Lourens van Niekerk

Quantum Computing

A brief introduction for the curious

Learning Objectives

- The state of quantum computing
- What are qubits and quantum circuits?
- The cool phrases: superposition and entanglement
- How does quantum computing physically even work?
- The Deutsch-Josza 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."

- Deutsch, *The Fabric of Reality* (1997)

Okay, still... what is Quantum Computing?

- Quantum computers utilizes quantum mechanics to solve complex problems
- From the ground up, it uses different physical representations to operate
  - ✗Bits $\longrightarrow$ ✓Qubits
  - ✗Transistors $\longrightarrow$ ✓e.g. Ions, photons
  - ✗Voltage $\longrightarrow$ ✓LASERS!
  - ✗Linear power $\longrightarrow$ ✓Exponential power
- 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)

A superconducting quantum computer

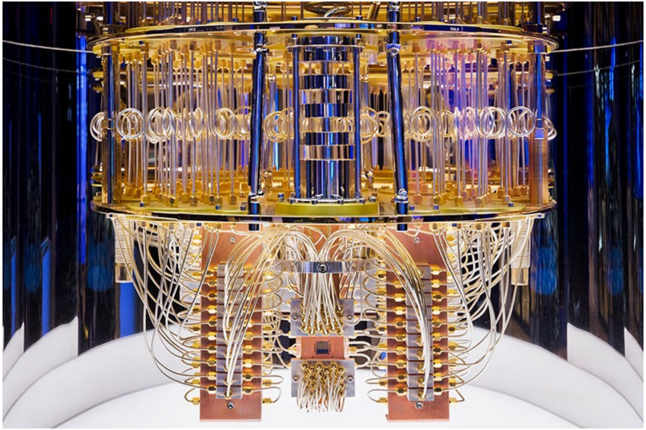


Figure: Source: IBM

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
- Quantum advantage attempts and disputes
  - Google in 2019 Kalai, Rinott, and Shoham, “Google’s Quantum Supremacy Claim: Data, Documentation, and Discussion”
  - IBM in 2023 Patra et al., “Efficient tensor network simulation of IBM’s largest quantum processors”
  - probably 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
- Promising fields are Quantum Chemistry, Combinatorial Optimization, Machine Learning

The qubit

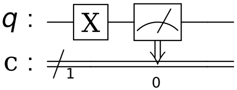
- The smallest unit of information in QC is the **qubit**
- Qubits are represented as two-dimensional $\mathbb{C}$-vectors $\vec{q}$ which are unitary
- Below two states of a qubit form its **Computational Basis**:
  $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- $|\psi\rangle$ known as **ket** notation
- Computations with these states are similar to classical computing (bits 0 and 1)
- All quantum states are a linear combination (**superposition**) of $|0\rangle$ and $|1\rangle$
- $|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha |0\rangle + \beta |1\rangle$ where $\alpha, \beta \in \mathbb{C}$
- (Unitary implies $|\alpha|^2 + |\beta|^2 = 1$)

Gates: Operations on qubits

- **Quantum gates** do operations on qubits (cf. logic gates in classical computing)
- Gates apply linear transformations on the statevector of qubits, i.e. matrices
- (Since statevectors need to remain unitary, gates have determinant 1, always!)
- Important 1-qubit gates are the **NOT gate (X)** and the **Hadamard gate (H)**
- $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow X\left|0\right\rangle = \left|1\right\rangle$ $\qquad H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \Rightarrow H\left|0\right\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$
- Other notable gates: Pauli $\{X, Y, Z\}$, Phase $P(\phi)$, and Universal $U(\theta, \phi, \lambda)$
- The inverse of any gate is shown with $\dagger$ e.g. $GG^\dagger = I = G^\dagger G$ for some gate G
- The Hadamard and Pauli gates are their own inverses, for example

Quantum Circuits and Measurement

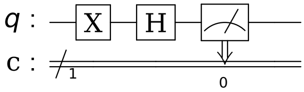
- **Quantum Circuits** can represent quantum computations

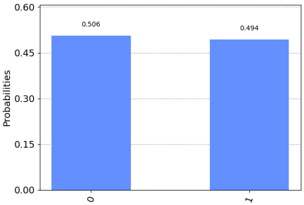


- Contain the quantum & classical registers, gates and **measurements**
- Measuring qubits collapses them into computational bases i.e. $|0\rangle' s$ and $|1\rangle' s$
- These bases can then be turned into bits for classical computing
- Which one they collapse into depends on a probability distribution
- If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then prob. collapse to $|0\rangle : \mathbb{P}_{|\psi\rangle}(|0\rangle) = |\alpha|^2$
- (For greater detail on these probabilities, look at the *Comprehensive-Notes* pdf)

Superposition measurement

- E.g: $|\psi\rangle = HX|0\rangle = H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{-1}{\sqrt{2}}|1\rangle \Rightarrow$ Chances of $|0\rangle$ or $|1\rangle$ are 50\50

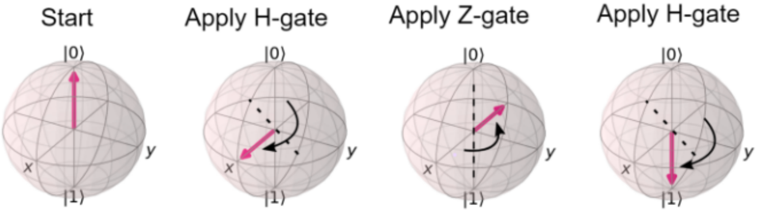


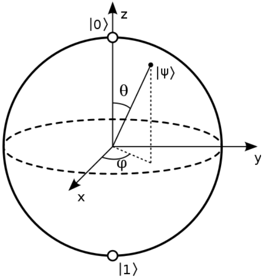


- Even without X, we just made a coin toss
- (This is Schrödinger's cat: We are officially Quantum now!)

Bloch Sphere

- **Bloch Sphere**: Useful representation of qubit statevectors (for our brains)
- Can represent any statevector with 2 angles: $0 \leq \theta \leq \pi$ and $0 \leq \varphi \leq 2\pi$





Multiple qubit states

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

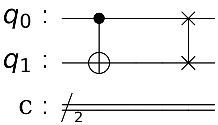
$$|a\rangle \otimes |b\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

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

- Prob. distribution of measuring 2 qubits:
  $|a_0 b_0|^2 + |a_0 b_1|^2 + |a_1 b_0|^2 + |a_1 b_1|^2 = 1$
- For example: $|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |01\rangle$
- Can be extended to n qubits: $|q_1\rangle \otimes |q_2\rangle \otimes \ldots \otimes |q_n\rangle$
- Grows exponentially, only $\sim$30-40 qubits can be simulated classically

MORE gates: 2-qubit gates

- We got the **CNOT**(*controlled-not*) gate
- $\oplus$ is the **target** and $\bullet$ is the **control**
- Also got the **SWAP** gate (X—X)

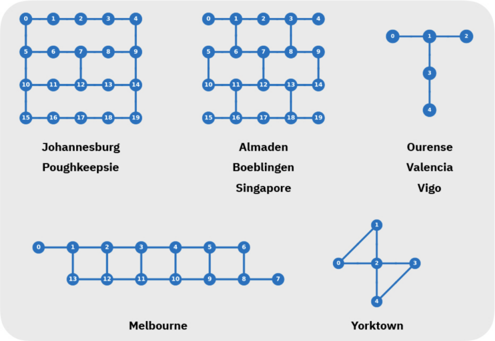


- $CNOT = CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & X \end{bmatrix}$ $\qquad SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- Note: Above form of CX only if applied to $|q_0 q_1\rangle$, depends on control and target
- ANOTHER note: *Qiskit's* qubit ordering of a circuit top-down: $|q_1 q_0\rangle$

SWAPs

- In QC, qubits aren't always connected; SWAPs let them be
- SWAPs are expensive $$$; want optimal connectivity (some QC have all-2-all)

IBM's 10 Quantum Device Lineup



Qiskit model and OpenQASM

- Qiskit contributors, *Qiskit: An Open-source Framework for Quantum Computing* is the quantum toolkit developed by IBM
- Can be installed as package to Python
- Uses OpenQASM Cross et al., “OpenQASM 3: A broader and deeper quantum assembly language” as its Intermediate Represenation to communicate with the QPUs

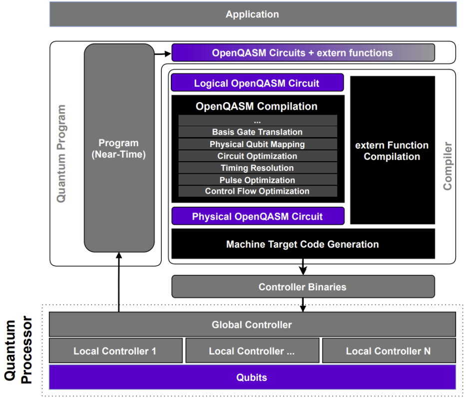


Figure: Compilation Model. Source: OpenQASM paper

Gate structure

- All n-qubit gates can be decomposed into 1- & 2-qubit gates (so stick to those)
- Thus 'n-qubit gates' in a circuit would refer to a subset of gates (for perspective)
- 2-qubit gates generally have the control\target form (except SWAPs)
- $\begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & G \end{bmatrix}$ with 1-qubit gate G
- Don't let names like control and target fool you; the control qubit can also be changed (this is called **phase kickback**)

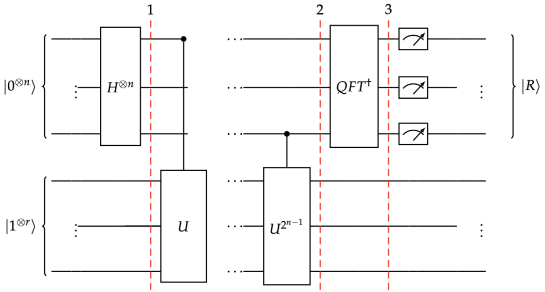
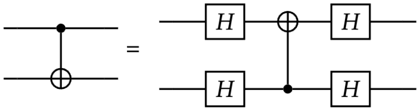


Figure: Shor's algorithm, for example

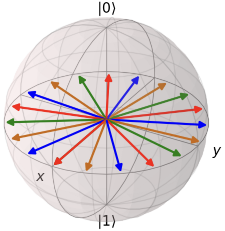
Phase kickback

- Important tool for algorithms, namely the quantum phase estimation
- Key attribute setting quantum apart from classical
- The beauty of many quantum algorithms lies in saving the solution in the phase of qubits via controlled-gates
- Represents a reverse of roles:



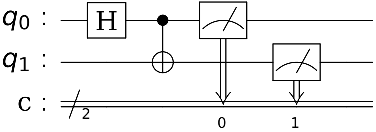
Phases

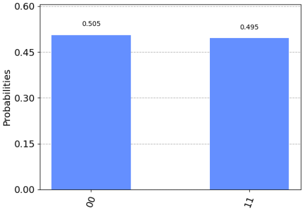
- You might wonder: 'if $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ with $\alpha, \beta \in \mathbb{C}$, what about $\begin{bmatrix}-1\\ 0\end{bmatrix}$ or $\begin{bmatrix}i\\ 0\end{bmatrix}$ or $\begin{bmatrix}\frac{1+i}{\sqrt{2}}\\ 0\end{bmatrix}$?'
- Basically... look at the *Comprehensive Notes* (we don't care much about)
- Multiplying any $\mathbb{C}$-unit (a **global phase**) by a statevector, does not change it for our purposes
  - The Bloch sphere still looks the same
  - First entry in the statevector can be $\in \mathbb{R}$
- Some statevectors have same measurement prob. distribution
- They differ by a **relative phase** (comp. effect!)
- Purpose of phase gate $P(\phi) = \begin{bmatrix}1 & 0\\ 0 & e^{i\phi}\end{bmatrix}$



Entanglement

- Consider the following circuit:





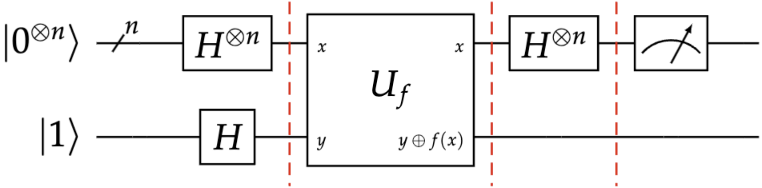
- Each qubit is equally likely to be measured as $|0\rangle$ and as $|1\rangle$
- However, both qubits will always be in the same state after measurement
- This is called **entanglement**
- Einstein's spooky action at a distance: More Quantum weirdness!
- What happens: $CNOT(I \otimes H)\,|00\rangle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

Deutsch-Josza algorithm overview

- First proposed by David Deutsch and Richard Josza in 1992 David Deutsch, “Rapid solution of problems by quantum computation”
- One of the first quantum algorithms that show quantum can trump classical
- Exponentially faster than any deterministic classical algorithm
- JUST LOOK AT THAT SMILE! SO PROUD

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 generalised problem:



- Remember: Quantum gates must be reversible

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

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-Josza extended algorithm

- Let's look again at Deutsch's algorithm:
  - Prepare (input) qubit-0 in $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ state
  - Prepare (output) qubit-1 in $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ state
  - Apply the oracle
  - Apply the Hadamard gate to qubit-0
  - Measure: If $|0\rangle$, the function is constant, else $|1\rangle$ if it is balanced
  - Oracles: constant implies 0 or 1. Balanced implies identity or negate
- Your job is to increase this to 3 or more input qubits (your job, not mine)
- The math of this general version can be found in the Comprehensive Notes
- Now balanced can be many more possibilities. Suppose 3 input qubits and 1 output qubit. Example balanced oracle:
  - $f(000) = f(101) = f(111) = f(100) = 0$
  - $f(110) = f(011) = f(010) = f(001) = 1$
  - You should measure $|000\rangle$ for constant, and anything else for balanced

What does GWDG offer?

- Our QC team provides HPC-QC software solutions, research and consulting
- SCC, Emmy and Grete have QC containers to run simulations on
  - Qiskit, Qulacs, Cirq, Qsim, Qibo, QuTip...
- Bachelors/Masters theses are available!
- We are engaged in publishing new research regarding NISQ software
- We provide courses and workshops (like this one!)
- We do not have our own quantum computer
- Our webpage is `https://gwdg.de/en/community-pages/qc-intro/` for more info

Further reading

- 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:
  - Quantum Computing: An Applied Approach by Jack D. Hidary
  - Quantum Computing verstehen von Matthias Homeister
- Brief overview of the field (link to arxiv.org):
  Quantum Computing in the NISQ era and beyond, John Preskill

References

Cross, Andrew W. et al. "OpenQASM 3: A broader and deeper quantum assembly language". In: *ACM Transactions on Quantum Computing, Volume 3, Issue 3, 2022* 3.3 (Apr. 30, 2021), pp. 1–50. ISSN: 2643-6817. DOI: 10.1145/3505636. arXiv: 2104.14722 [quant-ph].

David Deutsch, Richard Josza. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), pp. 553–558. ISSN: 2053-9177. DOI: 10.1098/rspa.1992.0167.

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

Kalai, Gil, Yosef Rinott, and Tomer Shoham. "Google's Quantum Supremacy Claim: Data, Documentation, and Discussion". In: (Oct. 2022). DOI: 10.48550/ARXIV.2210.12753. arXiv: 2210.12753 [quant-ph].

Patra, Siddhartha et al. "Efficient tensor network simulation of IBM's largest quantum processors". In: (Sept. 2023). DOI: 10.48550/ARXIV.2309.15642. arXiv: 2309.15642 [quant-ph].

Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

Exercises

Have fun with the exercises :)