Quantum Algorithms

Andreas Klappenecker *Texas A&M University*

Lecture notes of a course given in Spring 2003. Preliminary draft.

Preface

Quantum computing provides a fresh perspective on information processing. Some quantum algorithms have the promise to possibly provide an exponential speed-up over classical deterministic and randomized algorithms, which explains the massive worldwide efforts to build a viable quantum computer. However, this is by no means the only motivation. Quantum computing has serious repercussions on classical computing as well.

These lecture notes provide a rapid introduction to the main ideas behind quantum algorithms. The subject matter is not *difficult*, but dramatically *different* from its classical counterpart. We provide numerous very simple exercises that are designed to ease the transition into the quantum realm. Solving the exercises will help to gain an active working knowledge.

Our approach is largely based on the quantum circuit model, which is easy to understand. This model abstracts from the nature and the dynamics of the physical system realizing the quantum computer. The advantage of this approach is that within an extremely short period of time it will be possible to cover interesting algorithms.

The course requires some background in linear algebra. The books *Linear Algebra* by Serge Lang and *Linear Algebra Done Right* by Sheldon Axler are excellent sources to review such material.

Please note that this is a preliminary draft. The lecture notes are incomplete, and all parts are subject to change. The material should be read in conjuction with the books *Quantum Computation and Quantum Information* by Michael Nielsen and Isaac Chuang and *Classical and Quantum Computation* by Alexei Kitaev, A.H. Shen, and M.N. Vyalyi.

If you read these notes, then you have accepted a contract: you agree to communicate all errors to me. If you do not want to burden yourself with this task, then do not read further.

Andreas Klappenecker
College Station, Texas

Chapter 2

Quantum Circuits

Quantum computing can be based on various different computational models. The most accessible one is the quantum circuit model, which specifies a sequence of operations that manipulate the state of the quantum computer at discrete time steps. The basic rules of this model are surprisingly simple. This chapter introduces the basic properties of quantum states, quantum gates, and measurements.

§1 Quantum States

A bit has two distinguishable states, denoted by 0 and 1. A classical computer manipulates a set of bits, which form the memory of the computer. The memory of a quantum computer is based in a similar way on the notion of a **quantum bit**, **qubit** for short. A qubit has two clearly distinguishable states, denoted by $|0\rangle$ and $|1\rangle$. The possible states of a qubit are not exhausted by these two possibilities. In general, the state of a qubit is of the form $a|0\rangle+b|1\rangle$, where a and b are complex numbers satisfying $|a|^2 + |b|^2 = 1$.

The states $|0\rangle$ and $|1\rangle$ should be understood as basis vectors of a complex two-dimensional vector space. We can associate with these states the basis vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.1}$$

The state $a|0\rangle + b|1\rangle$ is a linear combination of these two basis vectors, and is represented by the vector $(a, b)^t$. The operations of the quantum computer manipulate these vectors by linear transformations or by measurements.

The value of a quantum bit is always 0 or 1, never anything else. If a qubit is in the state $a|0\rangle + b|1\rangle$, then this means that the value 0 is observed with probability $|a|^2$, and the value 1 with probability $|b|^2$. A measurement in the computational basis returns the value 0 or 1 according to this rule, and sets the qubit to the state $|0\rangle$ or $|1\rangle$, respectively. A consequence is that if the measurement is repeated, then it will return the same value.

It is easy to construct a state that yields a fair coin-flip. Choose the state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Then 0 and 1 are both observed with probability $(1/\sqrt{2})^2 = 1/2$. The resulting state after the measurement is $|0\rangle$, if the measurement result was 0, and $|1\rangle$ otherwise.

Exercise 2.1 Assume that a qubit is in the state $\frac{1}{\sqrt{10}}|0\rangle + \frac{3}{\sqrt{10}}|1\rangle$. What is the probability to observe 0, or 1, respectively?

Exercise 2.2 Assume that a qubit is in the state $\frac{i}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$. What is the probability to observe 0, or 1, respectively?

A memory consisting of n quantum bits has 2^n basis states, which are denoted by $|0\cdots 00\rangle, |0\cdots 01\rangle, |0\cdots 10\rangle, \ldots, |1\cdots 11\rangle$. The state of the memory is a linear combination of these basis states. Denote by $\mathbf{F}_2$ the finite field with two elements 0 and 1. An arbitrary state of the memory is of the form

$$\sum_{k\in\mathbf{F}_2^n} a_k|k\rangle, \quad \text{with} \quad \sum |a_k|^2 = 1.$$

If we read out the memory by a measurement in the computational basis, then we will observe the result k, a string of n bits, with probability $|a_k|^2$. The scalar coefficients a_k are called **probability amplitudes** or, simply, **amplitudes**.

Exercise 2.3 What is the probability of observing 11, if the memory is in the state $\frac{1}{2}|00\rangle - \frac{1}{2}|10\rangle + \frac{i}{\sqrt{2}}|11\rangle$? In what state is the memory once we have observed 11?

Exercise 2.4 Describe all possible states of a system of two quantum bits such that a measurement in the computational basis yields 00 with probability 1/2, and the results 01 and 11 both with probability 1/4.

Any quantum system with at least two different basis states can basically store a quantum bit, and finding appropriate storage media for a quantum computer is a very active area of current research. The linear combination of basis states reflects the superposition principle of quantum mechanics. It should be noted that only the measurement process introduces randomized behavior in quantum algorithms. All other operations of a quantum computer are completely deterministic.

§2 A Single Quantum Bit

The operations of a quantum computer allow reading, writing, or manipulating the content of the memory, and therefore serve the same purpose as the operations of a classical computer. The main distinction is that the operations of a quantum computer are formulated to be conformant with the laws of quantum mechanics. We explain in this section the basic operations on a single quantum bit, and introduce some convenient notations.

The input operation of a quantum computer can prepare the memory in any basis state. As a result, each quantum bit is either in the state $|0\rangle$ or in the state $|1\rangle$, but not in a superposition of these basis states. The actual computation is done by applying simple operations, called **quantum gates**, which allow to manipulate the content of the memory. The result of the computation is determined by **measurement operations**.

If the memory consists of a single quantum bit, then the operations are particularly easy to understand. We recall some mathematical vocabulary to ease our discussion. If $x = (x_{m-1}, \dots, x_0)^t$ and $y = (y_{m-1}, \dots, y_0)^t$ are vectors in $\mathbf{C}^m$, then

$$\langle x|y\rangle = \overline{x}_{m-1} y_{m-1} + \cdots + \overline{x}_0 y_0$$

defines a **hermitian product**. We follow the convention that hermitian products are anti-linear in the first argument, and linear in the second.

Exercise 2.5 Show that the hermitian product is positive definite, that is, $\langle x|x\rangle \geq 0$ for all $x \in \mathbf{C}^m$, and $\langle x|x\rangle > 0$ if $x \neq 0$.

If $x \in \mathbf{C}^m$, then the **norm** of x is defined by $\|x\| = \sqrt{\langle x|x\rangle}$. A vector x with norm $\|x\| = 1$ is called a **unit vector**. Let $U\colon \mathbf{C}^m \to \mathbf{C}^m$ be a linear map. If $\langle\, Ux|Uy\,\rangle = \langle x|y\rangle$ holds for all $x, y \in \mathbf{C}^m$, then U is called **unitary**.

Exercise 2.6 Show that a complex $m \times m$ matrix U is unitary if and only if $U^{-1} = \overline{U}^t$; that is, the inverse of a unitary matrix is obtained by transposing the matrix and conjugating the matrix entries.

Exercise 2.7 A quantum state is a unit vector. Show that if a linear map M maps each unit vector $x \in \mathbf{C}^m$ to a unit vector Mx, then M has to be unitary. This property explains the relevance of unitary maps in quantum computing.

Exercise 2.8 Let $\{u_0, \dots, u_{m-1}\}$ and $\{v_0, \dots, v_{m-1}\}$ be orthonormal bases of $\mathbf{C}^m$. Let U be a linear map such that $v_i = Uu_i$ for $i = 0, \dots, m-1$. Show that U is unitary.

We have now the terminology to describe the operations on a single quantum bit. A **quantum gate** changes the state of a single qubit by applying an arbitary unitary map U. We use the following graphical notation for such a quantum gate:

$$-\boxed{U}-$$

The horizontal line represents the evolution of the quantum bit over time. The time flow is from left to right. The box represents a quantum gate, which applies a unitary map U to the state of the qubit.

The quantum gate is unitary, hence, in particular, linear. This means that the action of the gate is completely determined by its behavior on the base states $|0\rangle$ and $|1\rangle$. Suppose that the quantum gate U changes the input state $|0\rangle$ to $m_{00}|0\rangle + m_{10}|1\rangle$ and the input $|1\rangle$ to $m_{01}|0\rangle + m_{11}|1\rangle$. If the input is a linear combination $a|0\rangle + b|1\rangle$, then the gate U will change this state to

$$\begin{aligned} & a\,(m_{00}|0\rangle + m_{10}|1\rangle) + b\,(m_{01}|0\rangle + m_{11}|1\rangle) \\ & = (am_{00} + bm_{01})|0\rangle + (am_{10} + bm_{11})|1\rangle. \end{aligned}$$

The result of this computation can be expressed in the standard basis (2.1) by the following matrix vector product:

$$\begin{pmatrix} a\,m_{00} + b\,m_{01} \\ a\,m_{10} + b\,m_{11} \end{pmatrix} = \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

The most familiar example is given by a **not gate**, which changes $|0\rangle$ to $|1\rangle$ and vice versa. This quantum gate can be described by the unitary matrix

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

If we apply this quantum gate twice, then we recover the input. Graphically, we obtain the rule

$$-\boxed{X}-\boxed{X}- \quad = \quad \text{———}$$

Another operation on one quantum bit is given by the **Hadamard gate**

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

This operation has the following effect:

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad \text{and} \quad H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

Exercise 2.9 Calculate $H(H|0\rangle)$ and $H(H|1\rangle)$ by evaluating the expressions in parentheses. Use linearity to obtain the result. Compare your result to the matrix H^2.

The product of two unitary matrices is a unitary matrix. Therefore, instead of applying gate A and then gate B, we can apply a single quantum gate BA. This way we obtain the rule

$$-\boxed{A}-\boxed{B}- \quad = \quad -\boxed{BA}-$$

The order of the matrices changes because the time flow in a quantum circuit is from left to right. However, the matrices act on column vectors; hence, applying BA means that A is applied first.

Exercise 2.10 Simplify the circuit, and determine a single unitary matrix Z that is the result of applying the Hadamard gate H, then the not gate X, then again the Hadamard gate H:

$$-\boxed{H}-\boxed{X}-\boxed{H}- \quad = \quad -\boxed{Z}-$$

Exercise 2.11 Find a unitary 2×2 matrix R such that

$$-\boxed{R}-\boxed{R}- \quad = \quad -\boxed{X}-$$

In other words, R should satisfy $R^2 = X$.

Numerous other unitary matrices are used in quantum algorithms. The rotation matrices

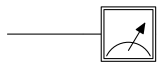
$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

and the Pauli matrices σ_x, σ_y, and σ_z are popular choices:

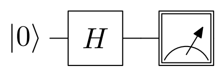
$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Exercise 2.12 Show that the product of any two Pauli matrices is – up to a multiplication by a scalar – either a Pauli matrix or the identity matrix. Memorize the definition of the Pauli matrices.

An output is obtained by measuring the state of the quantum bit. The **measurement operation** of a quantum bit in the state $a|0\rangle + b|1\rangle$ yields output 0 with probability $|a|^2$, and output 1 with probability $|b|^2$. The state is, in general, changed by the measurement operation. If 0 is observed, then the state is set to $|0\rangle$, and if 1 is observed, then the state is set to $|1\rangle$. We depict a measurement of the quantum bit by a meter sign:



The operations obtained so far allow us to derive a quantum circuit simulating an unbiased coin flip. This circuit produces output 0 with probability 1/2, and output 1 with probability 1/2. We initialize the quantum bit with the state $|0\rangle$, then apply the Hadamard gate, and measure the result:



The Hadamard gate changes the state to $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$; hence, the measurement produces the output with the desired probability.

Exercise 2.13 Design a quantum circuit that simulates a biased coin flip. The circuit should produce output 0 with probability 1/3, and output 1 with probability 2/3.

§3 Quantum Gates

We need operations that enable the interaction between different quantum bits. The **xor gate** or **controlled-not gate** acts on two distinct quantum bits. Suppose that the memory contains two quantum bits, then the controlled-not gate operates on the basis states of the system as follows:

$$\begin{array}{lcl} |00\rangle & \mapsto & |00\rangle, \\ |01\rangle & \mapsto & |01\rangle, \\ |10\rangle & \mapsto & |11\rangle, \\ |11\rangle & \mapsto & |10\rangle. \end{array}$$

If we extend this operation linearly, then the quantum state

$$a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$$

will be mapped by this controlled-not gate to

$$a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|11\rangle + a_{11}|10\rangle.$$

Exercise 2.14 The xor gate is a unitary map. Determine the associated unitary matrix with respect to the computational basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Choose the basis vectors in this order.

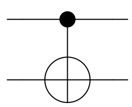
Exercise 2.15 Suppose that a controlled-not gate is applied to the state $\frac{1}{2}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{\sqrt{2}}|11\rangle$. What is the resulting state?

Controlled-not gates can be generalized to an arbitrary number $n \geq 2$ of quantum bits. A **controlled-not gate** with control bit at position i and target bit at position $j \neq i$ is a unitary map, which is determined by

$$|x_{n-1} \cdots x_1 x_0\rangle \mapsto |y_{n-1} \cdots y_1 y_0\rangle,$$

where x_k and y_k are elements of $\{0, 1\}$, such that $x_k = y_k$ for all $k \neq j$, and the target bit $y_j = x_i \oplus x_j$ is the result of adding x_i to x_j modulo 2. We denote this controlled-not gate by $\Lambda_{i,j}(X)$.

A controlled-not gate $\Lambda_{1,0}(X)$ acting on two quantum bits is depicted in the graphical notation for quantum gates by

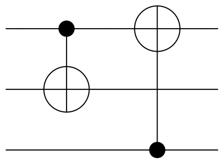


The two horizontal lines represent the two quantum bits. The most significant bit (the bit at position 1) is shown on top, and the least significant bit (the bit at position 0) is shown at the bottom. The black dot $\bullet$ depicts the control bit of the quantum gate, and the crossed circle $\oplus$ depicts the target bit.

To illustrate, assume that we have three quantum bits, which are initially in the state

$$\frac{1}{2}|001\rangle + \frac{1}{\sqrt{2}}|110\rangle + \frac{1}{2}|111\rangle.$$

Suppose that this state is processed by the quantum circuit



The time flow is from left to right. The first controlled-not gate $\Lambda_{2,1}(X)$ negates the quantum bit in the middle, if the most significant bit is set. The resulting intermediate state after applying the first controlled-not gate is

$$\frac{1}{2}|001\rangle + \frac{1}{\sqrt{2}}|100\rangle + \frac{1}{2}|101\rangle.$$

The second controlled-not gate $\Lambda_{0,2}(X)$ is controlled by the least significant bit, and the target bit is the most significant bit. The intermediate state is changed by this controlled-not gate to

$$\frac{1}{2}|101\rangle + \frac{1}{\sqrt{2}}|100\rangle + \frac{1}{2}|001\rangle.$$

Exercise 2.16 Design a quantum circuit consisting of controlled-not gates, which realizes the unitary map

$$|00\rangle \mapsto |00\rangle, \quad |01\rangle \mapsto |10\rangle, \quad |10\rangle \mapsto |01\rangle, \quad |11\rangle \mapsto |11\rangle.$$

We relegate the discussion of further multi-qubit operations to the next chapter. We focus instead on operations, which act locally on a single quantum bit. It turns out that single-qubit operations and controlled-not gates allow to fully program a quantum computer. Therefore, all other operations can be expressed in terms of these elementary operations. We make a digression and explain tensor products, which provide the proper framework to understand the data structure of the memory.

Let V and W be finite-dimensional complex vector spaces. The tensor product $V \otimes W$ is a vector space, which is spanned by linear combinations of elements $v \otimes w$ such that $v \in V$ and $w \in W$. The product $v \otimes w$ is defined such that it satisfies the additive relations

$$\begin{aligned}(v_1 + v_2) \otimes w &= v_1 \otimes w + v_2 \otimes w \quad (2.2)\\ v \otimes (w_1 + w_2) &= v \otimes w_1 + v \otimes w_2 \quad (2.3)\end{aligned}$$

and the balancing relations

$$c(v \otimes w) = (cv) \otimes w = v \otimes (cw) \quad (2.4)$$

for each v, v_1, v_2 in V, each w, w_1, w_2 in W, and each complex number c.

We can formally construct this vector space $V \otimes W$ as follows. Form the vector space A of all linear combinations of elements (v, w) with $v \in V$ and $w \in W$. Consider the subspace B of A, which consists of all linear combinations of the elements

$$\begin{aligned}(v_1 + v_2, w) - (v_1, w) - (v_2, w),\\ (v, w_1 + w_2) - (v, w_1) - (v, w_2),\\ c(v, w) - (cv, w), \quad c(v, w) - (v, cw),\end{aligned}$$

for $v, v_1, v_2 \in V$, $w, w_1, w_2 \in W$, and $c \in \mathbf{C}$. We define the tensor product $V \otimes W$ to be the quotient space A/B. The image of the element (v, w) of A in $V \otimes W$ is denoted by $v \otimes w$.

We emphasize that not every element of $V \otimes W$ is of the form $v \otimes w$ for some $v \in V$ and $w \in W$. However, every element of $V \otimes W$ can be expressed as a sum $\sum_{i,j} v_i \otimes w_j$ of such tensor products, with $v_i \in V$ and $w_j \in W$.

Exercise 2.17 Give an example of a vector in $\mathbf{C}^2 \otimes \mathbf{C}^2$ that cannot be written in the form $v \otimes w$ with $v, w \in \mathbf{C}^2$. Prove your result.

It might be helpful to re-iterate the construction. We started with two finite-dimensional vector spaces V and W. We constructed a giant vector space A with basis $\{(v, w) \mid v \in V, w \in W\}$. The generators of B were chosen such that the quotient space $V \otimes W = A/B$ satisfies the relations (2.2)–(2.4). It is easy to see that $V \otimes W = A/B$ is a finite-dimensional vector space, even though A and B are infinite-dimensional.

Exercise 2.18 Let V and W be complex finite-dimensional vector spaces. Let $\{e_1, \ldots, e_m\}$ be a basis of V and $\{f_1, \ldots, f_n\}$ be a basis of W. Show that $\{e_i \otimes f_j \mid 0 \leq i < m, 0 \leq j < n\}$ generates $V \otimes W$.

The exercise shows that $\dim(V \otimes W) \leq \dim(V) \dim(W)$. In fact, it is possible to show that equality holds, which proves that the generating set in the previous exercise is a basis of $V \otimes W$.

Let V and W be as in Exercise 2.18. Suppose that A is a linear map on V, and B is a linear map on W. Let $A \otimes B$ denote the linear map on $V \otimes W$, which is determined by

$$(A \otimes B)(e_i \otimes f_j) = Ae_i \otimes Bf_j.$$

This uniquely determines the values of $A \otimes B$ on other elements of $V \otimes W$ because the elements $e_i \otimes f_j$ are a basis.

Exercise 2.19 Let A and B be the matrices

$$A = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}, \qquad B = \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix}$$

representing linear maps with respect to the basis $\{e_0, e_1\}$. Determine the matrix $A \otimes B$ with respect to the basis $\{e_0 \otimes e_0, e_0 \otimes e_1, e_1 \otimes e_0, e_1 \otimes e_1\}$.

The tensor product plays a significant role in quantum computing. Recall that the state space of a single quantum bit is given by $\mathbf{C}^2$. In quantum mechanics, the state space of a joint quantum system is described by the tensor product of the state spaces of its parts. Consequently, a compound system of n quantum bits has the state space

$$\mathbf{C}^2 \otimes \cdots \otimes \mathbf{C}^2 \qquad (n \text{ factors}).$$

This is a 2^n-dimensional complex vector space, hence isomorphic to $\mathbf{C}^{2^n}$. The isomorphism is explicitly given by the linear map

$$|x_{n-1}\rangle \otimes \cdots \otimes |x_1\rangle \otimes |x_0\rangle \longmapsto |x_{n-1} \cdots x_1 x_0\rangle,$$

where $x_i \in \{0,1\}$, $0 \leq i < n$. We will use this isomorphism freely, and switch from one representation to the other, whichever is more convenient. We will silently identify the two notations and write $|00\rangle = |0\rangle \otimes |0\rangle$, etc.

Exercise 2.20 By convention, the basis vectors associated with the basis $|0\rangle$ and $|1\rangle$ of $\mathbf{C}^2$ are

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Derive the vectors of $\mathbf{C}^4 \cong \mathbf{C}^2 \otimes \mathbf{C}^2$ associated with

$$|00\rangle = |0\rangle \otimes |0\rangle, \quad |01\rangle = |0\rangle \otimes |1\rangle, \quad |10\rangle = |1\rangle \otimes |0\rangle, \quad |11\rangle = |1\rangle \otimes |1\rangle.$$

Exercise 2.21 Which vector is associated with $(a_0|0\rangle + a_1|1\rangle) \otimes (b_0|0\rangle + b_1|1\rangle)$, assuming the above convention for the basis vectors?

Suppose we have a memory with n quantum bits. Let U be a unitary 2×2 matrix. We define a **single-qubit gate** U acting on the quantum bit at position i to be the unitary map $\mathbf{1}_{2^{n-i-1}} \otimes U \otimes \mathbf{1}_{2^i}$. Alternatively, one can describe the action of the gate by

$$|x_{n-1}\rangle \otimes \cdots \otimes |x_i\rangle \otimes \cdots \otimes |x_0\rangle \mapsto |x_{n-1}\rangle \otimes \cdots \otimes U|x_i\rangle \otimes \cdots \otimes |x_0\rangle,$$

where $x_i \in \{0,1\}$, $0 \leq i < n$. All tensor components remain unchanged with the exception of $|x_i\rangle$, which is replaced by $U|x_i\rangle$.

Let us illustate this definition in the case of two quantum bits. Suppose that we apply the Hadamard gate H on the least significant bit, that is, the gate acts on the quantum bit at position $i = 0$. The unitary map associated with this gate is represented by the matrix

$$\mathbf{1}_2 \otimes H \otimes \mathbf{1}_1 = \mathbf{1}_2 \otimes H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}. \tag{2.5}$$

This matrix is the tensor product of the identity matrix $\mathbf{1}_2$ and the Hadamard matrix H.

The alternative description is even easier to grasp. Indeed, the state $|00\rangle = |0\rangle \otimes |0\rangle$ is mapped to

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

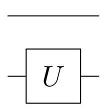
Note that this vector corresponds to the first column of the matrix (2.5). The state $|01\rangle = |0\rangle \otimes |1\rangle$ is mapped to

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

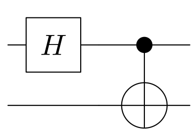
and corresponds to the second column of the matrix (2.5). The result of the input $|10\rangle$ and $|11\rangle$ is obtained in a similar way, and we leave these two cases to the reader.

Exercise 2.22 Suppose that the memory consists of two qubits. Determine the matrix corresponding to the Hadamard gate acting on the most significant qubit.

The graphical notation for single-qubit gates is similar to the single quantum bit case. A single-qubit gate U acting on the least significant bit in a system of two quantum bits is depicted by



Exercise 2.23 Determine the action of the circuit



on the input $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Explain why the resulting states form an orthonormal basis.

§4 Measurements

We need to have a way to obtain the value of a quantum bit. The quantum circuit model allows measuring an individual quantum bit with respect to the computational basis. We define these measurement operations in this section and discuss some possible extensions.

Assume that we have a memory consisting of n quantum bits. Suppose that the memory is in the quantum state

$$v = \sum_{x \in \mathbf{F}_2^n} a_x |x\rangle, \qquad a_x \in \mathbf{C}.$$

The state vector v is, as always, assumed to be of unit norm, $\|v\| = 1$. A **measurement** of the quantum bit at position i yields the result $k \in \{0,1\}$ with probability

$$\sum_{x \in \mathbf{F}_2^n \text{ with } x_i = k} |a_x|^2.$$

The measurement changes, in general, the state vector. If k is observed, then the resulting state of the memory is given by $\frac{1}{\|v_k\|} v_k$, where

$$v_k = \sum_{x \in \mathbf{F}_2^n \text{ with } x_i = k} a_x |x\rangle.$$

Let us illustrate the effect of this operation in the case of two quantum bits. Suppose that the memory is in the state

$$v = \frac{1}{2}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle + \frac{1}{2}|11\rangle.$$

If we measure the qubit at position $i = 0$, then we will observe 0 with probability $(1/2)^2 + (1/\sqrt{2})^2 = 3/4$, and 1 with probability $(1/2)^2 = 1/4$. Note that $v_0 = \frac{1}{2}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$ and $v_1 = \frac{1}{2}|11\rangle$. Therefore, if we observe 0, then the memory will be in the state

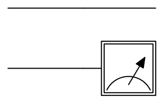
$$\frac{1}{\|v_0\|} v_0 = \frac{2}{\sqrt{3}} v_0 = \frac{1}{\sqrt{3}}|00\rangle + \frac{2}{\sqrt{6}}|10\rangle,$$

and if we observe 1, then the memory will be in the state

$$\frac{1}{\|v_1\|} v_1 = 2 v_1 = |11\rangle.$$

Exercise 2.24 Let $v = \frac{1}{3}|00\rangle + \frac{\sqrt{3}}{3}|01\rangle + \frac{\sqrt{5}}{3}|10\rangle$. If we measure the least significant bit, what is the probability to observe 0, respectively 1? Determine the resulting states v_0 and v_1 of the memory.

The graphical notation for a measurement is the meter sign. For instance, the measurement of the least significant quantum bit is depicted by

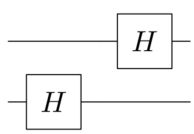


The reader familiar with quantum mechanics will notice that many more types of measurements are, in principle, possible. However, the practical ways to measure quantum bits are typically rather limited. Although quantum physics allows us to measure with respect to any orthonormal basis, we limit ourselves here to the computational basis. If we could perform a measurement with respect to a totally arbitrary orthonormal basis, then there would be no need for quantum gates. The quantum algorithm would then simply consist of a measurement in the appropriate basis.

§5 Examples

We give in this section some tiny examples, which illustrate the notions that we have introduced so far. We will mainly discuss some small quantum circuits, which do not necessarily have any purpose other than illustrating the effect of quantum operations. The superficial examples given here allow us, nonetheless, to illustrate some common tricks of the trade. We will discuss some more meaningful examples in the next chapter.

Example 1. The first example illustrates how the Hadamard gates can be used to generate quickly a superposition of all possible input states. Suppose that the Hadamard gate is applied to both quantum bits, first on the least significant bit, then on the most significant bit:



Therefore, the action on the state vector is given by $(H\otimes\mathbf{1}_2)(\mathbf{1}_2\otimes H)$. Suppose that the input is $|00\rangle = |0\rangle\otimes|0\rangle$. The intermediate state after applying the first gate is

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

The final state after applying the second gate is

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

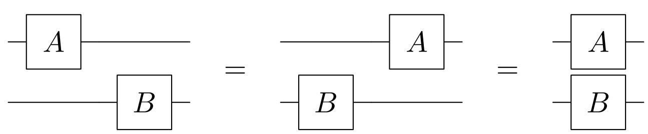
We can expand the right hand side using the bilinear relations of the tensor product, and obtain the simpler form

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

We could have obtained the same result by applying the gate on the most significant qubit first, and then the gate on the least significant bit; or even by applying both gates at the same time.

Exercise 2.25 Suppose that A_1 and B_1 are $n \times n$ matrices, and A_2 and B_2 are $m \times m$ matrices. Show that $(A_1 \otimes A_2)(B_1 \otimes B_2) = (A_1 B_1) \otimes (A_2 B_2)$.

A consequence of this exercise is that if we have two quantum gates, which affect disjoint sets of quantum bits, then we can execute these gates in arbitary order. Indeed, we have $(A \otimes \mathbf{1}_m)(\mathbf{1}_n \otimes B) = (\mathbf{1}_n \otimes B)(A \otimes \mathbf{1}_m)$. We can even execute these operations in parallel, because $(A \otimes \mathbf{1}_m)(\mathbf{1}_n \otimes B) = A \otimes B$. Therefore, gates acting on different quantum bits are often denoted on top of each other, as shown on the right, to make the graphical notation more compact:



These rules are also useful when one attempts to simplify quantum circuits.

Example 2. Engineering a specific quantum state is a frequent subtask in the design of quantum algorithms. For instance, suppose that we need to prepare four quantum bits in the state

$$\frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1111\rangle.$$

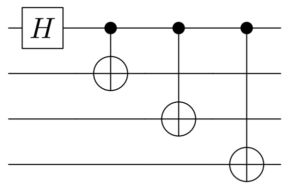
Assume that the quantum bits are initially in the state $|0000\rangle$. We can apply the Hadamard gate on the most significant qubit to obtain the state

$$\frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1000\rangle.$$

Applying controlled-not gates on the three least significant qubits as target qubits, with the most significant bit as a control bit, yields the desired state

$$\frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1111\rangle.$$

Indeed, if we apply the three controlled-not gates to the state $|0000\rangle$, then this state remains unchanged, and if we apply the three controlled-not gates to $|1000\rangle$, then we get $|1111\rangle$; the result follows by linearity of the quantum gates. In graphical notation, the quantum circuit is given by



Exercise 2.26 Design a quantum circuit that prepares the superposition of all basis states with even parity for a system of three quantum bits, namely the state

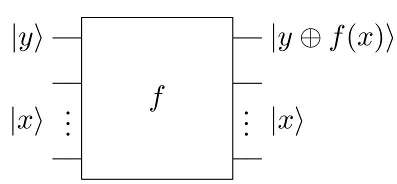
$$\frac{1}{2}|000\rangle + \frac{1}{2}|011\rangle + \frac{1}{2}|101\rangle + \frac{1}{2}|110\rangle.$$

Assume that the memory is initially in the state $|000\rangle$.

Example 3. Suppose that we have a boolean function $f\colon \mathbf{F}_2^n \to \mathbf{F}_2$. A quantum circuit implementing f has to be realized by a unitary map. This can be accomplished, for instance, by implementing the map

$$|y\rangle \otimes |x\rangle \mapsto |y \oplus f(x)\rangle \otimes |x\rangle$$

on $n+1$ qubits, where $x \in \mathbf{F}_2^n$, and $y \in \mathbf{F}_2$. The most significant bit is the output bit, and the n lowest significant bits are the input bits. The result of $f(x)$ is added modulo 2 to the output bit. The result is a quantum circuit of the form



The linearity of the circuit allows to evaluate f for any linear combination of the basis states. Assume that all $n+1$ quantum bits are initialized with state $|0\rangle$. We apply the Hadamard gate to all n input bits. The resulting state is

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \mathbf{F}_2^n} |0\rangle \otimes |x\rangle,$$

a superposition of all possible inputs. If we apply the circuit implementing the function f, then we obtain as a result

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \mathbf{F}_2^n} |f(x)\rangle \otimes |x\rangle.$$

Thus, the circuit evaluates the function f for all possible inputs at once.

Exercise 2.27 Design a quantum circuit that implements the parity function $f(x_2, x_1, x_0) = x_2 \oplus x_1 \oplus x_0$. Show how this circuit can be used to generate the state

§6 Summary

The state space of a memory with n quantum bits is given by the complex vector space $\mathbf{C}^{2^n} \cong \mathbf{C}^2 \otimes \cdots \otimes \mathbf{C}^2$. We choose, once and for all, a fixed orthonormal basis of this vector space, and call it the computational basis. Its basis vectors are denoted by $|0\cdots 00\rangle$, $|0\cdots 01\rangle, \cdots, |1\cdots 11\rangle$. An arbitrary state of the memory is of the form

$$\sum_{x \in \mathbf{F}_2^n} a_x |x\rangle, \quad \text{where} \quad \sum_{x \in F_2^n} |a_x|^2 = 1. \tag{2.6}$$

A measurement of the quantum bit at position i yields the result $k \in \{0, 1\}$ with probability $\sum_{x_i = k} |a_x|^2$. If k is observed, then the resulting state after the measurement is $\frac{1}{\|v_k\|} v_k$, where v_k denotes the vector

$$v_k = \sum_{x \in \mathbf{F}_2^n, x_i = k} a_x |x\rangle.$$

A single-qubit gate is determined by a matrix $U \in \mathcal{U}(2)$ and a bit position i. Such a gate modifies the state of the memory by applying the unitary matrix $\mathbf{1}_{2^{n-i-1}} \otimes U \otimes \mathbf{1}_{2^i}$. A controlled-not gate $\Lambda_{i,k}(X)$ is specified by its action on the basis vectors

$$\Lambda_{i,k}(X)|x_{n-1} \cdots x_1 x_0\rangle = |y_{n-1} \cdots y_1 y_0\rangle,$$

where $y_j = x_j$ for all $j \neq k$, and $y_k = x_i \oplus x_k$.