Liquid Studio
HELSINKI

QUANTUM COMPUTING

1.29.2019

HASSAN NASERI, TIM LEONHARDT

accenturetechnology

CONTECTS

Repository: https://goo.gl/tsDH7V

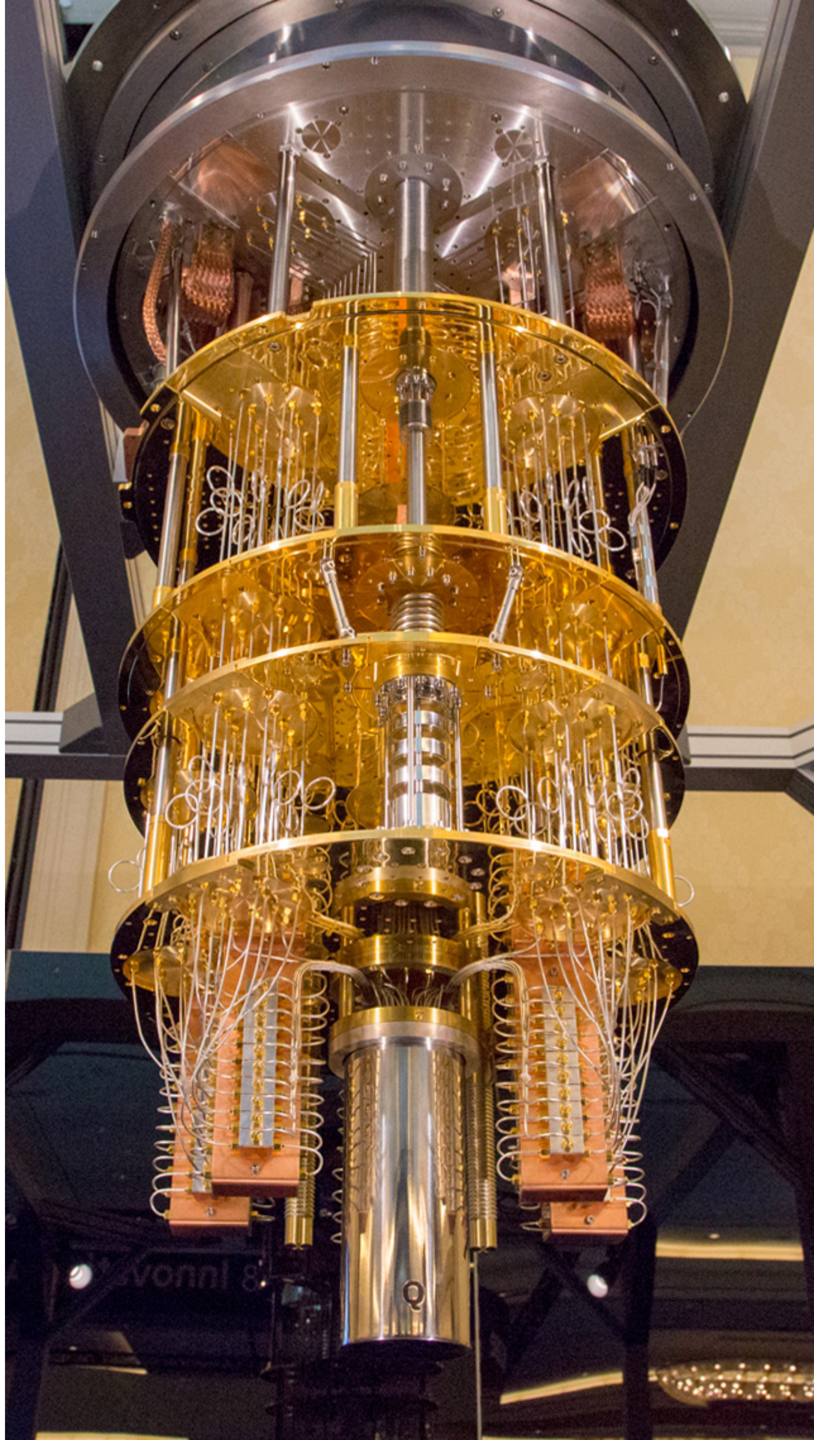
WHAT IS QUANTUM COMPUTING?

Quantum computer is not a faster computer!

In a quantum computer:

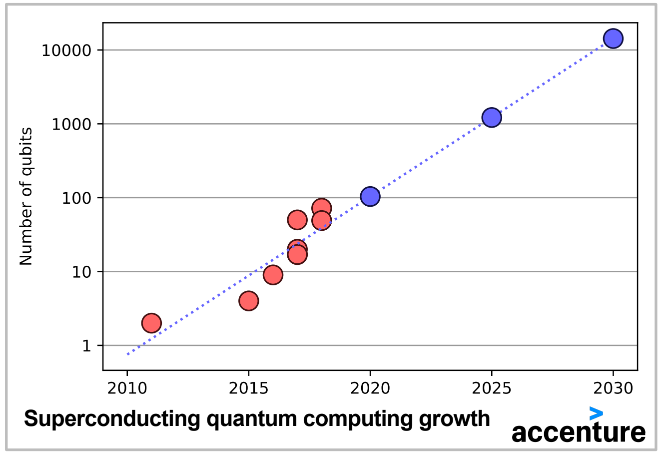
1. The information is stored in a quantum system.
2. The computation is governed by quantum-mechanical phenomena.

It is fundamentally different from digital computers and .

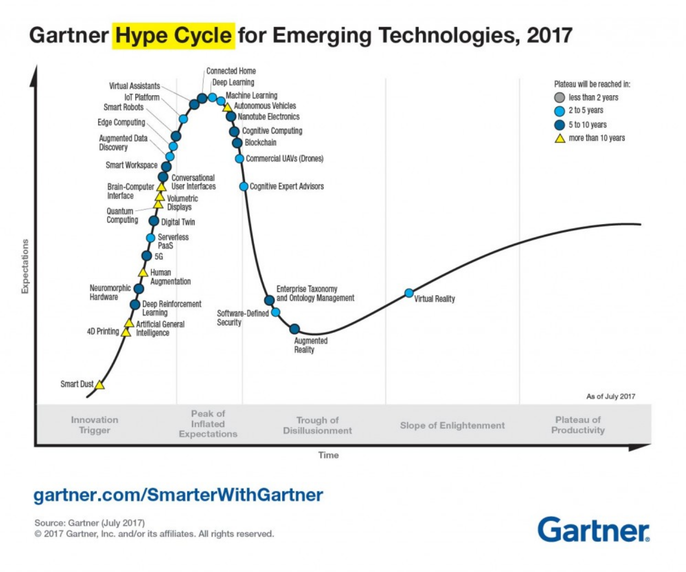


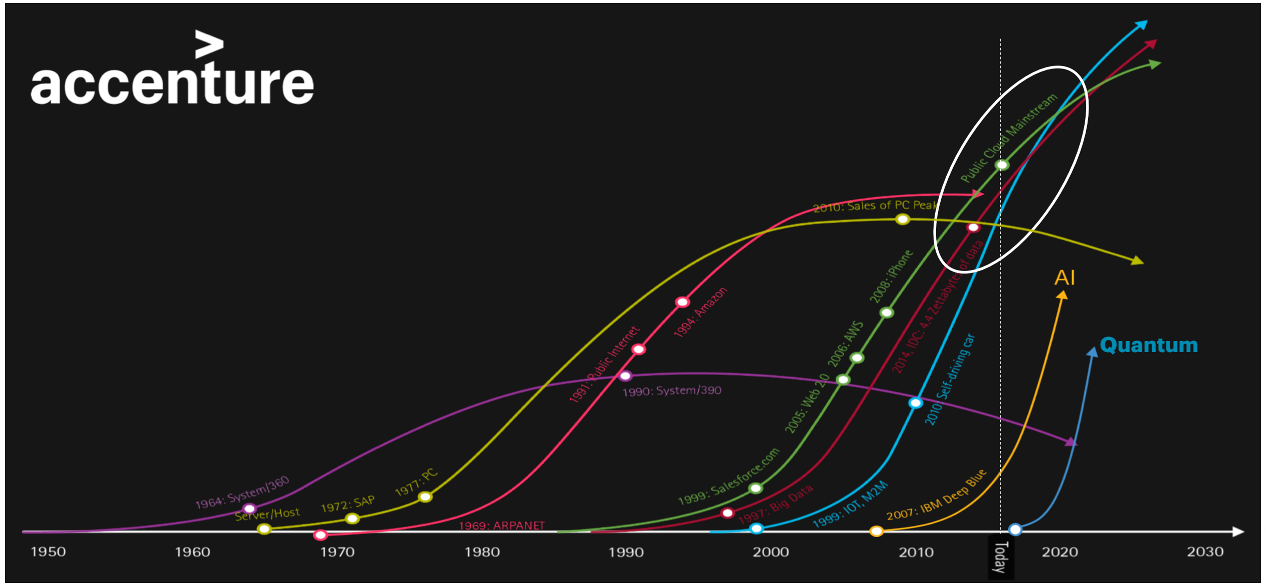
QUANTUM COMPUTING
TIMELINE

[1959]	Richard Feynman: “There's Plenty of Room at the Bottom”
[1973]	Charles H. Bennett (IBM): introduce reversible logic
[1994]	Peter Shor (Bell Labs): an important algorithm to cryptography
[2000]	Alamos National Laboratory: first working 7-qubit NMR
[2007]	**D-Wave** : 8-qubit quantum annealing computer
[2016]	**IBM**: public launch of IBM Q Experience and a 5 qubit universal quantum computer.
[2017 – 2018]	**IBM**: unveils a 50-qubit quantum computer. **Microsoft**: releases Quantum SDK and a 40-qubit simulator. **Google**: announces a 72-qubit quantum chip **Intel**: announces a 49-qubit superconducting test chip.



Exponential growth of superconducting quantum computing technology





FUTURE IS QUANTUM!

The promise of outperforming every classical computer to an extraordinary degree for certain tasks.

Change the world by solving problems that are intractable with today's technology.

Very likely to disrupt every industry in near future.

Market Situation:

- Public funds (2018): **US ($1.3B), EU (€1.1B), UK (£240M), China (>$10B)**, etc.
- Private sector: **IBM, Google, Intel, Microsoft**, and many startups.
- Big consultants, banks and industry: **Deloitte, KPMG, JPMorgan Chase, DAIMLER, Barclays**, **Accenture,** etc.

TWO MAIN TECHNOLOGIES

Adiabatic Quantum Computer

- Also known as **quantum annealing**
- **Approximate answers** to optimization problems.
- Cannot run standard quantum algorithms.
- Much larger machines available.

Universal Quantum Computer

- Can solve **any classical or quantum algorithm** (E.g. IBM, Google).
- Quantum **Gate Model** / Quantum Circuit.
- Provides quantum **speedup for certain algorithms** developed.
- Still very limited hardware.

D-Wave The Quantum Computing Company™

1QBit

IBM intel Microsoft Google

IT IS ALL ABOUT COMPLEXITY

Computationally complex problems require more space and time.

The difference in complexity can make a problem intractable.

There are many intractable problems (usually approximated!)

Example:

- A UPS truck has to deliver **400 packets in Helsinki**. The TSP has complexity $O(2^{400})$

A quantum computer is not generally faster or parallel.

Quantum speedup:

- There are problems which are classically hard and quantumly easy!
- **Quantum speedup requires the existence of a corresponding quantum algorithm.**

An optimal traveling salesperson tour through Germany's 15 largest cities. It is the shortest among 43,589,145,600 possible tours visiting each city once.

CONTECTS

THE FUTURE OF COMPUTING IS NOT DIGITAL!

Quantum computing is not electronic.

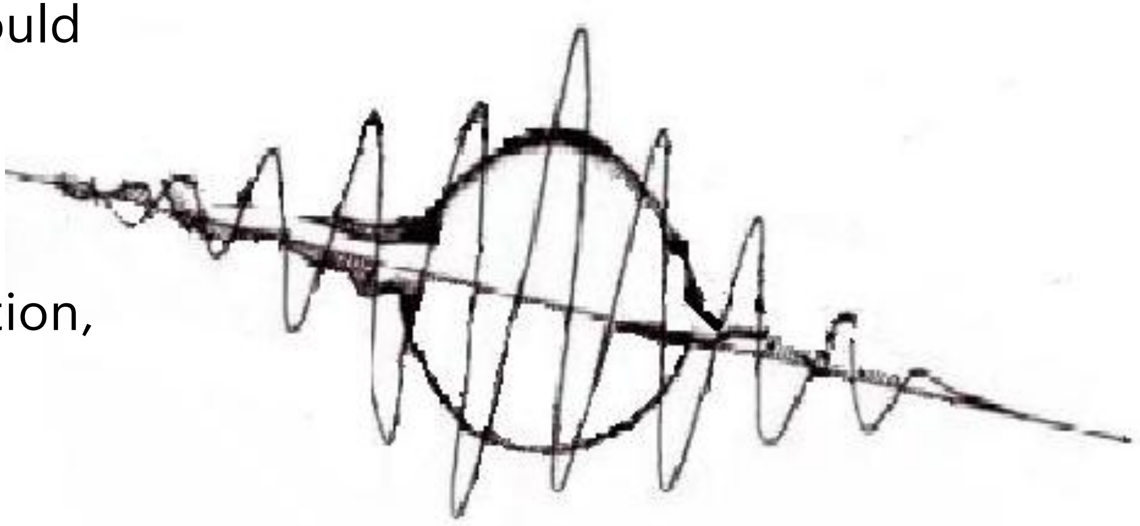
- It is not based on our current technology of electronics, semiconductor, ...
- There are few technologies under development.
- The winning technology is not clear yet, but it should be one.

Quantum computing is not digital.

- It can be considered a type of analogue computation, i.e., not discretized.

It is probabilistic.

It is quite noisy nowadays.



DO NOT WORRY ABOUT QUANTUM PHYSICS

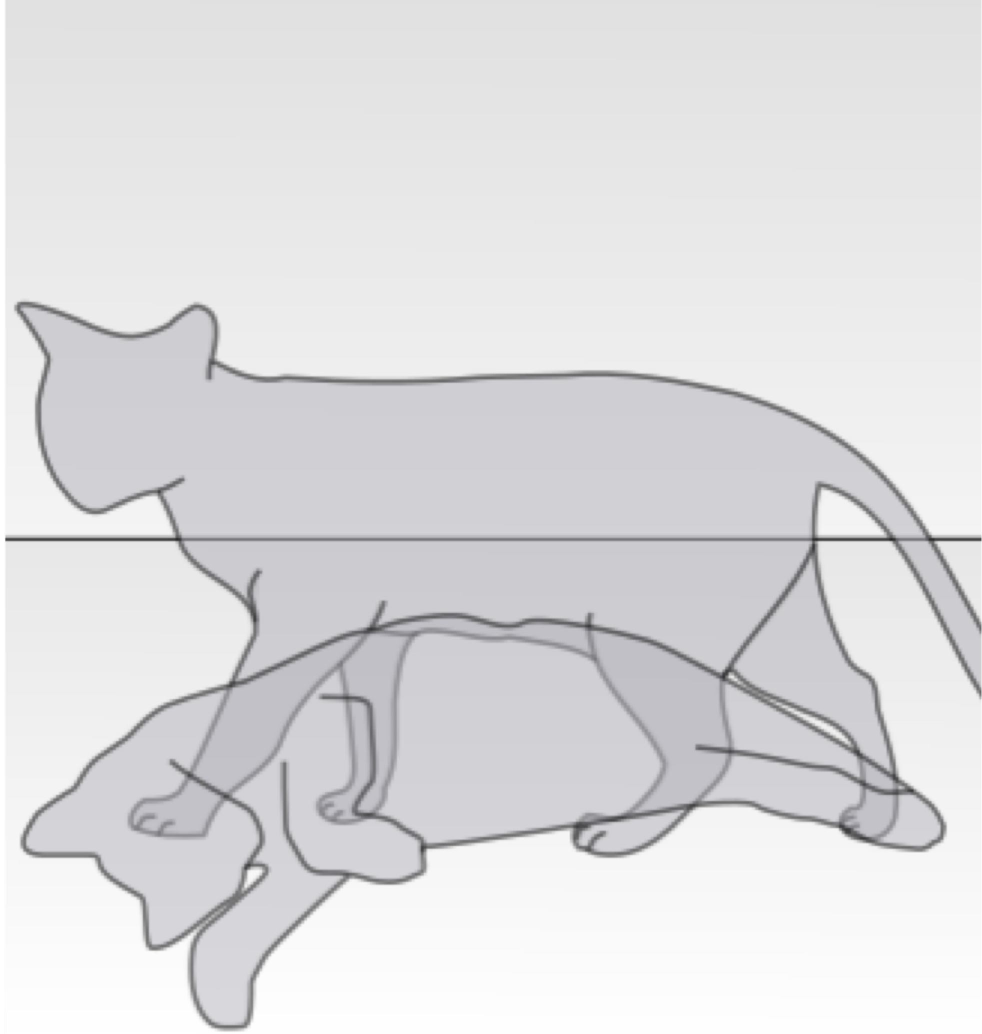
Superposition, Collapse, Entanglement, Teleportation

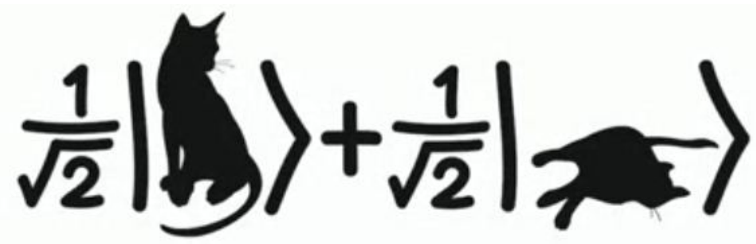
The physics can be difficult and subject to different interpretations.

Mathematics are simple and do not require interpretation.

Two important laws:

- Quantum operations are reversible.
- Measurement is irreversible

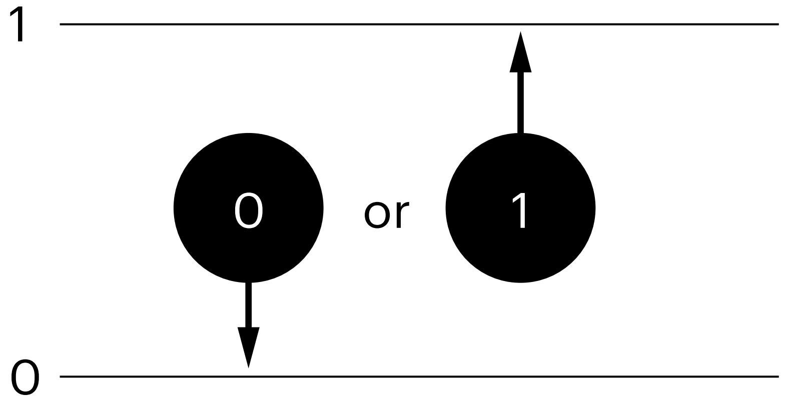




SOME TECHNICAL DETAILS!
WHAT IS A QUBIT?

A SUPERPOSITION OF TWO QUANTUM STATES.

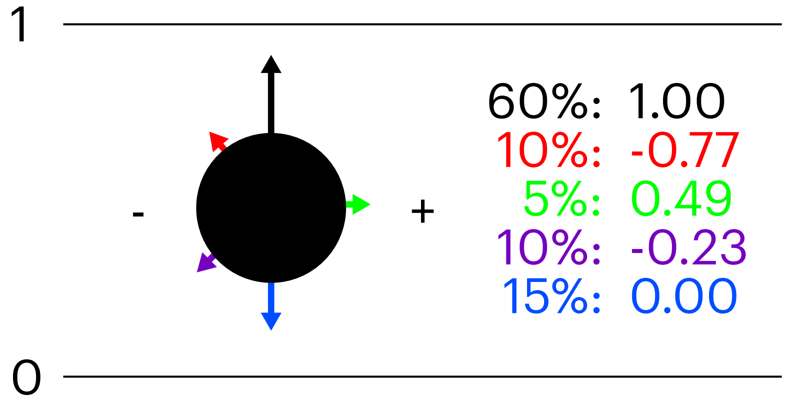
Classical



Discrete number of possible states: 0 or 1.

Deterministic: repeated computations on the same input will lead to thesame output.

Quantum

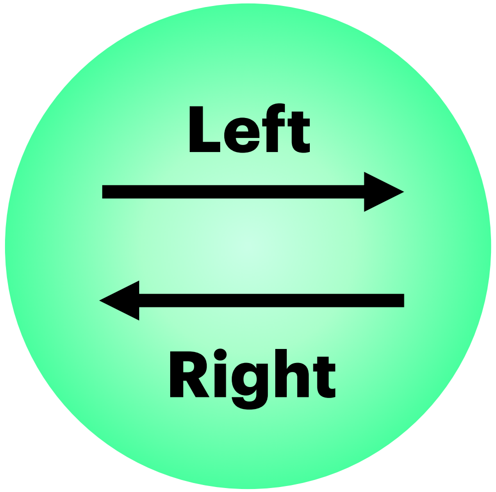


Infinite (continuous) number of possible states.

Probabilistic: measurements on superposed states yield probabilistic answers (our confidence in these answers builds up through repeated computations) then reduced to 0 or 1.

MATHEMATICAL REPRESENTATION

A Qubit is a Linear combination of two or more states.



$$q_0 = a|L\rangle + b|R\rangle$$

$$a|0\rangle + b|1\rangle \rightarrow \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\mathrm{Prob}(q_0 = |L\rangle) = a^2$$

$$\mathrm{Prob}(q_0 = |R\rangle) = b^2$$

$$a^2 + b^2 = 1$$

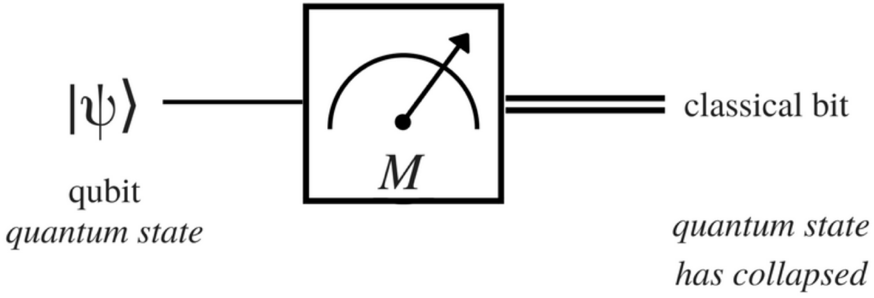
MEASUREMENT

- **The state of a qubit is not accessible from outside.**

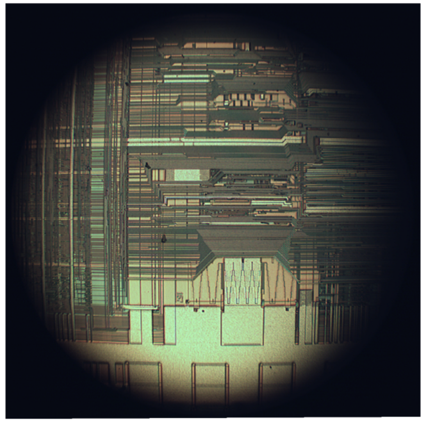
- **Measurement is done in a basis.**
- **Standard basis $|0\rangle, |1\rangle$.**

- **The state of a qubit collapses to one of the basis states.**

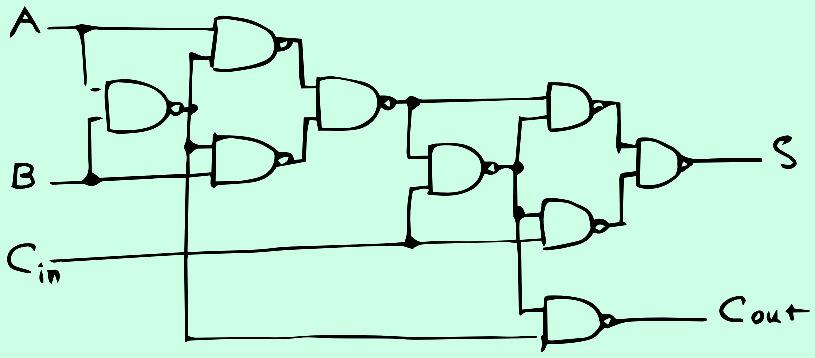
- **Measurement is irreversible, hence not a quantum gate.**



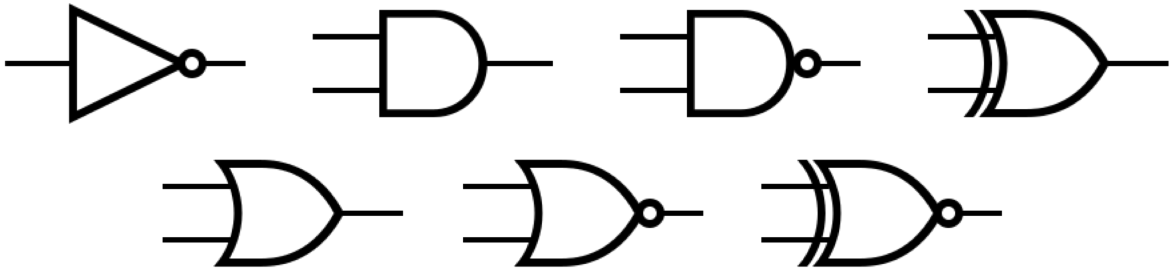
CLASSICAL COMPUTING: MODERN DIGITAL COMPUTERS



- **Electronic**: using electrons to process information
- **Digital**: information units are bits (0/1)
- Silicon-based transistor logic (MOSFET)
- Basic gates: AND, OR, NAND, …
- **Universal gates**: NAND, NOR
- Most system use a single universal



Full adder made of NAND gates



QUANTUM GATE MODEL
UNIVERSAL QUANTUM COMPUTER

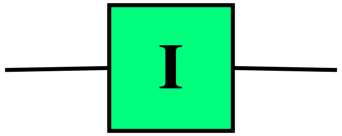
1. **A universal quantum computer made from simple gates.**
2. **Most common gates: 1-qubit and 2-qubit gates**
3. **Quantum logic gates are reversible.**
4. **Quantum logic gates are represented by unitary matrices.**

5. **Due to this inherent randomness (and also noise), the process is repeated and measured many times to find the probabilities of each state.**

IDENTITY AND HADAMARD GATES

- **Identity**
  - Does not change the state

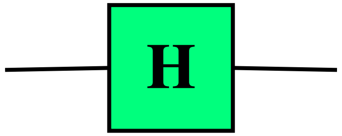
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$I \times \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

- **Hadamard Gate**

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



$$H \times \begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} a+b \\ a-b \end{bmatrix}$$

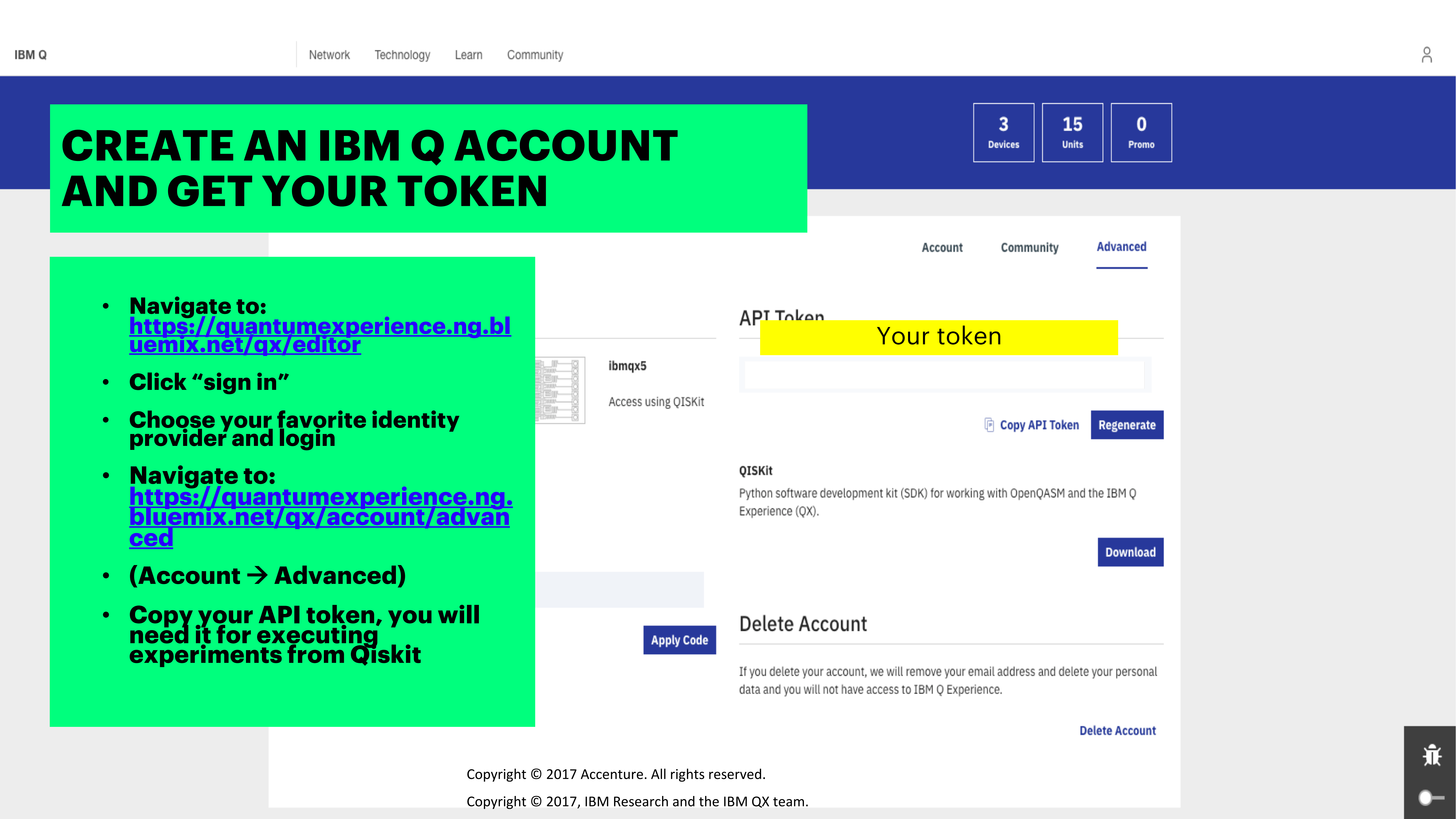
CONTECTS

DIFFERENT PROGRAMMING OPTIONS FOR IBM Q SYSTEM

- **Graphical interface (IBM Q – Composer)**
- **OpenQASM**
  - Open Quantum Assembly Language
- **Qiskit (Python SDK)**
- **Qiskit-Aqua (Python library)**

CREATE AN IBM Q ACCOUNT AND GET YOUR TOKEN

- **Navigate to: https://quantumexperience.ng.bluemix.net/qx/editor**
- **Click “sign in”**
- **Choose your favorite identity provider and login**
- **Navigate to: https://quantumexperience.ng.bluemix.net/qx/account/advanced**
- **(Account → Advanced)**
- **Copy your API token, you will need it for executing experiments from Qiskit**



QUANTUM PROGRAMMING – QISKIT

- **QISKIT (Quantum Information Science Kit) is an open-source quantum computing framework.**

- **It is a Python SDK.**

- **Backends: IBM Q computers, local and remote simulators**

- **The starting point for writing code is the QuantumCircuit.**

- Tutorial: https://qiskit.org/documentation/quickstart.html
- SDK reference: https://qiskit.org/documentation/_autodoc/qiskit.html

QISKIT SETUP
JUPYTER NOTEBOOK

Navigate to

- Repo: **https://goo.gl/tsDH7V**
- MyBinder: https://mybinder.org/v2/gh/HassanNaseri/quantum-computing-handson/master

Open file: **hello_qiskit.ipynb**

```
QX_TOKEN =
"728624125ddfb3acc6485060d3c3bbd63e8c1373e65b3e8d1f2
9fd91417c75f7efa60cccd73f17669da651e24669c8a2865e7c1fc
efc01a3879dd5c735003244"
QX_URL = "https://quantumexperience.ng.bluemix.net/api"
from qiskit import register
register(QX_TOKEN, QX_URL)
```

Registring your credentials

- Needs API Token and the QX URL.
- Needs to be done only once.

Import the Qiskit SDK

```
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import available_backends, execute
```

QISKIT CODE EXAMPLE – HELLO WORLD

Hello World!

1. Create registers
2. Create circuit
3. Add Hadamard gate
4. Add measurement
5. Plot circuit
6. Execute
7. Get results and print

Source code here:

https://mybinder.org/v2/gh/HassanNaseri/quantum-computing-handson/master

```
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import available_backends, execute
```

```
print("Available backends: ", available_backends())
```

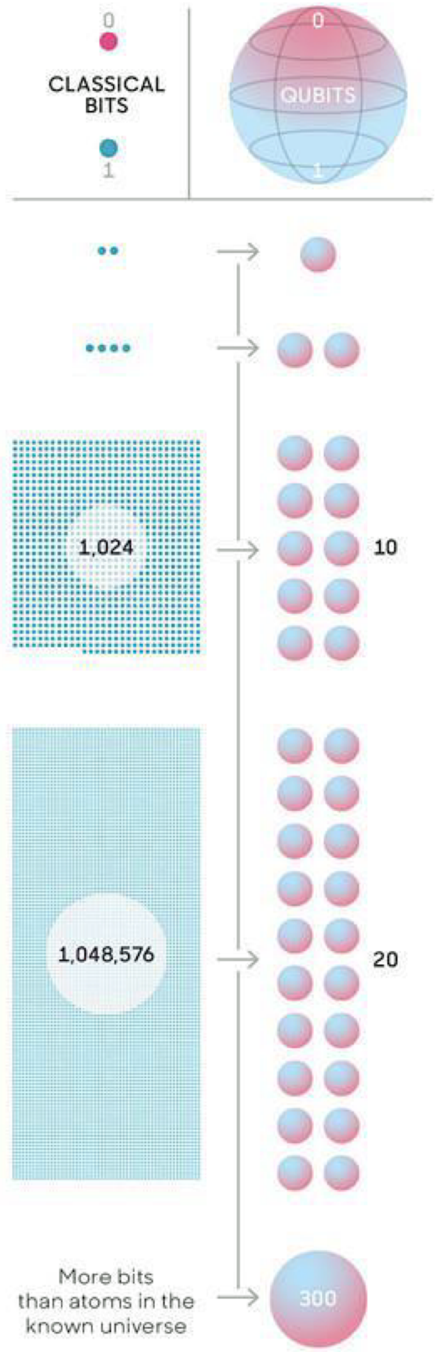
```
# Create a Quantum Register with 1 qubit. The initial state is |0>
q = QuantumRegister(1)
# Create a Classical Register with 1 bit.
c = ClassicalRegister(1)
# Create a Quantum Circuit
qc = QuantumCircuit(q, c)

# Add Hadamard gate to the qubit, putting it in superposition state.
qc.h(q)

# Add a Measurement gate to see the state.
qc.measure(q, c)

plot_circuit(qc)
```

```
job_sim = execute(qc, "local_qasm_simulator")

sim_result = job_sim.result()

print("simulation: ", sim_result)
print(sim_result.get_counts(qc))
```

CONTECTS

QUANTUM COMPUTING ALGORITHMS

- **The space of a quantum system grows exponentially.**
- **Quantum speedup requires quantum algorithms.**
- **It is difficult to design a new quantum algorithm.**
- **Different categories of algorithms:**
  - Exact algorithms
  - Approximate techniques
  - Hybrid algorithms
  - Quantum communication
- **Quantum algorithm provide exponential or quadratic speedup.**



QUANTUM ALGORITHMS – FOR UNIVERSAL COMPUTERS

Three categories of quantum algorithms:

1. **Quantum Fourier transform (QFT) (exponential speedups)**
   - **Shor's algorithm:** factorization of prime numbers
   - **Quantum algorithm for linear systems of equation**
2. **Amplitude amplification (quadratic speedups)**
   - **Grover's algorithm**: searches an unstructured database
3. **Quantum walks**
   - Graph optimization algorithms.

It is difficult to design a new Quantum algorithm!

APPLICATION AREA : CRYPTOGRAPHY

Cryptography Threat

- **RSA public key cryptosystem is based on complexity of solving a math problem.**
- Shor’s algorithm on a large quantum computer can break it in a short time.
- Such a computer could be available in 10 years.
- Many other cryptosystems are also at risk.

Responses

- Improving our current security practices
- Post-quantum cryptography
- Quantum communications

How much time do we have?

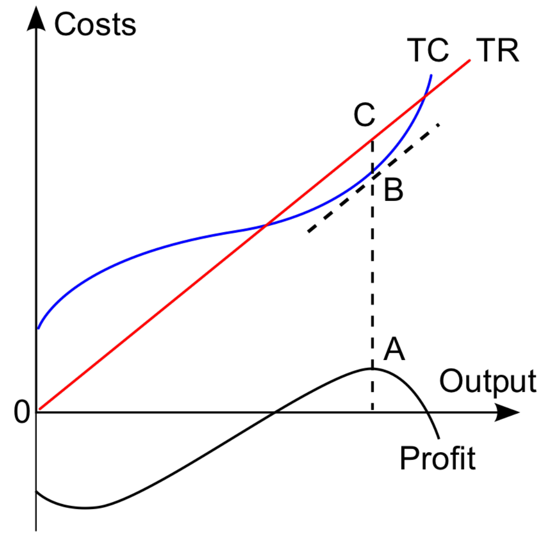
- How long would it take for a bank to plan and execute a security upgrade on legacy systems?
- What will happen to the archive of digitally-signed documents in a legal department?
- Do we have future-proof practices for data protection?

APPLICATION AREA : OPTIMIZATION AND MACHINE LEARNING

Mathematical optimization deals with finding the best solution to a problem.

- Examples: Portfolio optimization, Risk minimization

Optimization methods are the building blocks of machine learning and AI solutions.



Quantum computing has the promise of:

- Solving problems which are classically infeasible .
- Obtaining a substantial speedup in optimization.

Example techniques:

- **Quantum data fitting (regression)**
- **Quantum semidefinite programming**
- **Quantum combinatorial optimization**
- **Quantum support vector machine**

EXAMPLE - PORTFOLIO OPTIMIZATION

Portfolio optimization can easily become an intractable combinatorial optimization problem.

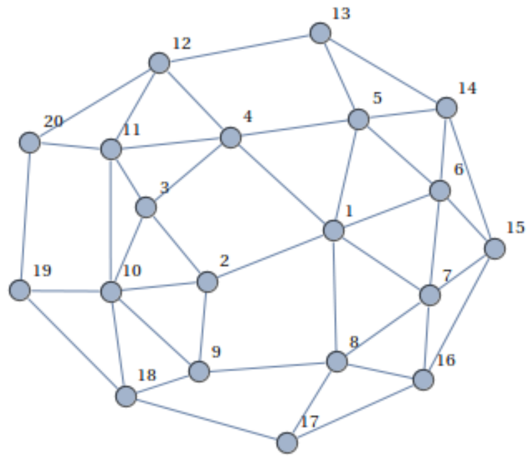
Example: Find portfolios containing only uncorrelated assets to minimize risk.

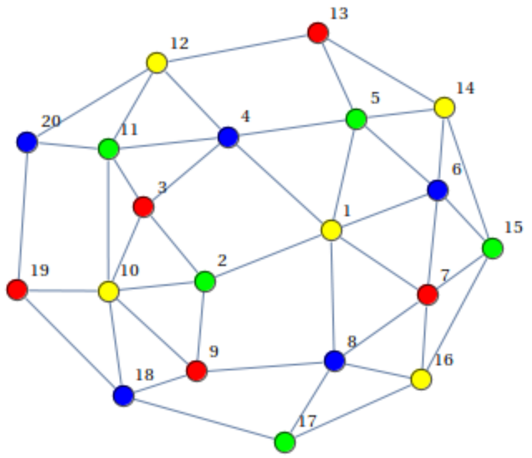
We represent the problem as a graph, where:

- Each asset corresponds to one node
- Each positive correlation is one edge
- Each node color corresponds to one portfolio

Graph coloring problem: assign a color to each node avoiding a similar color on connected nodes.

Technique: Quantum annealing on a D-Wave machine



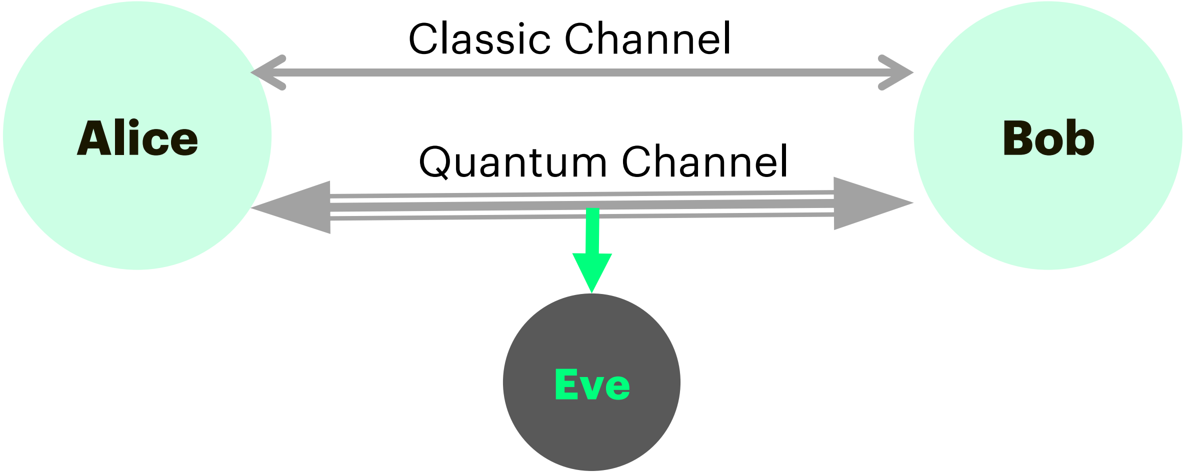


APPLICATION AREAS – QUANTUM KEY DISTRIBUTION

In Cryptography a secret key is used to encrypt/decrypt messages.

Key distribution is a mechanism to share a key between two parties in a secure manner.

Quantum entanglement provides a mechanism to distribute keys – completely tamper proof.



Current Status

- Currently available for point to point cases.
- Experiment: Shanghai to Beijing via satellite.

CONTECTS

ONE QBIT GATES CONT.
PAULI GATES AND S-GATE

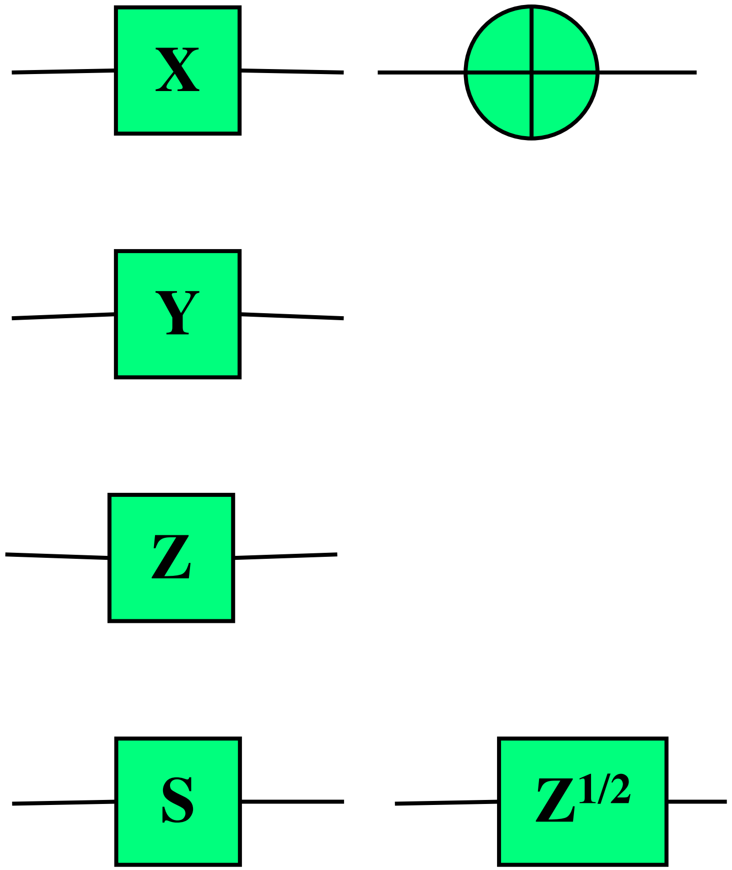
Pauli Gates

- X-gate: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- Y-gate: $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
- Z-gate: $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

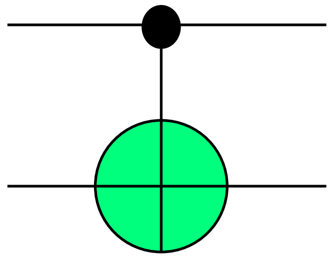
$I^2 = H^2 = X^2 = Y^2 = Z^2 = I$

S-gate

- $S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$



TWO-QUBIT GATES



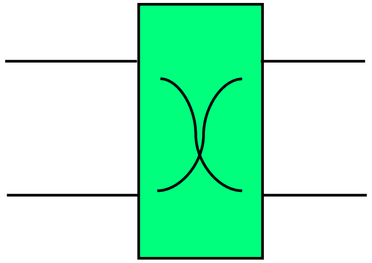
With respect to the basis:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

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

CNOT GATE (A UNIVERSAL GATE)

- It can be used to entangle and disentangle EPR states.
- Any quantum circuit can be simulated to an arbitrary degree of accuracy using a combination of CNOT gates and single qubit rotations.



SWAP GATE

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

VECTOR SPACE AND ENTAGLEMENT

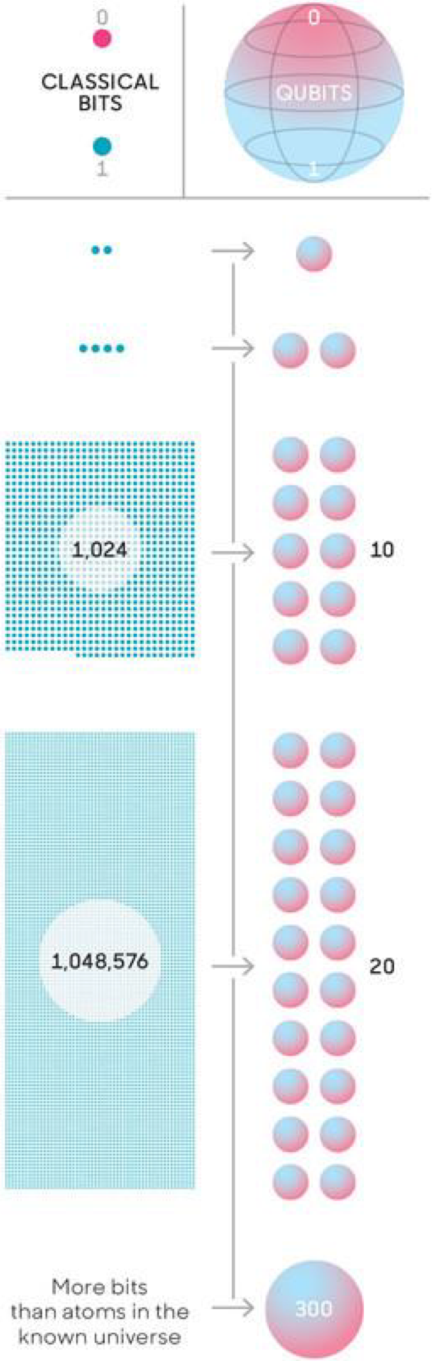
- **When multiple qubits interact the resulting state is in a vector space that contains all combinations of individual states.**

- **N qubits -> 2^N basis states**

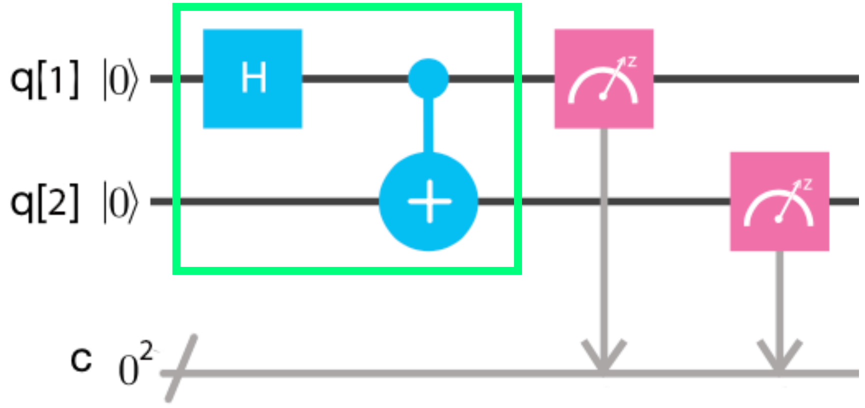
- **Vector representation of two qubits in their joint vector space**

$$v_{00}|00\rangle + v_{01}|01\rangle + v_{01}|01\rangle + v_{11}|11\rangle \rightarrow \begin{bmatrix} v_{00} \\ v_{01} \\ v_{01} \\ v_{11} \end{bmatrix}$$

- $|ab\rangle$ **is the basis vector representing a state**



QUANTUM CIRCUIT



Quantum Circuit is composed of

- Quantum and classic registers
- Quantum gates
- Measurement gates

$$|\Phi^+\rangle = \mathrm{CX} \cdot H_1 |00\rangle$$

$$|00\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

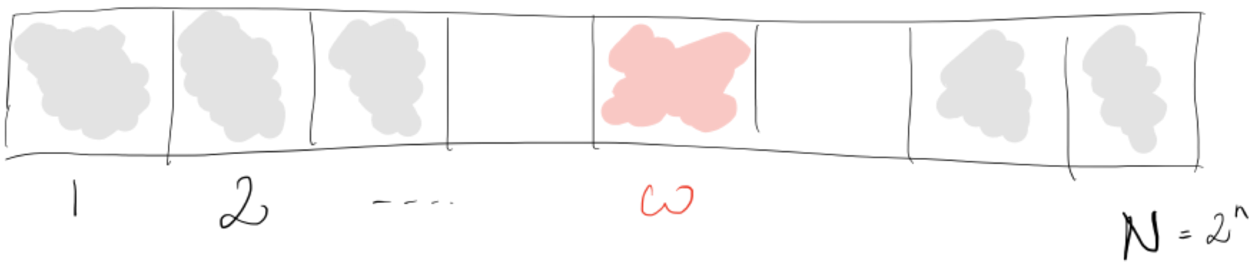
Example: Bell states

- The first Bell state can be created using a Hadamard gate and a CNOT gate.
- It creates and entanglement.
- Measurement of one qubit will assign one of two possible values to the other qubit instantly.

CONTECTS

PROBLEM SETUP
UNSTRUCTURED DATABASE SEARCH

- **We are searching for one item with a unique property in a large list of N items where there is no specific structure (e.g. order) in the dataset.**



- A list of N items. Only one has a **unique property** that we wish to locate; aka winner w.
- We have a check function to test if a given entry is winner or not, aka Oracle.
- Example: checking boxes one by one to find the only one box that contains an apple!

A CLASSICAL EXAMPLE
UNSTRUCTURED DATABASE SEARCH

- Find code in GitHub or in myBinder
  - **https://github.com/HassanNaseri/quantum-computing-handson**
  - **https://mybinder.org/v2/gh/HassanNaseri/quantum-computing-handson/master**

- Open the Jupyter Notebook file: **search_classic.ipynb**

- **You can run Jupyter Notebook files cell by cell.**

- The goal is to find the **shelf** number for a given fruit name
- We have access to **black_box_check(key, value)** function.

black_box_check(key, value)

- arguments
  - **key**: string
  - **value**: 4-bit uint
- return
  - NOT(**value**) if **value=winner**
  - **value** otherwise

CLASSICAL STRUCTURED DATABASE SEARCH
EXERCISE #1

- **Task 1:** Complete the code by giving value to

  Maximum_number_of_steps =

- **Task 2:** Complete the code by writing the second for loop

  Same loop as before except for using **random_indices** in the for statement.

- How many iterations are needed in average?

black_box_check(key, value)

- arguments
  - **key**: string
  - **value**: 4-bit uint
- return
  - NOT(**value**) if **value=winner**
  - **value** otherwise

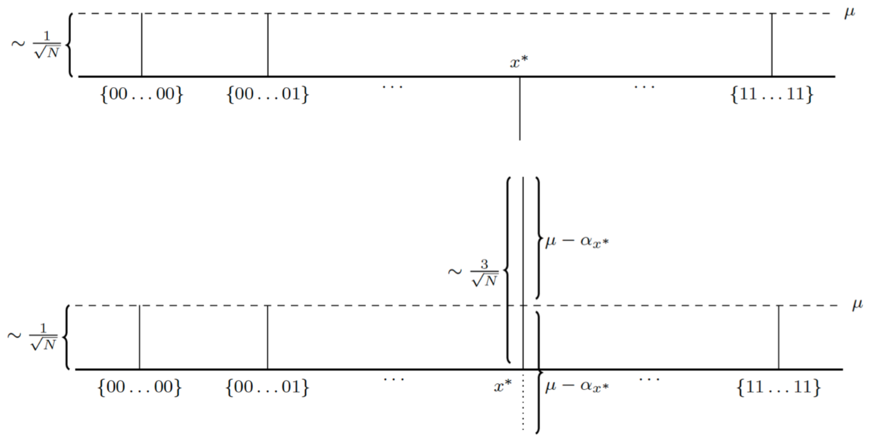
GROVER'S ALGORITHM

- **Amplitude amplification technique**
- **Unstructured search (Classical complexity is in the order of N)**
- **A very generic problem in computer science, optimization, …**
- **Quadratic speedup, $\sqrt{N}$ evaluations (e.g. 1000 instead of 1M operations)**

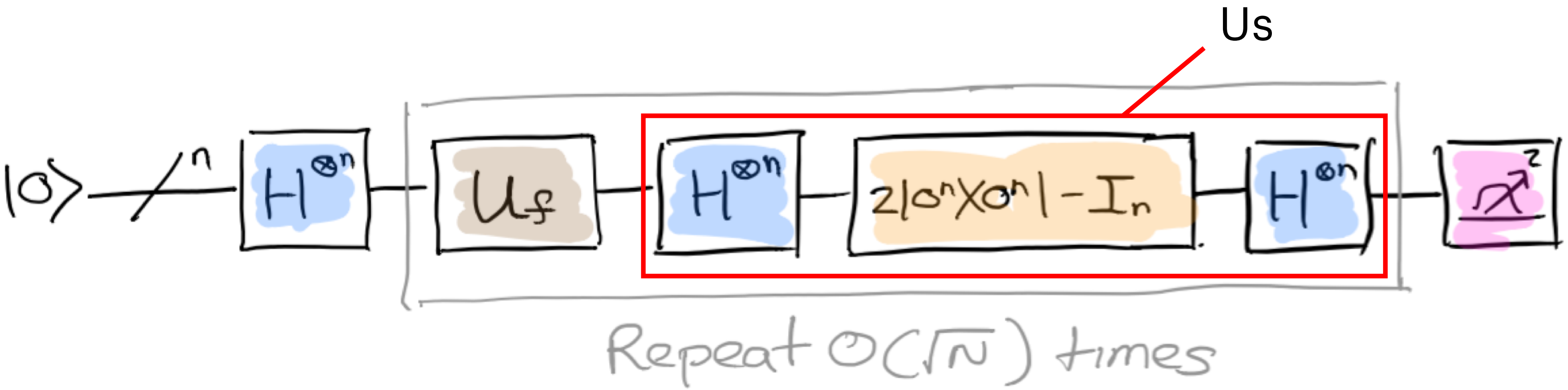
- **IBM Q - User Guide**

https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/004-Quantum_Algorithms/070-Grover's_Algorithm.html

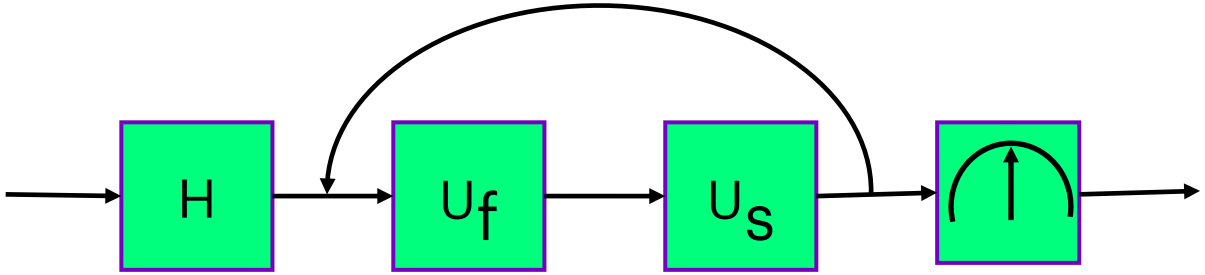
GROVER'S ALGORITHM OVERALL PROCESS



1. **Initialization**
2. **Uf: Black box search**
3. **Us: Reflection about average**
4. **Goto step 2** and repeat $\sqrt{N}$ times.



GROVER'S ALGORITHM EXERCISE #3



Navigate to

https://mybinder.org/v2/gh/HassanNaseri/quantum-computing-handson/master

Open file: **grover_1.ipynb**

Complete the code by calling two functions:

- black_box_check(circuit, key, q)
  - **circuit**: quantum circuit object
  - **key**: string containing the unique property of interest (fruit name)
  - **q**: quantum register for input/output
- reflection_about_average(circuit, q)
  - **circuit**: quantum circuit object
  - **q**: quantum register register for input/output

CONTECTS

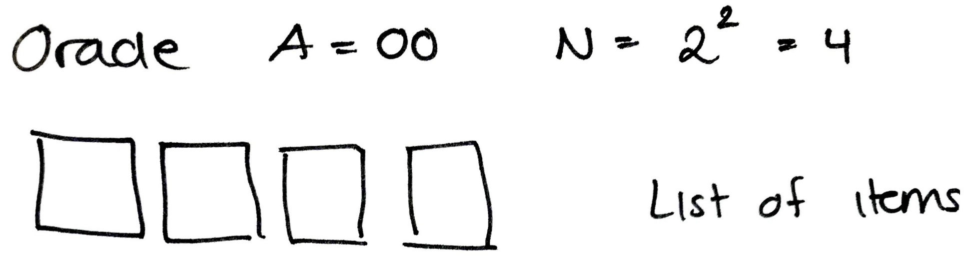
QISKIT EXAMPLE: THE ORACLE
HOW TO ENCODE A LIST OF ITEMS IN A QUANTUM COMPUTER?

Let's look at U_f or "the Oracle" for 2 qubit case

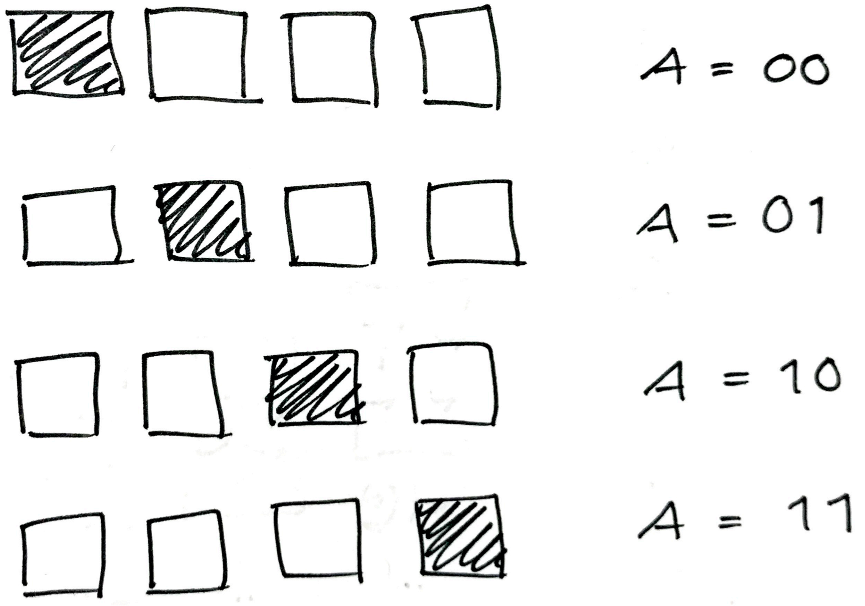
$$U_f\,|q\rangle = \begin{cases} -|q\rangle & if\ \ q = w \\ \ \ |q\rangle & otherwise \end{cases}$$

- The oracle knows the correct answer and flips the phase for that state.
- The oracle is a black box fucntion.

- Example of oracle for w = |00⟩:
- https://algassert.com/quirk#circuit={"cols":[["X"],[1,"H"],["•","X"],[1,"H"]]}
- https://algassert.com/quirk#circuit={"cols":[["Z^%C2%BD","Z^%C2%BD"],[1,"H"],["•","X"],[1,"H"],["Z^%C2%BD","Z^%C2%BD"]]}
- https://algassert.com/quirk#circuit={"cols":[["H","H"],["Z^%C2%BD","Z^%C2%BD"],[1,"H"],["•","X"],[1,"H"],["Z^%C2%BD","Z^%C2%BD"],["H","H"]]}



Four possible Oracles:



GROVER'S: THE LOGIC

- We start with a uniform superposition state.
- Oracle only changes the phase, but probabilities are not changed.
- If measured will collapse to any one of the basis states with the same probability of $1/N = 1/2^n$
- Assume that winner is $|w\rangle = |10\rangle$. We want an algorithm (transformation) that maps state $|s\rangle$ to state $|w\rangle$.

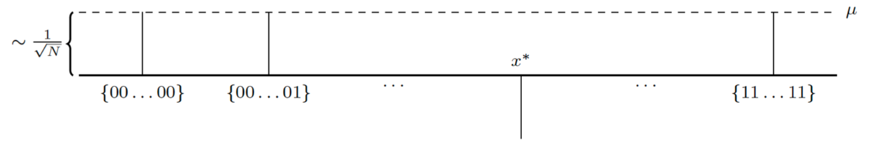
$$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

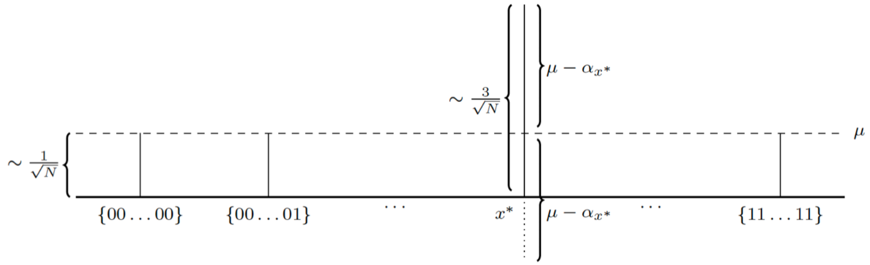
GROVER'S: AMPLITUDE AMPLIFICATION REFLECTION ABOUT THE AVERAGE

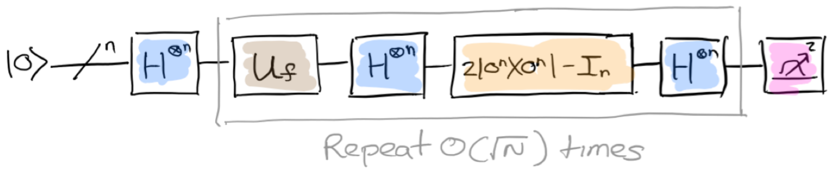
- Amplifies the amplitude of the marked item and shrinks the other amplitudes.
- Measurement collapses to the correct solution with high probability.
- A nice geometrical interpretation in terms of two reflections.

- **step 0** Initialize the uniform superposition, $|\psi_0\rangle = |s\rangle$.
- **step 1** Apply the oracle U_f : reflection about origin (|w⟩ => -|w⟩).
- **step 2** Apply rotation Us : reflection about the average.
- **Goto step 1** and repeat $\sqrt{N}$ times.

- **The transformation UsUf boosts the amplitude of |w⟩.**
- $\sqrt{N}$ iterations increase the winner probability N times.



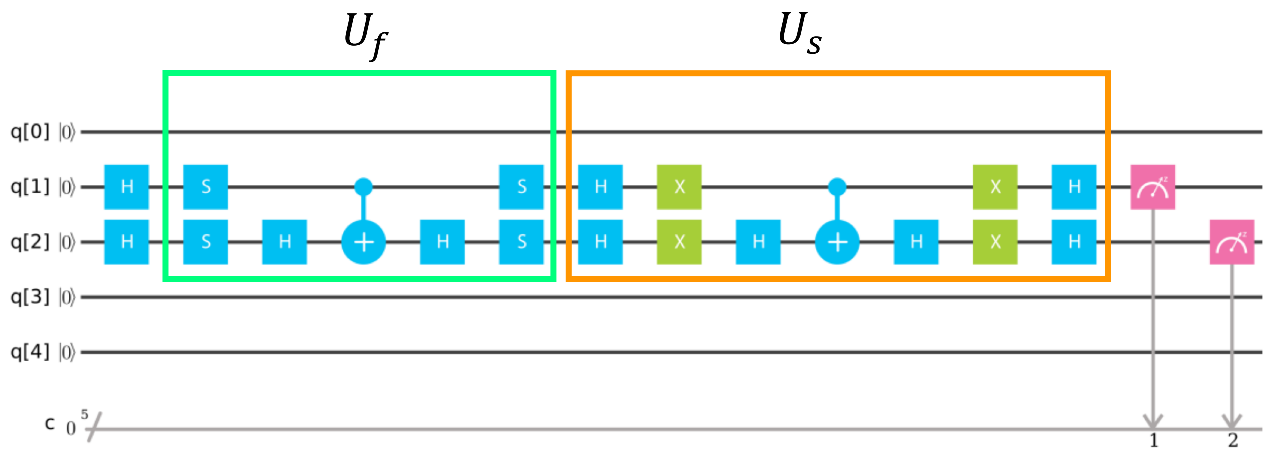


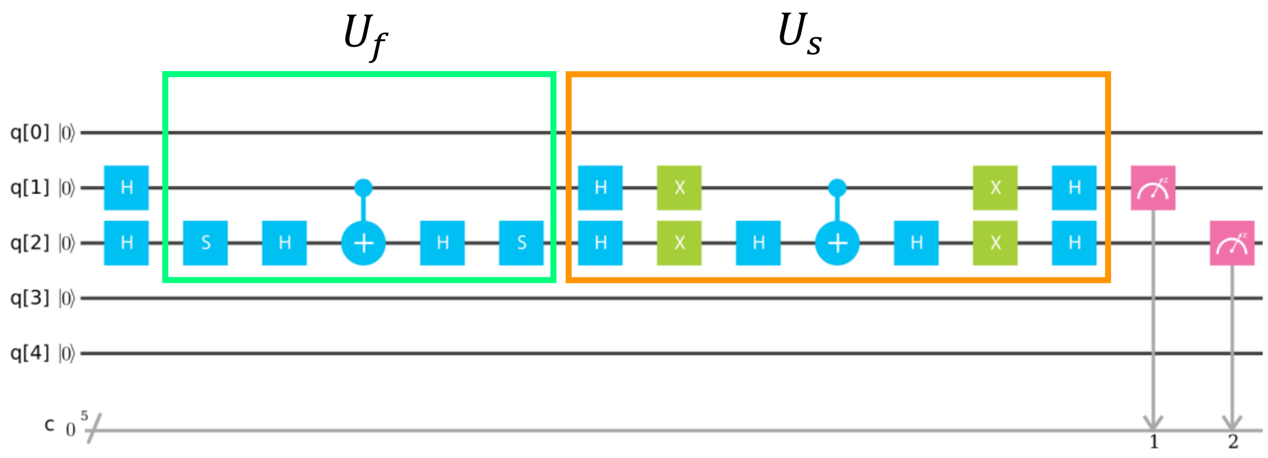


GROVER'S: IMPLEMENTATION

Implement this using QISKIT or Graphic composer.

- **Initialize**: Hadamard
- **Uf**: using control-Z gate, made from CNOTs
- **Us:** using control-Z gate.



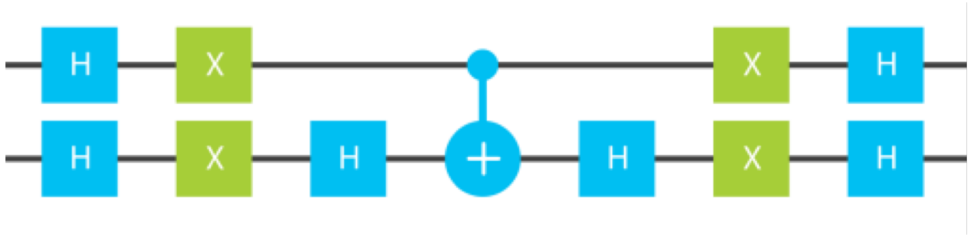


Grover's algorithm N=2, A=01

Examples

- Grover N=2 A=00
- Grover N=2 A=01

GROVER'S ALGORITHM EXERCISE #4



Navigate to

https://mybinder.org/v2/gh/HassanNaseri/quantum-computing-handson/master

Open file: **grover_2.ipynb**

Complete the code by implementing the function:

- **reflection_about_average(circuit, q)**
  - **circuit**: quantum circuit object
  - **q**: quantum register register for input/output

Gates needed:

- H : Hadamard
- X : Pauli X-gate
- CX: CNOT GATE

How to implement CNOT

circuit.cx(q[0], q[1])

SOLUTION

```
def reflection_about_average(circuit, q):
    # Reflection about average for amplitude amplification
    circuit.h(q)
    circuit.x(q)
    circuit.h(q[1])
    circuit.cx(q[0], q[1])
    circuit.h(q[1])
    circuit.x(q)
    circuit.h(q)
```

GROVER'S: THE QUANTUM TRICK!

- **We did not check for each shelf separately.**
- **We did not ask if apple is in shelf number 1.**
- **We asked if the apple in shelfs: 1 + 2 + 3 + 4 with 25% probability each!**
- **The exact input is:**

$$\frac{1}{2}\left|00\right\rangle + \frac{1}{2}\left|01\right\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

$$\begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

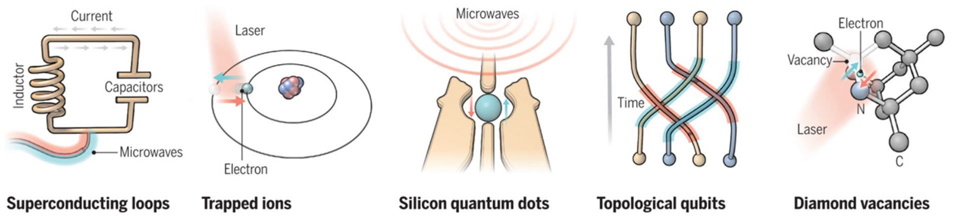
- **The output is:**

$$0|00\rangle + 0|01\rangle + 1|10\rangle + 0|11\rangle$$

- **The measurement almost always collapses to the correct state:** $|10\rangle$**.**

CONTECTS

CONCLUSIONS: HARDWARE AND SOFTWARE



- **There is no standard architecture or de facto technology yet.**
- **Current computer are very limited:**
  - They are very few and fragile.
  - The have limited capacity.
  - The coherence time is short.
  - The technology is not fault-tolerant.
- **The progress has just started and it is believed to grow very fast!**

IBM Q™ QISKit Google AI Cirq Q# Forest™ SDK

Plenty of things can/should be done before the ultimate hardware is ready.

Software development tools:

- IBM Q Experience
- Qiskit SDK
- Qiskit Aqua and other repositories: Aqua Chemistry, Aqua Optimization, Aqua Artificial Intelligence
- D-Wave: Internet API and programming libraries (C/C++, Python, and MATLAB)
- Microsoft: Quantum SDK, Q# (a new piece for Visual Studio) Azure simulator
- Google: Cirq (A python framework similar to Qiskit)

QUANTUM COMPUTING AT ACCENTURE

Quantum Computing Roadmap (next generation of computing)

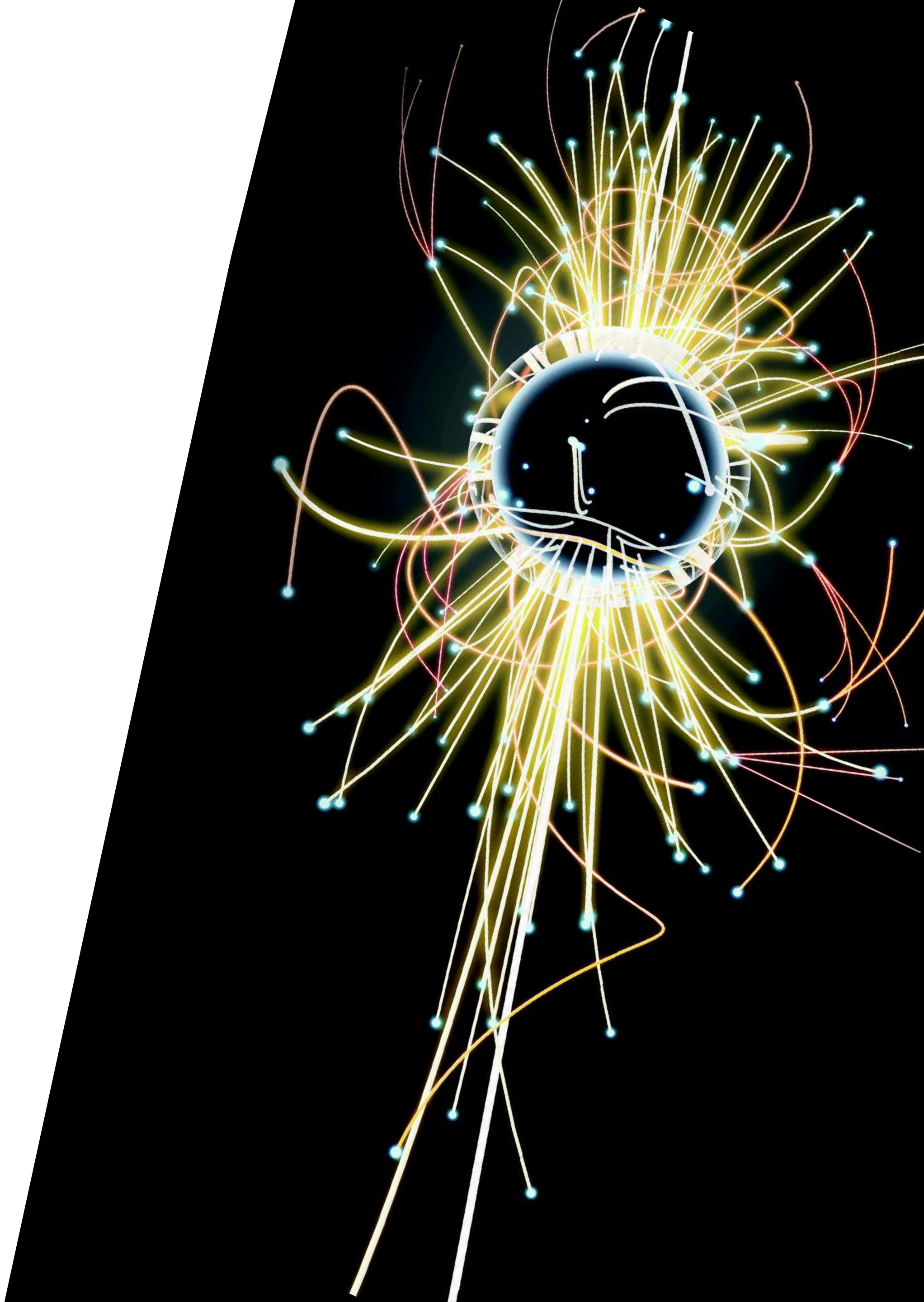
- End-users, Application Developer, Service Provides, Technology Developers

Quantum research teams in more than 20 countries

Partnership with 1QBit and development on D-Wave machines

Accenture-IBM Quantum Hub Project

- A global collaboration project among Accenture teams and IBM
- Develop business-driven prototypes on IBM quantum computers
- Access to latest IBM quantum computers and resources



SOME RESOURCES

- **A good introductory tutorial on quantum computing: https://people.cs.umass.edu/~strubell/doc/quantum_tutorial.pdf**
- **A video series on quantum computing (YouTube): https://www.youtube.com/watch?v=X2q1PuI2RFI&list=PL1826E60FD05B44E4**
- **IBM Q Experience user guide: https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/introduction.html**
- **A nice tool for learning (QUIRK): https://algassert.com/quirk**
- **A white paper from Accenture Labs (Innovating with Quantum Computing): https://www.accenture.com/t00010101T000000__w__/br-pt/_acnmedia/PDF-45/Accenture-Innovating-Quantum-Computing-Novo.pdf**
- **Quantum Developer Week, Accenture Media Exchange: https://mediaexchange.accenture.com/media/t/0_s769log0**
- **Quantum Computing Lecture including Linear Algebra (The University of Chicago) http://people.cs.uchicago.edu/~razborov/teaching/QuantumComputing/notes.pdf**
- **Grover's algorithm: https://www.cs.cmu.edu/~odonnell/quantum15/lecture04.pdf**
- **For a curious mind: Richard Feynman's lecture on quantum mechanics: https://www.youtube.com/watch?v=OYp_Mq_-QOs**

THANK YOU!

QUESTIONS?

Hassan Naseri

Tim Leonhardt