end{aligned}$$

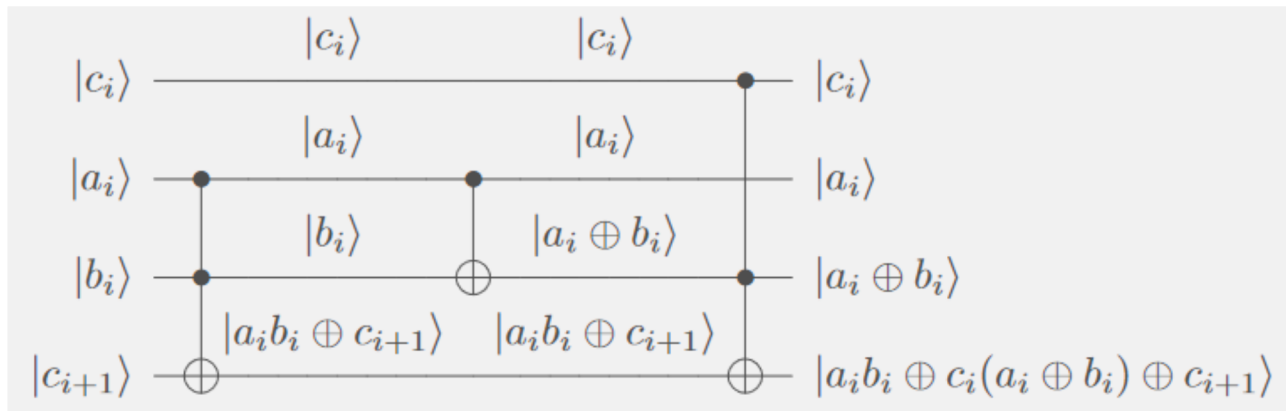
Quantum Sum

– We can implement the sum using two CNOT GATES.



Quantum Carry

– The quantum carry circuit is:



Quantum Ripple-Carry Adder

- For a detailed explanation along with circuit diagrams, directly read the section in the book.
- Draper's adder implemented in [QUIRK](#) and used to add $|a\rangle = |0111\rangle$ and $|b\rangle = |0011\rangle$. [Link](#) to circuit.

Circuit Complexity

- The quantum ripple-carry adder uses $4n - 2$ [TOFFOLI GATES](#) and $4n$ [CNOT](#) gates, which is linear in n , i.e., $\Theta(n)$, so the algorithm is *efficient*.

Adding in Superposition

- Our quantum ripple-carry adder circuit can be used to act on [SUPERPOSITIONS](#). It will calculate the results for both, but we will only be able to get/measure a certain result based on some probability.
- It is incorrect to think of a quantum computer as a massively parallel classical computer because we must measure and only get one result. It might be best to avoid the term "*parallel*" altogether when describing quantum computing.

Universal Quantum Gates

Definition

- A set of [QUANTUM GATES](#) that allows us to approximate any quantum gate to any desired precision is called a [UNIVERSAL GATE SET](#).
- Based on the context, we can infer if we're talking about classical or quantum universal gate sets.

Components of a Universal Gate Set

- [SUPERPOSITION](#): We must be able to produce superpositions.
- [ENTANGLEMENT](#): We must be able to entangle qubits. A [QUANTUM GATE](#) must act on at least two [QUBITS](#) to produce entanglement.
- Complex amplitudes: [CNOT GATE](#) and [H GATE](#) only contain real numbers, so they do not produce states with complex amplitudes.
- Contain more than the [CLIFFORD GROUP](#). Because of the [GOTTESMAN-KNILL THEOREM](#), Clifford group is only as powerful as a classical computer.

- It is unknown if these are sufficient requirements for a set of quantum gates to be universal. It may be that a set satisfies all of these properties, but is still not universal.

Examples of Universal Gate Sets

- {CNOT, ONE-QUBIT QUANTUM GATES}
- {CNOT, H, T}
- {CNOT, $R_{\pi/8}$, S}
- {TOFFOLI, H, S}
- HADAMARD GATE plus almost any two-qubit unitary.

Solovay-Kitaev Theorem

- The SOLOVAY-KITAEV THEOREM says that with any universal gate set, we can approximate a quantum gate on n QUBITS to precision ϵ using $\Theta(2^n \log^c(1/\epsilon))$ gates for some constant c .
- The dependence on the number of QUBITS 2^n is expected because an operator on n qubits is a MATRIX of $2^n \times 2^n$ entries. The dependence on the precision $\log^c(1/\epsilon)$ is great. The precision ϵ is the "distance" between the approximate quantum gate and the actual quantum gate, which we want to be small. So $1/\epsilon$ is big, but taking the LOGARITHM of it makes it small. The POLYLOG is also considered small. Thus this dependence means our approximation quickly converges on the actual quantum gate.

Quantum Computing without Complex Numbers

- We can express any COMPLEX NUMBER as two real numbers (x, y) and keep track of the fact that they play different roles.
- So, we can formulate all of quantum computing just in terms of real numbers. Then, a UNIVERSAL SET OF QUANTUM GATES technically does not need to produce states with complex amplitudes. For example, the following sets are also universal for quantum computing:
 - {TOFFOLI, any SINGLE-QUBIT BASIS-CHANGING GATE}
 - The CONTROLLED-HADAMARD GATE {CH}
 - {CNOT, any SINGLE-QUBIT GATE whose square is basis-changing}

Quantum Error Correction

Decoherence

- A **QUBIT** can experience full *bit flip errors* (rotations about the x -axis by $\pi = 180^\circ$) as well as partial bit flip errors (rotations about the x -axis by some angle).
- A qubit can also experience *phase flip errors* (rotations around the z -axis).
- **DECOHERENCE** is the process of small interactions of the **QUBIT** with the environment that move the qubit to a different location on the **BLOCH SPHERE**. This is the biggest obstacle in building large quantum computers.

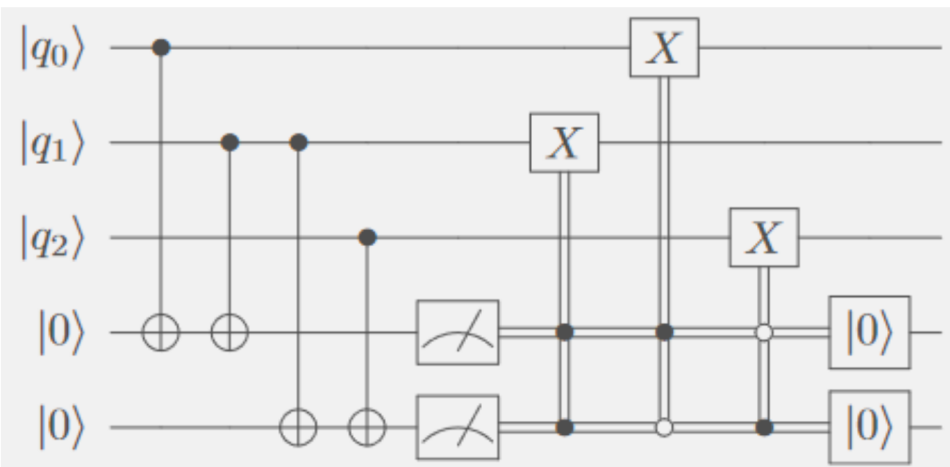
Bit-Flip Code

- We use three physical **QUBITS** to encode each **LOGICAL QUBIT**:

$$|0_L\rangle = |000\rangle, \quad |1_L\rangle = |111\rangle.$$

In general, a **LOGICAL QUBIT** is a **SUPERPOSITION** of $|0_L\rangle$ and $|1_L\rangle$: $\alpha|0_L\rangle + \beta|1_L\rangle$.

- Instead of measuring the qubit, which would collapse its superposition state, we use the **PARITY** of adjacent **QUBITS** to determine if a bit flip error has occurred. We can use **CNOT** gates to calculate the parity of qubits, and if a bit flip error is detected, we can apply **X GATE** to correct it.
- In case of partial bit flips, we use the following quantum circuit:



The first four columns are the **CNOTS** that calculate the parities of adjacent qubits. Then, we measure these parities, as shown by the meter symbols, which results in classical bits. We denote these classical bits/wires using double lines. We end with three **X GATES** conditioned on these classical bits/parities. If both parities are 1,

then q_1 flipped, so we apply an **X GATE** to it to correct it. If $\text{parity}(q_2, q_1) = 0$ and $\text{parity}(q_1, q_0) = 1$, then q_0 flipped, so we apply an X gate to it to correct it. Finally, if $\text{parity}(q_2, q_1) = 1$ and $\text{parity}(q_1, q_0) = 0$, then q_2 flipped, so we apply an X gate to it to correct it. We end by resetting the **ANCILLAS** to $|0\rangle$, indicated by the boxes with $|0\rangle$ in them.

- The **PRINCIPLE OF DEFERRED MEASUREMENT** states that intermediate measurements that are used to control operations can be moved after the operations, and the controls can be replaced by quantum controls.
 - Phrased another way, we can collapse and then do the controlled operations, or we can do the controlled operations in superposition, and then collapse.

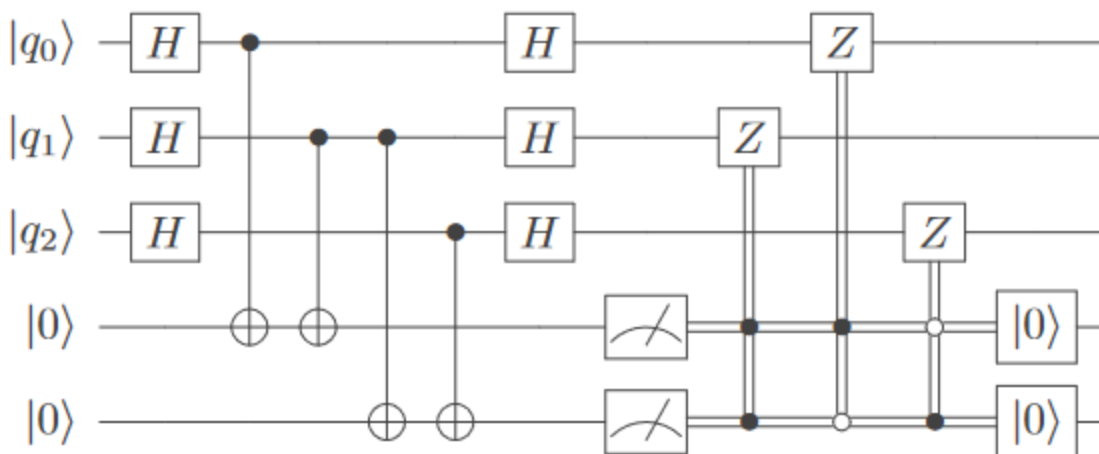
Phase-Flip Code

- For phase-flip errors, we use **LOGICAL QUBITS** using three $|+\rangle$ and $|-\rangle$, i.e.

$$|0_L\rangle = |+++ \rangle, \quad |1_L\rangle = |-- - \rangle,$$

so a general superposition is $\alpha|0_L\rangle + \beta|1_L\rangle$.

- To detect the phase flip error, we measure the parity of consecutive **QUBITS** in the **X-basis**, and then apply a **Z-GATE** if an error occurred.
- When we have a partial phase flip, the measurement forces it to be corrected or to become a complete bit flip, which we can correct by applying a **Z-GATE**. The quantum circuit for this procedure is shown below:



📍 Chapter 4 - Multiple Quantum Bits#Phase-Flip Code

Shor Code

- Shor code combines the phase-flip code and bit-flip code to correct both kinds of errors.
- We start off with the phase-flip code, i.e.

$$\begin{aligned}
 |0_L\rangle &= |+++ \rangle \\
 &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
 &= \frac{1}{2^{3/2}}(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle).
 \end{aligned}$$

Next, to correct the bit-flip error, we use bit-flip encoding to replace each of the three qubits, i.e., $|0\rangle \rightarrow |000\rangle$ and $|1\rangle \rightarrow |111\rangle$, so that each **LOGICAL QUBIT** is encoded using nine physical qubits:

$$|0_L\rangle = \frac{1}{2^{3/2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle).$$

Same goes for the $|1_L\rangle = |--\rangle$, i.e.

$$|1_L\rangle = \frac{1}{2^{3/2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle).$$

- Following that, the state of a general **LOGICAL QUBIT** is

$$\begin{aligned}
 \alpha|0_L\rangle + \beta|1_L\rangle &= \frac{\alpha}{2^{3/2}} \frac{1}{2^{3/2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \\
 &\quad + \frac{\beta}{2^{3/2}} \frac{1}{2^{3/2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle).
 \end{aligned}$$

- This encoding is called the **SHOR CODE**, named after its inventor, **PETER SHOR**.

Summary

- The state of multiple **QUBITS** is written as as a **TENSOR PRODUCT**.
- With n qubits, there are 2^n orthonormal basis states, and a general state is a **SUPERPOSITION** of these basis states.
- In a **PRODUCT STATE**, measuring one qubit cannot affect the others, while in an **ENTANGLED STATE**, measuring one qubit can affect the other qubits.
- A **QUANTUM GATE** on n qubits is a $2^n \times 2^n$ **UNITARY MATRIX**.
- A **UNIVERSAL SET OF QUANTUM GATES** can approximate any **QUANTUM GATE** to any desired precision.

- Quantum bits can suffer from both bit-flip and phase-flip errors, but they can be corrected.