Bilkent University
Spring 2001
PHYS/EEE 532 Lecture Notes
Quantum Computing and Communications

Erdal Arıkan
Alexander Shumovsky

Chapter 1

Overview of Quantum Computation

1.1 What is a quantum computer?

A quantum computer is a device designed to take advantage of distinctly quantum phenomena in carrying out a computational task. Classical systems appear to follow a single evolutionary trajectory as determined by the initial conditions and the system dynamic equations. In contrast, a quantum system can be put in a superposition of multiple states and can evolve along multiple trajectories simultaneously, only to choose a definite state at the time of measurement. A quantum computer tries to exploit this parallelism that classical computers do not utilize.

We must hasten to point out that in a very fundamental sense quantum computers are not any more powerful than classical computers: they cannot compute anything that a classical computer cannot. This is simply because the operation of a quantum computer (or any quantum system) can be simulated—in principle if not in practice—by a classical computer. So, quantum computers do not expand the domain of computability. (The notion of computability is a highly technical subject which we shall define precisely in a later lecture.)

The main appeal of quantum computation is the potential that quantum computers can have enormous speed advantages over classical computers in solving certain problems. We shall see several examples of such problems in this course. Factoring large numbers is perhaps the best known such problem. In a very influential 1994 paper [1] Peter Shor showed that quantum computers can factorize large numbers efficiently.

Factoring large numbers is a long-studied problem and the most efficient classical algorithm has time complexity $\exp[c(\ln n)^{1/3}(\ln\ln n)^{2/3}]$ where $c \sim 1.9$ and n is the number whose prime factors are sought. For example, factoring a 400 digit number is beyond the power of present computers even with the best algorithms. The apparent difficulty of factoring is crucial in the very sensitive area of public key cryptography. Shor's quantum algorithm factors an integer n in time of order $(\ln n)^3$. This is an exponential improvement in the running time over the most efficient classical method known.

Quantum computers were first proposed by Feynman in 1982 who pointed out the difficulty of simulating a quantum system using classical computers. While such a simulation is difficult for a classical computer, the system itself in a sense carries out its own simulation effortlessly. E.g., simulating 100 photons each of which can be in one of two polarization states on a classical computer would require processing $2^{100} \sim 10^{30}$ complex numbers—a task well beyond the capability of any present or future computer.

A typical quantum algorithm (program, computation) consists of the following steps (see Fig. 1.1):

- Begin in some initial state $|\Psi_0\rangle$ of the machine that encodes the program input in some form.
- Apply a unitary transformation U to the machine state as directed by the program to reach a final state $|\Psi_f\rangle$.
- Perform a measurement on $|\Psi_f\rangle$ and print the result.

Note that a quantum algorithm is inherently probabilistic due to the randomness of the measurement outcome. You could run the same program twice and measure a different result.

We now consider the elements of quantum computation in more detail.

1.2 Qubit

The basic information bearing element of quantum computation is the *qubit* (quantum bit). A qubit is a two-dimensional quantum system, such as the polarization state of a photon or a hydrogen atom in its two lowest energy levels.

For computational purposes, we choose a standard basis for a qubit and denote the basis vectors by $|0\rangle$ and $|1\rangle$. What is special about this basis is that we assume our computer is equipped with the instrumentation to set the state of a qubit to $|0\rangle$ or $|1\rangle$ at our own choosing and measure it correctly if the qubit is in one of these states. We call this standard basis the *measurement basis*. The general state of a qubit can be represented as

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

where α, β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. If we 'measure' such a qubit, the result equals $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. Thus, a single measurement can reveal at most one bit of information about the prior state of the qubit.

We do not assume that our quantum computer is capable of setting the state of a qubit to any arbitrary state in one elementary step. That may take quite a number of steps because the coefficients α and β are in general infinite precision numbers.

A qubit can be written also in the form

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

and visualized as a point on the 'Bloch' sphere, see Fig. 1.2.

1.2.1 Multiple Qubits

The joint state space of two quantum systems is the tensor product of their individual state spaces. For two qubits, the joint state space is 4 dimensional and has the measurement

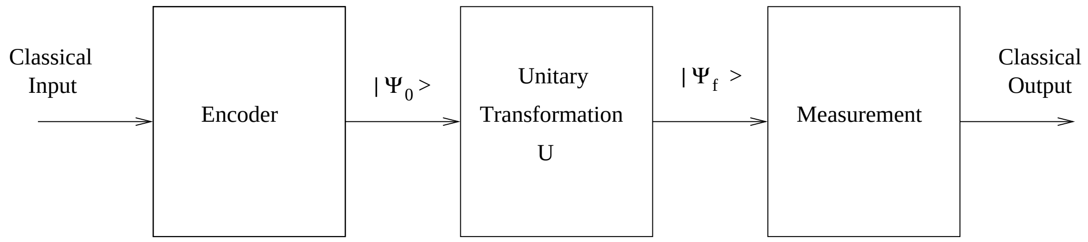

Figure 1.1: Block diagram of a quantum computer.

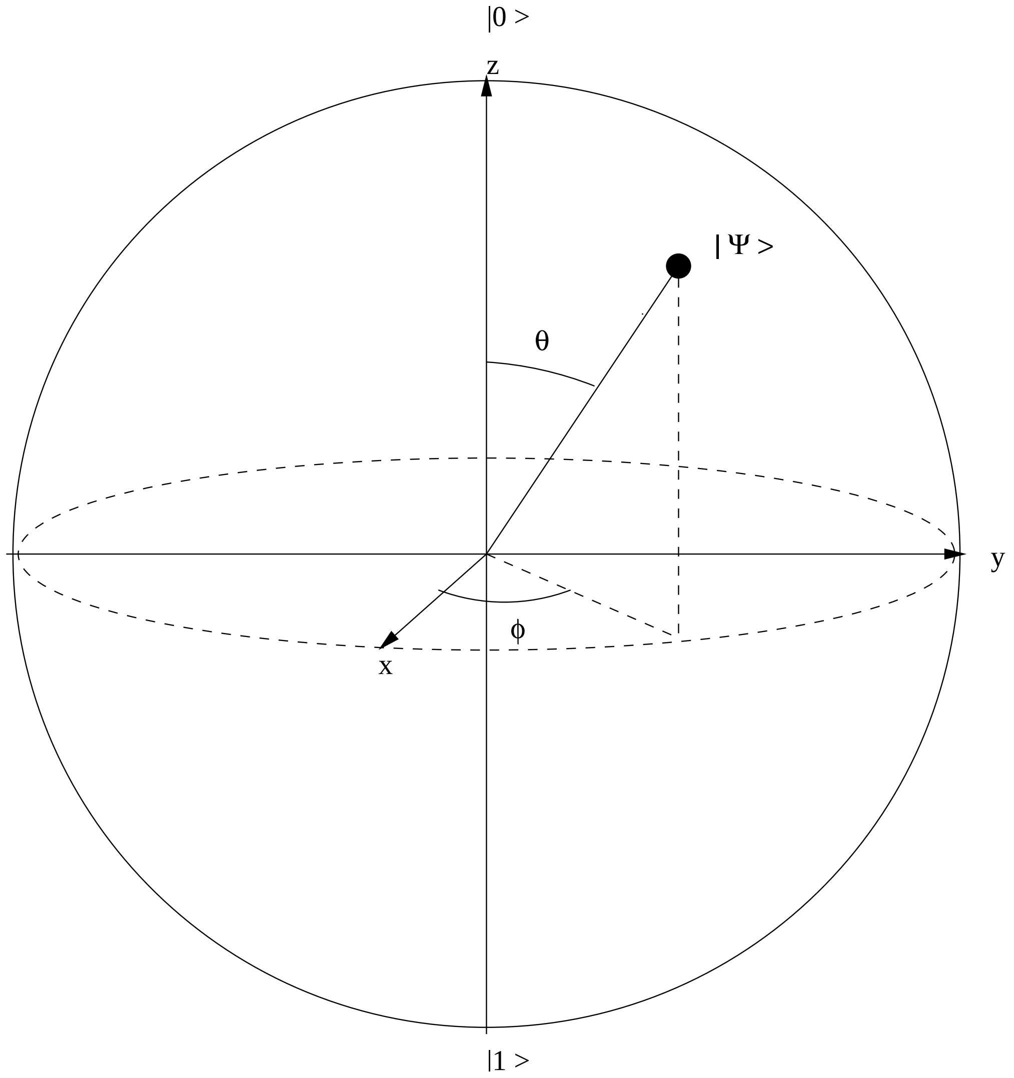

Figure 1.2: Bloch sphere representation of a qubit.

basis

$$\begin{aligned}
|0\rangle &= |00\rangle = |0\rangle \otimes |0\rangle \\
|1\rangle &= |01\rangle = |0\rangle \otimes |1\rangle \\
|2\rangle &= |10\rangle = |1\rangle \otimes |0\rangle \\
|3\rangle &= |11\rangle = |1\rangle \otimes |1\rangle
\end{aligned}$$

More generally, the state space of n qubits is 2^n dimensional and the measurement basis consists of vectors

$$|i\rangle = |i_1 i_2 \cdots i_n\rangle = |i_1\rangle|i_2\rangle \cdots |i_n\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle$$

where $0 \leq i \leq 2^n - 1$ is an integer and $i = 2^{n-1}i_1 + 2^{n-2}i_2 + \cdots i_n$ is the binary expansion of i. Any state of this system can be expanded as

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$$

where a_i are complex numbers such that $\sum_i |a_i|^2 = 1$.

1.3 Measurements

The subject of measurements will be discussed at length later. We consider here a few illustrative examples to introduce the subject.

Suppose we have a 2-qubit system in the state

$$\frac{1}{\sqrt{3}}(|00\rangle + |01\rangle - |11\rangle) \tag{1.1}$$

If we measure this system in the measurement basis for the two qubits, the outcome equals 00, 01, or 11 with probability 1/3 each. If the measurement result is 00, the state of the system after the measurement collapses to $|00\rangle$. (We ignore unimportant phase factors such as $e^{i\delta}$.)

If we measure only the first qubit of the state (1.1), the outcome equals 0 with probability 2/3, and 1 with probability 1/3. The state collapses to $(1/\sqrt{2})(|00\rangle + |01\rangle)$ on outcome 0, and to $|11\rangle$ on outcome 1.

Another possible measurement on the state (1.1) is $x_1 \oplus x_2$, the modulo(2) sum of the values of the two qubits. The possible outcomes are 0 and 1, occurring with probabilities 2/3 and 1/3, respectively. The state after measuring 0 is $(1/\sqrt{2})(|00\rangle - |11\rangle)$ and $|01\rangle$ after measuring 1.

1.4 Quantum Gates

Quantum gates are the functional building blocks of a quantum computer. We begin with the simplest quantum gates.

1.4.1 Single Qubit Gates

An example of a single qubit gate is the quantum NOT gate that inverts the states $|0\rangle$ and $|1\rangle$. Formally, the NOT gate is a unitary transformation X such that

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

The matrix representation of X in the measurement basis is

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

An important property of quantum operations is linearity. The NOT gate operates on a superposition state as follows:

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha X|0\rangle + \beta X|1\rangle = \alpha|1\rangle + \beta|0\rangle$$

In matrix notation,

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}.$$

Another useful single-qubit gate is the Hadamard gate

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

This gate transforms the measurement basis into the cross-polarization basis:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

and

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

It is a good exercise to visualize these operations on the Bloch sphere.

Some other other useful gates are

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The gates X, Y, Z are also called the Pauli (spin) matrices.

1.4.2 Controlled–NOT

CNOT is a two-qubit gate that transforms measurement basis vectors as follows.

$$|00\rangle \to |00\rangle;\ |01\rangle \to |01\rangle;\ |10\rangle \to |11\rangle;\ |11\rangle \to |10\rangle$$

or, symbolically,

$$|A, B\rangle \to |A, A \oplus B\rangle$$

The first qubit, A, is called the control qubit. Matrix representation for the CNOT gate is

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

where we use the lexicographic order for labeling the rows and columns beginning with the all zero vector—a convention we will use unless otherwise specified.

1.4.3 Universality

A very important result that we will return to prove later is the following.

Theorem 1 *The CNOT gate and single qubit gates are* universal *in the sense that any multiple qubit gate can be synthesized in terms of them.*

1.5 The Hadamard Transform

Many quantum algorithms that we shall discuss initialize the input state as an equally weighted superposition of all basis states. In this section we show how this is done.

Let $|x\rangle = |x_1, x_2, \ldots, x_n\rangle$ be an element of the measurement basis for an n-qubit system, such as $|00110 \cdots 010\rangle$. The Hadamard transform of $|x\rangle$ is defined as

$$H^{\otimes n}|x\rangle = H|x_1\rangle \otimes \cdots \otimes H|x_n\rangle$$

where

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

E.g., if we label the basis elements of a 2-qubit system as

$$\begin{aligned} |0\rangle &= |00\rangle = |0\rangle \otimes |0\rangle \\ |1\rangle &= |01\rangle = |0\rangle \otimes |1\rangle \\ |2\rangle &= |10\rangle = |1\rangle \otimes |0\rangle \\ |3\rangle &= |11\rangle = |1\rangle \otimes |1\rangle \end{aligned}$$

the Hadamard transform matrix in this basis is

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

Let $H_n = H^{\otimes n}$ denote the Hadamard transform for an n-qubit system. Then,

$$H_{n+1} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}$$

It is easy to verify that

1. $H_n^2 = I_n$, thus H_n is its self-inverse and unitary.
2. The Hadamard transform of $|0^n\rangle$ is the uniform superposition of all basis states

$$H_n|0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

3. The Hadamard transform of a general basis state $|x\rangle = |x_1, \ldots, x_n\rangle$ is given by

$$H_n|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} |z\rangle$$

   where the sum is over all states $|z\rangle = |z_1, \ldots, z_n\rangle$ and $x \cdot z = \sum_{i=1}^{n} x_i z_i$. We leave the verification of this as an exercise.

1.6 Deutsch's Problem

We are ready to discuss our first quantum algorithm. This relates to a problem paused by Deutsch. Consider a function f that takes n (classical) bits $i_1, \ldots, i_n$ to a single bit, 0 or 1, such as the parity function

$$f(i_1, \ldots, i_n) = i_1 \oplus \cdots \oplus i_n.$$

Assume that we know nothing about the function f except that it is either a constant function or an even function. An even function is one that takes the values 0 and 1 equally often, 2^{n-1} times each. An example of an even function is the parity function. We wish to determine whether the unknown function f is constant or even.

Of course, as stated, this problem cannot be solved. There is in addition an *oracle* that will help us. An oracle can be classical or quantum.

A classical oracle for this problem can be thought of as a black box that on input $i_1, \ldots, i_n$ prints out $f(i_1, \ldots, i_n)$. It is easy to see that the solution of Deutsch's problem requires $2^{n-1} + 1$ calls to such an oracle in the worst case. Thus the best classical algorithm has exponential running time as a function of n in the worst case.

Can a quantum computer speed things up with the aid of a quantum oracle? A quantum oracle for this problem can be thought of as a unitary transformation U_f that operates on $n + 1$ qubits as follows

$$U_f|i_1, \ldots, i_n\rangle|d\rangle = |i_1, \ldots, i_n\rangle|d \oplus f(i_1, \ldots, i_n)\rangle$$

for all $i_1, \ldots, i_n, d \in \{0, 1\}$. The operation U_f is a unitary transformation; it preserves orthogonality as can be verified easily. So, as any unitary transformation, U_f can be implemented using quantum gates. It turns out that we can solve Deutsch's problem with just one call to such a quantum oracle.

The Deutsch-Josza Algorithm

1. Initialize the registers

$$|0\rangle^{\otimes n}|1\rangle$$

2. Apply Hadamard transform to each qubit

$$\rightarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

3. Call the oracle U_f

$$\rightarrow \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)}|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

4. Apply (inverse) Hadamard transform to the first register

$$\rightarrow \frac{1}{\sqrt{2^n}} \sum_z \sum_x \frac{(-1)^{x\cdot z+f(x)}}{\sqrt{2^n}}|z\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

5. Measure the first register to determine the final outcome

$$\rightarrow z$$

Decide that f is constant if $z = 0^n$; otherwise, decide that f is even.

To see why the algorithm works, note the following.

1. The oracle action at step 3 produces

$$\begin{aligned} U_f|x\rangle(|0\rangle - |1\rangle) &= |x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= \begin{cases} |x\rangle(|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ |x\rangle(|1\rangle - |0\rangle) & \text{if } f(x) = 1 \end{cases} \\ &= (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle) \end{aligned}$$

2. The probability of observing state $|z\rangle$ in the final step is

$$|\frac{1}{2^n} \sum_x (-1)^{(x\cdot z+f(x))}|^2$$

For $z = 0^n$, this probability is

$$|\frac{1}{2^n} \sum_x (-1)^{f(x)}|^2 = \begin{cases} 1 & \text{if } f \text{ is constant} \\ 0 & \text{if } f \text{ is even.} \end{cases}$$

Thus, the algorithm determines whether f is constant or even with probability 1.

This is an example where quantum computation has a provable exponential speed-up over the best classical algorithm. Notice however that if we try k values of f using the classical oracle, the probability of not being able to determine whether f is constant or even is 2^{k-1}, which becomes very small quickly. So, this is not truly a difficult problem to solve classically, if we allow a certain amount of error. Deutsch's problem and the Deutsch-Jozsa algorithm are still significant because it was the first problem that clearly demonstrated the potential of quantum computation. The real interest in quantum computation began with Shor's invention of the factoring algorithm.

Bibliography

[1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. 35th Annual Symp. on Foundations of Computer Science*, Santa Fe., NM, November 20-22, 1994, IEEE Computer Society Press, pp. 124-134.