Chapter 14

Quantum computing

14.1 Quantum bits and quantum gates

In 1982 Feynman suggested that the problem of exponential complexity of simulating a quantum system can be solved by using quantum mechanics itself for computing, thus laying the foundation for the field of quantum computing [1]. Just as there are many ways to build a classical computer and lots of different conventions, also quantum computers could be designed in many different ways. Since we don't have any large scale quantum computer yet we are free to choose a design that is simple from a theoretical point of view.

14.1.1 Quantum bits

The basic memory element is typically chosen as the quantum bit, or qubit for short – a two-level system like a spin-1/2, where we associate the up spin state with the 0 bit and the down spin state with the 1 bit:

$$|0\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{14.1}$$

$$|1\rangle = |\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{14.2}$$

Just like for quantum spin-1/2s the quantum bit can exist in an arbitrary superposition of these two states:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{14.3}$$

where the normalization condition requires that $|\alpha|^2 + |\beta|^2 = 1$. While such a state needs an infinite number of classical bits to be described accurately (think of the binary representation of α and β), a measurement will only give a single bit of information, either 0 or 1. A register of N qubits can store the wave function of N spin-1/2s. This gives an exponential advantage in memory use. However, since we can only do one measurement on each qubit only N bits of information can ever be read out. This is the first restriction we have to face when devising quantum algorithms, for which a clever use of these quantum bits needs to be devised.

Gate Name	Symbol	Matrix Form
Pauli-X (NOT)	X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y	Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z	Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard gate	H	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Phase gate	S	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
T gate or $\pi/8$ gate	T	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi 4} \end{pmatrix}$
Rz(θ) gate	Rz(θ)	$\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$

Table 14.1: List of commonly used single-qubit gates

14.1.2 Quantum gates

Since quantum mechanical time evolution is unitary (apart from measurements that collapse the wave function), we can only perform unitary operations on quantum bits and measurements. This is the second big restriction.

Just as for classical computers it is convenient to build a quantum circuit from a set of quantum gates that act on a limited set of qubits. Classical circuits are typically built from a set of gates that include OR, AND, NOT, XOR and more. However, in principle only the NAND (not-AND) gate is necessary since all other gates can be built from it. The NAND gate is thus called universal: any classical computation can be done purely with NAND gates, and any boolean function can be written in terms of NAND gates. It still makes sense however to consider more types of gates when building circuits, to make the design of circuits easier.

For quantum circuits one similarly often uses a larger set of gates than is strictly necessary. In the following we will discuss a set of typically used one and two qubit gates and will then discuss which ones are strictly necessary.

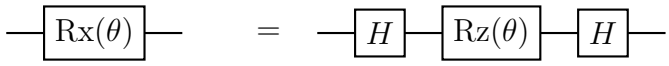
14.1.2.1 Single qubit gates

A few remarks may be useful. The X gate is the quantum analog of a classical NOT gate. The Hadamard gate (H) squares to the identity and is essentially a ninety degree rotation around the y axis, rotating a state aligned with z to x. The T gate is also called $\pi/8$ gate since it can – up to an irrelevant global phase – be written as

$$T = e^{i\pi/8} \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}. \tag{14.4}$$

The Rz gate performs a rotation around the z axis in spin space. Similar rotations around the x and y axis are performed by the Rx and Ry gates. For example, a

rotation around the x axis can be performed by swapping z and x by a Hadamard gate, performing a rotation around the z axis, and then rotating back:

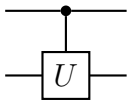


(14.5)

14.1.2.2 Two-qubit gates

A set of common two-qubit gates are controlled gates, consisting of a control qubit and a target qubit. The controlled version CU of a single qubit gate U (any of the list above) performs the single qubit operation U on the target qubit only if the control qubit is set to 1.

The quantum circuit for such a gate is:

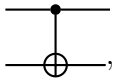


(14.6)

Denoting the matrix representation of the gate U as $\mathcal{U}$, the matrix representation of CU in a basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ is

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\mathcal{U}$}} \\ 0 & 0 & & \end{array}\right). \tag{14.7}$$

Maybe the most important two-qubit gate is the controlled-NOT-gate (CNOT), which is the same as a controlled-X gate. It is typically drawn as:

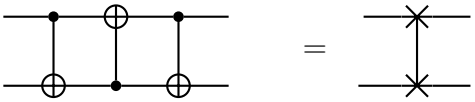


(14.8)

its matrix representation is

$$\text{CNOT} \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{14.9}$$

Other two-qubit gates can be built from single qubit gates and the CNOT gate. For example, the SWAP gate, which swaps the states of two qubits, can be built from three CNOT gates as:



(14.10)

In matrix representation the previous circuit corresponds to:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{14.11}$$

The last gate corresponds to the SWAP operation, indeed one can immediately verify that its action corresponds to

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{pmatrix} = \begin{pmatrix} c_{00} \\ c_{10} \\ c_{01} \\ c_{11} \end{pmatrix}. \tag{14.12}$$

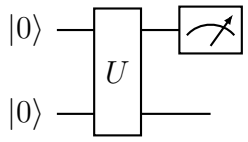
14.1.2.3 Universal gate sets

Of the above gates just the Hadamard, $\pi/8$ and CNOT gates are sufficient to implement any quantum circuit. All the other gates can be built from these gates, similar to the NAND gate being universal for classical computing.

The tricky part is how to represent arbitrary rotations using a discrete gate set. The Solovay-Kitaev algorithm allows to approximate arbitrary rotations to within any desired accuracy ϵ, with just poly$(\log(1/\epsilon)$ gates.[1] Better algorithms for approximation of rotations have recently been invented and this is still an interesting field of research.

14.1.3 Measurements

Measurements in quantum circuits can be done on an arbitrary number of qubits, and the conventional setting of quantum computing is to consider measurements of qubits in the computational basis, thus corresponding to measuring the spin in the Z direction. Diagrammatically, measurements are indicated as in the figure below, that shows a measurement of the uppermost qubit:

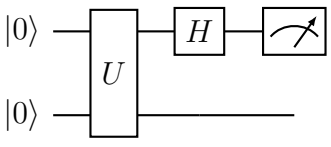


(14.13)

with a probability of observing one of the two possible outcomes ($+1$ or -1) given in this case by

$$P(s_1 = \pm 1) \quad = \quad \sum_{s_2=\pm 1} |\langle s_1, s_2 | \hat{U} | 0, 0 \rangle|^2 \tag{14.14}$$

Measurements in other directions are possible after applying suitable rotations of the spin, for example measuring in the X direction can be done by performing a rotation through the Hadamard gate. The circuit below shows a measurement of this type:

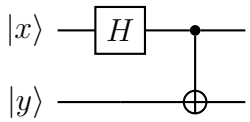


(14.15)

[1] The notation poly(x) indicates an effort that is polynomial in x.

14.1.4 Example: Creating entangled quantum states

As a first simple quantum algorithm that can be constructed using the build blocks discussed above, we will consider the task of creating entangled quantum states. Those can be very useful, for instance, in quantum communication tasks. One of the most prototypical entangled states are Bell states, also called EPR pairs after the Einstein-Podolsky-Rosen paper. These are produced by the following circuit:



$$= \qquad |\beta_{x,y}\rangle \tag{14.16}$$

Starting from an initial state $|\psi_0\rangle = |x\rangle\,|y\rangle$, where x and y are either 0 or 1 and not in superposition, the circuit produces a final state

$$|\beta_{xy}\rangle \equiv \frac{|0,y\rangle + (-1)^x|1,\overline{y}\rangle}{\sqrt{2}}. \tag{14.17}$$

For pure initial states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ we obtain the four Bell states called $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$ and $|\beta_{11}\rangle$. The four possible outcomes are summarized in the Table below:

In	Out
$\|00\rangle$	$\frac{\|00\rangle+\|11\rangle}{\sqrt{2}}$
$\|01\rangle$	$\frac{\|01\rangle+\|10\rangle}{\sqrt{2}}$
$\|10\rangle$	$\frac{\|00\rangle-\|11\rangle}{\sqrt{2}}$
$\|11\rangle$	$\frac{\|01\rangle-\|10\rangle}{\sqrt{2}}$

Table 14.2: Input-Output table for the circuit creating Bell states.

14.2 Simulating the dynamics of quantum systems

14.2.1 Time evolution of a quantum spin model

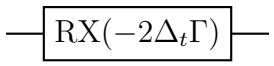
Exponential speedup over classical computers can be obtained for the simulation of the dynamics of quantum systems. As an example, we will consider once more the transverse field Ising model. Notation-wise, we will adopt a calligraphic notation for the Hamiltonian to avoid confusion with the Hadamard gate and we will also use the notation for Pauli matrices normally adopted in quantum computing: $\hat{Z}_i = \hat{\sigma}_i^z$, $\hat{X}_i = \hat{\sigma}_i^x$, $\hat{Y}_i = \hat{\sigma}_i^y$, thus our Hamiltonian is

$$\hat{\mathcal{H}} = -\Gamma \sum_i \hat{X}_i + \sum_{\langle i,j \rangle} J_{ij} \hat{Z}_i \hat{Z}_j. \tag{14.18}$$

In order to simulate the time evolution on a quantum computer we have to use a Trotter decomposition just like in the classical case, and again have the choice between simpler low-order approximations or more accurate high-order ones that are more complex but also more accurate.

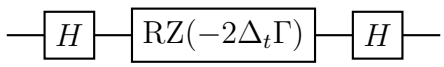
In this sense, the situation is entirely analogous to what we have seen when doing exact time evolution of quantum systems in the first lectures. The big advantage of quantum computers shows in the implementation of the individual terms of the Trotterized time evolution, which is now much easier. First, we need just N qubits instead of 2^N complex numbers in a classical code and requires only $\mathcal{O}(N)$ instead of $\mathcal{O}(2^N)$ operations.

The time evolution under the transverse field term $e^{i\Delta_t \Gamma \sigma_i^x}$ is trivial to implement, since it is just a rotation around the x axis, implemented by an RX(θ) gate with an angle $\theta = -2\Delta_t\Gamma$:



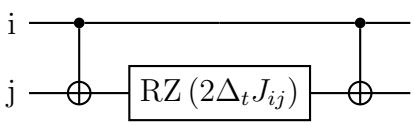
(14.19)

If we the quantum hardware does not offer an RX gate, but only (for example) arbitrary rotations around the z directions through the RZ(θ), then a basis transformation will be needed. It is easy to convince oneself that the Hadamard gate is the unitary matrix that transforms from the Z basis to the X basis, thus the Trotter step associated to the dynamics a single spin under the transverse field can be written as :



(14.20)

The Ising term is a 2-spin coupling and requires a 2-qubit gate. To implement $e^{-i\Delta_t J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z}$ one needs to rotate by an angle $-\Delta_t J_{ij}$ if the two spin values are the same and $+\Delta_t J_{ij}$ if they differ. The following simple circuit can realize this operation:



(14.21)

Similar circuits can be designed for other quantum models, as we will do in the exercises for a quantum Heisenberg model.

14.3 Quantum Phase Estimation

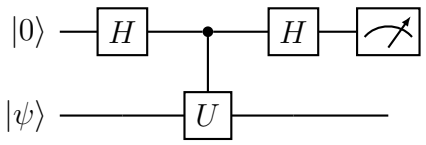
14.3.1 Measuring energies from phases

We now discuss an algorithm to estimate the energy of a given quantum state. The most straightforward way of measuring the energy of a quantum state $|\psi\rangle$ is to measure all the terms that make up the Hamiltonian $\hat{\mathcal{H}}$ and thus evaluate $\langle\psi|\hat{\mathcal{H}}|\psi\rangle$. However, this approach has several disadvantages. The wave function $|\psi\rangle$ gets destroyed with every measurement and we get only N bits of information. As this approach is similar to Monte Carlo sampling, we need $\mathcal{O}(\epsilon^{-2})$ measurements and thus $\mathcal{O}(\epsilon^{-2})$ preparations of the wave function $|\psi\rangle$ to measure the energy to an accuracy ϵ.

An alternative is to measure the phase which a state $|\psi\rangle$ picks up under time evolution with $\hat{\mathcal{H}}$. Let us first assume that $|\psi\rangle = |E_n\rangle$ be an eigenstate $|E_n\rangle$ of $\hat{\mathcal{H}}$ with eigenvalue E_n. Under time evolution $e^{-i\hat{\mathcal{H}}t}|\psi\rangle = e^{-iE_nt}|E_n\rangle$ the state picks up a phase E_nt. Measuring this phase would thus allow to measure the energy.

14.3.2 Quantum phase estimation algorithm

But, how do we measure the phase under time evolution? At first sight the phase is not an observable quantity. However, we can set up an "interference experiment" to determine the phase. We add an auxiliary qubit and perform the evolution under $\hat{U} = e^{-i\hat{\mathcal{H}}t}$ only if the auxiliary qubit is on:



(14.22)

Let us analyze step by the step what this circuit does. First, we start from $|0\rangle|\psi\rangle$ and apply a Hadamard gate to the auxiliary qubit, giving

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle\right) \tag{14.23}$$

We then apply the evolution controlled by the auxiliary qubit:

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + |1\rangle\hat{U}|\psi\rangle\right). \tag{14.24}$$

If $|\psi\rangle$ is an eigenstate $|E_n\rangle$ we pick up the corresponding phase $\phi = E_nt$

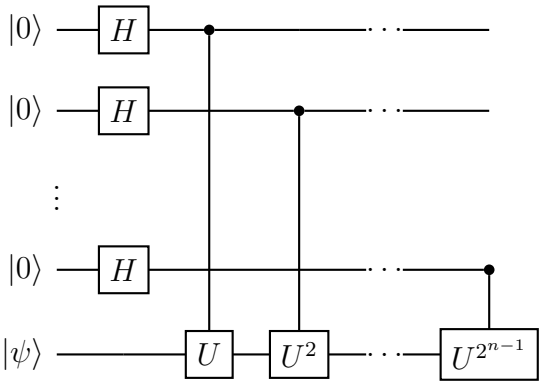
$$\frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + e^{-i\phi}|1\rangle|\psi\rangle\right). \tag{14.25}$$

Measuring now would not give us any information since the phase cannot be determined from a direct measurement. However, we can interfere the two cases by another Hadamard transform on the auxiliary qubit, obtaining

$$\frac{1}{2}\left[\left(1 + e^{-i\phi}\right)|0\rangle|\psi\rangle + \left(1 - e^{-i\phi}\right)|1\rangle|\psi\rangle\right]. \tag{14.26}$$

Now we measure the auxiliary qubit and we notice that the probability of measuring 0 is $(1+\cos\phi)/2 = \cos(\phi/2)^2$. By repeating then this experiment many times we can experimentally reconstruct (just counting how many times we measure 0) the probability of measuring 0, and consequently reconstruct ϕ. Using the histogram method, this probability can be determined to an accuracy of $1/\sqrt{M}$, if M measurements are realized. Thus, one needs to go through at least $M \sim 2^{2m}$ independent rounds of measurements to obtain m accurate binary digits of ϕ.

A more efficient version of this algorithm is due to Kitaev. We only sketch here the idea of this improved approach, that performs time evolutions with times $2^k t$ ($k = 1, \ldots, n$), that are powers of 2, by adding n auxiliary qubits rather than one as in the simpler approach discussed previously. One starts by preparing the state as given by the circuit below:



(14.27)

This is a generalization of the case with a single auxiliary qubit, and we can readily see that the output state of the circuit above is

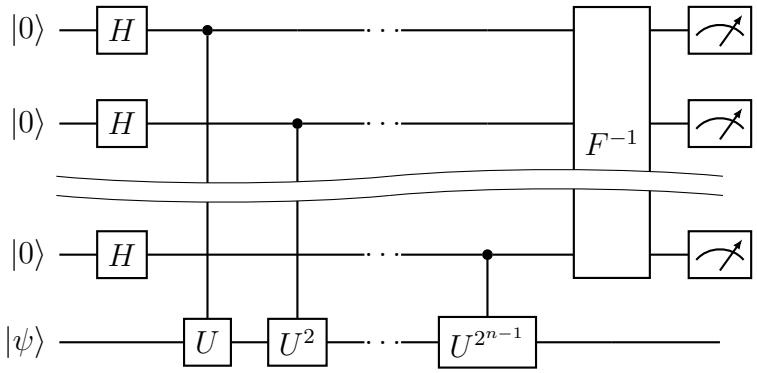
$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{-i\phi}|1\rangle\right) \otimes \left(|0\rangle + e^{-i2\phi}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{-i2^{n-1}\phi}|1\rangle\right) \otimes |\psi\rangle. \tag{14.28}$$

We therefore end up with a register containing the Fourier transform of the phase:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{-i\phi k} \, |k\rangle \, |\psi\rangle \tag{14.29}$$

Employing an inverse quantum Fourier transform F^{-1} (which is a unitary operation and can be implemented efficiently on a quantum computer) we measure a binary representation $\pi x/2^n$ of the phase ϕ.

The resulting algorithm is written in diagrammatic form below:



(14.30)

The main advantage of Kitaev's phase estimation approach is that it allows to reconstruct the phase much more accurately than using a single auxiliary qubit, since we can determine its value "digit-by-digit" in binary form through the inverse Fourier transform.

14.3.3 Quantum phase estimation to find the exact eigenvalues

It is especially interesting to use the quantum phase estimation algorithm in the case when $|\psi\rangle$ is not an exact eigenstate of the Hamiltonian, but rather a generic state. We can then expand it in the the eigenbasis of $\hat{\mathcal{H}}$:

$$|\psi\rangle \;=\; \sum_{l=0}^{2^N-1} c_l |E_l\rangle. \tag{14.31}$$

Since all the operations we have carried on in the quantum phase estimation are linear, it is not hard to convince one-self that if we apply the controlled time evolutions to an approximate state we get

$$\frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^N-1} c_l \left(|0\rangle + e^{-iE_l t}|1\rangle\right) \otimes \left(|0\rangle + e^{-i2E_l t}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{-i2^{n-1}E_l t}|1\rangle\right) \otimes |E_l\rangle =$$

$$= \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^N-1} \sum_{k=0}^{2^n-1} c_l e^{-i(E_l t)k} |k\rangle |E_l\rangle. \tag{14.32}$$

Further applying the inverse Fourier transform to this state then will return frequencies that correspond to the exact eigen-energies of the Hamiltonian! The intensity of these peaks will be proportional to the coefficients $|c_l|^2$. We therefore see that if we have access to a reasonable approximation of the ground state $|\psi\rangle$ such that its overlap with the exact ground state is not exponentially small, then by performing quantum phase estimation over such state will give us the exact energies of the system. In this sense, QPE is a very powerful technique. The important caveat, however, is that in general it is not easy to prepare a simple state $|\psi\rangle$ that has a sizable overlap with the exact ground state. In general, finding such an initial state is an exponentially hard task, however we will analyze strategies to do so in the next lecture.

Bibliography

[1] Feynman, R.P., 1982. Simulating physics with computers. Int J Theor Phys 21, 467–488.

[2] Kaye, P., Laflamme, R., Mosca, M., 2007. An Introduction to Quantum Computing, 1st edition. ed. Oxford University Press, Oxford.

[3] Rieffel, E.G., Polak, W.H., 2011. Quantum Computing: A Gentle Introduction, Scientific and Engineering Computation. MIT Press, Cambridge, MA, USA.