Quantum Computing
Lecture 4

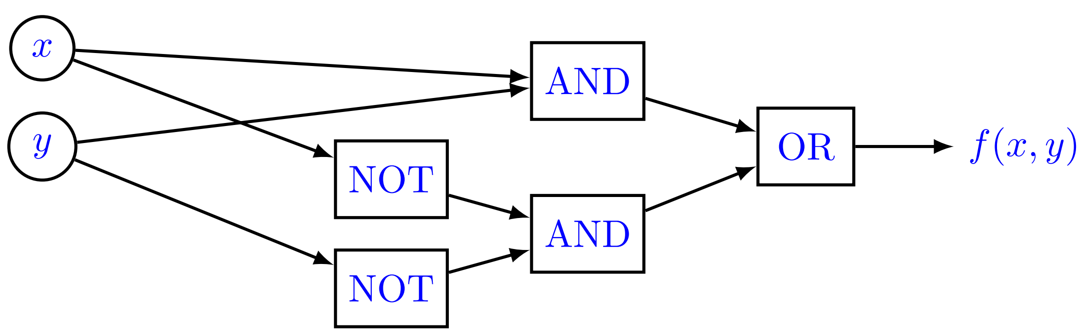
The Model of Quantum Computation

Maris Ozols

Boolean circuits

Logical gates model elementary computational steps in digital electronic circuits. E.g., $\mathrm{XOR}(x,y) = x \oplus y$, $\mathrm{NOT}(x) = x \oplus 1$, $\mathrm{AND}(x,y) = xy$.

Any Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be expressed in terms of these. E.g., equality $f(x,y) := (x = y)$ can be expressed as follows:



Universal sets of logical gates:

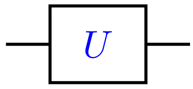
$$\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\} \qquad \{\mathrm{AND}, \mathrm{NOT}\} \qquad \{\mathrm{NAND}\}$$

Note: we take FANOUT or copying for granted!

Single-qubit gates

A 1-qubit gate is a 2×2 unitary. It can be written as follows:

$$U = \sum_{i,j \in \{0,1\}} U_{i,j} |i\rangle\langle j|$$



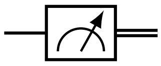
Example: The quantum analogue of logical NOT is the Pauli gate $X = |0\rangle\langle 1| + |1\rangle\langle 0| = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$:

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

Example: The Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ changes basis between the standard basis $\{|0\rangle, |1\rangle\}$ and the Hadamard basis $\{|+\rangle, |-\rangle\}$:

$$H|0\rangle = |+\rangle \qquad H|+\rangle = |0\rangle$$
$$H|1\rangle = |-\rangle \qquad H|-\rangle = |1\rangle$$

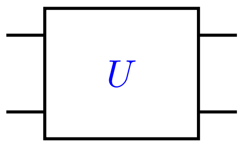
The standard basis measurement of a qubit is given by $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ and is not unitary. It has quantum input and classical output:



Multi-qubit quantum gates

A 2-qubit gate is a 4×4 unitary. It can be written as follows:

$$\begin{aligned} U &= \textstyle\sum_{i,j,k,l \in \{0,1\}} U_{ij,kl} |i,j\rangle\langle k,l| \\ &= \textstyle\sum_{i,j,k,l \in \{0,1\}} U_{ij,kl} |i\rangle\langle k| \otimes |j\rangle\langle l| \end{aligned}$$

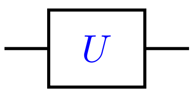


Example: $|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| + |11\rangle\langle 11|$ is the 2-qubit identity.

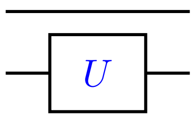
An n-qubit gate is a $2^n \times 2^n$ unitary. It has n input and n output qubits.

If a 1-qubit gate is applied locally on one of two qubits, we can write this as a 2-qubit unitary:

$U \otimes I$



$I \otimes U$

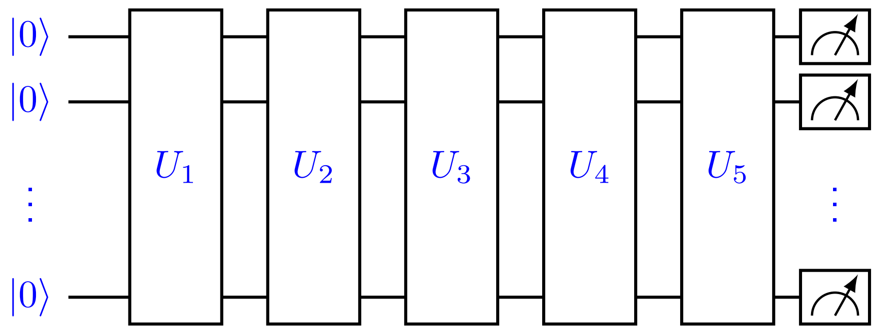


If a 2-qubit gate is applied locally on qubits 1 and 3, we can write this as a 3-qubit unitary:

$$\sum_{i,j,k,l \in \{0,1\}} U_{ij,kl} |i\rangle\langle k| \otimes I \otimes |j\rangle\langle l|$$

Quantum circuits

An n-qubit quantum circuit is a sequence of n-qubit unitary operations, followed by the measurement in the standard basis:



Typically the initial state is $|00\ldots0\rangle = |0\rangle^{\otimes n}$ and we measure all qubits in the standard basis at the end.

Fact: While one can imagine more general circuits with intermediate measurements, all measurements can always be deferred to the end and converted into independent standard basis measurements for each qubit.

Locality and uniformity

Computation is local if it consists of elementary operations that act only on a few bits or qubits at a time.

Example: Any Boolean formula can be expressed using logical gates with at most two input bits (e.g., AND and NOT).

We would like quantum circuits to be local as well:

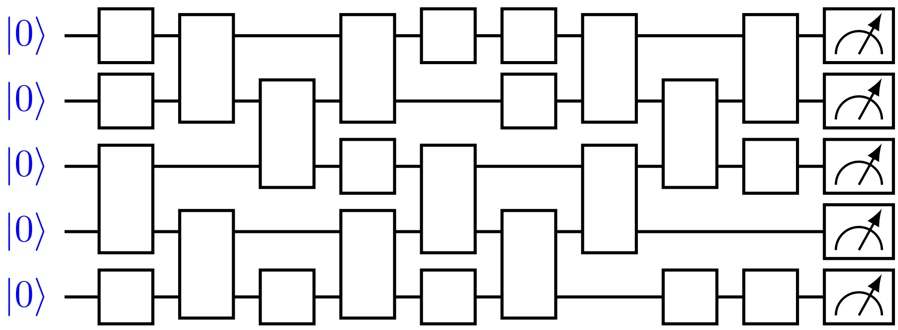
- it is hard to make more than two systems interact simultaneously
- we might have only a finite number of different types of interactions
- we cannot afford to store $2^n \times 2^n$ matrices

To solve a computational problem, we need a family of circuits: one for every input size. But how do we find these circuits in the first place?

A uniform family of circuits is one that can be produced by a deterministic Turing machine in polynomial time. This rules out circuits that might implement a look-up table for some difficult problem.

Quantum algorithm

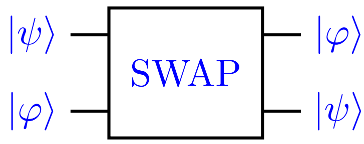
A quantum algorithm is an infinite family $C_1, C_2, \ldots$ of quantum circuits, where C_n acts on n qubits and consists of a finite sequence of 1-qubit and 2-qubit gates: $C_n = (U_1, U_2, \ldots, U_{L(n)})$ where $L(n)$ denotes the number of gates in the circuit. The map $n \to C_n$ must be efficiently computable (e.g., in polynomial time on a deterministic Turing machine).



Quantum SWAP gate

How can we exchange two qubits? We would like a two-qubit unitary gate such that for any $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^2$:

$$\mathrm{SWAP}(|\psi\rangle \otimes |\varphi\rangle) = |\varphi\rangle \otimes |\psi\rangle$$



In particular, this should work for the standard basis:

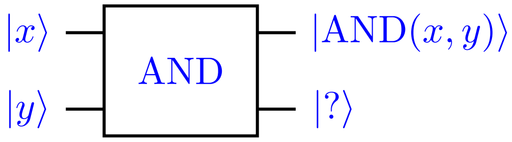
$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |10\rangle \\ |10\rangle &\mapsto |01\rangle \\ |11\rangle &\mapsto |11\rangle \end{aligned} \qquad \mathrm{SWAP}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \\ c \\ b \\ d \end{pmatrix}$$

The same matrix works for any states! If $|\psi\rangle = \binom{x}{y}$ and $|\varphi\rangle = \binom{z}{t}$ then

$$|\psi\rangle \otimes |\varphi\rangle = \begin{pmatrix} x\begin{pmatrix} z \\ t \end{pmatrix} \\ y\begin{pmatrix} z \\ t \end{pmatrix} \end{pmatrix} = \begin{pmatrix} xz \\ xt \\ yz \\ yt \end{pmatrix} \qquad |\varphi\rangle \otimes |\psi\rangle = \begin{pmatrix} z\begin{pmatrix} x \\ y \end{pmatrix} \\ t\begin{pmatrix} x \\ y \end{pmatrix} \end{pmatrix} = \begin{pmatrix} xz \\ yz \\ xt \\ yt \end{pmatrix}$$

Quantum AND gate?

Let's try to find a unitary that computes the logical AND of two bits. Given $x, y \in \{0,1\}$, it should output $\mathrm{AND}(x,y)$ on the first qubit:



What should it output on the second qubit? Some arbitrary states:

$$
\begin{aligned}
|00\rangle &\mapsto |0\rangle|\psi_1\rangle \\
|01\rangle &\mapsto |0\rangle|\psi_2\rangle \\
|10\rangle &\mapsto |0\rangle|\psi_3\rangle \\
|11\rangle &\mapsto |1\rangle|\psi_4\rangle
\end{aligned}
$$

A unitary gate must preserve orthogonality: since $|00\rangle$, $|01\rangle$, $|10\rangle$ are orthogonal, so must be the three output states $|\psi_1\rangle$, $|\psi_2\rangle$, $|\psi_3\rangle$, but it is not possible to have three mutually orthogonal states in $\mathbb{C}^2$!

Computing a Boolean function reversibly

If $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function, the map $|x\rangle \mapsto |f(x)\rangle$ is not reversible in general and hence not unitary.

One way of making the map reversible is by keeping the input:

$$|x\rangle \mapsto |x, f(x)\rangle$$

However, this is not unitary as the output dimension is larger than the input dimension. We can make the two dimensions match as follows:

$$|x, 0\rangle \mapsto |x, f(x)\rangle$$

However, this map is not fully defined. What is the image of $|x, 1\rangle$? What is the pre-image of $|x, y\rangle$ if $y \neq f(x)$? A unitary version of f is

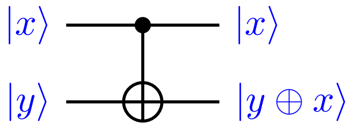
$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$$

for all $x \in \{0,1\}^n$ and $y \in \{0,1\}$. Note that $U_f^2 = I$ so U_f is self-inverse.

Controlled gates

The CNOT (controlled NOT) is a 2-qubit gate that flips the second qubit only when the first qubit is in state $|1\rangle$ (below $x, y \in \{0,1\}$):

$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |01\rangle \\ |10\rangle &\mapsto |11\rangle \\ |11\rangle &\mapsto |10\rangle \end{aligned} \qquad \mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

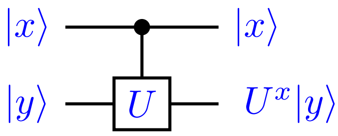


It is a quantum analogue of $f(x) = x$ where $x \in \{0,1\}$. Note that $\mathrm{CNOT}|x,0\rangle = |x,x\rangle$, so it copies x "in the standard basis".

Problem: Do we also have $\mathrm{CNOT}|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$ for any $|\psi\rangle \in \mathbb{C}^2$?

If U is a single-qubit unitary, then the controlled U gate is

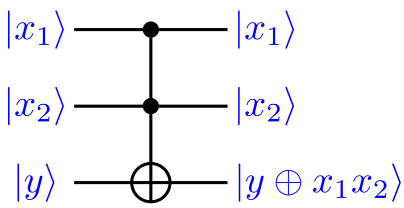
$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |01\rangle \\ |10\rangle &\mapsto |1\rangle \otimes U|0\rangle \\ |11\rangle &\mapsto |1\rangle \otimes U|1\rangle \end{aligned} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{U}} \\ 0 & 0 & & \end{pmatrix}$$



Another way of writing it down is as follows: $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$.

Toffoli gate

The Toffoli gate is a 3-qubit gate that flips the last qubit only when the first two qubits are 1:



It is a reversible implementation of $\mathrm{AND}(x_1, x_2) = x_1 x_2$. In fact,

$$y \oplus x_1 x_2 = \begin{cases} \mathrm{AND}(x_1, x_2) & \text{if } y = 0 \\ \mathrm{XOR}(y, x_1) & \text{if } x_2 = 1 \\ \mathrm{NOT}(y) & \text{if } x_1 = x_2 = 1 \\ x_1 & \text{if } y = 0 \text{ and } x_2 = 1 \end{cases}$$

Toffoli is universal for reversible classical computation as it can implement any Boolean function reversibly given enough copies of $|0\rangle$ and $|1\rangle$.

Getting rid of junk...

There is still one problem: we consume the 0's and 1's in the process and produce some junk data that contain values from intermediate steps:

$$|x, 0^n 1^m, 0\rangle \longmapsto |x, \mathsf{junk}(x), f(x)\rangle$$

The "uncomputing" trick:

$$\begin{aligned} |x, 0^n 1^m, 0, 0\rangle \quad &\xmapsto{U_f} \quad |x, \mathsf{junk}(x), f(x), 0\rangle \\ &\xmapsto{\text{CNOT}} \quad |x, \mathsf{junk}(x), f(x), f(x)\rangle \\ &\xmapsto{U_f^\dagger} \quad |x, 0^n 1^m, 0, f(x)\rangle \end{aligned}$$

where CNOT copies the classical bit $f(x)$ to the last register and $U_f^\dagger$ corresponds to running the Toffoli circuit U_f backwards.

Summary: Given enough copies of $|0\rangle$ and $|1\rangle$, Toffoli gates can reversibly compute any Boolean function while still returning back the original copies of $|0\rangle$ and $|1\rangle$ (except for one bit that contains $f(x)$).

Universal sets of quantum gates

Fact: While Toffoli gate cannot be implemented by reversible classical 2-bit gates, it can be implemented by 2-qubit unitary gates.

Fact: Any $2^n \times 2^n$ unitary operation on n qubits can be implemented by a sequence of 2-qubit operations.

Fact: Any unitary operation can be implemented exactly by a combination of CNOTs and single qubit operations.

Fact: Any unitary operation can be approximated to any required degree of accuracy using only gates from the set $\{\text{CNOT}, H, T\}$ where

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

This can serve as our finite set of gates for quantum computation.

Deutsch's problem (XOR)

Let $f : \{0,1\} \to \{0,1\}$ be one of the 4 possible Boolean functions:

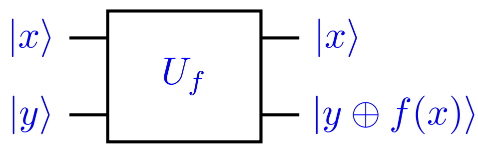
	f_{00}	f_{01}	f_{10}	f_{11}
$f(0)$	0	0	1	1
$f(1)$	0	1	0	1

f_{00} and f_{11} are constant
f_{01} and f_{10} are balanced

Problem: How many calls to f are required to determine whether f is constant or balanced (or equivalently, for computing $f(0) \oplus f(1)$)?

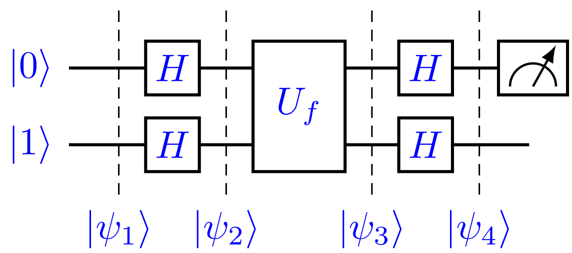
Classically this requires two calls to the function f.

One call suffices quantumly if we are given the quantum black box:



Deutsch's algorithm

Phase kick-back: If $x \in \{0,1\}$ then $X^x|-\rangle = (-1)^x|-\rangle$. Similarly, if $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$ then $U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle$.



Step-by-step analysis:

$$|\psi_1\rangle = |0\rangle|1\rangle$$

$$|\psi_2\rangle = |+\rangle|-\rangle = \tfrac{1}{\sqrt{2}} \textstyle\sum_{x\in\{0,1\}} |x\rangle|-\rangle$$

$$|\psi_3\rangle = \tfrac{1}{\sqrt{2}} \textstyle\sum_{x\in\{0,1\}} (-1)^{f(x)}|x\rangle|-\rangle$$

$$= (-1)^{f(0)} \underbrace{\tfrac{1}{\sqrt{2}}\big(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle\big)}_{|+\rangle \text{ or } |-\rangle \text{ depending on } f(0)\oplus f(1)} |-\rangle$$

$$|\psi_4\rangle = (-1)^{f(0)}|f(0) \oplus f(1)\rangle|1\rangle$$

Summary

- **Pauli X:** like logical NOT: $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$
- **Hadamard:** $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ flips between $\{|0\rangle, |1\rangle\}$ and $\{|+\rangle, |-\rangle\}$
- **Quantum circuit:** a sequence of unitary gates followed by the standard basis measurement
- **Locality:** all gates should act on at most 2 qubits
- **Uniformity:** there should be an efficient way of producing the circuit
- **Quantum SWAP:** $\mathrm{SWAP}|\psi\rangle|\varphi\rangle = |\varphi\rangle|\psi\rangle$
- **Quantum Boolean function:** $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$
- **Controlled NOT:** $\mathrm{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$
- **Toffoli gate:** controlled CNOT, universal for reversible computation
- **Junk:** can be erased reversibly by uncomputing
- **Universal quantum gate set:** $\{\mathrm{CNOT}, H, T\}$
- **Phase kick-back:** $X^x|-\rangle = (-1)^x|-\rangle$
- **Deutsch's algorithm:** $U_f|+\rangle|-\rangle = (-1)^{f(0)}|(-1)^{f(0)\oplus f(1)}\rangle|-\rangle$, two queries classically, only one quantumly