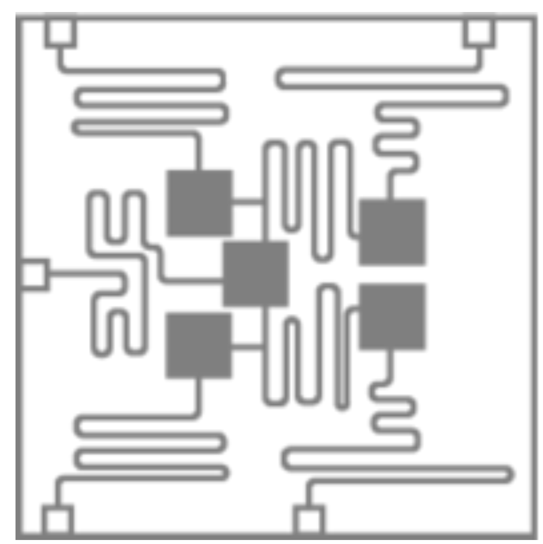
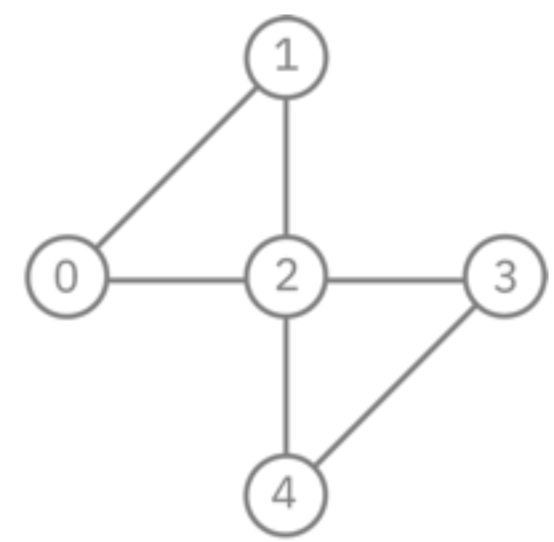
Quantum Information Processing

Budapest University of Technology and Economics
2019 Spring

Ákos Budai, András Pályi, Zoltán Zimborás

Lecture 1
Basics: quantum information, python, qiskit





IBM Quantum Experience

today: “Hello World” on a q-computer

Quantum Computing - what is it?

Quantum computing

From Wikipedia, the free encyclopedia

Quantum computing is computing using quantum-mechanical phenomena, such as superposition and entanglement.[1] A **quantum computer** is a device that performs quantum computing. Such a computer is different from binary digital electronic computers based on transistors. Whereas common digital computing requires that the data be encoded into binary digits (bits), each of which is always in one of two definite states (0 or 1), quantum computation uses quantum bits or qubits, which can be in superpositions of states. A quantum Turing machine is a theoretical model of such a computer, and is also known as the universal quantum computer. The field of quantum computing was initiated by the work of Paul Benioff[2] and Yuri Manin in 1980,[3] Richard Feynman in 1982,[4] and David Deutsch in 1985.[5]

Quantum Computing - why should anyone care?

QC could be useful

- algorithms solving computational problems can be slow or fast
- for example, prime factorization is a problem for which only slow classical algorithms are known
- prime factorization is important in information technology & security
- there is a fast quantum algorithm for prime factorization (Shor)

People are interested in QC

- many experimental research groups are trying to build and improve quantum computer prototypes
- private funding in quantum information technology increased a lot in the past few years (IBM, Google, Intel, Microsoft; Rigetti, Q-Ctrl, etc)

Quantum computers do exist

- prototype quantum computers that are available for anyone do exist, e.g., IBM Quantum Experience (small, noisy, not useful yet)

Schedule of this course

Szerda		
February 6.	lecture 01 (today)	Basics
February 13.	lecture 02	Algorithms 1
February 20.	lecture 03	Decoherence 1
February 27.	lecture 04	Decoherence 2
March 6.	lecture 05	Decoherence 3
March 13.	lecture 06	Algorithms 2
March 20. Tavaszi szünet		
March 27.	lecture 07	Algorithms 3
April 3.	lecture 08	Algorithms 4
April 10.	lecture 09	Algorithms 5
April 17.	lecture 10	Quantum Error Correction 1
April 24.	lecture 11	Quantum Error Correction 2
May 1. Munka ünnepe		
May 8.	lecture 12	Bell inequalities, teleportation
May 15.	lecture 13	

Course website on fizipedia.bme.hu

https fizipedia.bme.hu/index.php/Kvantuminformatika

Quantum Information Processing [szerkesztés]

Course Information, 2019 Spring [szerkesztés]

- **Lecturers:** András Pályi, Zoltán Zimborás
- **Responsible lecturer:** András Pályi
- **Language:** English
- **Location:** F3212
- **Time:** Wednesdays, 12:15-13:45

Details [szerkesztés]

- One goal is to provide an introduction to basic concepts of quantum information theory and computing. Another goal is to provide hands-on experience in programming an actual quantum computer. That is, the basic concepts, gadgets, algorithms, etc., should be implemented and run by the students themselves during the course and as homework. We will use the quantum computers of the IBM Quantum Experience project, which are available via the cloud for anyone.
- Lectures will combine conventional, frontal presentation, and programming exercises. Therefore, the location is a computer lab. Of course, students are welcome to use there own laptop computers.
- The main resource used for the course is the online documentations of (1) the quantum computers available through the IBM Quantum Experience project [1], and (2) the Qiskit quantum computing framework [2].

List of topics [szerkesztés]

- Basics: quantum information, python, and qiskit (Lecture 1, AP)
- Bernstein-Vazirani algorithm (Lecture 2, ZZ)
- Density matrix. State tomography. Process Tomography. Relaxation. Dephasing. Decoherence. (Lectures 3-5, AP).
- Quantum algorithms: Deutsch, Grover, Shor, quantum simulation (Lectures 6-9, ZZ)
- Classical, hybrid, and quantum error correction using the repetition code. (Lectures 10-11, AP)
- Bell inequalities. Quantum teleportation. (Lecture 12, ZZ)

Classical bits, gates, circuits

truth tables

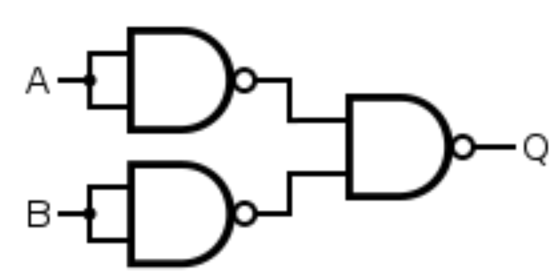
INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- the value of a c-bit is 0 or 1
- *operations, gates:* a c-logical gate maps n c-bits to m c-bits; e.g., NOT, AND, OR, XOR.
- single-bit gate: $n = m = 1$
- there is only one non-trivial single-bit gate: NOT
- two-bit gate: $n = 2$, $m = 1$, e.g., AND, OR, XOR
- c-gates are not necessarily reversible: e.g., any $n > m$ gate is irreversible
- *c-circuit*: an arrangement of "wires" and gates
- *universal gate set*: a set of gates that allows to construct circuits for any algorithm
- exercise: construct a c-circuit that adds two single-bit numbers using only the NAND gate

OR can be built from NANDs:



Quantum bits

1. *quantum bit, qubit, q-bit, qbit*: two-level quantum system
2. state of a qubit: $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$
3. α_0, α_1 are called *amplitudes*; they are complex numbers
4. $|0\rangle$ and $|1\rangle$ are the *qubit basis states*
5. normalization condition: $|\alpha_0|^2 + |\alpha_1|^2 = 1$
6. alternative notation (*vector notation* or *spinor notation*):

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ \alpha_0 |0\rangle + \alpha_1 |1\rangle \equiv \alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

7. realizations: electron spin, nuclear spins (e.g., H-1, C-13), superconducting circuits, etc.

Dynamics of a qubit

1. time-dependent Schrodinger equation: $\dot{\psi}(t) = -\frac{i}{\hbar} H(t)\psi(t)$.
2. for a qubit, $H(t)$ is a 2x2 Hermitian matrix
3. Hamiltonian can be expressed with Pauli matrices

$$H(t) = \sum_{j=0}^{3} c_j(t)\sigma_j$$

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

4. dynamics for a time-independent Hamiltonian: $\psi(t) = \exp\left(-\frac{i}{\hbar} H t\right)\psi(0) \equiv U(t)\psi(0)$
5. $U(t)$ is a unitary matrix, called the *propagator*
6. dynamics for a time-dependent Hamiltonian is also unitary: $\psi(t) = \mathcal{T}\exp\left(-\frac{i}{\hbar}\int_0^t dt' H(t')\right)\psi(0) \equiv U(t)\psi(0)$

Measurement (`readout') of a qubit

1. $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$
2. the probability of measuring 0 is $P_0 = |\alpha_0|^2$
3. the probability of measuring 1 is $P_1 = |\alpha_1|^2 = 1 - P_0$
4. if the outcome of the measurement is 0, then the state changes to $|0\rangle$
5. if the outcome of the measurement is 1, then the state changes to $|1\rangle$

Geometrical representation of a qubit: the Bloch sphere

1. we can parametrize the qubit state with three angles, γ, θ, ϕ:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = e^{i\gamma}(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

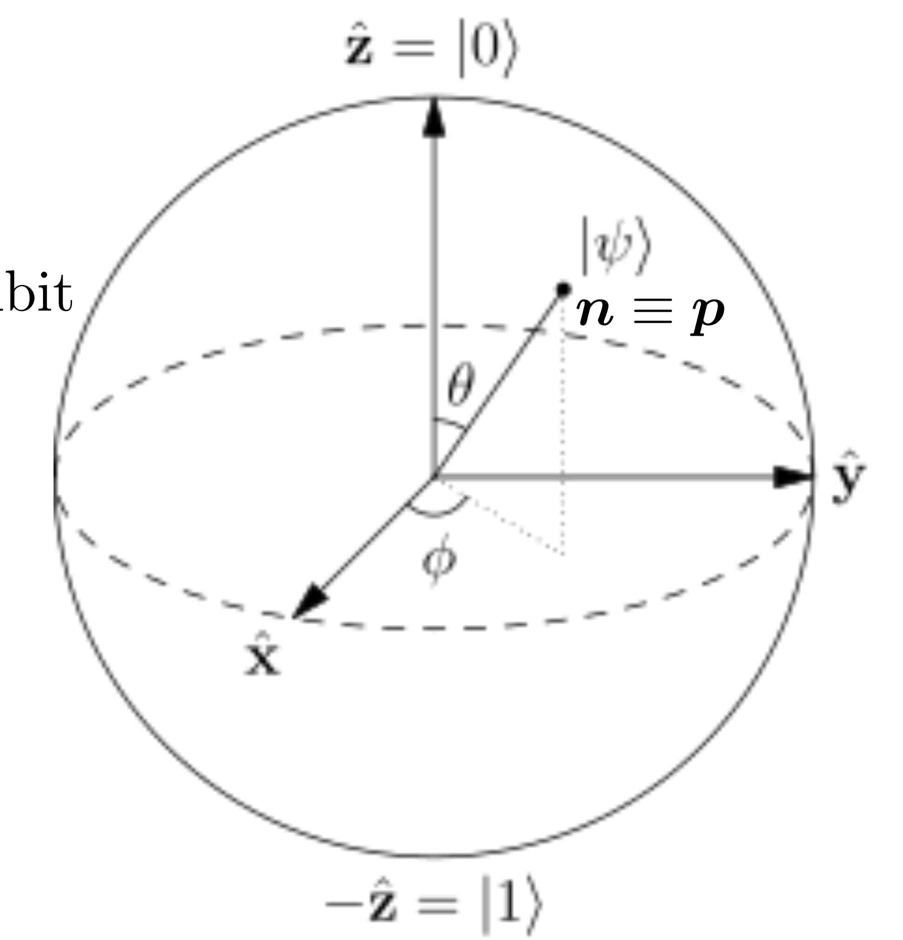
2. angle γ has no physical significance
3. the qubit state can be mapped to the surface of a unit sphere (*Bloch sphere*):

$$|\psi\rangle \mapsto (\theta,\phi) \mapsto \boldsymbol{n} = \left(\, \sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta \,\right)$$

4. another mapping, seemingly different, but actually identical to $\boldsymbol{n}$:

$$\boldsymbol{p} = \langle\psi|\boldsymbol{\sigma}|\psi\rangle\,,$$

   where $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$.
5. $\boldsymbol{n} \equiv \boldsymbol{p}$ is called the *Bloch vector* or the *polarization vector* of the qubit



More qubits

1. states of two qubits: $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$
2. normalization condition: $\sum_{x\in\{0,1\}^2} |\alpha_x|^2 = 1$
3. a single-qubit state can be represented on the Bloch sphere; does not work for multiple-qubit states
4. measurement of one qubit: e.g., of the first one: $P_0 = |\alpha_{00}|^2 + |\alpha_{01}|^2$, and the post-measurement state after measuring 0 is

$$|\psi_{\rm pm}\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{P_0}}$$

5. example for a two-qubit product state:

$$|\psi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

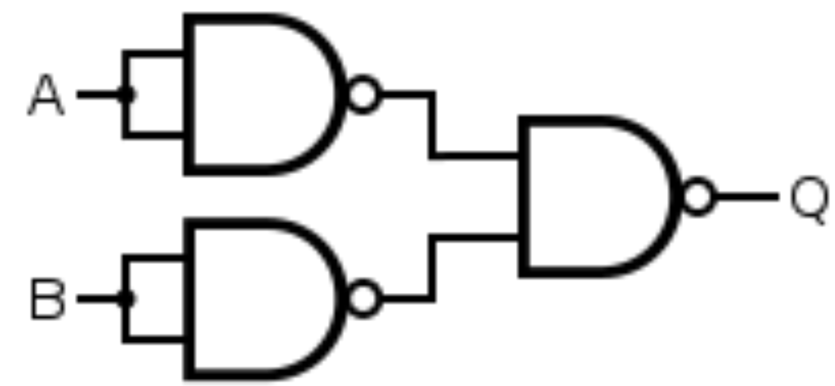
6. example for a two-qubit entangled state:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

7. the state of n qubits is described by 2^n amplitudes

1-qubit quantum gates

c-circuit



1. q-circuit: an arrangement of "wires" and quantum gates
2. q-gates: unitary operations on a few qubits (reversible, unlike c-gates)
3. 1-qubit gate example: q-NOT (usually called the X *gate*):

$$|\psi_1\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \;\mapsto\; |\psi_2\rangle = \alpha\,|1\rangle + \beta\,|0\rangle$$

   matrix representation of this gate: $X \equiv \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

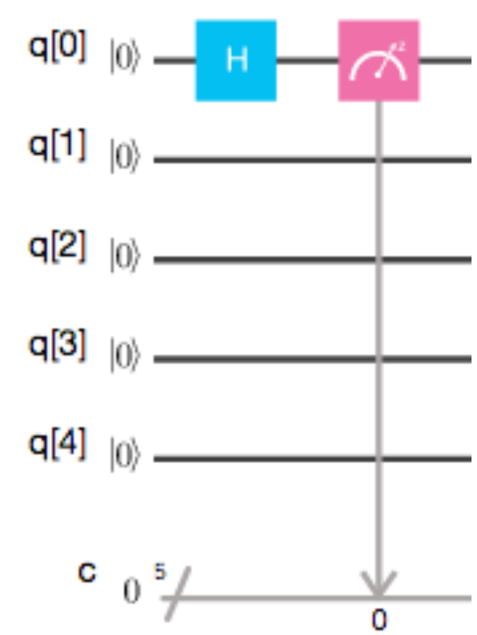
4. further 1-qubit gate examples:

$$\text{Z gate: } Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

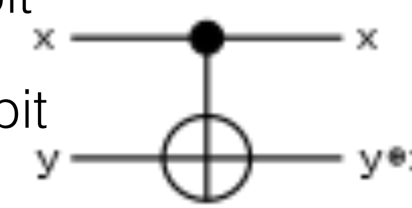
$$\text{Hadamard gate: } H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

5. each 1-qubit gate generates a bijective map of the Bloch sphere to itself
6. exercise: determine the transformations generated by 1-qubit gates listed above

q-circuit



2-qubit quantum gates

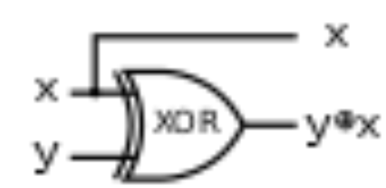


input		output	
x	y	x	y+x
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

1. 2-qubit gate example: *controlled-NOT* or *CNOT*
   with the basis-state ordering $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, it is represented by

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

   it could be represented by a 'classical' truth table



input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

2. 1-qubit gates together with CNOT form a unversal q-gate set

IBM Quantum Experience

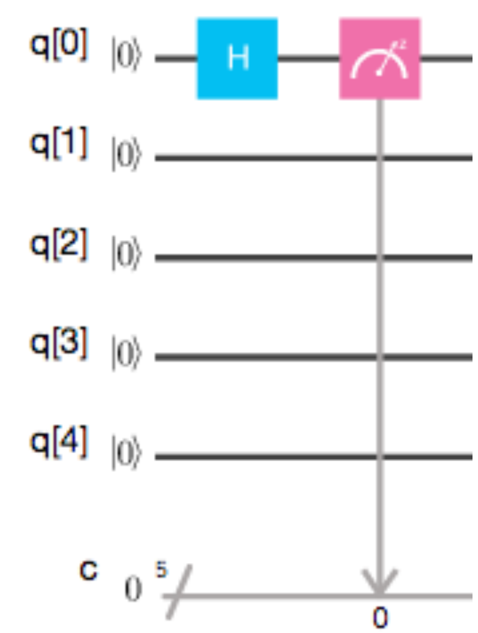
1. *Quantum Hello World.*
   Do a Hadamard gate and measure the qubit afterwards. Use 10 shots.

   (a) Define the circuit in the composer. Run the simulator. How many times do you measure the state 1?

   (b) Define the circuit in qiskit in a jupyter notebook. Run the circuit in your notebook on your local simulator. How many times do you measure the state 1?

   (c) Homework: Run the circuit on a quantum computer, using the composer.

   (d) Homework: Run the circuit on a quantum computer, using qiskit in a jupyter notebook.



Step 1: launch Anaconda Navigator (bottom left corner)

Step 2: launch jupyter notebook

Step 3: open new notebook (python 3)

Your notebooks will be temporarily stored in
`C:\Users\diak\Documents\`

For permanent storage, make a folder
`D:\NeptunCode\`
and copy your notebooks there
or copy your notebooks to your pendrive.

2. *Draw a circuit.*
   Visualize the simple circuit above in qiskit in a jupyter notebook.

3. *Draw a histogram.*
   Plot the histogram of the data obtained above, using qiskit in the jupyter notebook. (counts of 0, counts of 1)

4. *Bell state.*
   Your goal is to prepare the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ by acting on the first qubit with a Hadamard gate, and acting with a CNOT on the two qubits, and measuring the two qubits. Draw the circuit on a piece of paper. Use 1024 shots in what follows.

   (a) Compose the circuit in the composer. Run the circuit from the composer on the simulator.

   (b) Define the circuit with qiskit in a jupyter notebook. Run the circuit.

   (c) Plot the histogram of the measured data.

5. *Plot a function.*
   Plot the sine function in the $[0, 4\pi]$ interval in a jupyter notebook.

6. *Exponential decay.*
   Generate a noisy exponential decay curve: sample the function $f(t) = e^{-t}$ between the interval $[0, 10]$ in steps 0.1, add a normally distributed random contribution to each data point with a standard deviation of 0.1, and fit an exponential function, $g(t) = Ae^{-t/T_1} + c$, to the noisy data. What are the three values of the parameters A, T_1, c obtaind from the fit? How do they relate to the parameters ($A = 1$, $T_1 = 1$, $c = 0$) of the original $f(t)$ function? Plot the noiseless data set, the noisy data set, and the fitted curve, in the same graph.

7. *Rabi oscillations on the Bloch sphere.*
   The Rabi formula states that the time evolution of the polarization vector of a resonantly driven qubit reads

$$\boldsymbol{p}(t) = \begin{pmatrix} \sin\theta(t)\cos(\phi(t)) \\ \sin\theta(t)\sin(\phi(t)) \\ \cos\theta(t) \end{pmatrix}, \tag{1}$$

   where $\phi(t) = \omega_L t$ and $\theta(t) = \Omega t$. Plot this time evolution as a 3D parametric plot, with $\Omega = 1$, $\omega_L = 10$, in the time window $t \in [0, \pi]$.