QuIC Seminar 1

The Circuit Model of Quantum Computing

Contents

1.1 Quantum Bits

On classical computers, information is stored in a *binary digit*, known as a *bit*. The values of a bit can be either 1 or 0. Depending on the context, we may respectively think of these values as "on" or "off," "true" or "false," "high voltage" or "low voltage." Although classical computers can perform extremely sophisticated and complex tasks, at the lowest level they all work by processing bits.

Quantum computers store information differently. Rather than storing information in binary digits, quantum computers store information in the state of a two-level quantum system known as a *quantum bit*, or *qubit* for short. The formal definition of a qubit depends on whether you ask a physicist, mathematician, or computer scientist. We offer three equivalent definitions below.

Definition 1.1 (Quantum Bit (Qubit)). Physically, a **quantum bit**, or **qubit**, is a closed quantum system that has two energy levels, the ground state (lower energy level) $|0\rangle$ and the first excited state $|1\rangle$.

Mathematically, a qubit $|\psi\rangle$ is a two-element unit vector over the complex numbers—i.e.,

$$|\psi\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix} \tag{1.1}$$

such that $|\alpha|^2 + |\beta|^2 = 1$ where $|\cdot|$ denotes the modulus $|\alpha| = \alpha^*\alpha$.

Computationally, a qubit is the fundamental unit of information on a quantum computer.

In linear algebra, we have the notion of a *basis* for a vector space. The vector space (more specifically, Hilbert space) for qubits is $\mathbb{C}^2$. The states of a qubit $|0\rangle$ and $|1\rangle$ form a basis for this Hilbert space, known as the *computational basis*.

Definition 1.2 (Computational Basis). The **computational basis**, known as the standard or natural basis in other contexts, consists of the vectors $|0\rangle, |1\rangle \in \mathbb{C}^2$ where

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1.2}$$

With the definition of the computational basis, it is a well-known result from linear algebra that any arbitrary vector can be written as a linear combination of basis vectors. We state this result as a formal theorem below.

Theorem 1.1 (Qubit as Linear Combination of Basis Vectors). Any qubit $|\psi\rangle = [\alpha\ \beta]^T$ can be written as a linear combination of computational basis vectors, i.e.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{1.3}$$

Proof. Immediate from (1.2). □

What is Theorem 1.1 saying? Precisely what we said was different between quantum bits and classical bits before—quantum bits can exist in basis states or linear combinations of basis states. We also refer to such linear combinations in quantum mechanics as *superpositions*. Indeed, the superposition principle of quantum mechanics states:

Theorem 1.2 (The Superposition Principle). Let $|\psi_1\rangle$ and $|\psi_2\rangle$ be solutions of the Schrödinger equation

$$i\hbar\frac{\partial\Psi}{\partial t} = H\Psi. \tag{1.4}$$

Then, any linear combination $\alpha|\psi_1\rangle + \beta|\psi_2\rangle$, where $\alpha, \beta \in \mathbb{C}$, is also a solution.

Proof. Immediate from substitution. □

The superposition principle is just one of several reasons why quantum bits are different than classical bits. Subsequently, we will see *entanglement*, which concerns how multiple qubits interact with each other. For now, we will stick to the state of one qubit.

First, we remark that the notation $|\cdot\rangle$ is known as a "ket." The conjugate transpose of a ket, denoted $\langle\cdot|$, is known as a "bra." (This terminology is due to Dirac's bra-ket notation.) A ket is a column vector, whereas a bra is a row vector. Anything within the bra-kets serves as a label for a state, just as anything in bold (or underneath a directed arrow) serves as a label for a vector in mathematical contexts.

Definition 1.3 (Bras, Kets, Conjugate Transposes). Let $|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ be a **ket** (column vector). Then, the **conjugate transpose** (denoted by a dagger †) of $|\psi\rangle$ is a **bra** (row vector) given by

$$\langle\psi| := |\psi\rangle^\dagger = [\alpha^*\ \beta^*]. \tag{1.5}$$

Here, as in other places, $*$ denotes complex conjugation $(a + bi)^* = a - bi$.

We are often concerned about the overlap between two quantum states, for which we define the inner product.

Definition 1.4 (Overlap). The **overlap** between two quantum states $|\psi\rangle$ and $|\phi\rangle$ is defined as $|\langle\psi|\phi\rangle|$. Note that order matters in the **inner product** since $\langle\psi|\phi\rangle = \langle\phi|\psi\rangle^*$.

Two quantum states with zero overlap are said to be **orthogonal**.

Example 1: Computational basis vectors are orthogonal.
The computational basis vectors are orthogonal (as required for a basis). Indeed, it can be easily verified by direct computation that $\langle 0|1\rangle = \langle 1|0\rangle = 0$.

We can get the magnitude of a quantum state by computing the overlap with itself.

Definition 1.5 (Norm). The **norm**, or **magnitude**, of a quantum state $|\psi\rangle$ is defined as $\langle\psi|\psi\rangle$.

Note that any qubit has unit norm by Definition 1.1. For any state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, it is easy to verify that $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2$. Indeed,

$$\begin{aligned}\langle\psi|\psi\rangle &= (\alpha|0\rangle + \beta|1\rangle)^\dagger (\alpha|0\rangle + \beta|1\rangle) \\ &= (\alpha^*\langle 0| + \beta^*\langle 1|) (\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha^*\alpha\langle 0|0\rangle + \beta^*\beta\langle 1|1\rangle \\ &= |\alpha|^2 + |\beta|^2.\end{aligned}$$

Exercise 1: Is the quantum state $|0\rangle + |1\rangle$ normalized? If not, what is the normalization factor?

Solution 1: The state is not normalized, as it has magnitude $\sqrt{|1|^2 + |1|^2} = \sqrt{2}$. Hence, the normalization factor should be $1/\sqrt{2}$.

1.1.1 Visualizing Single Qubit States on the Bloch Sphere

Because qubits are normalized, we may write (1.3) equivalently as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{1.6}$$

where $0 \le \theta \le \pi$ and $0 \le \phi < 2\pi$.

Exercise 2: Verify that (1.6) has unit norm.

Solution 2: We have that $\langle\psi|\psi\rangle = \cos^2\theta/2 + e^{i\phi}e^{-i\phi}\sin^2\theta/2 = \cos^2\theta/2 + \sin^2\theta/2 = 1$.

Why does only one term in (1.6) have a *phase term* $e^{i\phi}$? It is because of the following theorem.

Theorem 1.3 (Global phase of quantum states is irrelevant.). The quantum states $|\psi\rangle$ and $e^{i\gamma}|\psi\rangle$ are equivalent in that they have the same norm and produce the same measurement distribution. We often say that these states are **equal up to global phase**.

Proof. Follows immediately from direct computation using the observation that $\left(e^{i\gamma}\right)^* = e^{-i\gamma}$. □

Because of this theorem, if there were a phase term on the $|0\rangle$ state in (1.6), we could factor it out to write a new phase $e^{i\phi'}$ on the $|1\rangle$ state. This new state would be equal to the old one up to global phase. Hence, we only write a *relative phase* on one of the terms in (1.6).

This representation leads to a convenient visualization of qubits. By interpreting θ and ϕ as polar and azimuthal angles in spherical polar coordinates, the state of any qubit can be seen to lie on the surface of a unit sphere in $\mathbb{C}^2$, known as the **Bloch sphere** and shown in Figure 1.1.

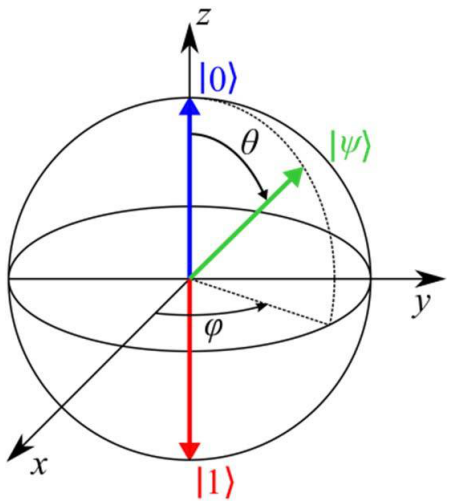

Figure 1.1: The Bloch sphere representation of a single qubit state.

1.2 Quantum Gates

We now have an understanding of how information is stored in a quantum computer. The way it is processed is by means of *quantum gates*. Before we discuss these in detail, it is useful to think of the classical analogues, i.e. logic gates. Logic gates in classical computers act on one or more bits to produce one or more bits of output. For example, the NOT gate flips the truth value of a classical bit. That is,

$$\text{NOT}(0) = 1 \qquad \text{and} \qquad \text{NOT}(1) = 0. \tag{1.7}$$

Other classical gates include the OR gate, which inputs two bits and evaluates to true if either of the input bits is true. Yet another classical gate is the AND gate, which inputs two bits and evaluates to true if both of the input bits is true.

A direct analogue to the classical NOT gate exists in quantum computer programming, namely the Pauli-X gate (also called the quantum NOT gate). This gate has the action that

$$X|0\rangle = |1\rangle \qquad \text{and} \qquad X|1\rangle = |0\rangle. \tag{1.8}$$

Quantum gates that act on a single qubit are known as, not suprisingly, *single qubit gates*. Gates that act on more than one qubit (multiple qubits) are known as *multi-qubit* gates. For instance, the CNOT gate is a multi-qubit gate that acts on two qubits, also called more simply a *two-qubit gate*. The Toffoli gate is an example of a *three-qubit gate*.

Definition 1.6 (Quantum Gate.). An n-qubit **quantum gate** is a unitary operator on n qubits.

For now, we'll restrict our discussion to single-qubit gates (i.e., $n = 1$). Why do we impose the restriction that quantum gates must be unitary operators? First, recall that an *operator* is a transformation (function) where the domain and co-domain are the same. That quantum gates are operators means that they send qubits to qubits. Why unitary? It is because of the normalization condition of qubits $\langle\psi|\psi\rangle = 1$. Suppose we evolve the qubit $|\psi\rangle$ under the operator U so that the new state is $U|\psi\rangle$. Then, the norm of the new state is given by $\langle\psi|U^\dagger U|\psi\rangle$. If U is unitary, then $U^\dagger U = UU^\dagger = I$ (the identity operator) so that $\langle\psi|U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle = 1$. Thus, unitary operators preserve norms.

Because quantum gates are unitary operators, there exists a matrix representation for them. Conventionally, matrix representations are given in the computational basis. Here, we recall from linear algebra that the ith column of an operator T is the action of T on the ith basis state—in Dirac notation, $T|i\rangle$.

Example 2: Matrix representation of X in the computational basis.
From (1.8), we know that the first column of X must be $|1\rangle$ (because $X|0\rangle = |1\rangle$). Similarly, the second column of X must be $|0\rangle$. Thus, a matrix representation for X in the computational basis is

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{1.9}$$

We emphasize that matrix representations depend *entirely* on what basis one is using.

Exercise 3: This exercise shows how matrix representations of quantum gates depend on the underlying basis.

(a) Verify that $\{|+\rangle, |-\rangle\}$, where $|+\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$ and $|-\rangle := (|0\rangle - |1\rangle)/\sqrt{2}$, form a basis for qubits (i.e., a basis for $\mathbb{C}^2$). This basis is known as the *+/- basis* or *Hadamard basis*.

(b) Prove that $X|+\rangle = |+\rangle$ and that $X|-\rangle = -|-\rangle$.

(c) Argue that the matrix representation for X in the Hadamard basis $\{|+\rangle, |-\rangle\}$ is

$$X_{HB} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{1.10}$$

where here the subscript HB emphasizes that we're in the Hadamard basis.

Solution 3: Follows from direct computation using (1.8).

Unless otherwise stated, all matrix representations of quantum gates will be given in the computational basis.

1.2.1 Common Single Qubit Gates

The quantum NOT gate X is one of four Pauli operators. These operators are given as follows.

Definition 1.7 (Pauli Operators). The **Pauli operators** in the computational basis are defined by

$$\sigma_0 \equiv I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \sigma_1 \equiv \sigma_X \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \sigma_2 \equiv \sigma_Y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \ \sigma_3 \equiv \sigma_Z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1.11}$$

Exercise 4: Prove that all four Pauli operators are unitary. Also prove that they are Hermitian.

Exercise 5: Compute the eigenvalues and eigenvectors of the Pauli operators.

Exercise 6: Prove that the Pauli operators are *self-inverse*, i.e. $\sigma_j^2 = I$ for all $j = 0, ..., 3$.

The Pauli operators (gates) lead to rotation operators when exponentiated. First, recall that the exponential of a matrix A is given by

$$e^A := I + A + A^2/2! + A^3/3! + \cdots = \sum_{n=0}^{\infty} A^n/n!. \tag{1.12}$$

(It can be proved that this infinite sum converges.) Supposing that A is self-inverse, i.e. $A^2 = A$, we can write (1.12) as

$$e^A = (1 + 1/2! + 1/4! + \cdots)I + (1 + 1/3! + 1/5! + \cdots)A. \tag{1.13}$$

If we instead consider e^{iAt} for some real parameter t, then we obtain

$$e^{iAt} = \cos(t)I + i\sin(t)A. \tag{1.14}$$

by recalling the Taylor series of $\cos(x)$ and $\sin(x)$ about $x = 0$.

Since Pauli operators are self-inverse, we can exponentiate them to obtain corresponding *single qubit rotation operators*.

Definition 1.8 (Single Qubit Rotation Gates.). The **single qubit rotation gates** are defined by

$$e^{-i\sigma_j\theta/2} = \cos(\theta/2)I + i\sin(\theta/2)\sigma_j \tag{1.15}$$

where $j = 1, 2, 3$, σ_j is a Pauli operator, and $0 \leq \theta < 2\pi$. Explicitly,

$$R_X(\theta) \equiv e^{-iX\theta/2} = \cos(\theta/2)I + i\sin(\theta/2)X = \begin{bmatrix} \cos\theta/2 & -i\sin\theta/2 \\ -i\sin\theta/2 & \cos\theta/2 \end{bmatrix}, \tag{1.16}$$

$$R_Y(\theta) \equiv e^{-iY\theta/2} = \cos(\theta/2)I + i\sin(\theta/2)Y = \begin{bmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{bmatrix}, \tag{1.17}$$

$$R_Z(\theta) \equiv e^{-iZ\theta/2} = \cos(\theta/2)I + i\sin(\theta/2)Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \tag{1.18}$$

There are many single qubit gates, but all can be simply expressed as a product of three rotation operators. We'll prove more theorems about quantum gates in upcoming seminars. For now, we highlight one of the most common gates used in quantum algorithms, the Hadmard gate H.

Definition 1.9 (Hadamard gate.). The **Hadamard gate** is a single-qubit gate with the action $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$.

Exercise 7: Verify that a matrix representation for the Hadamard gate is

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1.19}$$

in the computational basis.

Exercise 8: Prove that the Hadamard gate is hermitian, unitary, and self-inverse.

The Hadamard gate is useful in that it produces an equal superposition of basis states $|0\rangle$ and $|1\rangle$. This means that, if we perform the Hadamard gate and make a measurement, we have equal probability of measuring the zero state or the one state.

1.3 Measurements

In quantum mechanics, measuring a state means projecting it onto a given basis. Just as specifying matrix representations for quantum gates, measurements are basis dependent. If no basis is specified, "measuring" means "measuring in the computational basis." Here we offer a high-level discussion of what it means to make a measurement. The theory of quantum measurements is rather rich, and we will see more of it in upcoming seminars.

Because we project onto basis states when we make a measurement, we will only ever get two values (since there are two basis states for qubits, $|0\rangle$ and $|1\rangle$.) When we measure $|0\rangle$, we say we get the outcome 0, which is a classical bit. When we measure $|1\rangle$, we say that we get the outcome 1, which is again a classical bit. (This is why we use the labels "0" and "1" in the bra-kets.)

How do we know what outcome we will get? This comes down to the probabilistic interpretation of quantum mechanics due to Max Born. We restate *Born's Law* (also called *Born's Rule*) in the context of quantum computing as follows.

Definition 1.10 (Born's Law). Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ be a qubit. Then, the probability of measuring 0, $P(0)$, is

$$P(0) := \langle\psi|0\rangle\langle 0|\psi\rangle = |\langle 0|\psi\rangle|^2 = |\alpha|^2 \tag{1.20}$$

and the probability of measuring 1, $P(1)$, is similarly

$$P(1) := \langle\psi|0\rangle\langle 0|\psi\rangle = |\langle 1|\psi\rangle|^2 = |\beta|^2. \tag{1.21}$$

1.4 An Example Quantum Algorithm

These ingredients—qubits, gates, and measurements—are the building blocks of the *circuit model of quantum computing*. The circuit model consists of a collection of gates that act on a collection of qubits with measurements made on some or all qubits, known as a *quantum algorithm*. Other models of quantum computing exist, namely the adiabatic model of quantum computing. This model will not be discussed here but can be shown to be equivalent to the circuit model. The circuit model is sometimes also called the *gate model* of quantum computing or *digital quantum computing*.

There is a convenient graphical description for quantum algorithms that is preferable in many cases to mathematical formulas. Here we present *circuit diagrams* through a simple quantum algorithm known as the *quantum random bit generator*. In a circuit diagram, it looks like the following:

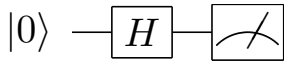

Figure 1.2: Quantum circuit diagram showing the random bit generator quantum algorithm.

This diagram says the following: first, start with a qubit in the $|0\rangle$ state; second, perform the Hadamard gate H on the qubit; and third, measure the qubit (in the computational basis). A single line in a quantum circuit diagram represents the evolution of a qubit (throughout time). Gates are denoted with specific symbols that usually refer to mathematical symbols used for the gates. Here, we represent the single qubit Hadamard gate with a box that has an H inside. Measurements are denoted by the "meter" symbol shown at the far right of Figure 1.2. Mathematically, we would represent this circuit simply as $H|0\rangle$ (and then measure).

How does the quantum algorithm produce a random bit? Recall from Definition 1.9 that $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. From Born's Law (Definition 1.10, when we measure this state in the computational basis we will get 0 with probability $P(0) = |1/\sqrt{2}|^2 = 1/2$ and we will get 1 with probability $P(1) = |1/\sqrt{2}|^2 = 1/2$. That is, we will get the outcome 0 or 1 uniformly at random. Thus, this circuit generates a random bit.

Exercise 9: Using n qubits, generalize the random bit generator to produce a random *number* between 0 and 2^n.

Exercise 10: Design a quantum algorithm that produces a 0 with probability $P(0) = 0.7$ and a 1 with probability $P(1) = 0.3$.