Christian Boehme

Quantum Computing (for the curious)

Learning Objectives

- Formulate small Quantum algorithms as Quantum circuits
- Implement small Quantum circuits with Qiskit
- Understand and implement Deutsch’s algorithm

Table of contents

What is Quantum Computing?

"Quantum computation ... will be the first technology that allows ... distributing components of a complex task among vast numbers of parallel universes, and then sharing the results."

— David Deutsch, The fabric of reality: the science of parallel universes (1997)

Deutsch, *The Fabric of Reality*

Again: What is Quantum Computing?

- A QC utilizes quantum mechanics to solve complex problems
- QC *may* be superior to classical computing in
  - Simulating Quantum systems (highly probable)
  - Combinatorial optimization problems (probable)
  - Cryptography / prime factorization (probable)
  - Machine learning and other fields (unknown)

State of Quantum Computing

- Noisy, intermediate-size Quantum era (**NISQ**)
- Quantum Computers with hundreds of Qubits
- High error rate (noise) due to external influences and imperfect control
- Full error tolerance in NISQ era unachievable due to overhead
- Real world Quantum advantage requires thousands or millions of Qubits
- Great time to do basic research
  - ... if interested in algorithms and/or Quantum mechanics
  - Availability of accessible frameworks and learning material

The Qubit

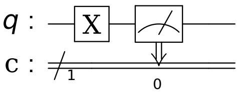
- The smallest unit of information in QC is the **Qubit**
- Qubits are represented as two-dimensional vectors $\vec{q}$
- Two states of the Qubit form its **Computational Basis**:
  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Computations with these states are similar to classic computing

Gates: Operations on Qubits

- Operations on Qubits are called **gates** (cf. logic gates in classical computing)
- Gates are linear transformations of the state vector of a Qubit, i.e. matrices
- E.g., see the NOT or **X** gate in action:
- $X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \times 1 + 1 \times 0 \\ 1 \times 1 + 0 \times 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$
- $X|1\rangle = |0\rangle$ (see exercise)

Quantum Circuits and Measurement

- **Quantum Circuits** are one way to represent quantum computations



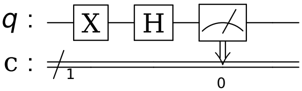
- Qubits are measured in the computational basis
- The result is either $|0\rangle$ *or* $|1\rangle$ and is stored in a classical bit

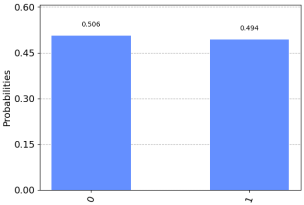
Superposition and the Hadamard gate

- Say "hello" to the **Hadamard** or **H** gate:
- $H\,|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\left[\begin{pmatrix}1\\0\end{pmatrix}+\begin{pmatrix}0\\1\end{pmatrix}\right] = \frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle$
- $H\,|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle$ (see exercise)
- After the H gate the Qubit’s state is a linear combination of basis states
- This is called **superposition**
- The H gate is it’s own inverse, applying it to $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ yields $|0\rangle$
- Applying H to $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$ yields $|1\rangle$

Superposition measurement

- Let’s measure our superposition:





- In a superposition $a\,|0\rangle + b\,|1\rangle$, $a, b \in \mathbb{C}$
  - the probability to measure Qubit state $|0\rangle$ equals $|a|^2$
  - the probability to measure Qubit state $|1\rangle$ equals $|b|^2$
- This also means $|a|^2 + |b|^2 = 1$
- (This is Schrödinger’s cat: We are officially Quantum now!)

Multiple Qubit states

- Qubit states are combined by tensor product, i.e. for two Qubits:

$$|a\rangle \otimes |b\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{pmatrix}$$

$$= a_0 b_0 |00\rangle + a_0 b_1 |01\rangle + a_1 b_0 |10\rangle + a_1 b_1 |11\rangle$$

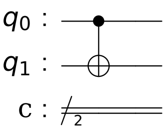
- Measuring 2 Qubits results in probabilities

$$|a_0 b_0|^2 + |a_0 b_1|^2 + |a_1 b_0|^2 + |a_1 b_1|^2 = 1$$

- For example: $|0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle$
- Can be extended to n Qubits: $|q_1\rangle \otimes |q_2\rangle \otimes ... \otimes |q_n\rangle$
- Grows exponentially, only 30+ Qubits can be simulated classically

The CNOT gate

- The **CNOT** or **XOR** gate is a 2 Qubit gate:



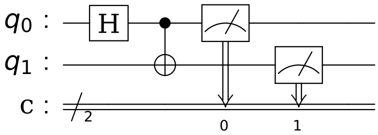
- (Note Qiskit's Qubit order: $|q_1 q_0\rangle$)
- Or, as a matrix:

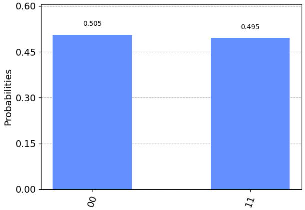
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix} = \begin{pmatrix} a_{00} \\ a_{11} \\ a_{10} \\ a_{01} \end{pmatrix}$$

- 2 Qubit gates have two outputs, as Quantum gates need to be **reversible**

Entanglement

- Consider the following circuit:

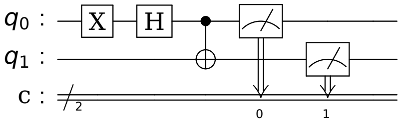




- Each Qubit is equally likely to be measured as $|0\rangle$ and as $|1\rangle$
- However, both Qubits will always be in the **same** state after measurement
- This is called **entanglement**
- Einstein's spooky action at a distance: More Quantum weirdness!
- What happens: $CNOT(I \otimes H)\,|00\rangle = CNOT[\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)] = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Introduction to Qiskit: Entanglement

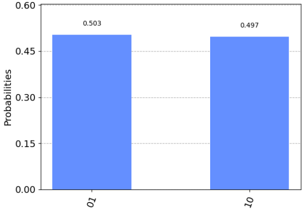
- Login at `https://jupyter-hpc.gwdg.de/hub/login`
- Upload the two Jupyter notebooks (`.ipynb`) from the `exercises` folder
- Run `pcpc_qc_exc_1-4.ipynb`
  - Is the probability distribution as you expected?
  - Can you write the corresponding superposition in terms of basis states $|00\rangle$, ...
- Entangle the two Qubits, but put the control Qubit in $|1\rangle$ state before applying the H gate
  - I.e., implement:



  - Think about the result: Is it what you expected? Why or why not?

Modifying entanglement

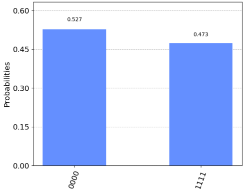
- There exists an entangled state where 2 Qubits are always in **different** states after measurement:



- Implement the Quantum circuit producing that state
- Hint: You need one additional gate

Optional: Extending entanglement

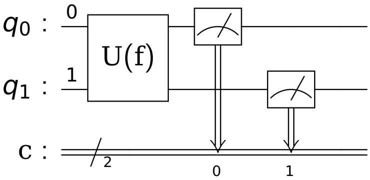
- Consider this maximally entangled state of 4 Qubits:



- Implement the Quantum circuit producing that state
- For this you need to extend the Quantum circuit to 4 Qubits:
  `circuit = QuantumCircuit(4)`
- Hint 1: Start with entangling 2 Qubits
- Hint 2: The first three Qubits are now in state $\frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|011\rangle$
- Hint 3: Switch $|011\rangle$ to $|111\rangle$ but **not** $|000\rangle$ to $|100\rangle$. What gate does this?

Deutsch's problem

- Consider an unknown function of a one bit input x
- The output $f(x)$ could either be **constant** 0 or 1 or depend on x (**balanced**)
- Two tests required classically to determine balanced vs. constant
- Quantum circuit for the problem:



- Remember: Quantum gates must be reversible

Quantum solution part 1: Phase kickback

- Our $U(f)$ gate transforms $|x\rangle\,|y\rangle$ to $|x\rangle\,|y \oplus f(x)\rangle$ ($\oplus$: plus, then modulo 2)
- Let's "cheat": Instead of using $|0\rangle$ for $|y\rangle$ we use $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- Then, if we apply our gate $U(f)$:
  $U(f)\,|x\rangle\,\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
  $= |x\rangle\,\frac{1}{\sqrt{2}}(|f(x)\rangle - |1 \oplus f(x)\rangle)$
  $= (-1)^{f(x)}\,|x\rangle\,\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- Now we have encoded information in the sign (or **phase**) of the input Qubit!
- This is called **phase kickback** and is used in many Quantum algorithms

Quantum solution part 2: Input superposition

- We will now use the function information encoded in the phase
- We initialize $|x\rangle$ as superposition of both possible values: $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- We can ignore the output Qubit and get, after applying $U(f)$:
  $(-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2}}((-1)^{f(|0\rangle)} |0\rangle + (-1)^{f(|1\rangle)} |1\rangle)$
- That means constant functions result in $|q_0\rangle = \pm\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- Balanced functions result in $|q_0\rangle = \pm\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- (Note that multiplying a Qubit state globally by -1 does not change it)
- Use the H gate and measure to get $|0\rangle$ for constant and $|1\rangle$ for balanced
- We have solved Deutsch's problem in one try!
- Deutsch-Josza extends this to input length n: One try vs. worst case $2^{n-1} + 1$

Implementing Deutsch's algorithm

- Deutsch's algorithm:
  - Prepare the input (0) Qubit in $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ state
  - Prepare the output (1) Qubit in $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ state
  - Apply the oracle
  - Apply the Hadamard gate to Qubit 0 and 1 each
  - Measure: If $|10\rangle$, the function is constant, else it is balanced
- Please implement this in `debase.py`
- Implement pre- and postprocessing as described
- Implement at least one constant and one balanced oracle
- Constant oracles are constant $|0\rangle$ and constant $|1\rangle$
- Balanced oracles are identity and negate

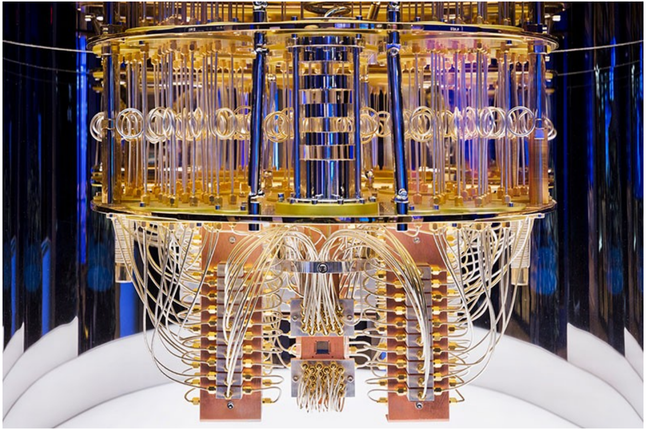
Optional: Extend Deutsch to Deutsch-Josza

- We will not cover the math here (but it's very similar to Deutsch)
- Tasks:
  - Use 3 instead of 1 input Qubits (4 Qubits in total)
  - Prepare the input and output Qubit states as before
  - You can reuse your constant oracle
  - Implement at least one balanced oracle
  - This should output $|0\rangle$ for half of the possible inputs, $|1\rangle$ for the other half
  - Extension (harder): Can you implement a random set of balanced oracles?
  - Implement postprocessing and measuring as before
  - You should measure $|1000\rangle$ for constant (and something else for balanced)

Further resources

- Recap linear algebra:
  Essence of linear algebra on YouTube
- Deutsch's algorithm explained using a state machine:
  Quantum Computing for Computer Scientists on YouTube
- Introduction for self study:
  Quantum computing for the very curious
- Everything on Qiskit, lots of tutorials: `https://qiskit.org/`
- Good textbooks:
  - Hidary, *Quantum Computing: An Applied Approach*
  - Homeister, *Quantum Computing verstehen : Grundlagen - Anwendungen - Perspektiven*
- Brief overview of the field (link to arxiv.org):
  Quantum Computing in the NISQ era and beyond, John Preskill

Questions?



Credit: IBM

References

Deutsch, David. *The Fabric of Reality*. Allan Lane, 1997.

Hidary, J.D. *Quantum Computing: An Applied Approach*. Springer International Publishing, 2019. ISBN: 9783030239213. URL: https://books.google.de/books?id=nkKoxQEACAAJ.

Homeister, Matthias. *Quantum Computing verstehen : Grundlagen - Anwendungen - Perspektiven*. 4. Auflage. 2017. ISBN: 978-3-658-10455-9 (Online). DOI: 10.1007/978-3-658-10455-9. URL: http://www.springerlink.com/content/978-3-658-10455-9.