Chapter 3

Quantum Circuit Model of Computation

3.1 Introduction

In the previous chapter we saw that any Boolean function can be computed from a reversible circuit. In particular, in principle at least, this can be done from a small set of universal logical gates which do not dissipate heat dissipation (so we have logical reversibility as well as physical reversibility. We also saw that quantum evolution of the states of a system (e.g. a collection of quantum bits) is unitary (ignoring the reading or measurement process of the bits). A unitary matrix is of course invertible, but also conserves entropy (at the present level you can think of this as a conservation of scalar product which implies conservation of probabilities and entropies). So quantum evolution is also both "logically" and physically reversible.

The next natural question is then to ask if we can use quantum operations to perform computational tasks, such as computing a Boolean function? Do the principles of quantum mechanics bring in new limitations with respect to classical computations? Or do they on the contrary bring in new ressources and advantages?

These issues were raised and discussed by Feynman, Benioff and Manin in the early 1980's. In principle QM does not bring any new limitations, but on the contrary the superposition principle applied to many particle systems (many qubits) enables us to perform parallel computations, thereby speeding up classical computations. This was recognized very early by Feynman who argued that classical computers cannot simulate efficiently quantum mechanical processes. The basic reason is that general quantum states involve a superposition of 2^n classical states :

$$|\psi\rangle = \sum_{b_1 \ldots b_n \in \{0,1\}^n} c_{b_1,\ldots,b_n} |b_1 \ldots b_n\rangle$$

Here the Hilbert space is $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 = (\mathbb{C}^2)^{\otimes n}$ A classical simulation of the evolution of $|\psi\rangle$ must perform essentially 2^n simulations for the evolution of each $|b_1 \ldots b_n\rangle$. On the contrary, the unitary quantum dynamics acts on $|\psi\rangle$ as a whole (or on each $|b_1 \ldots b_n\rangle$ in parallel). So physical devices performing a unitary dynamics on $|\psi\rangle$ can be viewed as devices performing a quantum computation.

Feynman developed the concept of quantum computation in the language of Hamiltonian dynamics. It turns out that this is not very practical if we are going to build a universal quantum computation which perform diverse tasks. But a classical universal computer can be represented by a circuit model built out of a given set of elementary gates acting in a recursive way on the input of the computation. Around 1985 David Deutsch showed that the same holds in the quantum case. Namely, any unitary evolution can be approximated well enough by some set of universal elementary quantum gates.

Nowadays it is the Deutsch model – the quantum circuit model – of a quantum computer that is the most popular and developed model for quantum computation. The subject of this chapter is to explain this model. There is also a notion of quantum Turing machine which is (analogous to classical Turing machines and) the natural and most convenient framework to discuss quantum complexity classes. It has been shown that the Quantum Turning machine model is equivalent to the quantum circuit model. Complexity issues are briefly discussed at the end of this chapter.

3.2 Quantum gates and their representations

As we will see the quantum circuits are built out of a small set of gates. For this reason it is useful to start by listing a few of the most important gates that we will encounter.

3.2.1 Single Qbit gates

- The three Pauli-gates $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$; $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ and $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

  For example, $|b\rangle \longrightarrow \boxed{\text{X}} \longrightarrow |\bar{b}\rangle$ is the quantum NOT gate.

Quantum mechanically, the input can also be a coherent superposition of the states $\{|0\rangle, |1\rangle\}$. For example,

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

- The Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. For example,

$$|b\rangle \longrightarrow \boxed{\text{H}} \longrightarrow H|b\rangle = \tfrac{1}{\sqrt{2}}\big(|0\rangle + (-1)^b|1\rangle\big)$$

- The "$\frac{\pi}{8}$ gate" $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ [Up to a global phase this is multiplicative by $e^{\pm i\pi/8}$]. We have for example

$$|b\rangle \longrightarrow \boxed{\text{T}} \longrightarrow e^{ib\pi/4}|b\rangle = \begin{cases} |0\rangle \to |0\rangle \\ |1\rangle \to e^{i\pi/4}|1\rangle \end{cases}$$

For superpositions we have : $\alpha|0\rangle + \beta|1\rangle \to \alpha|0\rangle + \beta e^{i\pi/4}|1\rangle$.

- The gate $S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad \begin{cases} |0\rangle \to |0\rangle \\ |1\rangle \to i|1\rangle. \end{cases}$

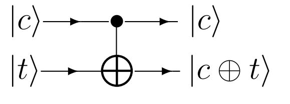
An important Lemma that we give here without proof is

Lemme 1 (Approximation of single Qbit gates by H and T.) *Any unitary single bit gate U can be approximated to arbitrary precision δ by a concatenation of "Hadamard H" and "$\frac{\pi}{8} - T$ gates". Moreover, if V is the concatenation of H and T gates approximating U and $||U - V|| < \delta$, we need at most $O(\ln\delta)$ gates H and T [This last statement is known as the Solevay-Kitaev theorem].*

Proof idea. The main idea of the proof is to represent U by a succession of rotations, themselves represented by Pauli-gates, themselves represented by H and T. It is not very difficult to prove that a circuit size $O(1/\delta)$ is sufficient. The Solovay-Kitaev result $O(\ln\delta)$ is more difficult. ■

3.2.2 Controlled two-bit gates.

- The CNOT gate (controlled not) is the prototypical two-bit gate:



control bit.
target bit.

In the basis

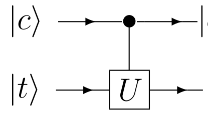
$$|0\rangle \otimes |0\rangle \ ; \quad |0\rangle \otimes |1\rangle \ ; \quad |1\rangle \otimes |0\rangle \ ; \quad |1\rangle \otimes |1\rangle$$

$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \qquad \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

the matrix representation is

$$CNOT = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

- The controlled-U gate where U is a single bit operation.

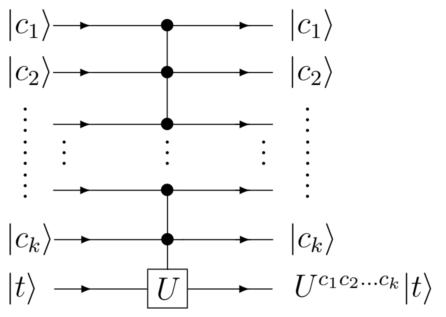


$$U^c|t\rangle = \begin{cases} |t\rangle \text{ if } c = 0 \\ U|t\rangle \text{ if } c = 1 \end{cases}$$

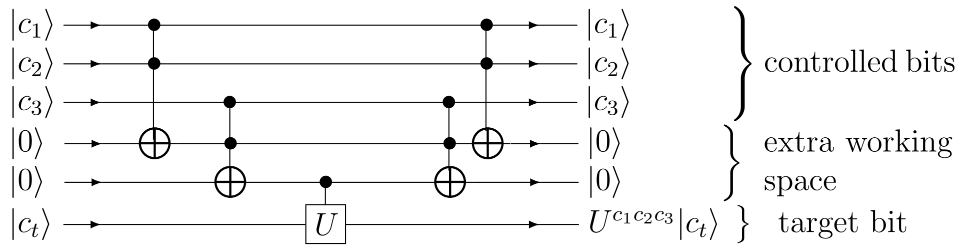
[Exercise: Build a CNOT with a controlled Z and two Hadamard H.]

3.2.3 Multi-controlled gates.

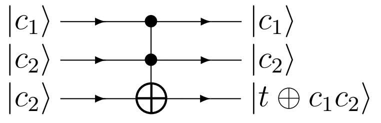
A generalization of the previous controlled-U gate is the multi-controlled-U



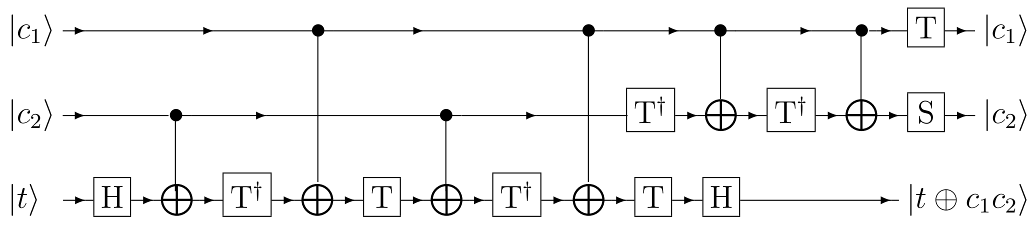
So U acts on the target bit if and only if all control bits are set to 1. By increasing the work space this gate can be represented by a concatenation of controlled-controlled-NOT and a controlled U. Indeed:



The controlled-controlled-NOT gate is also called Quantum Toffoli. Remarkably, it can be represented by one and two-bit quantum gates $\{T, S, H, CNOT\}$. Remember that classically this is not possible! The reader can check that



is equivalent to the following circuit:



Summarizing we arrive at the following lemma:

Lemme 2 *Any multibit-controlled gate U acting on N Qbits ($2^n \times 2^n$ matrix) can be represented by the set $\{T, S, H, CNOT, U\}$ where U acts on the last Qbit (2×2 matrix).*

3.3 A universal set of quantum gates

An important lemma that we give here without proof is:

Lemme 3 *Any unitary U acting on N-Qbits states, i.e., states in $(\mathbb{C}^2)^{\otimes n}$ can be decomposed as a finite product of "two level unitaries":*

$$U = U^{(i_1j_1)} \otimes U^{(i_2j_2)} ... U^{(i_Kj_K)}$$

where $U^{(ij)}$ acts from $(\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ on coordinates i, j and trivially on all other spaces.

For example if n = 4 we may have

$$U^{(14)} = \begin{pmatrix} a & 0 & 0 & b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ c & 0 & 0 & d \end{pmatrix} \text{ with } \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ unitary.}$$

By implementing suitable permutations of basis vectors with CNOT, we obtain:

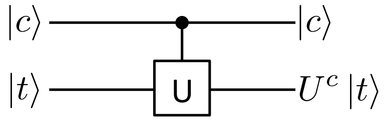
Lemme 4 *Any two level unitary $U^{(ij)}$ acting as U on coordinates i, j and as the identity on all others can be implemented by a concatenation of* CNOT *and a multi-controlled single bit U.*

Proof sketch for $n = 4$. Permutations bring $U^{(14)}$ to the form

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{U}} \\ 0 & 0 & & \end{array}\right)$$

These permutations can be realized with SWAP gates. The SWAP gates can themselves be realized with $\{T, S, H, CNOT\}$.

It remains to show this matrix is a controlled U:



Indeed,

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{U}} \\ 0 & 0 & & \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes U + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Now,

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes U\,|c\rangle \otimes |t\rangle = \begin{cases} 0 & \text{if } |c\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \\ |c\rangle \otimes U\,|t\rangle & \text{if } |c\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \end{cases}$$

and

$$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} \otimes \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} |c\rangle \otimes |t\rangle = \begin{cases} |c\rangle \otimes |t\rangle & \text{if } |c\rangle = \begin{pmatrix}1\\0\end{pmatrix} = |0\rangle \\ 0 & \text{if } |c\rangle = \begin{pmatrix}0\\1\end{pmatrix} = |1\rangle \end{cases} .$$

Thus,

$$\begin{aligned}\begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix} \otimes U + \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} \otimes \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} &= \begin{cases} |c\rangle \otimes |t\rangle & \text{if } |c\rangle = |0\rangle \\ |c\rangle \otimes U\,|t\rangle & \text{if } |c\rangle = |1\rangle \end{cases} \\ &= |c\rangle \otimes U^c\,|t\rangle \,. \end{aligned}$$

■

One can show that there exist N-Qbit unitary matrices ($2^n \times 2^n$ matrices) such that the decomposition obtained by lemmas 3 and 4 requires $O(e^n)$ gates. For some special problems such as factoring we will see that $O\big(\text{poly}(n)\big)$ suffices.

Combining lemmas 1, 2, 3, 4 we arrive at the following basic theorem on which the quantum circuit model of quantum computation is based:

Theorem 5 *Any $2^n \times 2^n$ unitary matrix U acting on $\mathbb{C}^2 \otimes ... \otimes \mathbb{C}^2 = (\mathbb{C}^2)^{\otimes n}$ can be represented to arbitrary accuracy by a concatenation of the finite set of single and two-bit gates $\{T, S, H, CNOT\}$.*

If the required accuracy is δ, one can argue that the maximal number of gates is of the form $O((\ln \delta) e^n)$. One can show that there exists unitary U for which it is not possible to have $O((\ln \delta)\text{poly}(n))$.

3.4 Deutsch model of quantum computation

The basic theorem just explained justifies the following model for quantum computation.

Definition 6 (A quantum circuit) *A quantum circuit is defined by the following:*

(a) A quantum circuit is a directed acyclic graph whore vertices are gates among the finite set $\{T, S, H, CNOT\}$. The wires "carry" single Qbits ($\alpha|0\rangle + \beta|1\rangle$).

(b) *The input is set to the simple tensor product state*

$$|0\rangle \otimes \cdots \otimes |0\rangle.$$

(c) *The output is the result of the unitary evolution operating on the input. The output is in general a state of the form*

$$|\psi\rangle = \sum_{c_1 \ldots c_n \in \{0,1\}^n} A(c_1 \ldots c_n)|c_1 c_2 \ldots c_n\rangle$$

(d) *Finally, a measurement is performed on $|\psi\rangle$ with an apparatus measuring in the basis $\{|0\rangle, |1\rangle\}^{\otimes n}$. The outcome of the measurement is "the result of the computation" $|c_1 \ldots c_n\rangle$ obtained with probability $|A(c_1 \ldots c_n)|^2$.*

A few remarks about this model are in order:

1. Acting on "Qtrits" instead of "Qbits" would not change anything fundamental (*e.g.* the order of magnitude with respect to n of the size or complexity of the circuit does not change).

2. Initializing with a different state can be accounted for by adding extra unitary gates in the initial stages of the circuit. Note that this may change the complexity. Note also that nature could potentially offer us initial states that inherently contain some high complexity.

3. Performing measurements in another basis simply amounts to first unitarily rotate the basis so this can be viewed as an adjunction of unitary gates at the end of teh circuit (just before the emasurement apparatus). Concerning complexity the same remarks as in previous item apply.

4. Performing measurements at intermediate stages instead of at the end does not change anything.

5. Other sets of universal gates exist. It may be surprising that in the classical case three bit gates are needed whereas this is not the case for quantum computation. But from a more physical point of view this is not surprising because the classical three bit gates can be viewed as an effect of "two-body interaction" [see Billiard-Ball-Model and Fredkim gate].

6. A quantum computation is reversible as long as the measurement has not been performed.

7. A reversible classical computation can be represented by a unitary operator. Indeed

$$\tilde{f}(x_1...x_n, y) = (x_1...x_n, y \oplus f(x_1...x_n))$$

can be represented by the unitary

$$U_f|x_1...x_n, y\rangle = |x_1...x_n, y \oplus f(x_1...x_n)\rangle.$$

That U_f is unitary, can easily verified by checking that it preserves the scalar product (exercises). Thus any classical reversible computation is included in the model of quantum computation.

8. The power of quantum computation comes from the simultaneous action of the unitary evolution on all 2^n strings $|c_1...c_n\rangle$. The complexity of the calculation is given by the size of the circuit. Since the result is obtained with some probability, typically one must repeat a certain number of times the computation to get a result with high probability (hopefully). This repetition may add to the complexity.