Quantum Computing (CST Part II)

Lecture 5: The Quantum Circuit Model

Information is physical.
Rolf Landauer

Resources for this lecture

Nielsen and Chuang chapter 4 contains a thorough introduction to the quantum circuit model (although this is rather more than is needed for this course).

Quantum circuits: the big picture

This lecture represents a shift in perspective from seeing quantum mechanical events as merely natural phenomena, to instead seeing them as executable operations in a programmable computer.

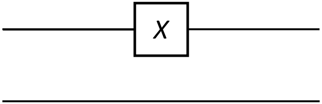
There is, however, a subtlety here: the postulates of quantum mechanics describe what will happen to a *closed* quantum system, however treating quantum phenomena as controllable and executable necessarily implies some opening of the system: we later plug this gap by considering noisy quantum systems.

Tensor networks

We have already seen that qubit states can be entangled (not separable), however we can apply separable operations even to entangled states. Consider:

- A two qubit state: $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$
- Performing a Pauli-X on the first qubit only.

From the previous notes on linear algebra and the postulates of quantum mechanics, we know that this yields a state, $|\psi'\rangle$, equal to $(X \otimes I)|\psi\rangle$. However, we can also consider a tensor network, with each wire representing a qubit:



As the Pauli-X is a "not" operation, we immediately get

$$|\psi'\rangle = \alpha|10\rangle + \beta|11\rangle + \gamma|00\rangle + \delta|01\rangle$$

Exercise: prove consistency with the matrix calculation.

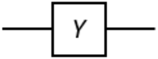
Quantum circuits: from matrices to gates

In the tensor network, we have that:

- Wires are qubits (possibly entangled).
- Gates are unitary matrices.

We have already met the Pauli and Hadamard single-qubit unitary matrices as well as the CNOT two-qubit unitary, and the *phase gate* $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ is also a useful primitive.

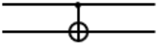
Pauli-X:


Pauli-Y:


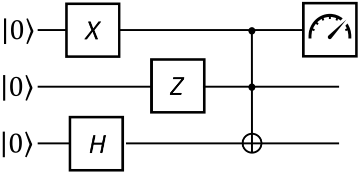
Pauli-Z:


Hadamard:


Phase:


CNOT:


Quantum circuits



A quantum circuit is a tensor network of n qubits, with three stages:

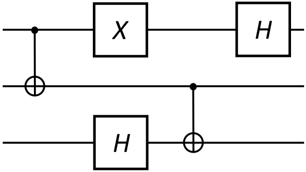
- Initialisation of all qubits in the $|0\rangle$ state (denoted $|0\rangle^{\otimes n}$).
- Some quantum gates, which represent unitary transformations.
- A final layer of measurements in the computational basis, on some or all of the qubits.

The matrix of a quantum circuit

As the quantum circuit (with the initialisation and measurement stages omitted) just represents a unitary evolution, we can express the whole thing as a matrix. We must follows the following two rules:

- Composition across wires is achieved by the tensor product.
- Composition along (sets of) wires is achieved by the normal matrix product, but right to left.

For example:



Is equal to:

$$(H \otimes I_4) \times (I_2 \otimes \mathsf{CNOT}) \times (X \otimes I_2 \otimes H) \times (\mathsf{CNOT} \otimes I_2)$$

where I_2 is the 2×2 identity, and $I_4 = I_2 \otimes I_2$ is the 4×4 identity.

Quantum computational power (1/2)

The quantum circuit model completely captures the postulates of quantum mechanics:

- The wires represent the state-space of a composition of 2-level quantum systems (qubits), which can be entangled – postulates 1 and 4.
- The gates are just a convenient way of writing down the unitary evolution – postulate 2.
- Measurement occurs (and it can be shown that this can always be deferred to the end of the circuit) – postulate 3.

Furthermore, there is no loss in generality in assuming that we can prepare the states as $|0\rangle^{\otimes n}$.

It follows that any computation leveraging the quantum nature of some physical system can, in principle, be expressed using the quantum circuit model.

Quantum computational power (2/2)

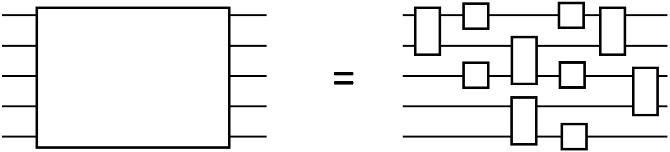
Additionally:

- Quantum computing generalises classical computing, and so any classical computation can be performed on a quantum computer.
- It has been shown that quantum computing does not violate the Church-Turing thesis – there is no problem that is solvable on a quantum computer that is not on a classical computer... what quantum computers give us is a more efficient way to do some computations.

Locality constrains the physical realisation of gates

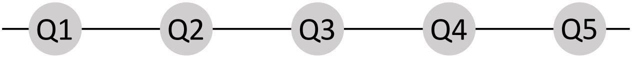
Unitary matrices of all dimensions exist, thus in principle quantum gates of all dimensions exist... however quantum computers live in physical space, and so it follows that it is physically unreasonable to assume that we can have an arbitrary number of qubits in a single operation (that is, that we can have gates of any size). In fact, usually we assume that we are only allowed to use single- and two- qubit gates.

It has been proven that two-qubit unitaries are universal, in the sense that any arbitrary n-qubit unitary can be decomposed as a product of two-qubit unitaries, e.g.:

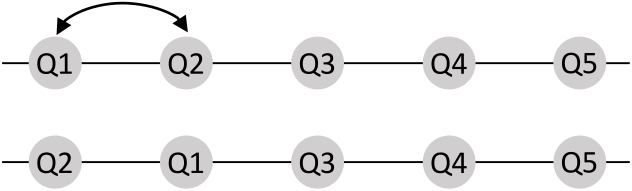


Qubits located in an array

Not only do we assume that we can only perform operations (gates) on one or two qubits, but in physical quantum computers two qubits that undergo a two-qubit gate must be physically adjacent. For example, the qubits may be laid out in a linear array:

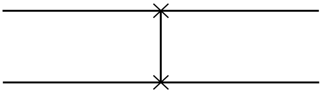


If a gate is to be executed on qubits 1 and 3, it is necessary to *swap* qubits 1 and 2 such that qubits 1 and 3 are adjacent:



The SWAP gate

Fortunately, this swapping can be achieved using the SWAP gate, which swaps the states of two qubits:



Let $|\psi_1\psi_2\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$, which corresponds to the vector $[\alpha, \beta, \gamma, \delta]^T$, we have that:

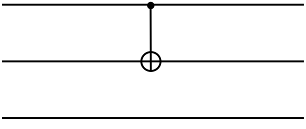
$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} \alpha \\ \gamma \\ \beta \\ \delta \end{bmatrix} = \text{SWAP} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix}$$

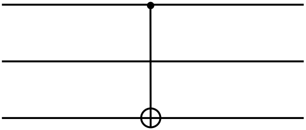
i.e., is equal to $|\psi_2\psi_1\rangle = \alpha\,|00\rangle + \gamma\,|01\rangle + \beta\,|10\rangle + \delta\,|11\rangle$.

SWAP can be constructed from three CNOT gates (**exercise sheet**).

Matrix representation of CNOT on non-adjacent qubits

Even though the existence of the SWAP gate is crucial for practical considerations, we continue to write down two-qubit operations on non-adjacent qubits. This raises the question of how to express them in matrix form. For example, consider the following





We know that we can express the left-hand circuit as CNOT $\otimes I_2$, but how would we express the right-hand circuit?

...we can just SWAP, do the CNOT on adjacent qubits and then SWAP back:

$$(I_2 \otimes \text{SWAP}) \times (\text{CNOT} \otimes I_2) \times (I_2 \otimes \text{SWAP})$$

How many one- and two-qubit gates do we need?

Previously, it was asserted that an arbitrary unitary operation could be decomposed into a product of one- and two- qubit unitaries. However, as a unitary is a matrix of complex numbers this leaves two possibilities:

- Either we require a continuum of two qubit unitaries (i.e., an infinite number of gates).
- Or we can construct arbitrary one- and two-qubit unitaries from a finite set of unitaries (a finite universal gate-set).

In fact, the latter is true, indeed we can *efficiently* approximate any circuit consisting of CNOT gates and single qubit unitaries to a desired accuracy ϵ:

The Solovay-Kitaev theorem implies that any circuit containing m CNOTs and arbitrary single qubit unitaries can be approximated to an accuracy ϵ by a circuit using a universal finite gate-set with $\mathcal{O}(m \log^c(m/\epsilon))$ gates, where $c \approx 2$.

A universal gate-set

Perhaps surprisingly, only three gates are needed to form a universal gate-set, two we have met: CNOT and H, and the third is:

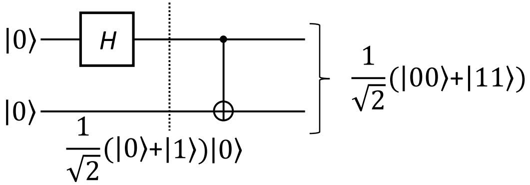
$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

The introduction of this T gate is, however, crucial, and the famous Gottesman-Knill theorem holds that any circuit consisting of just the gates we have met thus far $X, Y, Z, H, S,$ CNOT can be efficiently simulated on a classical computer.

We can see that the single-qubit gates we have met so far can be expressed in terms of H and T as follows:

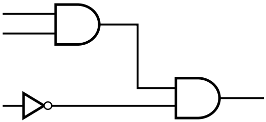
- $S = T^2$
- $Z = S^2$
- $X = HZH$
- $Y = iXZ = SXSZ$

Quantum circuit example 1: entangling two qubits



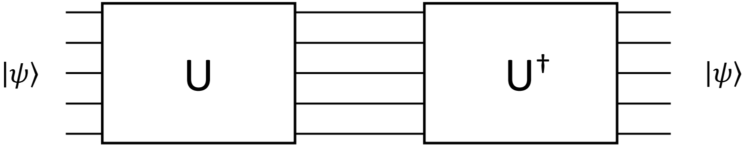
Comparison with classical logic circuits

By expressing quantum evolutions in circuit form, we can express physical phenomena in a manner that can be recognised as similar to classical logic circuits, with which we are all very familiar.



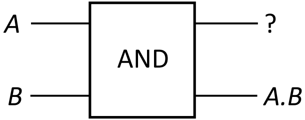
There are, however, two important distinctions:

- Quantum gates have exactly the same number of outputs as they have inputs.
- Moreover, as the gates represent unitary matrices, they are invertible.



An invertible AND gate?

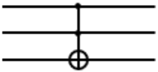
Consider the classical logic gate the "AND" gate. Clearly it is not invertible, as one input leads to two outputs. However, if we give the "AND" gate a second output, can we make it invertible? That is:



In fact we cannot – we have three occasions when the second output is zero $(A = 0, B = 0)$; $(A = 0, B = 1)$; $(A = 1, B = 0)$, and only one bit with which to distinguish them, so we can never reconstruct the inputs A and B from two outputs of which one is $A.B$.

The Toffoli gate

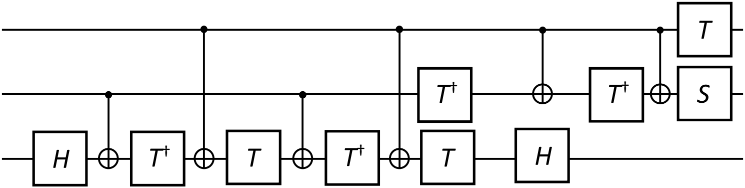
The Toffoli gate *does* provide a quantum generalisation of the classical AND gate, with three inputs and outputs.



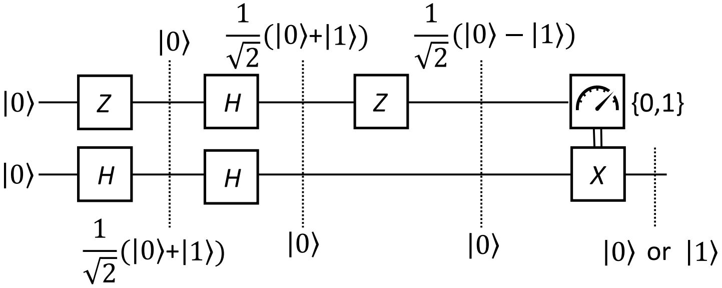
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

When the first two inputs are classical bits ($|0\rangle$ or $|1\rangle$), and the third is $|0\rangle$ the third output is the AND of the first two inputs.

Quantum circuit example 2: decomposing the Toffoli gate into two-qubit unitaries



Quantum circuit example 3: self-inverse nature of H and classical control



Summary

For the remainder of the course, it is crucial to be comfortable with manipulating quantum circuits. The main points to remember from this lecture are:

- Quantum circuits are tensor networks where the wires are qubits and the gates are one- or two- qubit unitary operations.
- Quantum circuits can be used to completely represent quantum computation, and the class of problems solvable on a quantum computer is exactly equal to that on a classical computer.
- CNOT, H, T is a universal gate-set, but for convenience we include X, Y, Z and S as primitives.
- Quantum gates are reversible, and the Toffoli gate generalises the classical AND gate.