A Research on Quantum Computer Architectures

Presentation By Team 9

Outline

Introduction

What is Quantum Computing?

The word "quantum", in quantum computer, originates from "quantum mechanics," a basic theory in physics. In brief, on the scale of atoms and molecules, matter behaves in a quantum manner. A quantum computer is a machine that performs calculations based on the laws of quantum mechanics, which is the behavior of particles at the sub-atomic level.

In other words it is the use of quantum-mechanical phenomena such as superposition and entanglement to perform computation.

Moore’s law: Since the 1960s, the number of components, e.g. transistors, in integrated circuits (ICs) has doubled approximately every two years. This exponential growth is described by Moore's law and it enabled exponential increase in calculation power.

Problem of classical Computer

Ex- **Optimization**

10 people gathered for dinner

Q: **How many different ways are there**

to configure 10 people around the table?

A: 10 Factorial i.e. 3628800 (3.6 million ways)

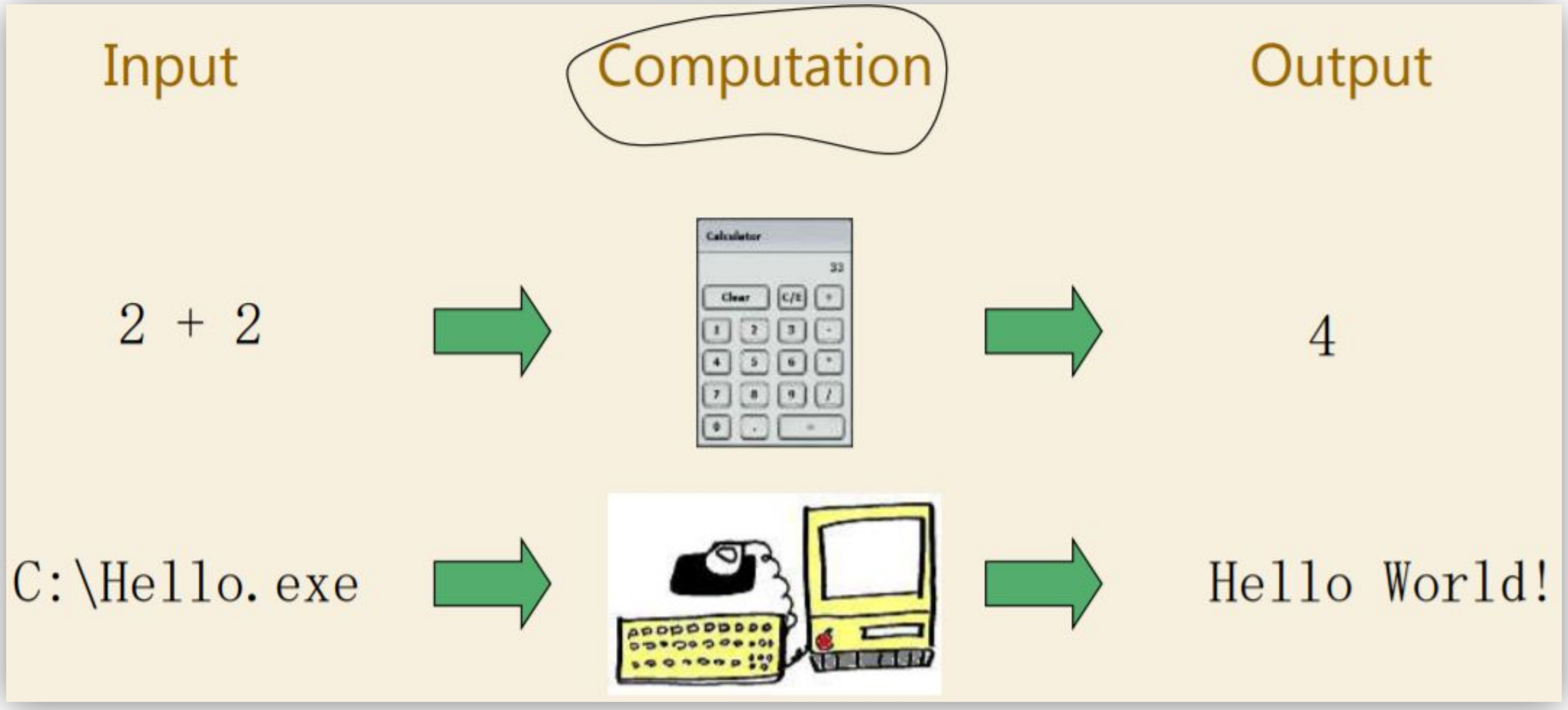
A classical computer will have to go through each of the 3.6 million ways individually and then compare them to figure out the best optimization.

Quantum Mechanical Computers

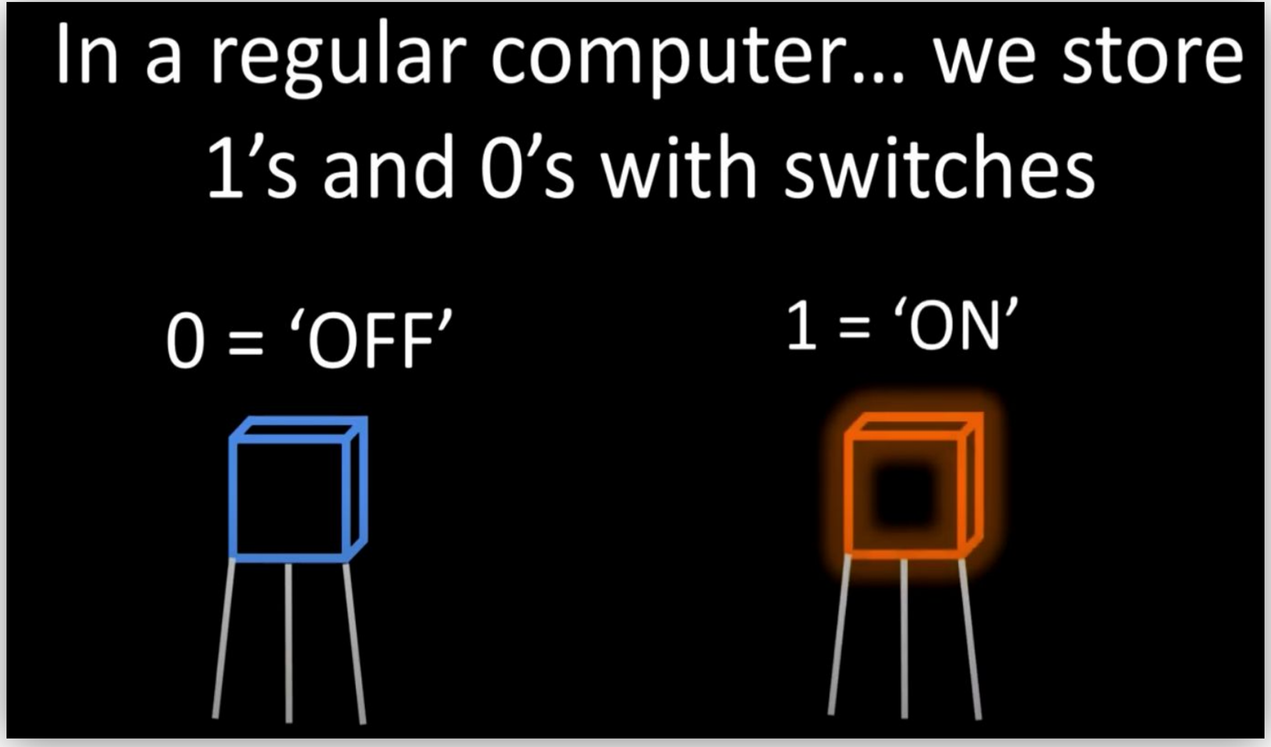
Our society runs on classical computers: memory-locations have specific value (0 or 1), processor acts on specific location, .

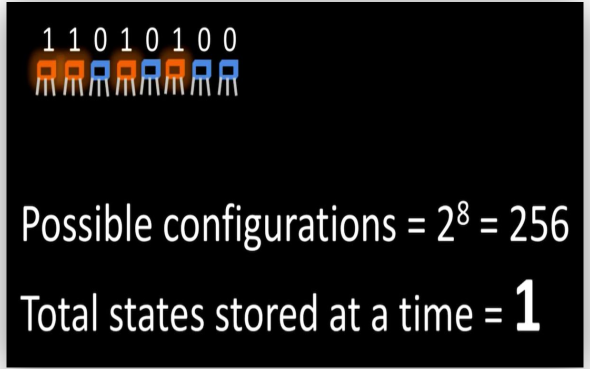
But our world is not classical, so let's study quantum-mechanical computers Why?
1. Enable further miniaturization (Moore's law)
2. Enable large speed-up for some important problems
3. The goal of computer science is to study the power of the best computing machines that Nature allows us

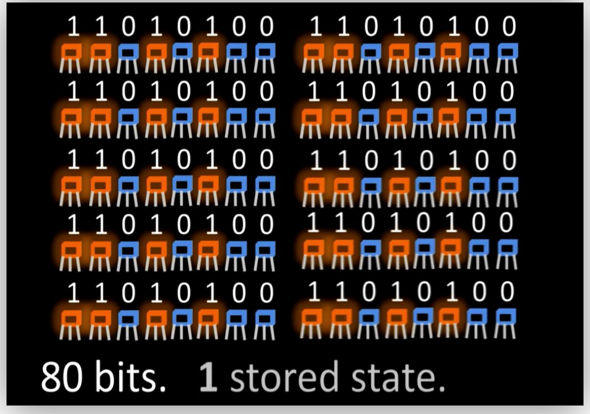
Classical Computation

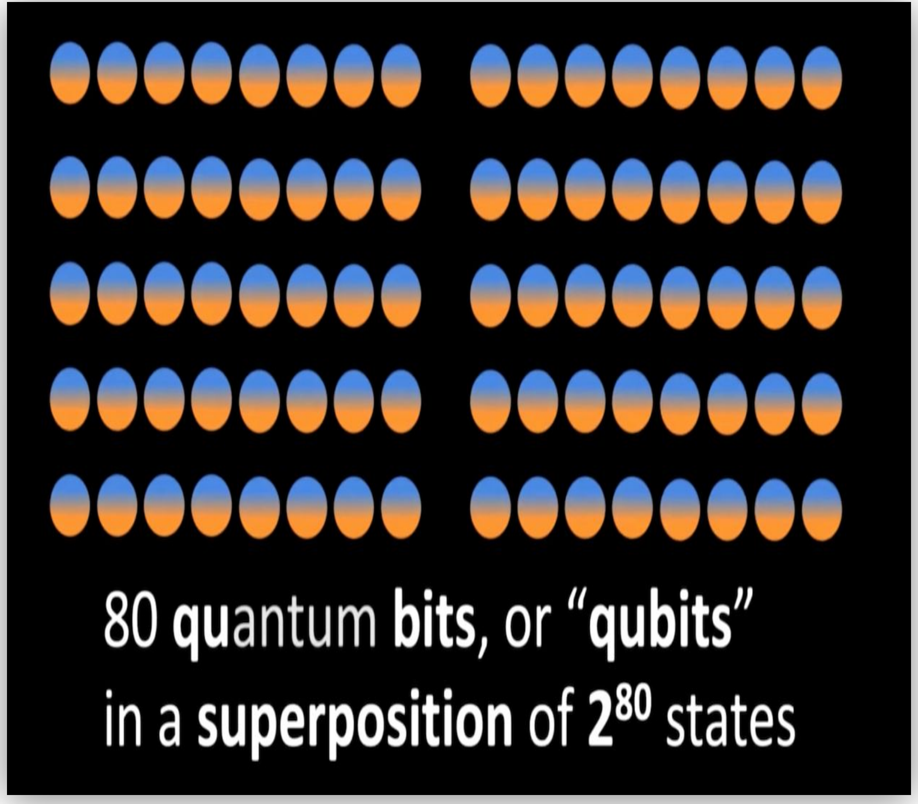


Binary Digit(Bit) - Transistors




1 1 0 1 0 1 0 0
Possible configurations = 2^8 = 256
Total states stored at a time = 1


1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
1 1 0 1 0 1 0 0
80 bits. 1 stored state.


80 quantum bits, or "qubits"
in a superposition of 2^{80} states

How big is 2^{80}?

1,208,925,819,614,629,174,706,176

Bigger than the number of atoms in the observable universe!

Quantum bits (Qubits)

In existing computers, all information is expressed in terms of 0s and 1s, and the entity that carries such information is called a "bit.“

- A bit can be in either a 0 or 1 state **at any one moment in time.**

- A quantum computer, on the other hand, uses a “**quantum bit**” or "**qubit**" instead of a bit.

- A qubit also makes use of two states (0 and 1) to hold information, but in contrast to a bit, In this state, a qubit can take on the properties of 0 and 1 **simultaneously** at any one moment

2 qubits: superposition of **4** possible basis states (00,01,10,11)
3 qubits: superposition of 8 possible basis states
.
.
.
n qubits: superposition of 2^n possible basis states
Described by a “wavefunction": vector of all 2^n amplitudes

Properties of Quantum Computing

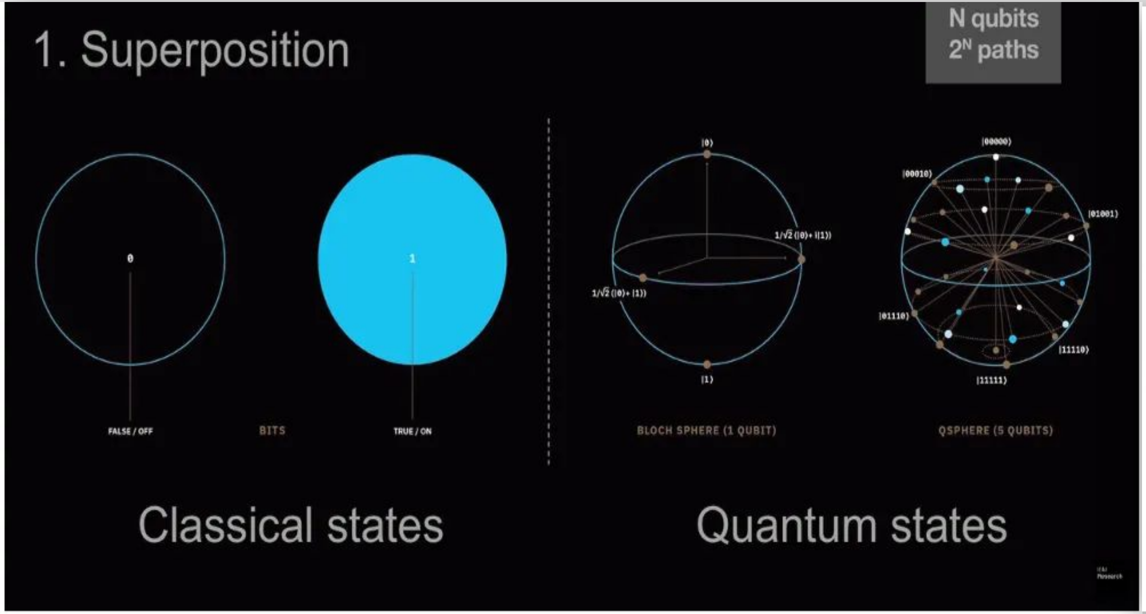
→ **Superposition** is essentially the ability of a quantum system to be in multiple states at the same time — that is, something can be “here” and “there,” or “up” and “down” at the same time.

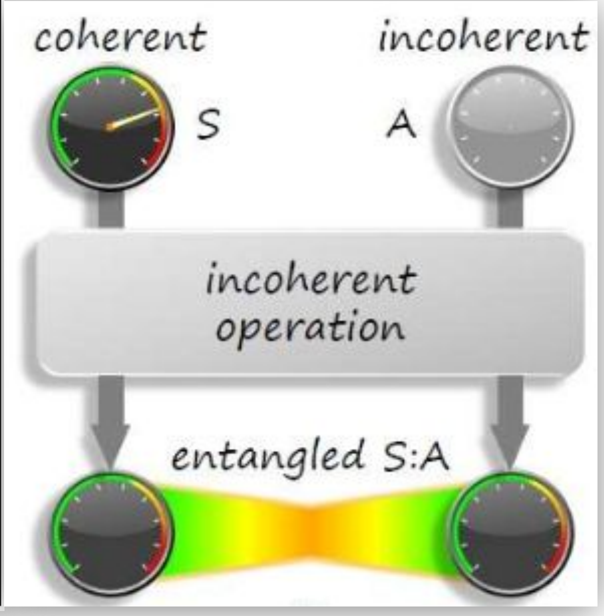
→ **Entanglement** is the ability of quantum systems to exhibit correlations between states within a superposition.

→ **Quantum Coherence** deals with the idea that all objects have wave-like properties. **Coherence** also lies at the heart of **quantum computing**, in which a qubit is in a superposition of the "0" and "1" states, resulting in a speed-up over various classical algorithms.

→ **Quantum interference**, a byproduct of superposition, is what allows us to bias the measurement of a qubit toward a desired state or set of states

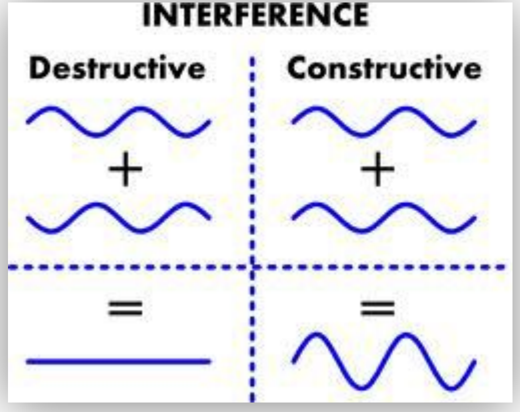
Entanglement & Superposition

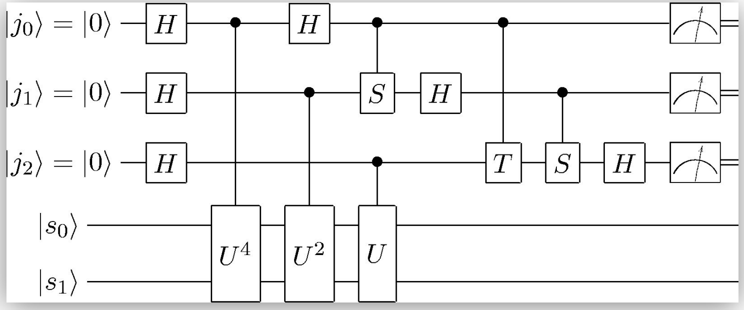




Laws of Quantum Computation:

Waves can strengthen or weaken each other:





Quantum computation = superposition + interference

1. Start with qubits in some simple state (e.g. all |0)
2. Run circuit of \elementary gates"
   creating the right interference,
   so the nal state has most of its
   weight on solutions to your problem

How Quantum Computer would solve factorial problem?

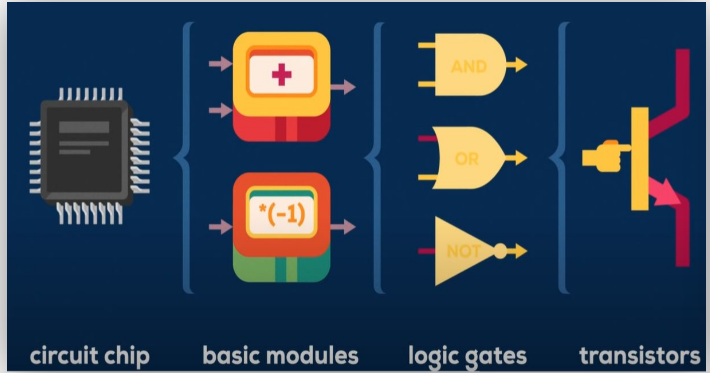
A quantum computer would:

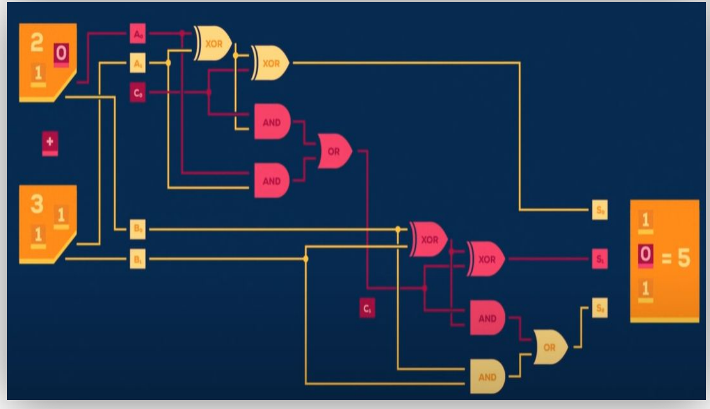
Take the qubits and go into a superposition of all possible states and configurations.

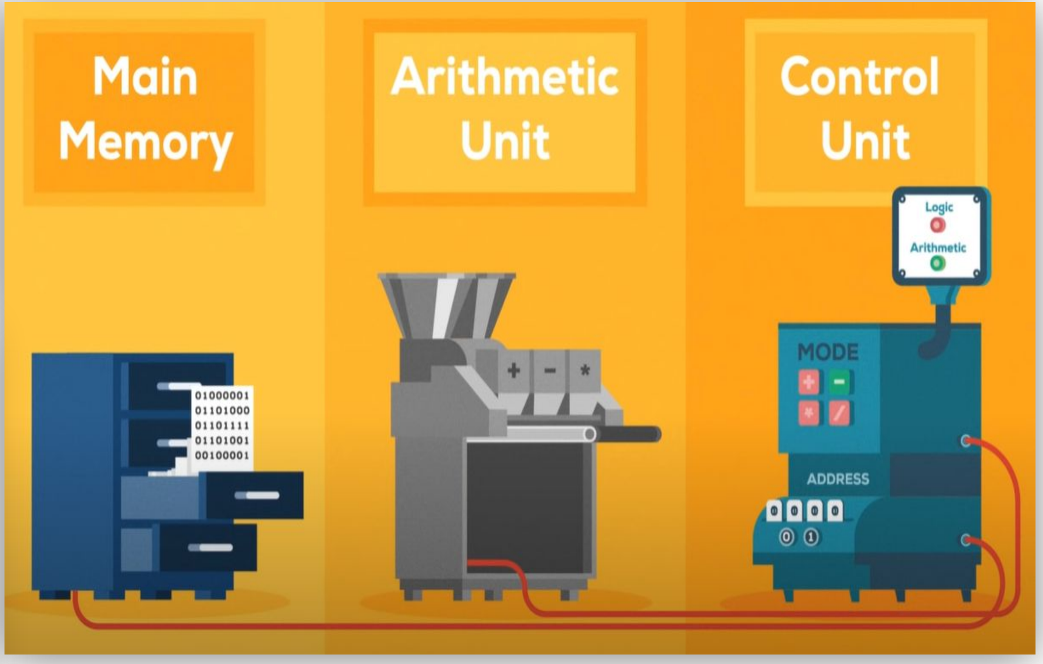
When this problem is encoded into a quantum computer, the encoder applies a phase on each of the states. When the ways are in phase, the amplitudes add. When they are out of phase, the amplitudes cancel. This is a similar idea to noise cancelling headphones, which create noise to phase out outer noise.

You use interference to amplify some answers, while cancelling the others, finally approaching one answer.

Logic gates of Classical Computer







Common Gates of Quantum Computers

Hadamard Gate(SRN gate):

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

Transforms |0 to (|0 +|1)/2
and |1 to (|0 -|1)/2

Pauli-Y gate:

$$Y=\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$$

Acts on a single qubit.

Transforms |1 to -i|0 and |0 to i|1

Pauli-X gate:

$$X=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

Acts on a single qubit.

Quantum equivalent of NOT gate.
Transforms |1 to |0 and |0 to |1

Pauli-Z gate:

$$Z=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

Acts on a single qubit.

Transforms |1 to -|1 and |0 remains unchanged.

Phase shift gate:

$$R_\theta=\begin{pmatrix}1 & 0\\ 0 & e^{i\theta}\end{pmatrix}$$

Acts on a single qubit.

Transforms |1 to ei |1 and |0 remains unchanged.

Modifies(rotates) the phase of quantum state by theta.

How does the use of qubits affect computation?

Classical Computation	Quantum Computation
Operations: logical	Operations: unitary
Valid operations:	Valid operations:

Classical Computation — 1-bit

NOT =

	in: 0	in: 1
out	1	0

Classical Computation — 2-bit

AND =

in \ in	0	1
0	0	0
1	0	1

(out)

Quantum Computation — 1-qubit

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\sigma_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \qquad H_d = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Quantum Computation — 2-qubit

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

The CNOT gate is two-qubit operation, where the first qubit is usually referred to as the control qubit and the second qubit as the target qubit. Expressed in basis states, the CNOT gate

Quantum computers: hype vs substance

Assume large quantum computers will be built in the next decades.

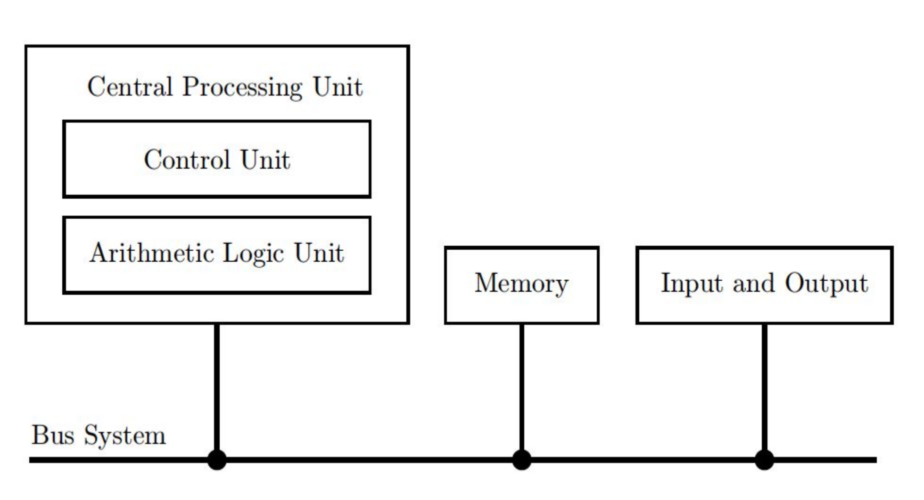
Where will they have a real impact?
→ Probably: Cryptography, optimization, simulation
→ Maybe: Machine learning
→ Forget about it: NP-hard problems (TSP, protein folding,. . .), ending climate change, nding ET, . . .

Outline

A Quantum von Neumann Architecture

Classical Computer Architecture

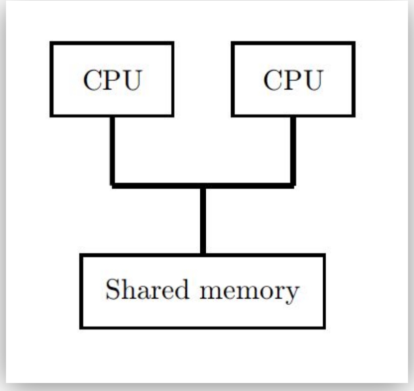
- Most modern computer architectures are based on the von Neumann architecture.
- The fundamental idea behind the von Neumann architecture is to divide the computer into individual parts
  - Arithmetic Logic Unit (ALU)
  - Control Unit (CU)
  - Bus System
  - Memory
  - Input and Output (IO)



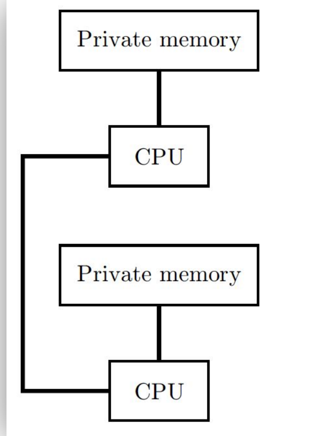
The von Neumann Computer Architecture

Parallelism in Classical Computing

- Speedup of computation in computers is achieved by multiple methods:
  - Increasing the clock speed of the CPU.
  - Executing multiple instructions simultaneously in parallel (Multiprocessor and Multicomputer) systems).



Multiprocessor system with a shared memory



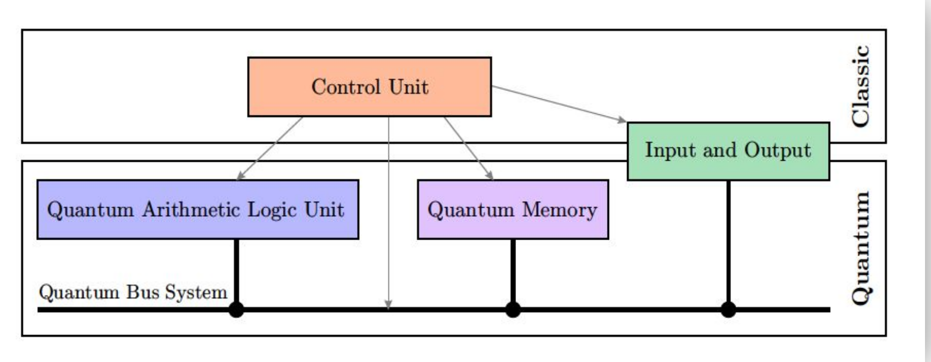
Multicomputer system with each processor having local memories

Quantum von Neumann Architecture

- The field of quantum computer architecture is still in its infancy.
- A general architecture for a quantum computer, like von Neumann architecture for classical computers is very challenging as the interactions for qubit manipulation vary greatly from technology to technology.
- Facilitating scalability in quantum computer hardware is one of the most challenging tasks of quantum computer architectures.
- In this paper, the quantum von Neumann architecture is introduced by combining the classical von Neumann architecture with the requirements of quantum computing.

Quantum von Neumann Architecture

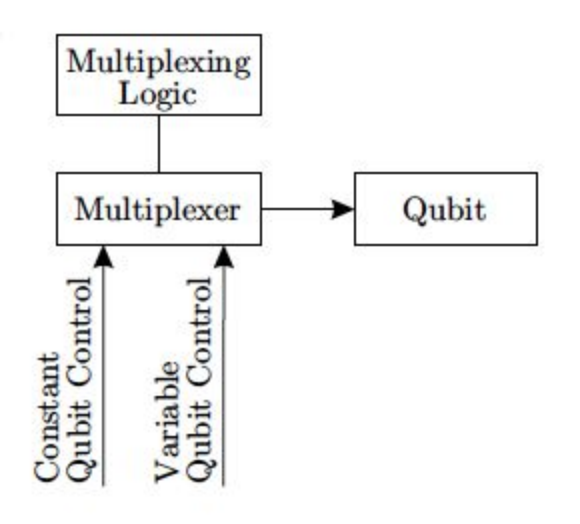
- The quantum computer requires a classical control unit which controls the quantum computer.
- A quantum bus system allows moving quantum information between the different parts of the quantum computer.
- The manipulation of the quantum information is executed in the quantum arithmetic logic unit (QALU).
- The quantum information is stored in the quantum memory which will rely on multiplexing technology.
- Lastly, an input and output region acts as an interface to the classical world in which the qubit state is initialized and/or detected.



The Quantum von Neumann architecture

Quantum Memory Region

- In classic computers, the components of a DRAM needed to store one bit of information are a field effect transistor (FET) and a capacitor.
- Digital Multiplexing logic controls the FET to access the DRAM cell. This lower hardware demand per bit results in big data storage capacities on DRAM chips.
- Large scale quantum computers have to achieve big quantum data storage capacities with low hardware demand in the quantum memory.
- This hardware demand can be reduced by utilizing multiplexing circuits.
- This Quantum memory stores an ion chain in a segmented Paul Trap, using a negative DC voltage at the position of the ion string and positive DC voltages surrounding it which forms an axial confinement for the ion string.

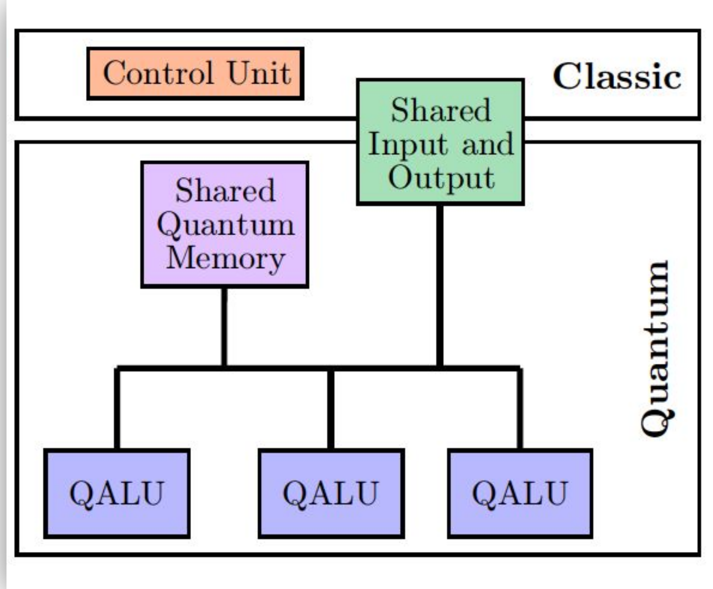


Multiplexing Quantum Memory

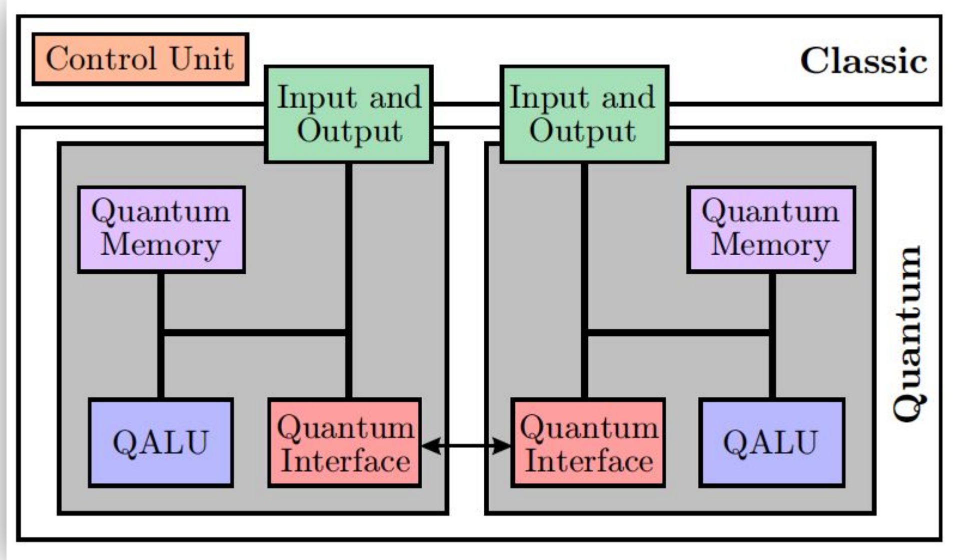
Quantum Information Transport

- Quantum information cannot be copied, therefore they must be transported by physically moving the qubits, quantum teleportation or through coupling to photons.
- This has to be performed with high fidelity to allow fault-tolerant QC.
- Atomic or molecular qubit systems enable quantum information transport via physical movement of the qubits from one location in space to another.
- Quantum teleportation requires an entangled qubit pair and is very hardware demanding. Quantum Information Processing through teleportation is possible in superconducting quantum computing systems.
- Mapping qubits to photons was also shown to be possible in atomic or molecular qubit systems.
- Due to the high hardware demand in the memory, quantum teleportation and mapping to photons will only be applicable in small and medium scale systems.

Parallelism in Quantum von Neumann architectures



A multi quantum processor system with a shared quantum memory



A multi quantum computer which couples multiple quantum computers via a quantum interface

The Quantum 4004

- The Intel 4004 was one of the world's first microprocessors and was the first microprocessor which customers could program themselves (4-bit CPU, Instruction Cycle time: 10.6 microseconds, Instruction Execution Time: 1 or 2 instruction cycles, able to directly address 32768 bits (4096 bytes) of memory)

- The Quantum 4004 is based on the presented von Neumann architecture with design parameters corresponding to the technical specification of the Intel 4004.

The Quantum 4004

- The Quantum 4004 was built with a simple hardware with one QALU, and its computation speed is 10 microseconds for a single qubit operation and 20 microseconds for two qubit operations.
- It is structured in 4 qubits per ion string, which are DFS encoded.
- Therefore, there are 8 qubits per ion string.
- The Quantum 4004 can work with up to 32768 qubit ions in a fault-tolerant way.

Different Types of Quantum Computing Architecture

- SuperConducting
- Ion Trap

Outline

Introduction - Swayanshu

A Quantum von Neumann Architecture - Samrid

Superconducting Quantum Computing Architecture - Samrid

Ion Trap based Quantum Computers - Selvakumar

D-Wave Quantum Computers - Balaviknesh

IBM Quantum Computers - Rishi

Google Quantum Computers - Swayanshu

Conclusion - Balaviknesh

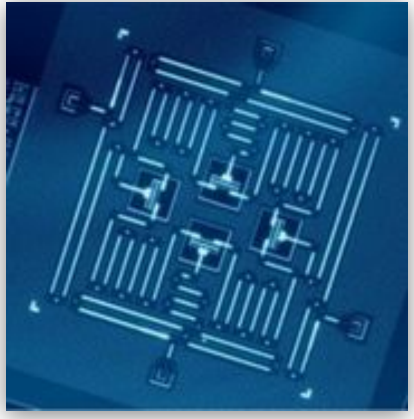
Superconducting Quantum Computing Architecture

Superconducting qubits

- Superconductors are elements, inter-metallic alloys, or compounds that conduct electricity without resistance below a certain temperature.
- Engineers print an artificial "quantum system" in a circuit for the qubits instead of using atoms to store quantum information.
- Semiconductor circuits are built for the qubits.
- Each qubit is an LC circuit, an inductor and a capacitor.
- Its energy state can be manipulated to represent a superposition of $0\rangle$ and$|1\rangle$.
- A microwave tone is sent to the resonator and the signal that is reflected back is analyzed.
- The amplitude and phase of the reflected signal can determine the state of the qubit.

IBM Q Superconducting processor

- The IBM Q, hold superconducting circuits to create a quantum system.
- As the circuitry contains superconductors, it requires extremely low temperatures to operate.
- The square box in the picture holds four superconducting transmon qubits, which are designed to have reduced sensitivity to charge noise.



Left - Inside IBM 50 superconducting qubits quantum computer
Right - enclosing the quantum computer

Pros and Cons

- Used by many industrial giants such as Google, IBM, Intel, D-Wave etc as there is an existing semiconductor industry to rely upon.

- However, they can collapse easily and must be kept very cold.

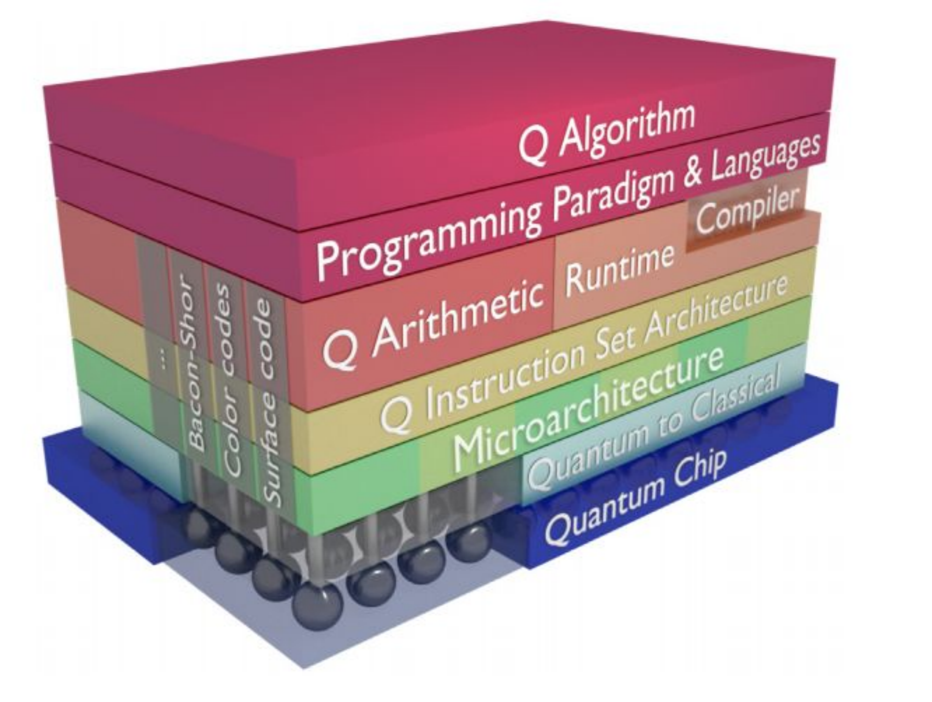
An Experimental Microarchitecture for a Superconducting Quantum Processor

- Research in quantum computing has been focused primarily on,
  - Devising high-level programming languages and compilers to describe and optimize quantum algorithms
  - Or building low-level quantum hardware

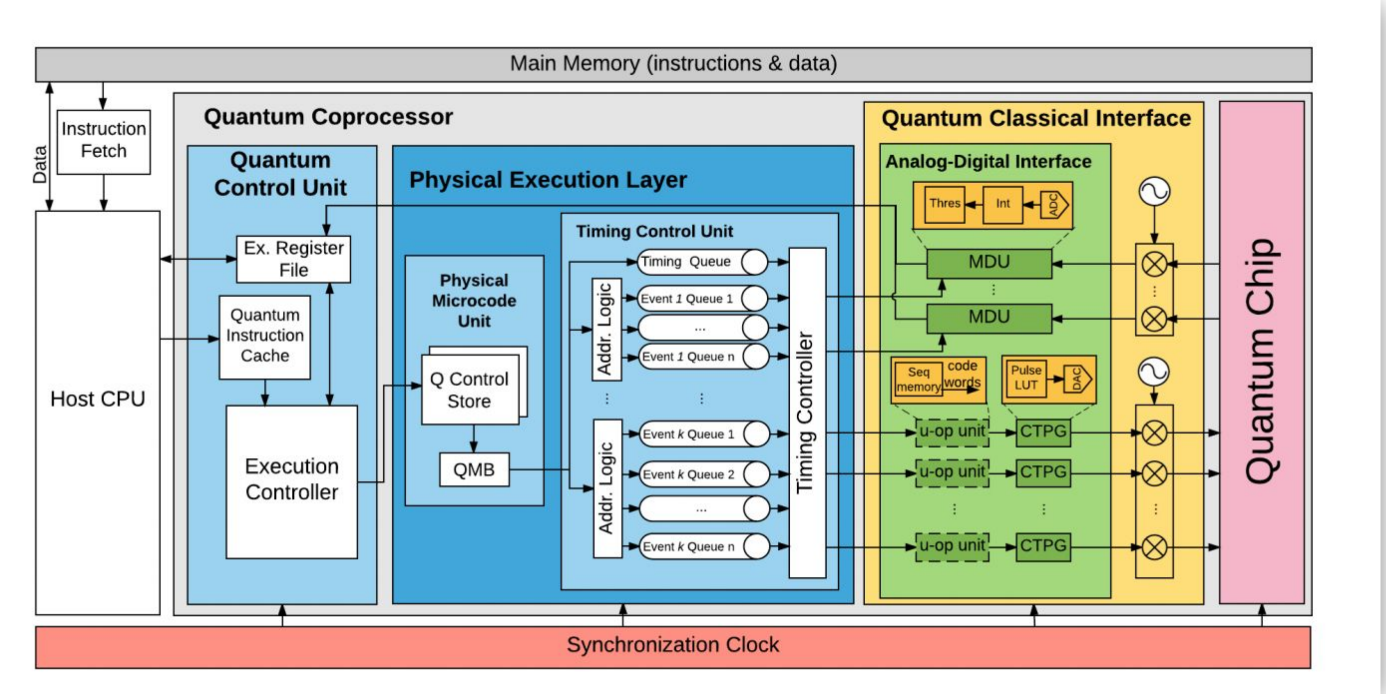
- Relatively little attention has been given to using the compiler output to fully control the operations on experimental quantum processors.

- This paper presents a prototype of a flexible control microarchitecture supporting quantum-classical mixed code for a superconducting quantum processor.

Quantum Computer System Stack



Overview of the Quantum MicroArchitecture



Codeword-Based event control

- The micro-operation unit and the codeword-triggered pulse generation unit translate codeword triggers into pulses representing quantum operations on the qubits with a fixed latency.

- From right to left analog measurement waveforms from the quantum processor are discriminated into binary results by the measurement discrimination unit.

- From right to left analog measurement waveforms from the quantum processor are discriminated into binary results by the measurement discrimination unit.

Queue Based Event Timing Control

- The timing control unit divides the microarchitecture into two timing domains: the non-deterministic timing domain and the deterministic timing domain.

- In the nondeterministic timing domain, the quantum control unit and physical execution layer execute instructions and feed quantum operations to the queues in an as-fast-as-possible fashion.

- In the deterministic timing domain, quantum operations in the queue are emitted to the analog-digital interface with deterministic and precise timing. To this end, queue-based event timing control is introduced.

Multilevel Instruction Decoding

- **Instruction Definition**
    - The quantum code is written with instructions in the Quantum Instruction Set (QIS).
    - QIS contains auxiliary classical instructions and quantum instructions.
    - Auxiliary classical instructions are used for basic arithmetic and logic operations and program flow control.
    - Quantum instructions describe which and when quantum operations will be applied on qubits.
    - By including auxiliary classical instructions, QIS can support feedback control based on measurement results and a hierarchical description of quantum algorithms which can significantly reduce the program.

Multilevel Instruction Decoding

- Instruction Decoding
    - To support a technology-independent quantum instruction set definition, a multilevel instruction decoding approach is adopted.
    - Quantum instructions, especially that for quantum gates, are successively decoded into quantum microinstructions, micro-operations and finally codeword triggers to control codeword-triggered pulse generation to generate pulses.
- Execution Controller
    - This unit executes the auxiliary classical instructions in the QIS and streams quantum instructions to the physical microcode unit.

Multilevel Instruction Decoding

- Physical Microcode Unit
  - Quantum instructions are translated into a sequence of microinstructions in the physical microcode unit based on the microprograms uploaded into the Q control store. The timing for each quantum operation is also determined at this stage.
- Micro-Operation Unit
  - At the micro-operation unit, each micro-operation is translated into a sequence of codeword triggers with predefined latency, which further makes associated codeword triggered pulse generation units generate primitive operation pulses.

Conclusion

- The QuMA processor prototype was successfully implemented on a FPGA.
- The microarchitecture was validated by performing a successful AllXY experiment on a superconducting qubit using a combination of the auxiliary classical instructions and QuMIS instructions which are generated by OpenQL.

Outline

- Introduction - Swayanshu
- A Quantum von Neumann Architecture - Samrid
- Superconducting Quantum Computing Architecture - Samrid
- **Ion Trap based Quantum Computers - Selvakumar**
- D-Wave Quantum Computers - Balaviknesh
- IBM Quantum Computers - Rishi
- Google Quantum Computers - Swayanshu
- Conclusion - Balaviknesh

Ion Trap based Quantum Computers

Ion Trap based Quantum Computers

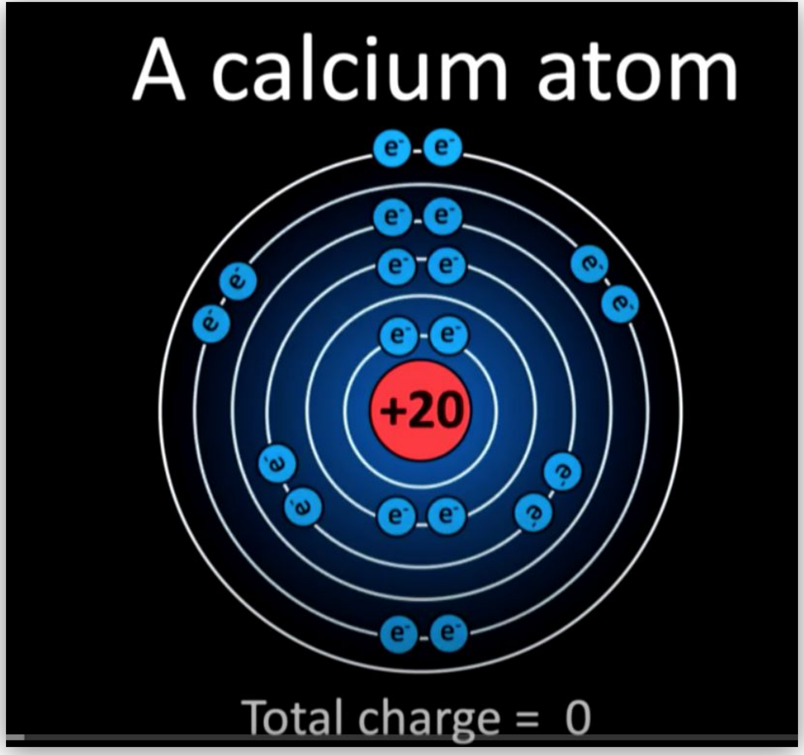
- Ions are just atoms which have lost or gained one or more electrons, thus acquiring an electrical charge(+ve or -ve).

- Ion trap based Quantum computers widely use Calcium, Strontium or Ytterbium ions trapped in an electromagnetic field as Qubits.

- The first demonstration of a universal set of quantum logic gates were performed with trapped ions.

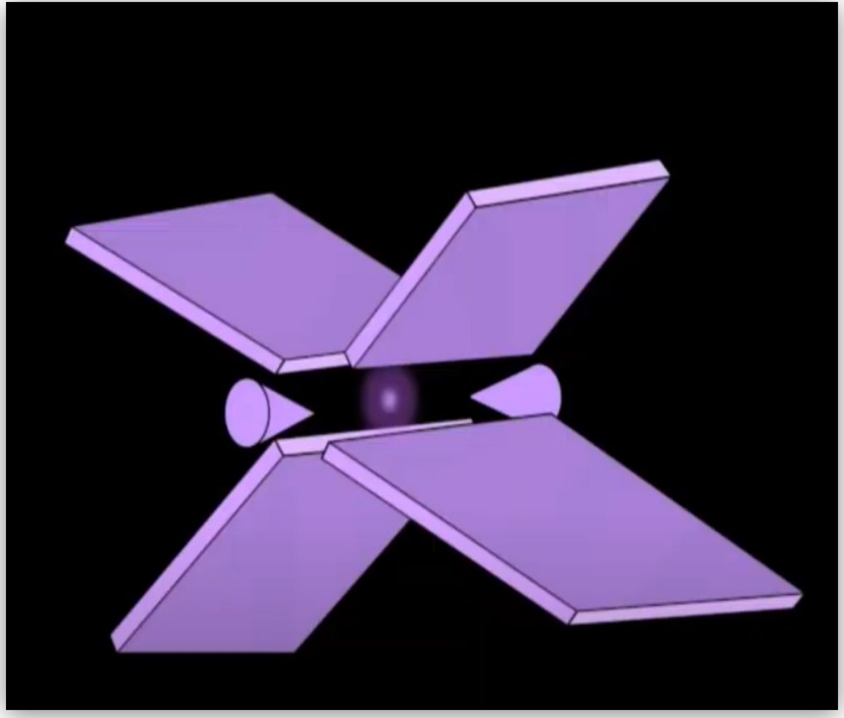
- The invention of the 3D quadrupole ion trap is attributed to **Wolfgang Paul** who shared the Nobel Prize in Physics in 1989 for this work.

Atom Vs Ion



- There are 20 electrons and 20 protons and it is a neutral atom.
- If we use a laser and push a single out electron out, it becomes a calcium ion, and it is positively charged.
- A charged atom is useful because we could use forces of attraction and repulsion to trap it!

Ion Trap - Working

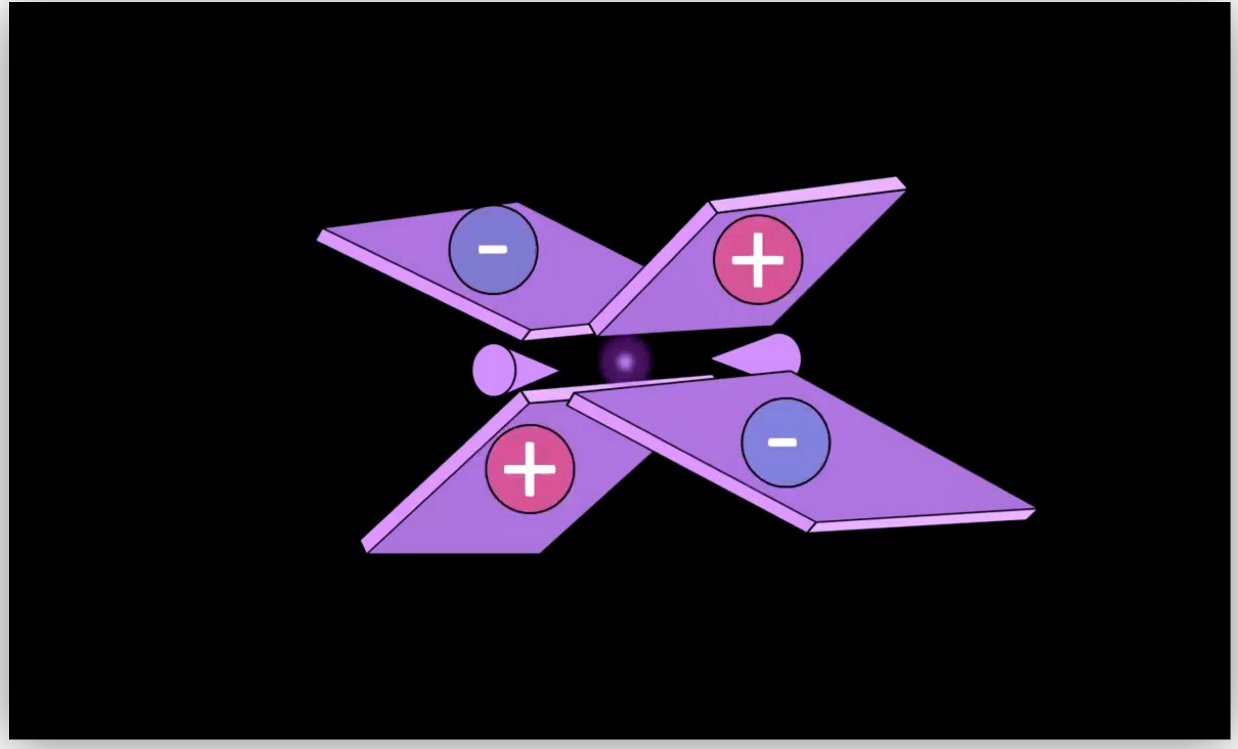


- An electrode is a solid electric conductor that carries electric current into non-metallic solids, or liquids, or gases, or plasmas, or vacuums.

- We surround the ion with two end cap electrodes, and four blade four blade electrodes.

- Then, we flip the charges of the blade electrodes continuously and trap the ion using the forces of attraction and repulsion.

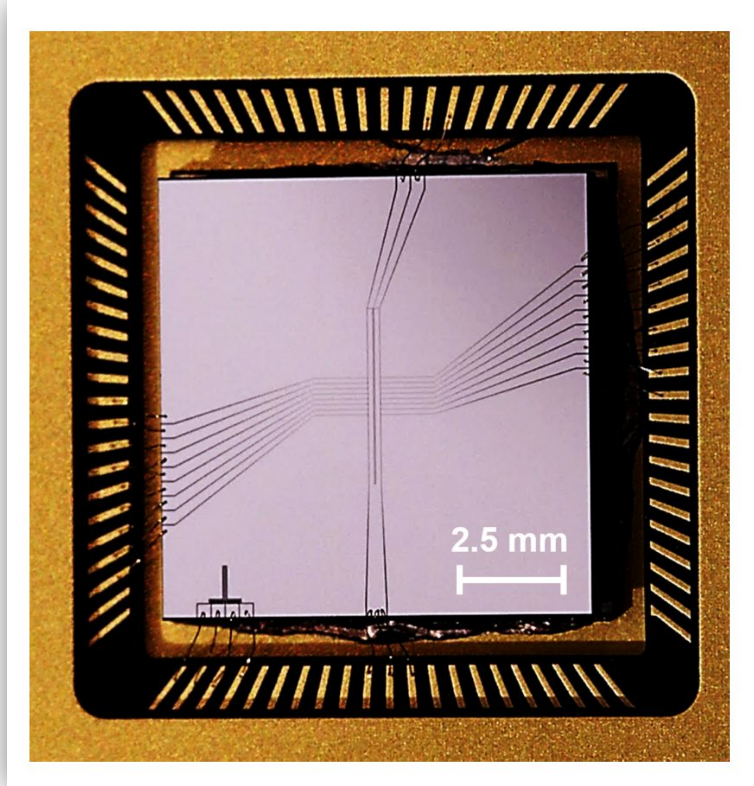
Ion Trap

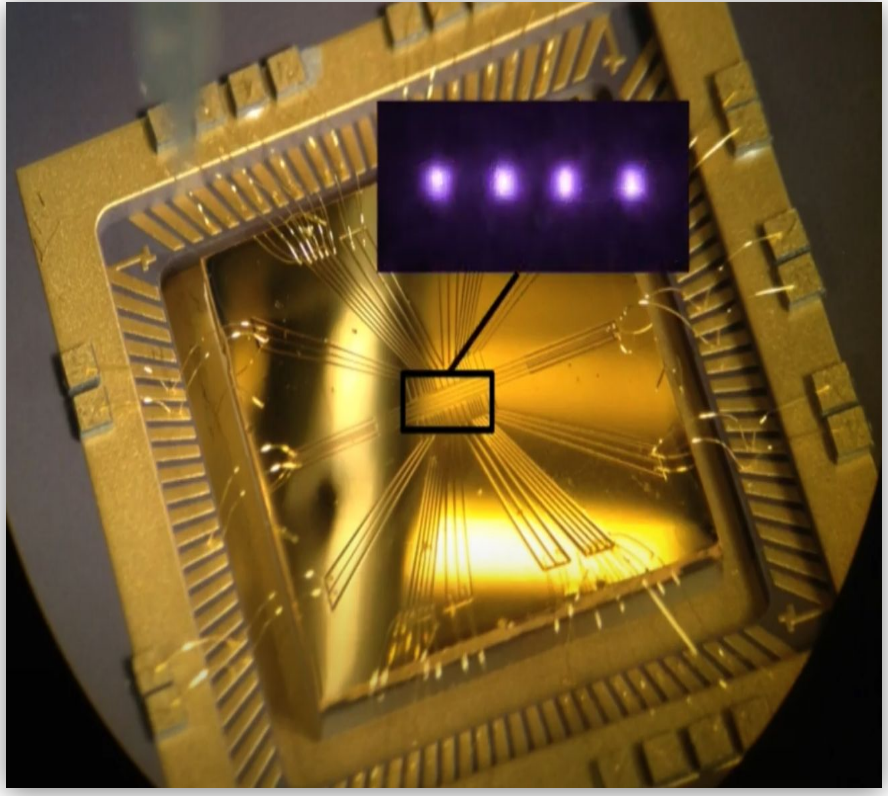


Ion Trap chip

- The flat electrodes can be projected in to chips.
- A single chip could contain many such electrodes setup and thus we can trap more than one ion using a single chip.
- It is extremely feasible to produce such chips on large scale.
- We can then connect such chips and build a large scale quantum processor, where quantum information can be transferred from one chip to another using photonic link system.

Ion Trap chip

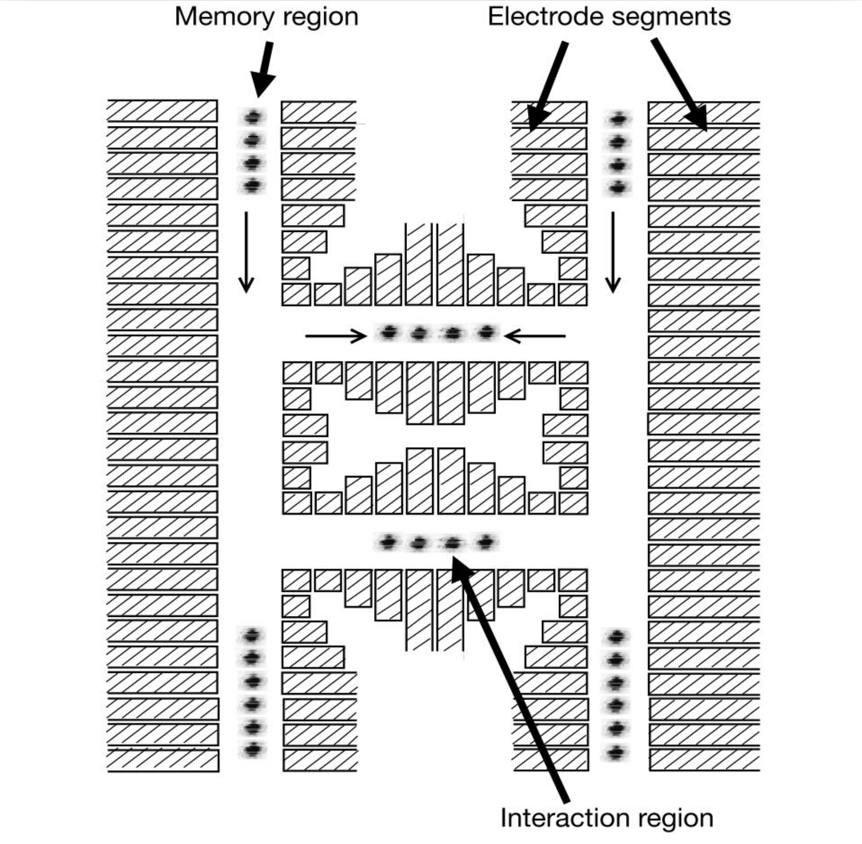




Quantum charge-coupled device (QCCD)

- An architecture consisting of a large number of interconnected ion traps.
- Trapped ions storing quantum information are held in the **memory region**.
- To perform a logic gate, we move the relevant ions into an **interaction region** by applying appropriate voltages to the electrode segments.
- In the interaction region, the ions are held close together, enabling the Coulomb coupling necessary for entangling gates.
- We can realize the trapping and transport potentials needed for the QCCD using a combination of radio-frequency (r.f.) and electric fields.
- Electrode structures for the QCCD are relatively easy to fabricate.

Quantum charge-coupled device (QCCD)



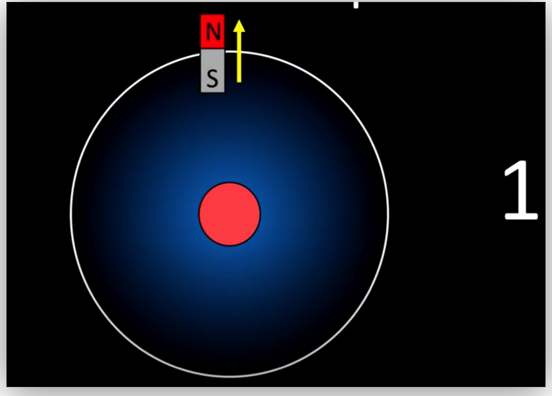
Trapped Ion as Qubit

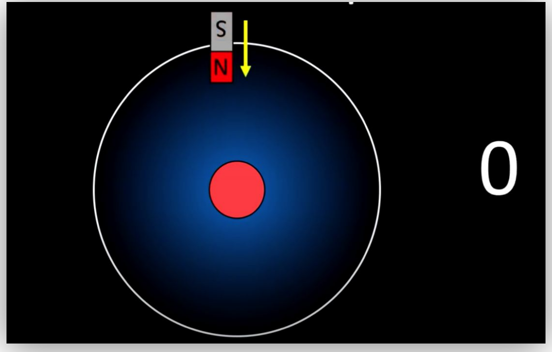
- The outermost electron of the trapped ion has a spin property which we could leverage to use the trapped ion as a Qubit.

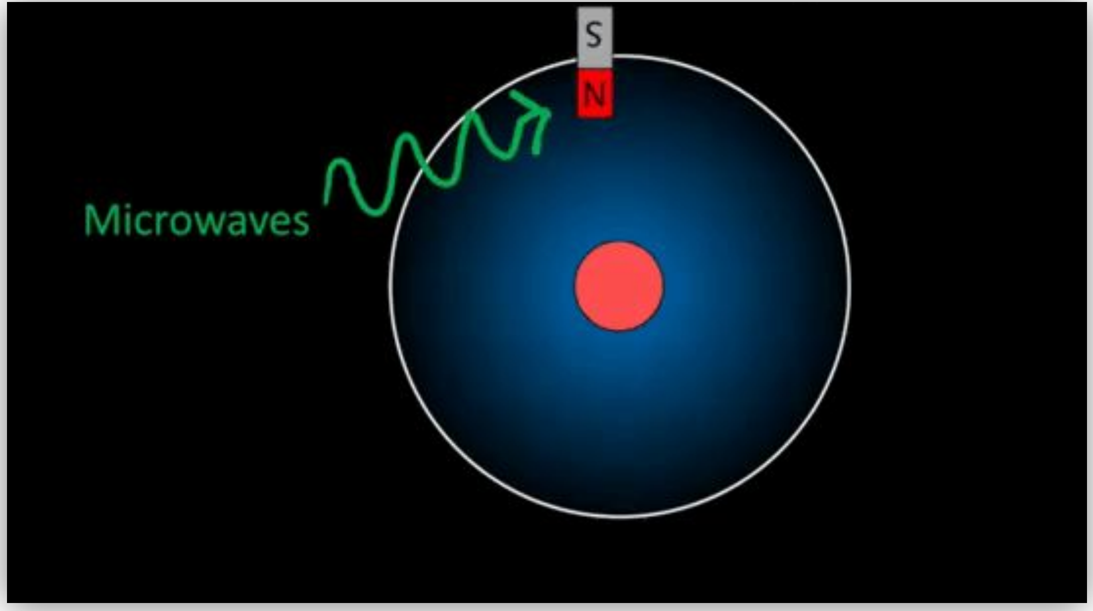
- To understand spin property, ee could think of the outermost electron as a bar magnet
  - If the magnet is pointing up(North-South), then we say it is storing “1”
  - If the magnet is pointing down(South-North), then we say it storing “0”

- To keep the ion in a superposition of both states, we apply microwaves to that electron and rotate the spin electron.

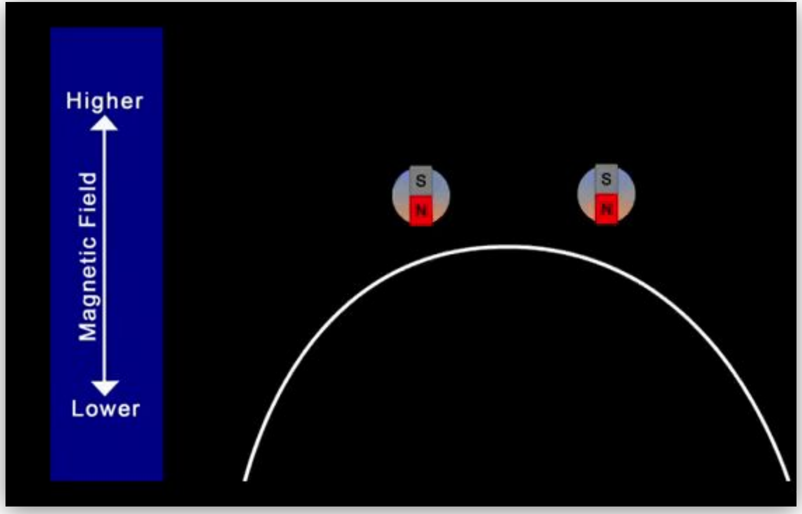
Trapped Ions as Qubit

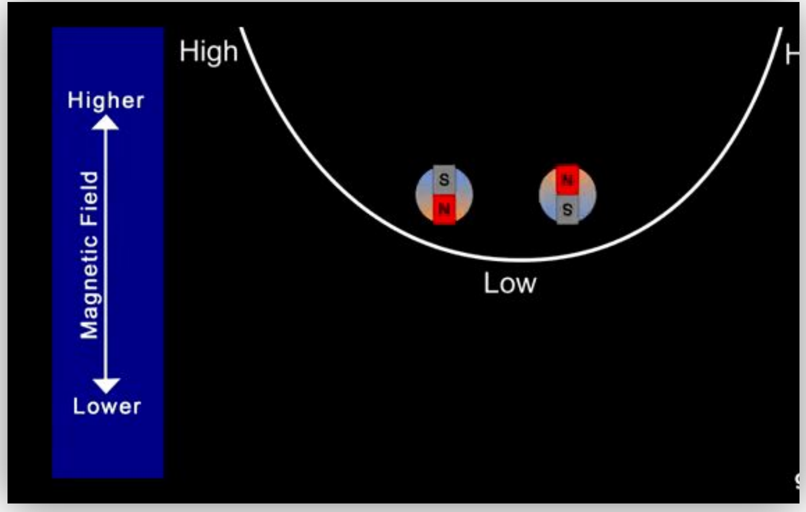




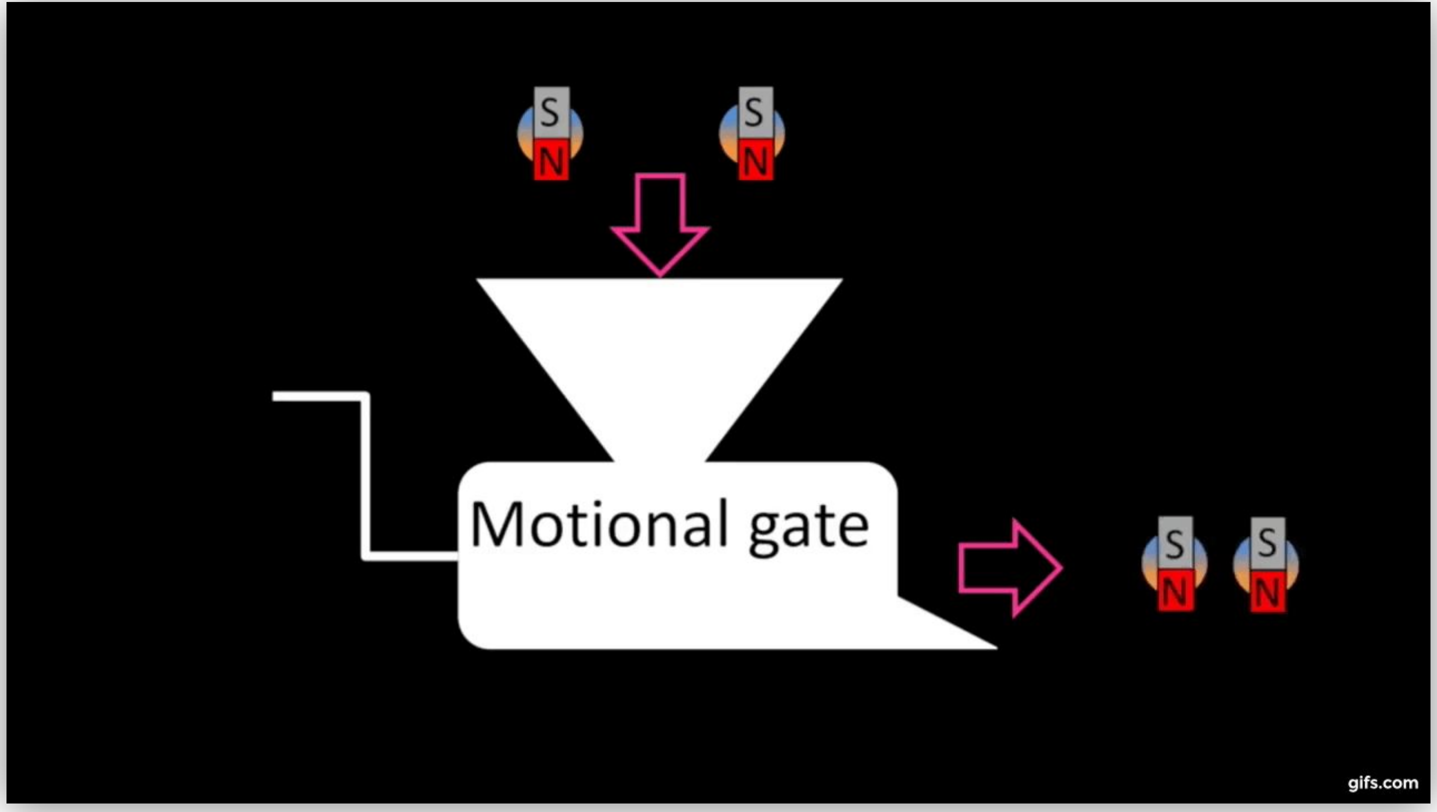


Moving Ions


Higher
Magnetic Field
Lower
S
N
S
N


High
Higher
Magnetic Field
Lower
S
N
N
S
Low

Motional Gate



Error correction in classical computer

- We simply replicate bits
- And then take maximum votes of final received bits to detect and correct the error.

Data sent	Data received	Error detection	Error correction
0 0 0	0 0 0	No	None
0 0 0	0 0 1	Yes	Last bit
0 0 0	0 1 0	Yes	Middle bit
0 0 0	1 0 0	Yes	First bit
1 1 1	1 1 1	No	None
1 1 1	1 1 0	Yes	Last bit
1 1 1	1 0 1	Yes	Middle bit
1 1 1	0 1 1	Yes	First bit

Error correction in Quantum computer

- We cannot replicate superposition states of qubits like we do in classical computing
- We also, cannot measure the quantum bits to know it’s state as it would make the Qubit lose it’s superposition state, and would not be useful for further computation.

Question 1	Question 2	Error correction
Yes	Yes	None
Yes	No	Last bit
No	No	Middle bit
No	Yes	First bit

Error Correction

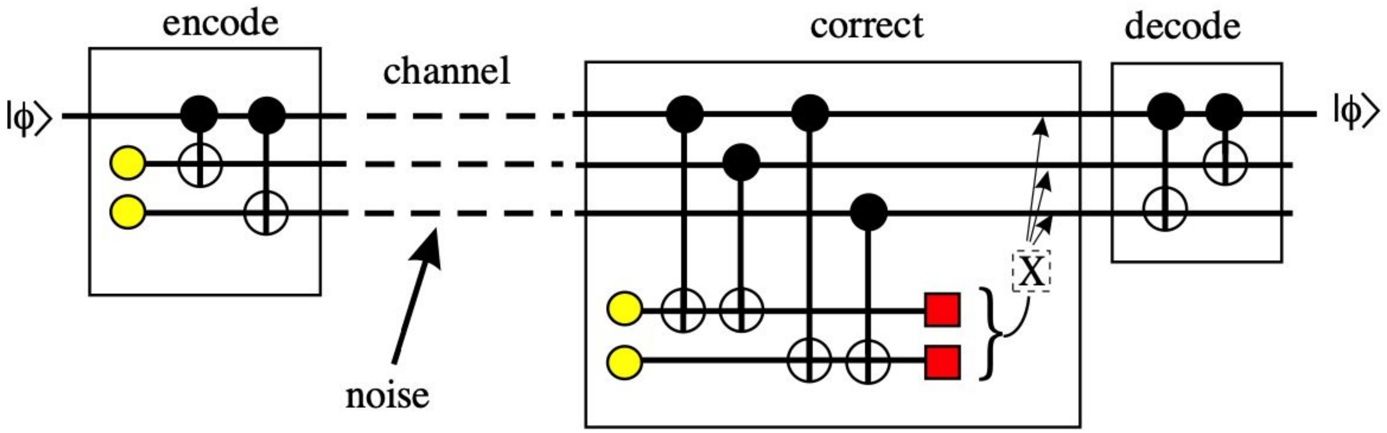
One of the main difficulties in realizing quantum computation is that **decoherence** tends to destroy the information in a superposition of states in a quantum computer, making long computations impossible.

Quantum error-correcting (QEC) codes are among the most promising approaches to deliver scalable and reliable fault-tolerant quantum computation.

Quantum error-correction protocols protect information by encoding it in an entangled state with **ancilla qubits.** In addition to the principal qubits and the quantum operations required for the calculation, further qubits and processing steps are needed to implement the error correction.

The goal is to achieve an acceptable error rate with a reasonable amount of overhead.

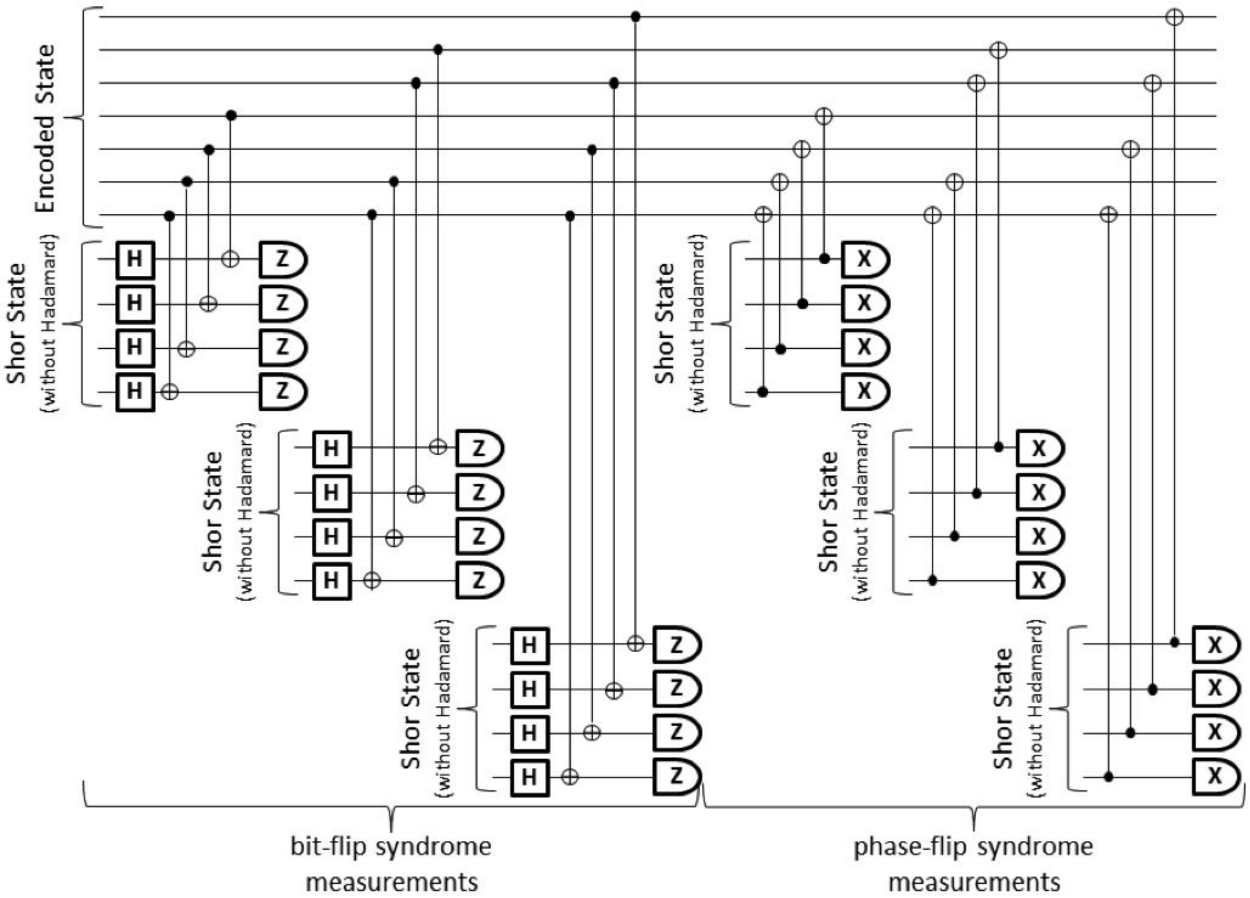
Three bit code



[[n,k,d]] error correction code notation

An $[[n,k,d]]$ code is a quantum error correction code which encodes k qubits in an n-qubit state, in such a way that any operation which maps some encoded state to another encoded state must act on at least d qubits. (So, for example, any encoded state which has been subjected to an error consisting of at most $\lfloor (d-1)/2 \rfloor$ $\lfloor (d - 1) / 2 \rfloor$ Pauli operations can in principle be recovered perfectly).

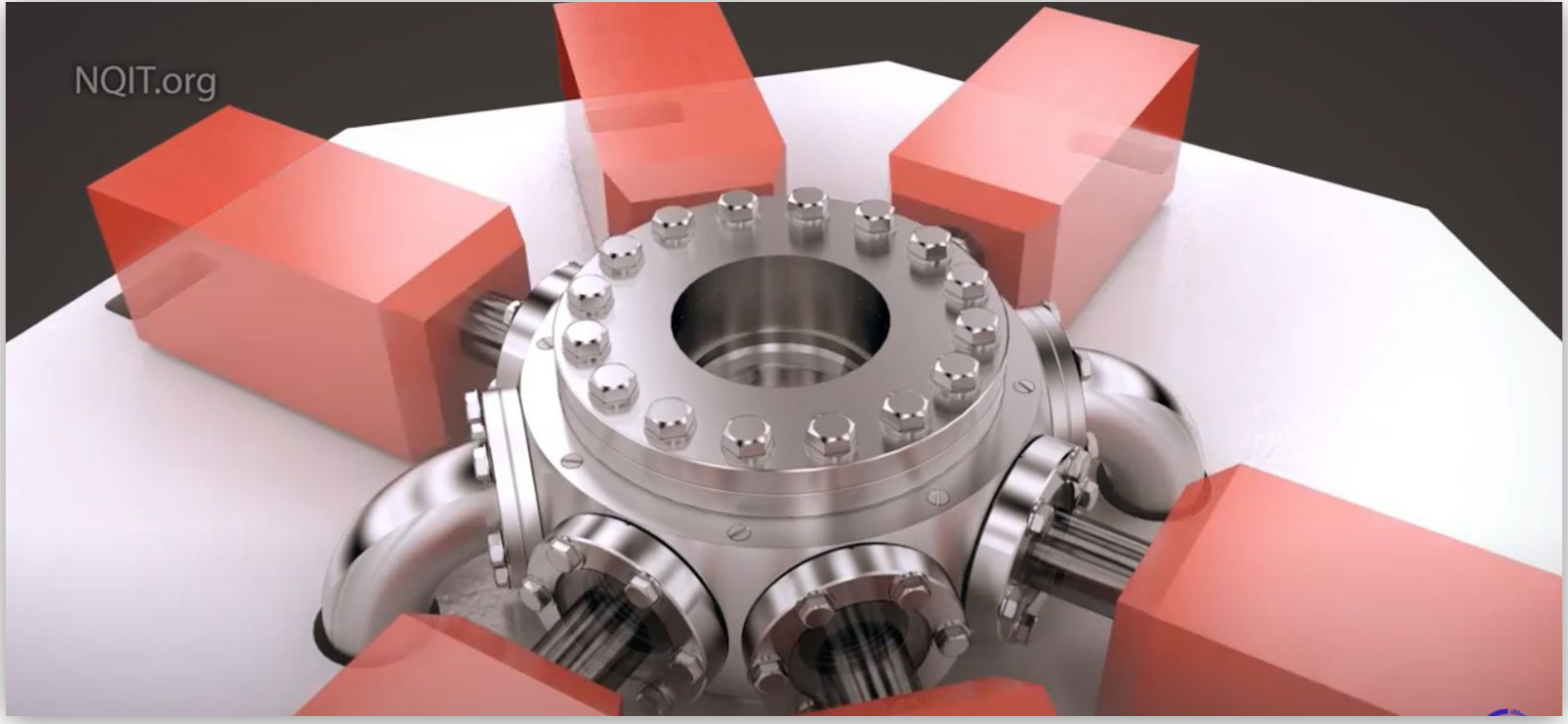
Fault tolerant syndrome measurements for the [[7,1,3]] code using Shor states



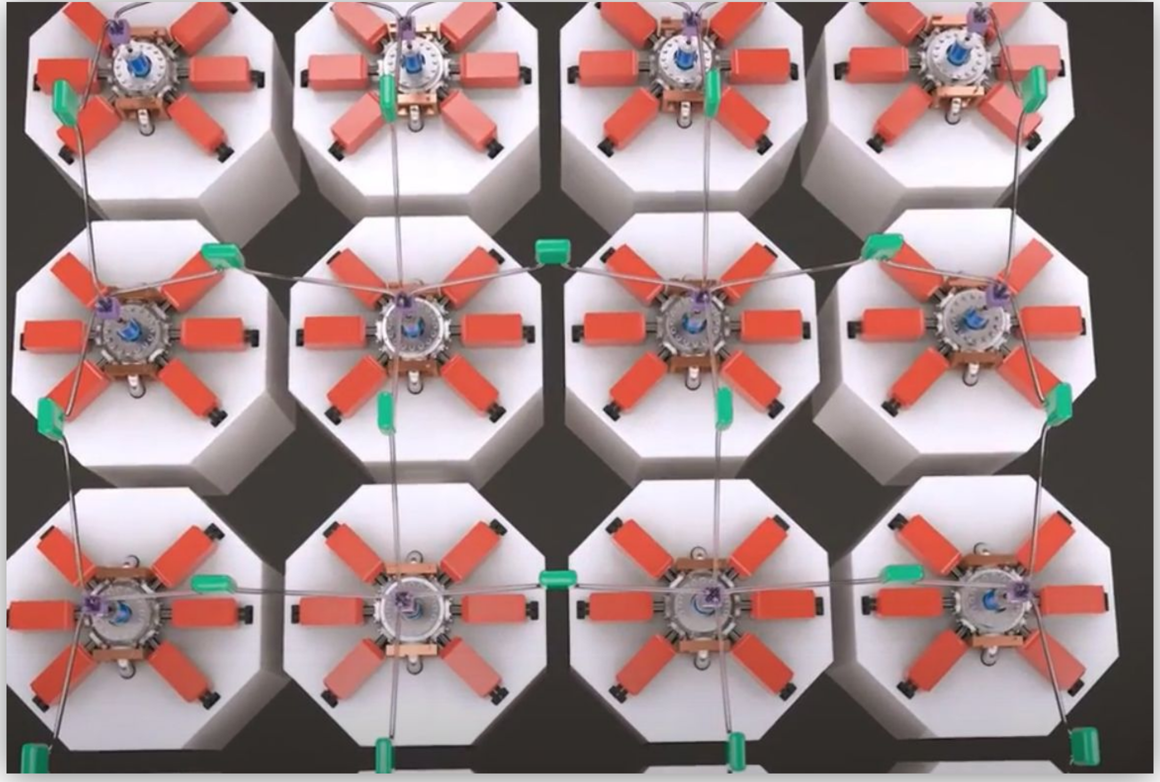
Ion Trap based Quantum processor - Components

- Traps
  - Ring trap - Used in early stages of ion trap researches.
  - Linear trap - The first linear trap held approximately 20 times the number of ions as that of a ring trap
  - Surface trap - The ion trap chip integrated with multiple ion trap arrays is divided into different regions
- Vacuum chamber
- Laser Systems
- Cooling System
  - Doppler Cooling - Two laser sources that are tuned slightly less than the characteristic difference between ground state and first high state.
  - Sideband Cooling - lasers tuned to the characteristic energy difference between the vibrational states and excited states.
- Photonic Link Systems - Entangling multiple quantum processors to form a single quantum system.

Ion Trap based Quantum Processor



Ion Trap based Quantum Processor



Applications

- **Safer airplanes**—Lockheed Martin plans to use its D-Wave to test jet software that is currently too complex for classical computers.
- **Discover distant planets**—Quantum computers will be able to analyze the vast amount of data collected by telescopes and seek out Earth-like planets.
- **Detect cancer earlier**—Computational models will help determine how diseases develop.
- **Help automobiles drive themselves**—Google is using a quantum computer to design software that can distinguish cars from landmarks.
- **Reduce weather-related deaths**—Precision forecasting will give people more time to take cover.
- **Cut back on travel time**—Sophisticated analysis of traffic patterns in the air and on the ground will forestall bottlenecks and snarls.
- **Develop more effective drugs**—By mapping amino acids,for example, or analyzing DNA-sequencing data, doctors will discover and design superior drug-based treatments.

Pros and Cons

- Trapped ions are more stable and have better connectivity to other qubits than their superconducting counterpart.
- Entangling groups of qubits in a shared trap is easy due to the Coulomb force.
- They need less overhead for error correction.
- Slow gate speed
- Difficult to scale lasers.

Outline

- Introduction - Swayanshu
- A Quantum von Neumann Architecture - Samrid
- Superconducting Quantum Computing Architecture - Samrid
- Ion Trap based Quantum Computers - Selvakumar
- **D-Wave Quantum Computers - Balaviknesh**
- IBM Quantum Computers - Rishi
- Google Quantum Computers - Swayanshu
- Conclusion - Balaviknesh

Quantum Computer Architecture in Today's World

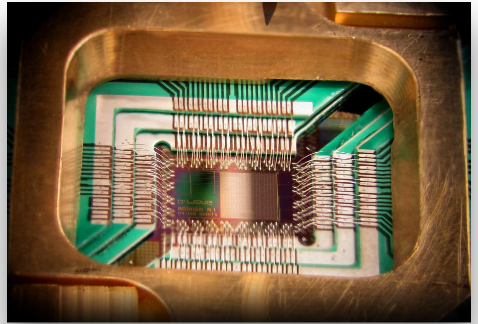
- D-Wave Systems
- IBM
- Google

D-Wave Quantum Computers

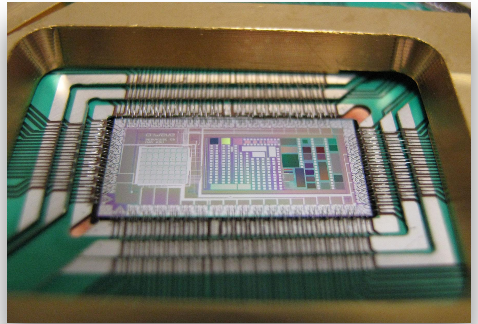
D-Wave

- Evolution
  - D-Wave is a Canadian Quantum computing Company
  - They developed the World's first Commercial Quantum Computer (D-Wave One)
  - It all started from 2011 - **D-Wave One** to upcoming **D-Wave Advantage** - Mid 2020

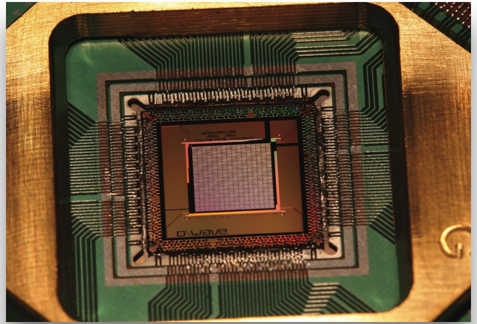
D-Wave Quantum Computers



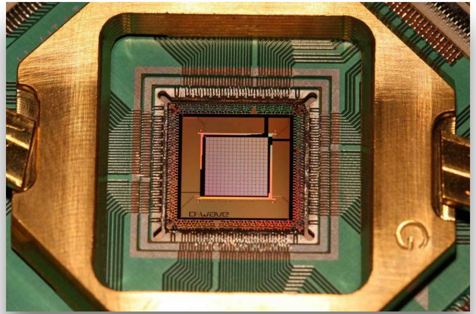
D-Wave One
128 Qubits
Rainer
2011



D-Wave 2x
1152 Qubits
W1K
2015



D-Wave Two
512 Qubits
Vesuvius
2013



D-Wave 2000Q
2048 Qubits
W2K
2017
(Active currently)

How D-Wave QPU Works?

There are many approaches in building the Quantum Computing, conventional way of building is **Quantum Gate** model (similar to classical Logic Gates). But D-Wave approaches through a method called **Quantum Annealing.**

Quantum Annealing

- A way of using the intrinsic effects of quantum physics to solve **Optimization** problems.

- In other words its also called as Energy minimization problem, Where we find the global minimum energy of something in the Quantum world.

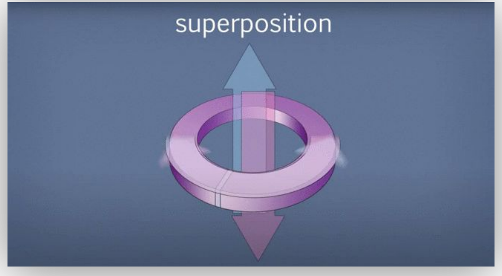
- Example: Finding the cheapest way to travel all the countries in the world with some constraints.

Quantum Annealing Vs Quantum Gates

- It harness the natural evolution of the Quantum States.
- No control over the evolution.
- Setup problem -> quantum states evolves -> Results(Final States)
- Solves Optimization problem

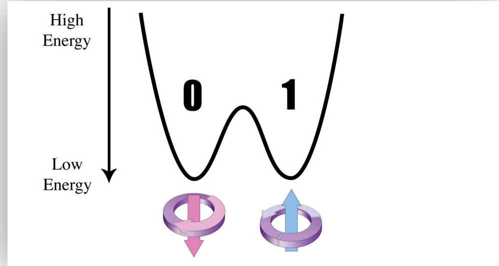
- Able to control/manipulate the evolution of the quantum states over time.
- More difficult and complex
- Solves bigger class of problems

Quantum Annealing in Action

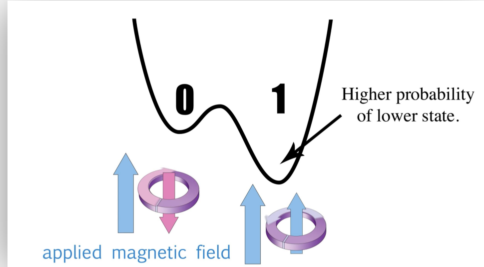


1) Qubit in Superposition State

3) Qubit after QA- either 0 or 1



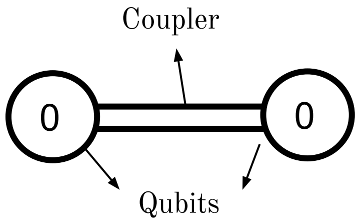
2) Energy Diagram - Superposition State



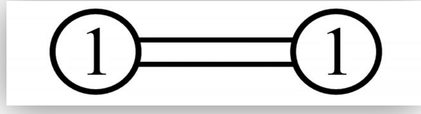
Amount of magnetic field applied on the Qubit is denoted as **Bias**

4) Energy Diagram - After Quantum Annealing

Quantum Annealing in Action

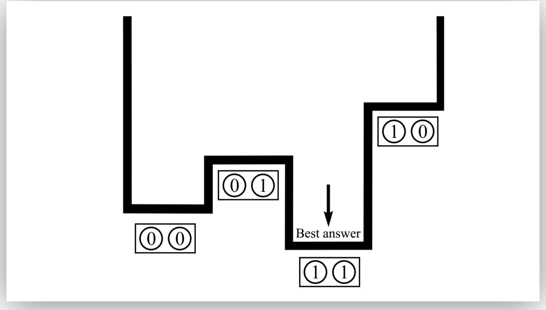


Higher **Coupling strength** - opposite

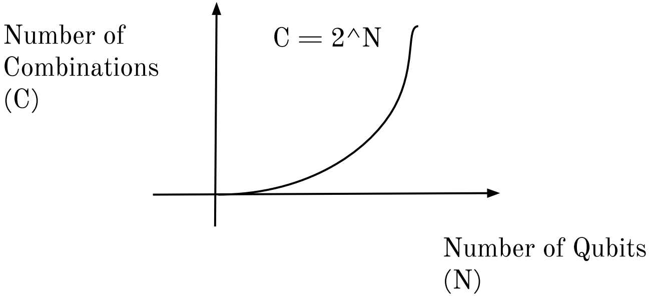


Low **Coupling strength** - Same

Entanglement



Energy diagram for 2 coupled qubits - 4 states (favouring (1,1) by applying **Bias**)



D-Wave 2000Q - QPU Architecture

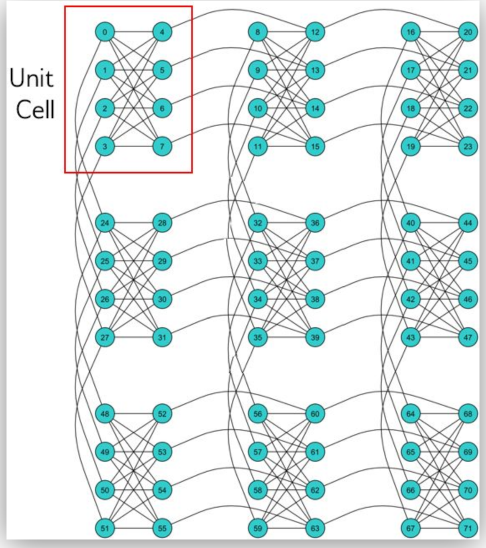
Chimera Topology 2048 Qubits

The D-Wave QPU is a lattice of interconnected qubits.

Qubits are connected to others by couplers

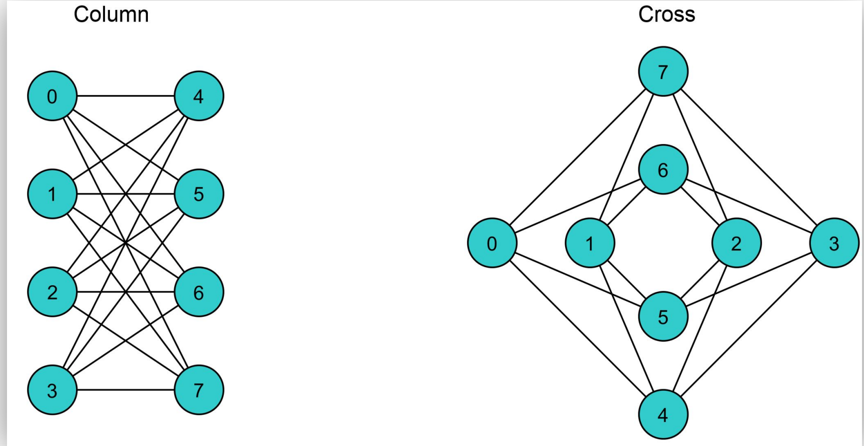
They are not fully connected, but they are interconnected with architecture called **Chimera.**

The D-Wave 2000Q QPU supports a C16 Chimera graph: its 2048 qubits are logically mapped into a **16x16** matrix of **unit cells of 8 qubits**. (16 * 16 * 8 = 2048)

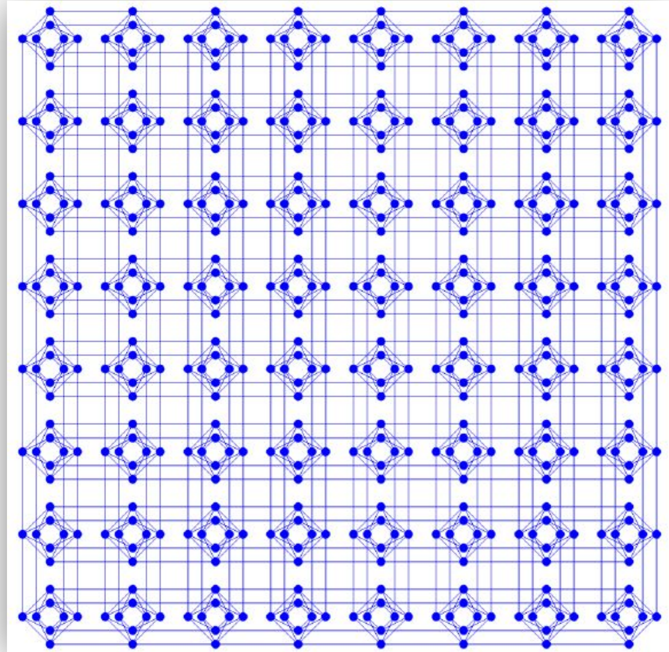


A 3 x 3 Chimera graph of interconnected Qubits. Dentes as C3

Chimera C16



Unit cell in Chimera graph



Complete C16 graph in D-wave 2000Q QPU

Working of QA in C16 QPU Architecture

- D-Wave System are designed to solve the Optimization of Binary variables Problem.
- Binary Variables can only have 0 or 1 / True or False

Simple Problem Representation - XNOR (Exclusive NOR)

0	0	True	Energy should be low
0	1	False	Energy should be high
1	0	False	Energy should be high
1	1	True	Energy should be low

Creating an Objective Function

$$f(s) = b_1 Q_1 + b_2 Q_2 + C_{1,2}\, Q_1 Q_2$$

In an Objective function,

Qubits - Variables {**Q_1,Q_2**}

Bias - Coefficient of Linear terms{**b_1 - bias for Q_1, b_2 - bias for Q_2**}

Coupling Strength - Coefficient of Quadratic terms {**$C_{1,2}$ - Coupling strength btw Q_1 & Q_2**}

To satisfy our goal of the XNOR problem, we need to set the bias and Coupling Strength

Setting Bias and Coupling strength - Obj Func

We have to **favor** the state (0,0) and (1, 1) and **penalize** the state (1,0) and (0,1), to do that lets **apply a bias 0.1** for both Qubits (i.e) **b1, b2 = 0.1**

Q_1	Q_2		Expected	$f(s) = b_1Q_1 + b_2Q_2 + C_{1,2}\ Q_1Q_2$	Applying $C_{1,2} = -0.2$
0	0	True	Min Energy	$(0.1 * 0) + (0.1 * 0) + C_{1,2} * 0 * 0 = 0$	$= 0$
0	1	False	> Min Energy	$(0.1 * 0) + (0.1 * 1) + C_{1,2} * 0 * 1 = 0.1$	$= 0.1$
1	0	False	> Min Energy	$(0.1 * 1) + (0.1 * 0) + C_{1,2} * 1 * 0 = 0.1$	$= 0.1$
1	1	True	Min Energy	$(0.1 * 1) + (0.1 * 1) + C_{1,2} * 1 * 1 = 0.2 + C_{1,2}$	$= 0.2 - 0.2 = 0$

To satisfy our XNOR problem, the final objective function is

$$\boxed{f(s) = 0.1Q_1 + 0.1Q_2 - 0.2Q_1Q_2}$$

Using QUBO for Constraint Problems

QUBO - Quadratic Unconstrained Binary Optimization

Let's consider the Traveling salesman Problem, in this particular Optimization problem, we need some constraints like **the salesperson can be there in Only in one city/place at a time, if not its is invalid. (Exactly 1)**

To solve this we need to create the Objective function, but this time using QUBO , find E(a,b,c) that is min when objective is True

1) $a + b + c = 1 \Rightarrow a + b + c - 1 = 0$

2) $a + b + c - 1 = 0 \Rightarrow (a + b + c - 1)^2 = 0$

3) $E(a,b,c) \Rightarrow 2ab+2ac+2bc-a-b-c+1$ ($a^2 = a$, $a=0$ or 1)

Where Quadratic coefficients - Coupling strength, Linear coefficients are Bias

a	b	c	Exactly1
0	0	0	False
0	0	1	True
0	1	0	True
0	1	1	False
1	0	0	True
1	0	1	False
1	1	0	False
1	1	1	False

Using QUBO for Constraint Problems

After applying the function E(a,b,c) to the table,

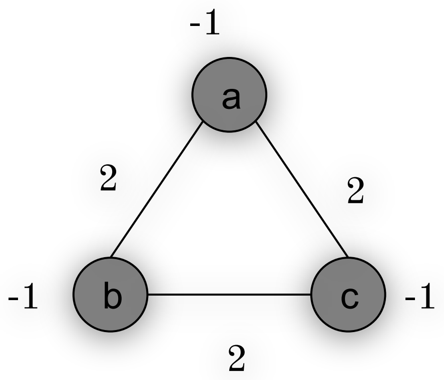
$$E(a,b,c) = 2ab + 2ac + 2bc - a - b - c + 1$$

a	b	c	Exactly1	Energy
0	0	0	False	1
0	0	1	True	0
0	1	0	True	0
0	1	1	False	1
1	0	0	True	0
1	0	1	False	1
1	1	0	False	1
1	1	1	False	4

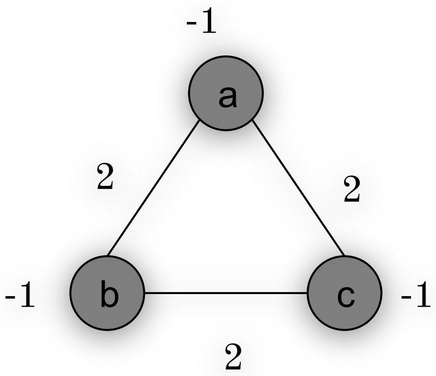
Converting QUBO into a graph

Our QUBO function

$$E(a,b,c)=2ab+2ac+2bc-a-b-c+1$$

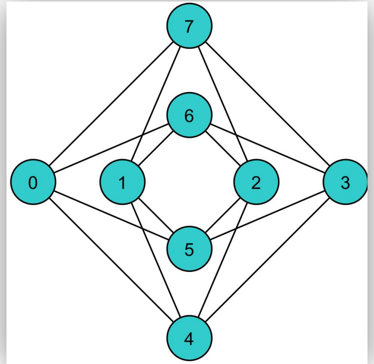


Minor Embedding



Our QUBO graph

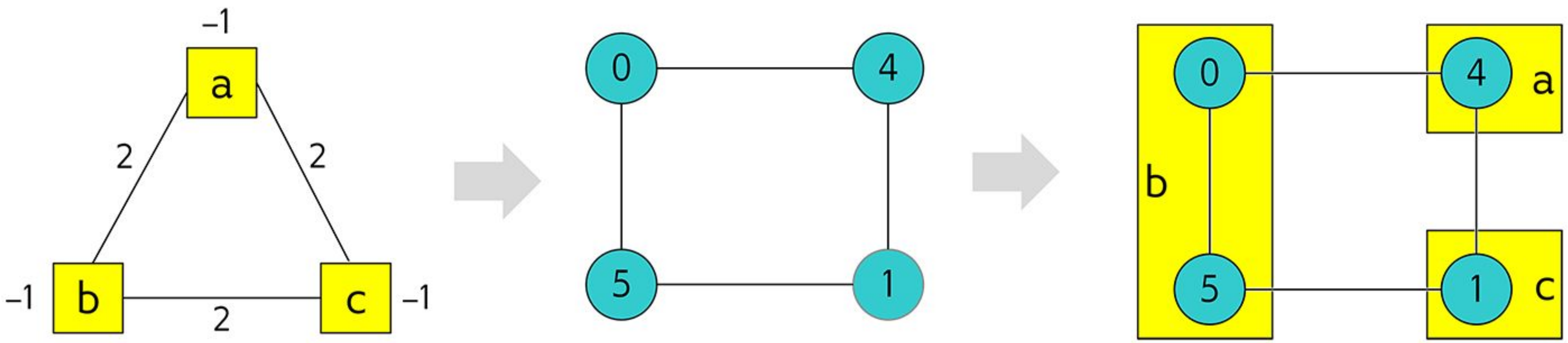
?



A unit cell in Chimera C16 graph

It's clear that we cannot form a triangle

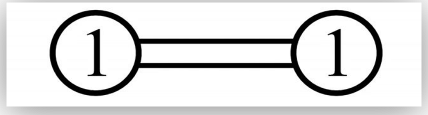
Benefit of Entanglement in Minor Embedding



We know that $E(a,b,c)=2ab+2ac+2bc-a-b-c+1$

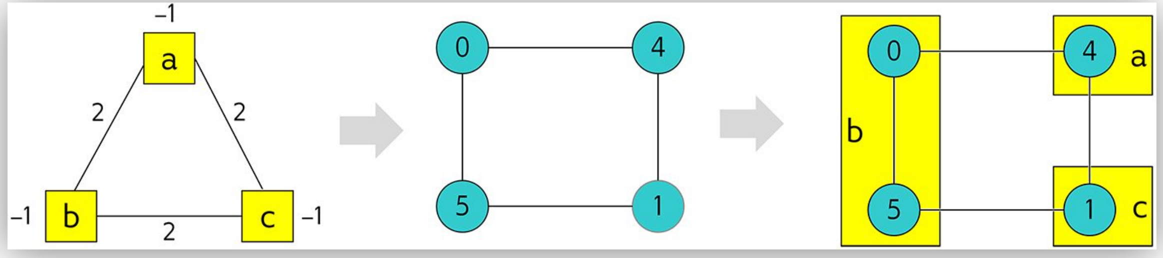
Qubit	0	5	4	1
variable	b	b	a	c
bias	?	?	-1	-1

Coupler	(0,4)	(5,0)	(4,1)	(1,5)
Strength	2	?	2	2



Setting low coupling strength - gives same state

Split and Normalize



- Evenly splitting Qubit 0 and 5 bias => -0.5
- Strong Negative coupling - C(5,0) => - 3
- Add 1.5 to Q0 and Q5 to compensate -3 Coupling Biases => 1
- Normalize everything by dividing with 3 => makes -1 as largest coupler strength

Qubit	**0**	**5**	**4**	**1**
variable	b	b	a	c
bias	0.33	0.33	-0.33	-0.33

Coupler	**(0,4)**	**(5,0)**	**(4,1)**	**(1,5)**
Strength	0.667	-1	0.667	0.667

Results

Possible returned values for our problem are:

$Q0 = 1$
$Q5 = 1$
$Q4 = 0$
$Q1 = 0$

Reverse the embedding process to get the solution to our original problem:

$a = Q4 = 0$
$b = Q0, Q5 = 1$
$c = Q1 = 0$

$(a,b,c)=(0,1,0)$
Other possible solutions are (1,0,0) and (0,0,1)

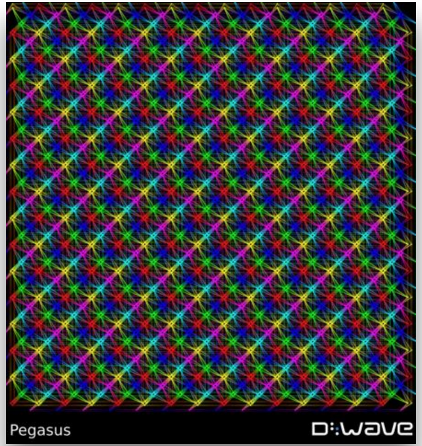
D-Wave Advantage (Mid 2020)

Pegasus P16

Announced on 2019 and planned for production on Mid 2020

Uses the Pegasus Topology

Runs on 5640 Qubits



D-Wave 5th Generation QPU - Pegasus

How is it possible ?

D-Wave

2011 - 128 Qubits

2013 - 512 Qubits

2015 - 1152 Qubits

2017 - 2048 Qubits

2020 - 5640 Qubits (Future Release)

Ten Years from 128 to 5640 Qubits

IBM

2016 - 5 Qubits
2018 - 20 Qubits
2019 - 53 Qubits

Four Years from 5 to 53 Qubits

Google

2017 - 20 Qubits
2018 - 72 Qubits
2019 - 53 Qubits

3 Years from 20 to 72 Qubits

Moore's Law: The principle that the speed and capability of computers can be expected to double every two years, as a result of increases in the number of transistors a microchip can contain.

Summary

- **Quantum Annealing is the reason for high scaling of qubits in D-Wave QPU over the time.**

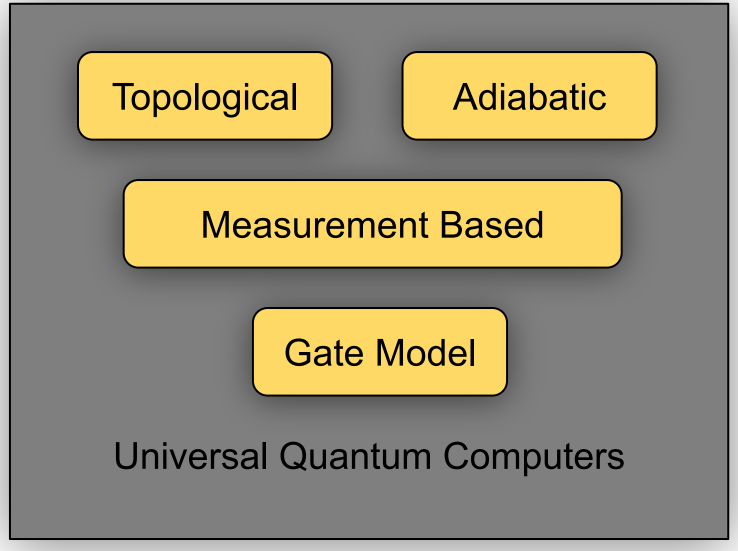
- Google states that the **D-Wave** machine is more than **10^8 times faster** than simulated annealing running on a single core.

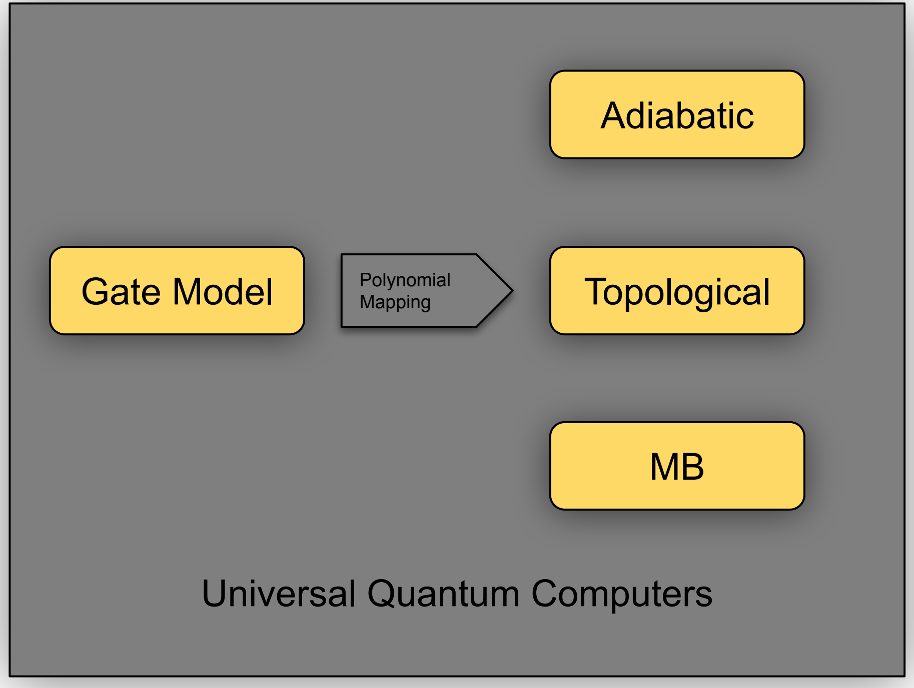
- At the same time **D-Wave cannot run Shor's algorithm** to crack RSA encryption or **Grover's algorithm** for faster search.

- Where as **Google and IBM quantum computers** which is based on Quantum gates, **has the advantage to run Shor’s, Grover’s and nearly 50 other Quantum algorithm** makes them the Universal Quantum computing.

- Also the same advantage of Google and IBM causes problem in designing the chip to improve coherence and Qubit reliability.

Universal Quantum Computing





Outline

Introduction - Swayanshu

A Quantum von Neumann Architecture - Samrid

Superconducting Quantum Computing Architecture - Samrid

Ion Trap based Quantum Computers - Selvakumar

D-Wave Quantum Computers - Balaviknesh

IBM Quantum Computers - Rishi

Google Quantum Computers - Swayanshu

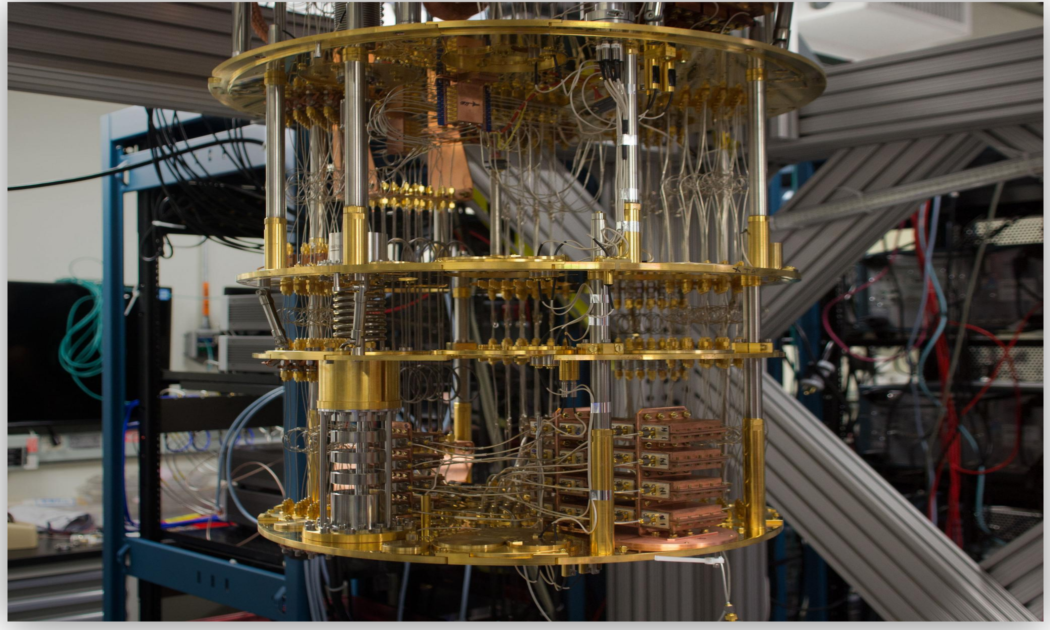
Conclusion - Balaviknesh

IBM Quantum Computers

Evolution

- IBM is a american technology company.
- In 2016, IBM releases quantum experience, an online interface to their superconducting systems.
- In 2017, IBM releases 17-qubit quantum computer.
- In 2017, IBM releases 50-qubit quantum computer.
- In 2019, IBM unveils IBM Q System one.
- In 2019, IBM unveils biggest quantum computer consisting of 53 qubits.

Inside an IBM Q Quantum Computing System



How IBM Q works?

- QPU's are based on quantum circuit and quantum logic gates based model of computing.
- It is based on superconducting quantum computing.

Quantum Circuit

- It is a model used for quantum computation.

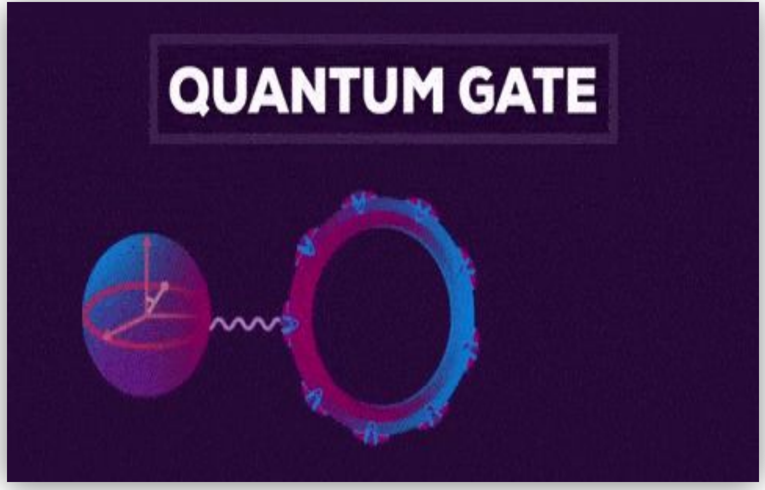
- Computation is a sequence of quantum gates.

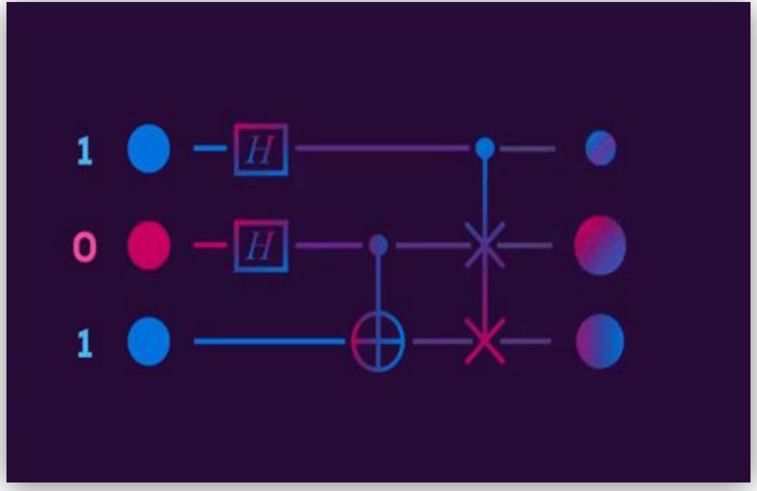
Quantum Logic Gates

- It is a basic quantum circuit operating on a smaller number of qubits.

- Quantum logic gates are reversible.

Quantum gates and Quantum Circuit





IBM: Superconducting Qubit Processor.

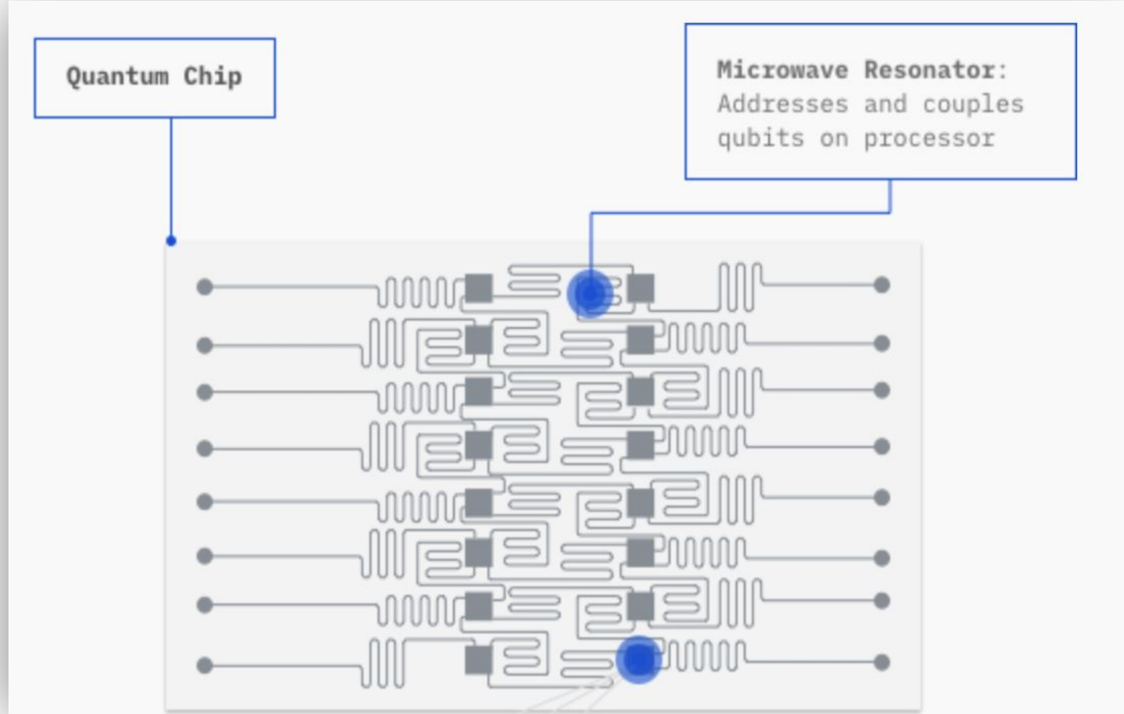
Superconducting Qubit:

- quantum information carrier.
- 70 µs lifetime, 50 MHz clock speed.

Microwave resonator:

- Read out of qubit states.
- Quantum bus, noise filter.

Superconducting Qubit Processor



IBM Qubit Processor Architecture

- IBM Q experience

  * 5 qubits (2016).

  *16 qubits (2017).

- IBM Q commercial

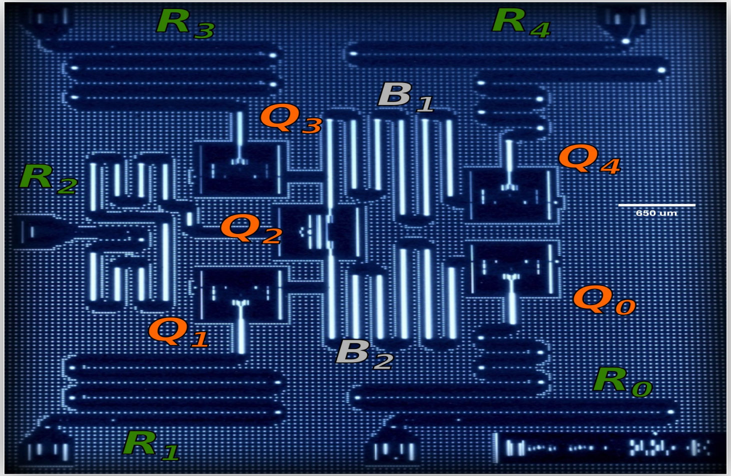
  *20 qubits.

  *50 qubits.

IBM Q5 Tenerife

- Connectivity provided by two coplanar waveguide resonators.

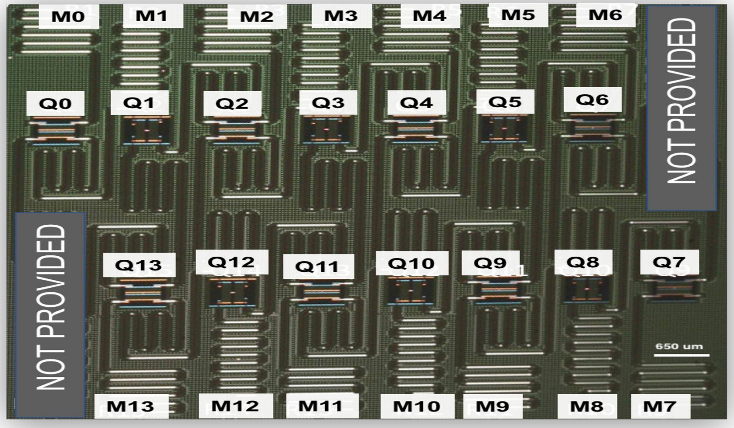
- Each qubit has a dedicated CPW for control and circuit.



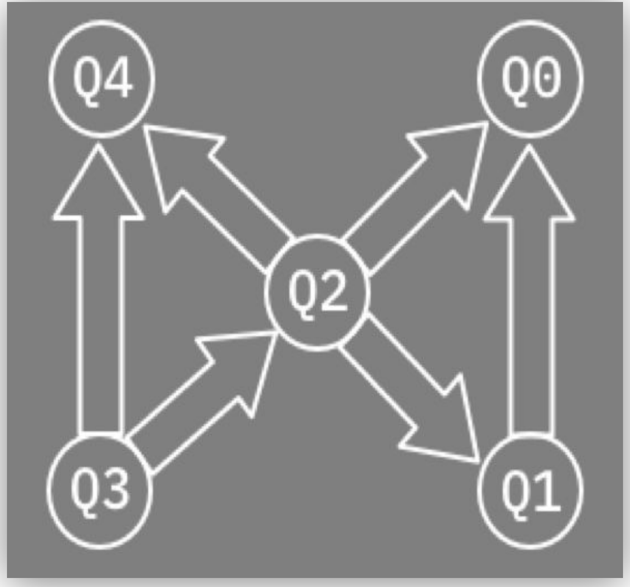
IBM Q16 Melbourne

- Connectivity provided by two coplanar waveguide resonators.

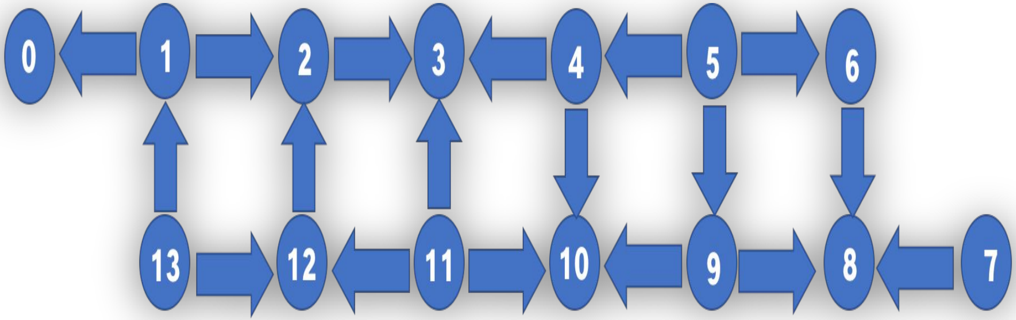
- Each qubit has a dedicated CPW readout resonator for control and circuit.



Layouts of qubits and Connections among them.



IBM Q5 Tenerife.



IBM Q16 Melbourne.

Evaluation of Quantum Computer Architecture.

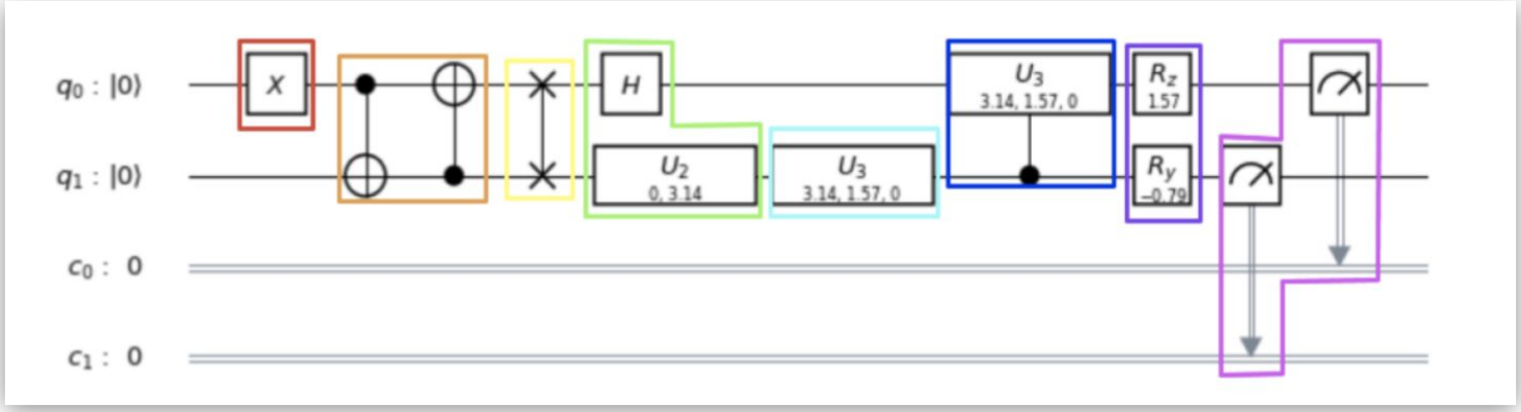
- Studies two data structures to simulate a quantum walk.
- Quantum walk is the quantum analog to the classical random walk.
- Provide polynomial speedup over classical algorithms.

Examples: element distinctness problem, evaluating NAND trees and triangle finding problem.

- First quantised version.
- Second quantised version.
- Compilation and running the circuits.
- Discussion.

Quantum Circuit

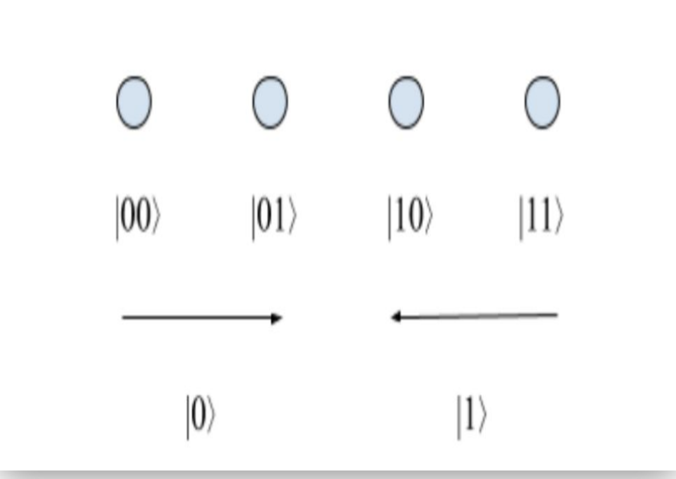
- Built using quantum composer or with the qiskit software framework.



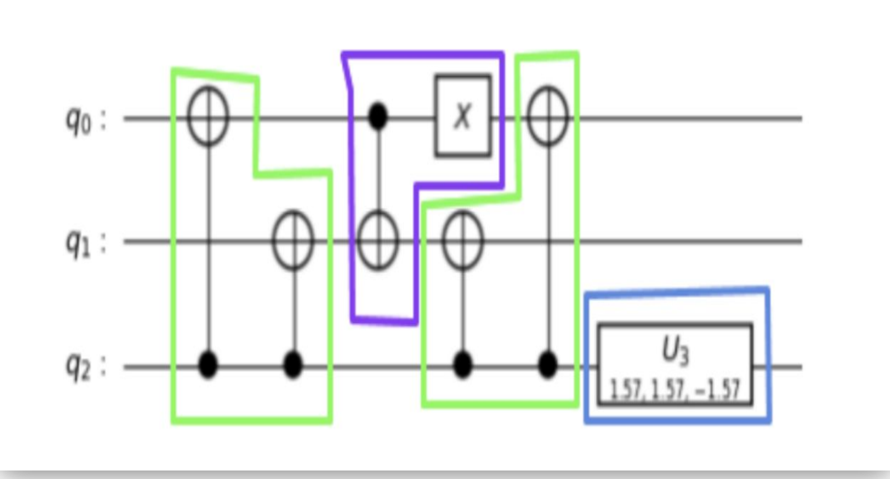
Representation of the quantum circuit.

First quantised version

- Uses $\log_2 N$ qubits to represent the particle position.
- Uses additional qubit to store velocity or direction of the particle.

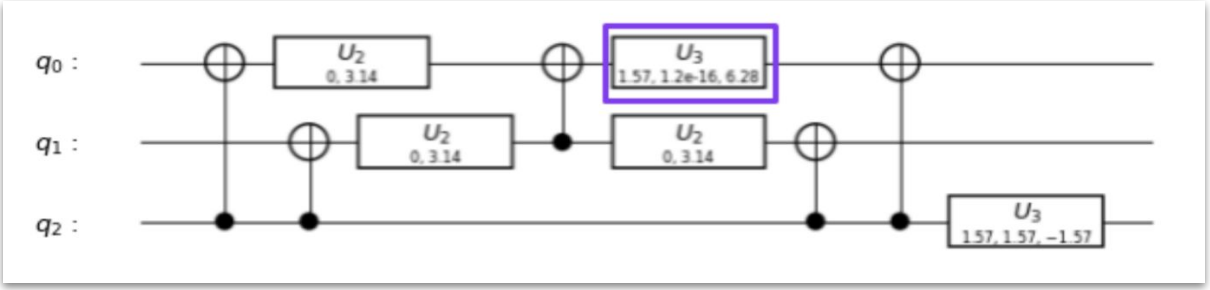


Visualization of first quantized version

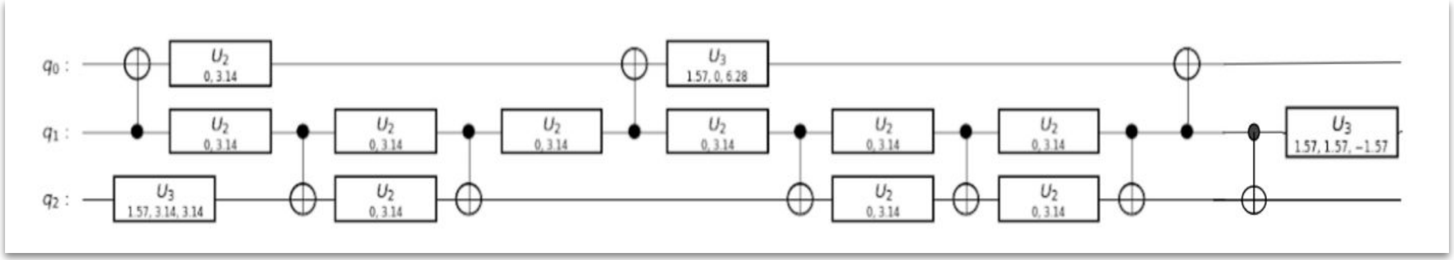


Circuit for one time step of first quantized version

Compiling the first quantized version



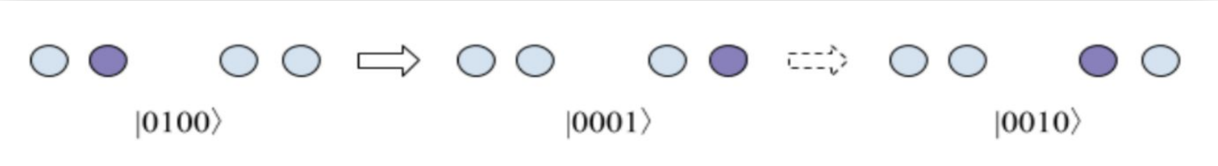
Circuit compiled by the IBM onto ibmqx4.



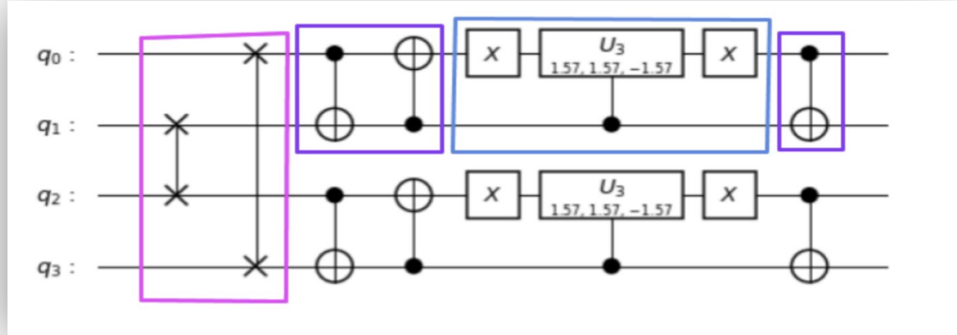
Circuit compiled by the IBM onto ibmqx16.

Second quantised version

- Uses 2N qubits organised into pairs.
- Each pair represents single site.
- Particle position within the pair indicates the direction.

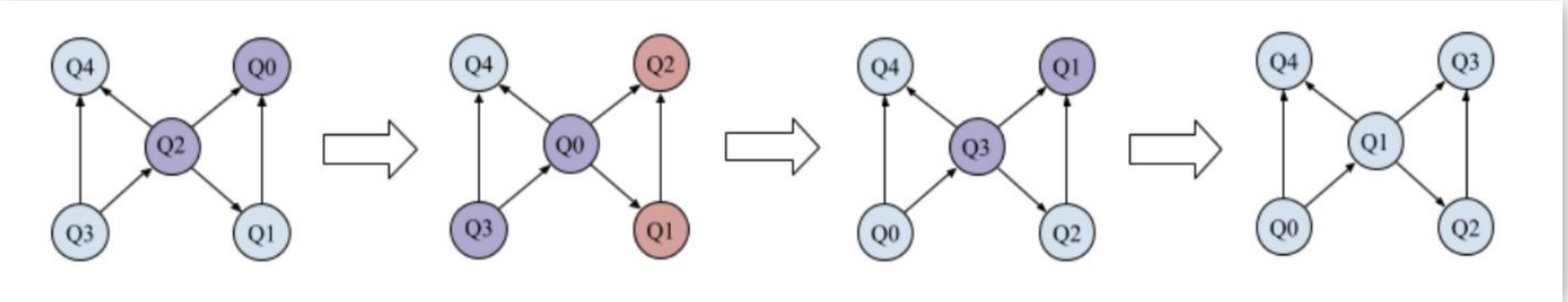


Visualization of second quantised version.

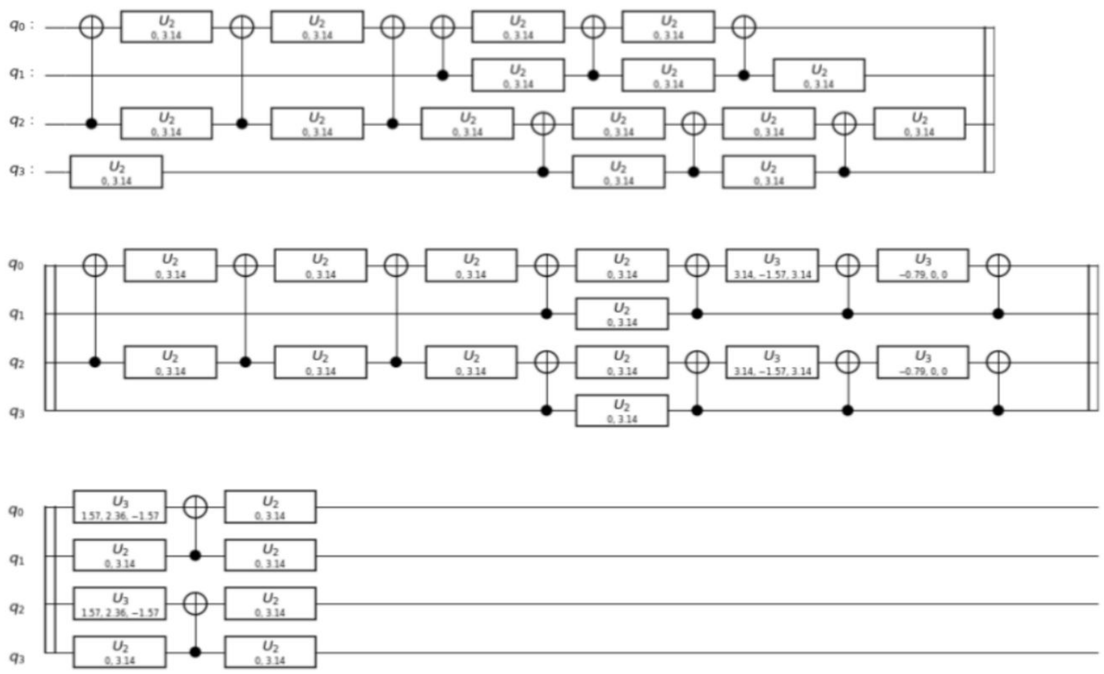


Circuit for one time step of second quantized version

Compiling the second quantized version



Steps required to swap qubits q0 with q3 and q1 with q2.



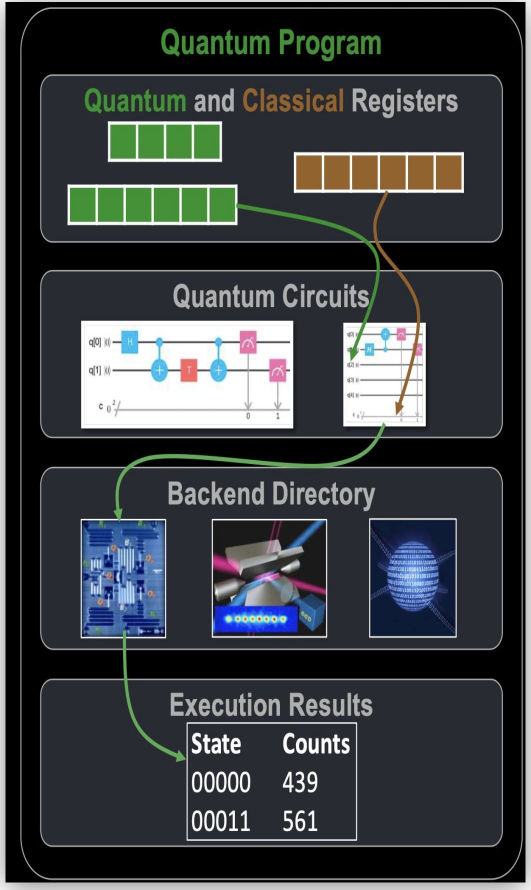
Circuit compiled by the IBM onto ibmqx4.

QISkit: Basic Workflow.

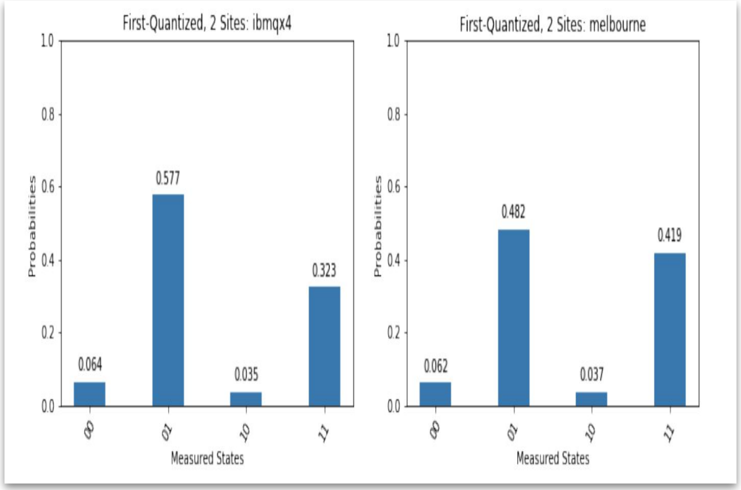
*Building quantum circuits.

*Compiling quantum circuits.

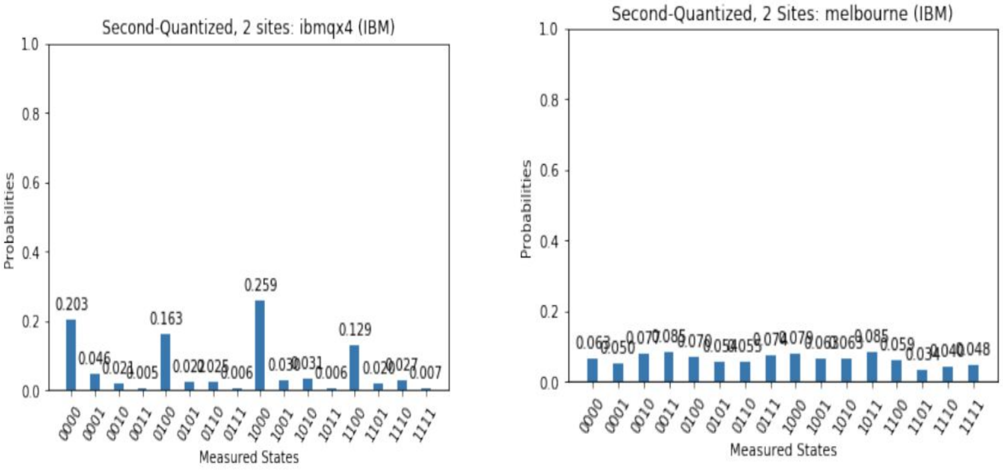
*Executing quantum circuits.



Running the circuits



Distribution of measured states for first quantized version.



Distribution of measured states for second quantized version.

Discussion's

ℓ1 Distances between Run Distributions and Expected Distributions with the ℓ1 Distance of the Uniform Distribution as Reference								
# Shots	Compiler	First Quantized				Second Quantized		
		2 Sites		4 Sites		2 Sites		4 Sites
		ibmqx4	melbourne	ibmqx4	melbourne	ibmqx4	melbourne	melbourne
1024	IBM	0.162±0.017	0.119±0.025	0.207±0.013	0.420±0.014	0.588±0.008	0.871±0.008	0.993±0.001
	Clara			0.313±0.011	0.786±0.014	0.551±0.012	0.746±0.017	0.988±0.003
2056	IBM	0.165±0.013	0.098±0.002	0.202±0.013	0.302±0.005	0.572±0.020	0.839±0.002	0.990±0.002
	Clara			0.312±0.008	0.762±0.036	0.553±0.005	0.746±0.014	0.985±0.005
5000	IBM	0.161±0.007	0.092±0.007	0.189±0.004	0.441±0.006	0.568±0.009	0.839±0.005	0.985±0.002
	Clara			0.310±0.008	0.673±0.029	0.555±0.007	0.787±0.008	0.987±0.001
Uniform Distribution		0.5		0.75		0.875		0.992

Grover’s Algorithm

* Background.
* Algorithm.
* Implementation on IBM’s 5-qubit computer.

Background

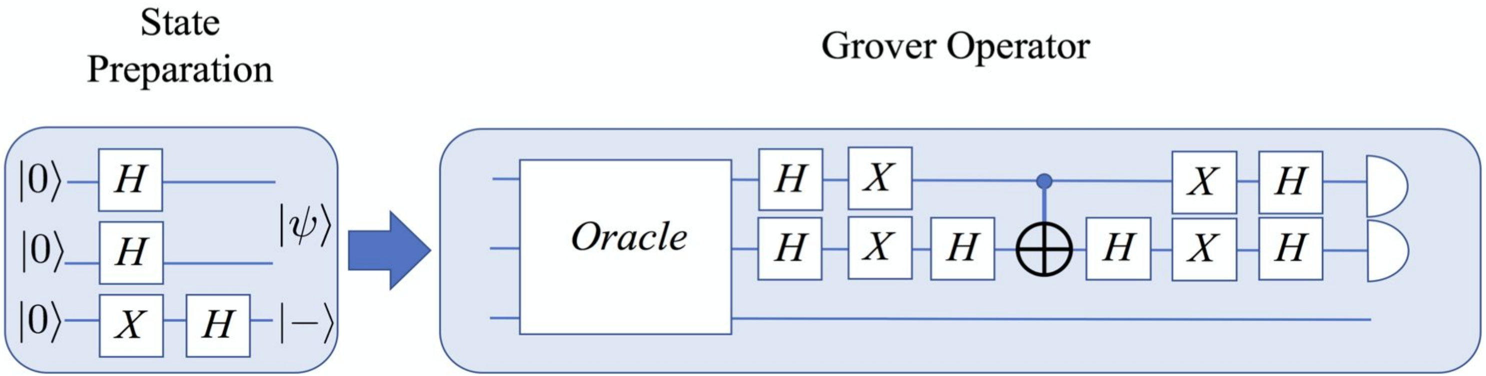
- Enables to find a specific item within a randomly ordered database of N items.
- Uses $O(\sqrt{N})$ operations.
- Provides a quadratic speedup over classical algorithm.
- Grover's operator- key piece of machinery in Grover's algorithm.

$G = (2|\psi|\rangle \langle\psi| - I)O$, ψ- Uniform superposition of all states. O- Oracle operator. Operator applied $(\pi\sqrt{N}/4)$ times.

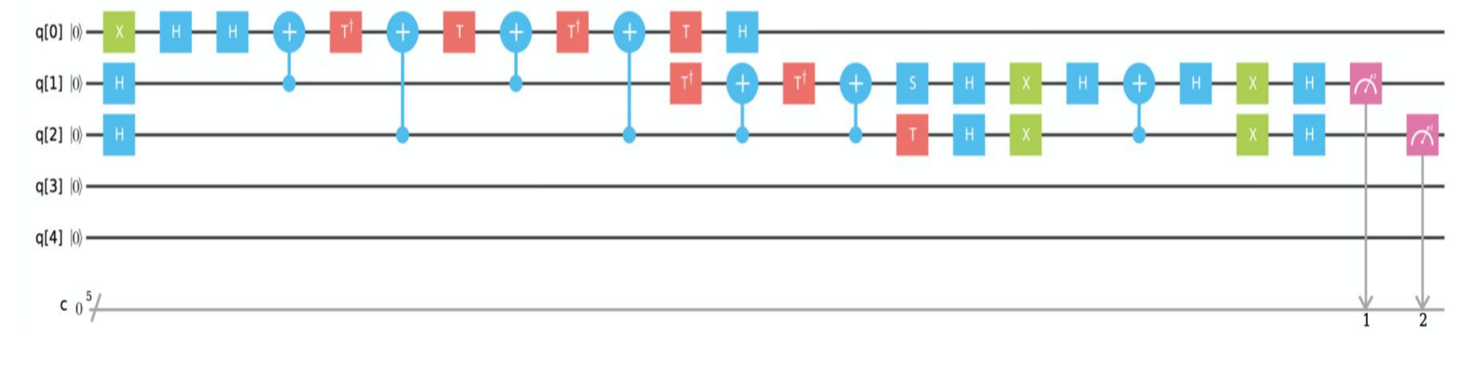
Algorithm Description

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}^* \\ 0 & \text{if } \mathbf{x} \neq \mathbf{x}^* \end{cases}$$

Behaviour of oracle: $|\mathbf{x}\rangle\,|q\rangle \rightarrow |\mathbf{x}\rangle\,|f(\mathbf{x}) \oplus q\rangle.$



Algorithm Implemented on IBM 5- qubit computer



Conclusion

- Quantum computer successfully completed the search with 65% success rate.
- Deeper and complex oracles provide less satisfactory results.

Conclusion

- IBM Q is used in the field of machine learning, simulating quantum materials, optimization of applications and financial sector.
- Provides cloud based software.
- Computation is not below the fault tolerant threshold.
- Operating the computer’s at very low temperatures.
- Challenge of programming a computer entirely different to the current systems.

Outline

Introduction - Swayanshu

A Quantum von Neumann Architecture - Samrid

Superconducting Quantum Computing Architecture - Samrid

Ion Trap based Quantum Computers - Selvakumar

D-Wave Quantum Computers - Balaviknesh

IBM Quantum Computers - Rishi

Google Quantum Computers - Swayanshu

Conclusion - Balaviknesh

Google Quantum Computers

Google Quantum Computers

Introduction - Google Quantum Computer - Bristlecone - Their Work

Architecture

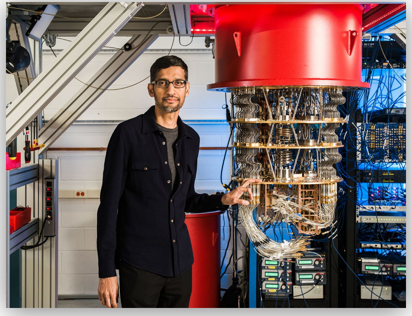
Shor's Algorithm - Circuit diagram - Qutip

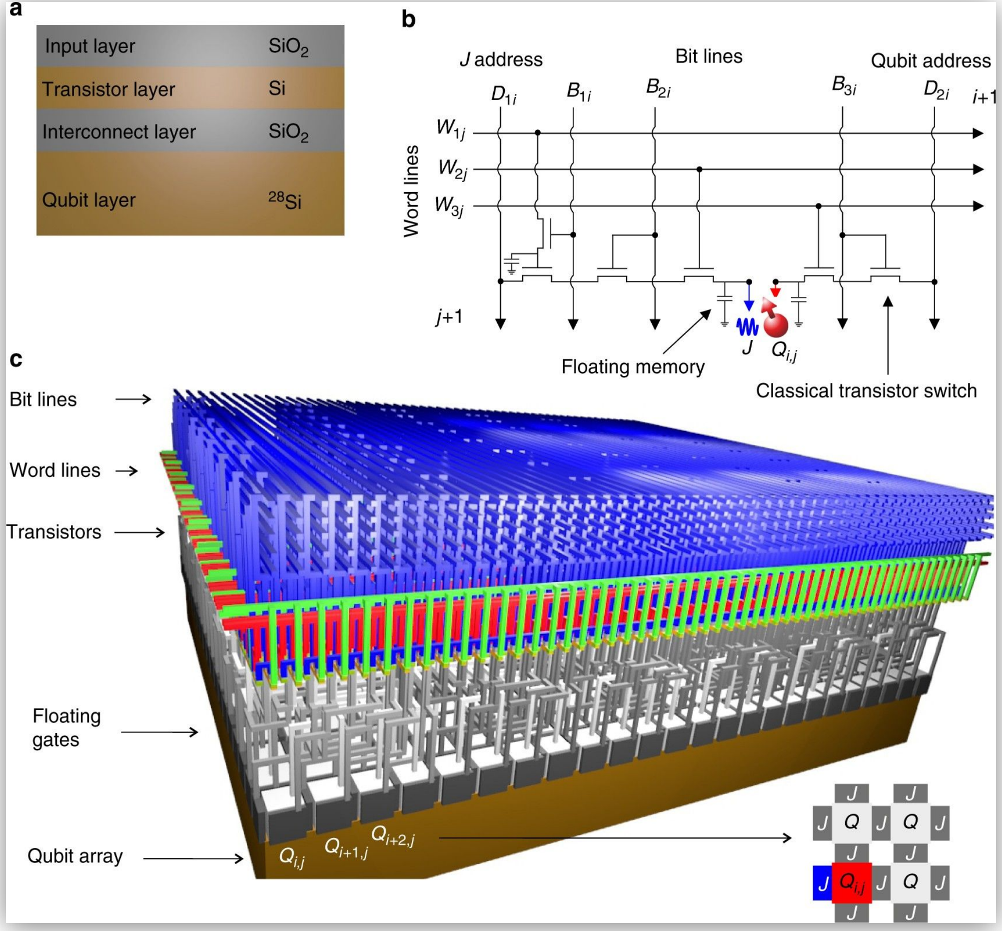
Quantum Supremacy

Architecture

They displayed the hardware Architecture for quantum Computers. Detail architectural overview is Still on Testing. Only 2 Quantum processors they explained i.e.

1. Sycamore
2. Bristlecone





How to create a Qubit?

from sympy.physics.quantum.qubit import Qubit

>>> Qubit(0,0,0)

o/p - |000⟩

o/p - |000⟩

>>> q = Qubit('0101')

>>> q

o/p - |0101⟩

Represent a state and then go back to its qubit form:

>>> from sympy.physics.quantum.qubit import matrix_to_qubit, Qubit

>>> from sympy.physics.quantum.gate import Z

>>> from sympy.physics.quantum.represent import represent

>>> q = Qubit('01')

>>> matrix_to_qubit(represent(q))

o/p - |01⟩

Wigner function

This function allows one to study the classical limit, offering a comparison of the classical and quantum dynamics in phase space.

This is the code used to create a coherent state.

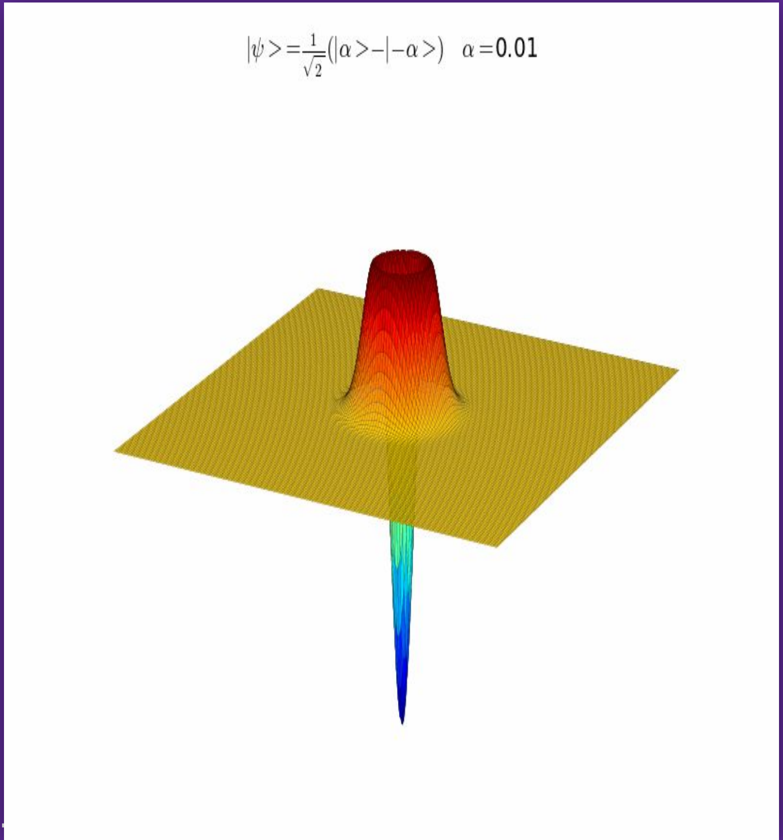
```
N= 20 #Size of Hilbert Space

alpha= 3 #amplitude of the coherent state

D = displace(N,alpha) # Displacement operator

vac= basis(N,0).unit() # Vacuum state

psi=(D*vac).unit() # create a coherent state. The unit() ensures normalization.
```



Google Quantum Computers - QUTIP

Quantum tool box in python: http://qutip.org/tutorials.html

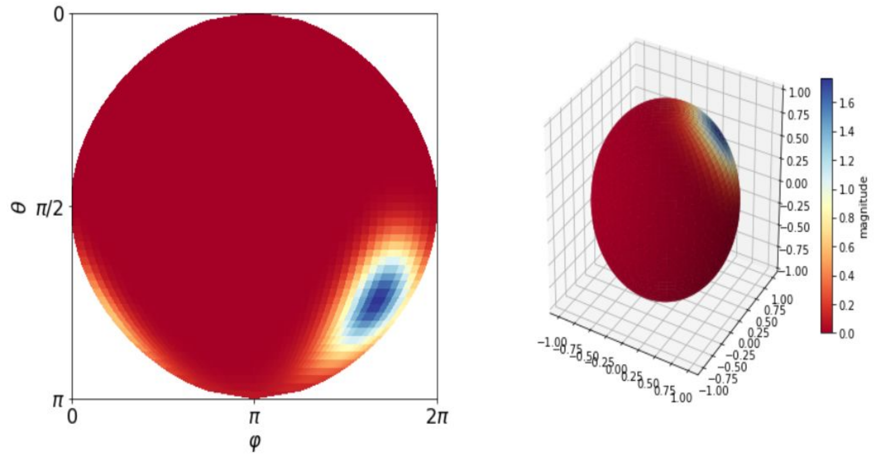
Ex- Plot spin-Wigner functions

```
W, THETA, PHI = spin_wigner(psi, theta, phi)
```

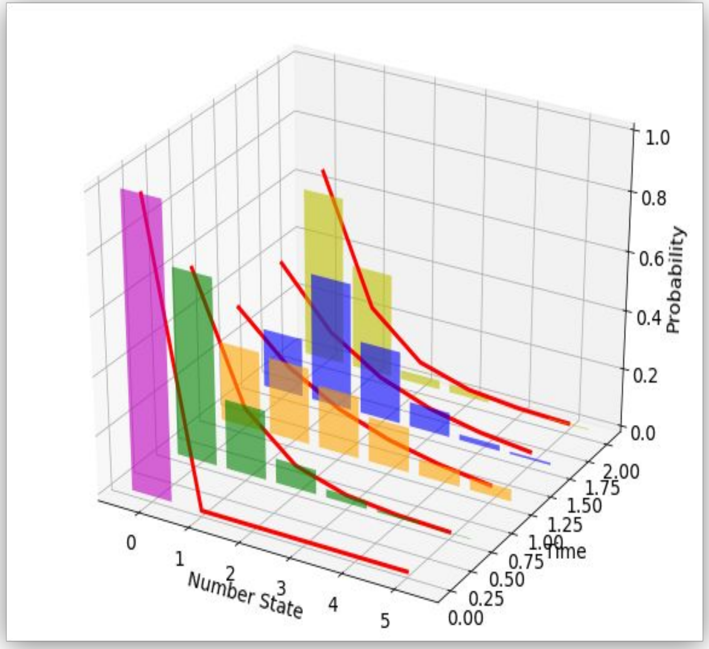
```
fig = plt.figure(figsize=(14,6))

ax = fig.add_subplot(1, 2, 1)
f1, a1 = plot_spin_distribution_2d(W.real, THETA, PHI, fig=fig, ax=ax)

ax = fig.add_subplot(1, 2, 2, projection='3d')
f2, a2 = plot_spin_distribution_3d(W.real, THETA, PHI, fig=fig, ax=ax)
```



Tri-linear Probability Visualization



Shor’s algorithm

Motivation:

Highly important for cryptography, as it can factor large numbers much faster than classical algorithms (polynomial instead of exponential)

Use: It is a Quantum algorithm, to find the prime factors of any given integer N.

When observed or measured, the particle will manifest itself in one observable state or the other with respective probability encoded in the superposition.

Any state of the system is modeled by a unit-length vector in the Hilbert space C2n

Ex-

$7^0 \bmod 15 = 1$

$7^1 \bmod 15 = 7$

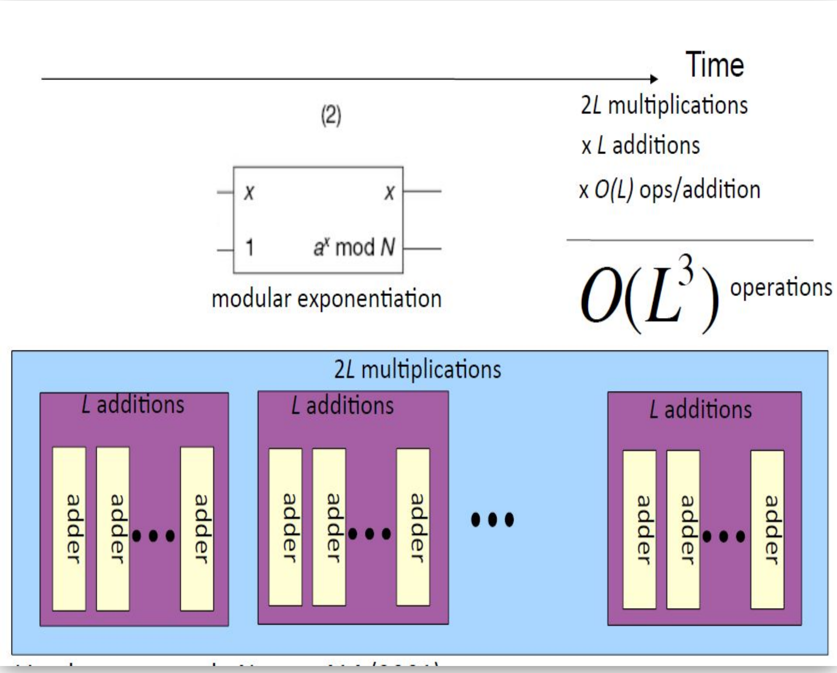
$7^2 \bmod 15 = 4$

$7^3 \bmod 15 = 13$

$7^4 \bmod 15 = 1$

$\vdots$

Circuit for Shor's Algorithm



2-qubit QC: $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$

$$\Rightarrow |\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle \mapsto |0\rangle, |1\rangle, |2\rangle, |3\rangle$$

N-qubit quantum computer $\longrightarrow$ 2^n states $|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle$

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \qquad \sum_{i=0}^{2^n-1} |\alpha_i|^2$$

What did the Google team do?

Completed a problem within 200s as compared to 10,000y on a supercomputer;

→ Programmable Superconducting Processor 54- qubits processor – Sycamore;

→ Quality control over the qubits;

"... improved two-qubit gates with enhanced parallelism that reliably..."

"... new type of control knob that is able to turn off interactions between neighboring qubits..."

→ Sensitive computational benchmark!

What is Supremacy?

"In quantum computing, quantum supremacy is the potential ability of devices to solve problems that classical computers practically cannot.[1] "

Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5

Received: 22 July 2019

Accepted: 20 September 2019

Published online: 23 October 2019

Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G. S. L. Brandao[1,4], David A. Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[1,5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Sergey Knysh[1], Alexander Korotkov[1,8], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[9], Salvatore Mandrà[3,10], Jarrod R. McClean[1], Matthew McEwen[5], Anthony Megrant[1], Xiao Mi[1], Kristel Michielsen[11,12], Masoud Mohseni[1], Josh Mutus[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Murphy Yuezhen Niu[1], Eric Ostby[1], Andre Petukhov[1], John C. Platt[1], Chris Quintana[1], Eleanor G. Rieffel[3], Pedram Roushan[1], Nicholas C. Rubin[1], Daniel Sank[1], Kevin J. Satzinger[1], Vadim Smelyanskiy[1], Kevin J. Sung[1,13], Matthew D. Trevithick[1], Amit Vainsencher[1], Benjamin Villalonga[1,14], Theodore White[1], Z. Jamie Yao[1], Ping Yeh[1], Adam Zalcman[1], Hartmut Neven[1] & John M. Martinis[1,5]*

Google Chief Sundar Pichai

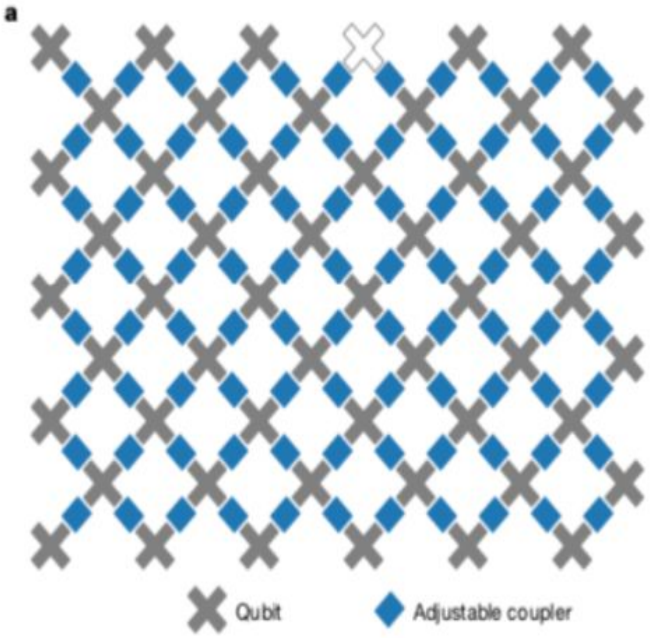
Chief Scientists:

John Martinis - Quantum Hardware

Sergio Boixo - Quantum Computing Theory

1. ^ a b Preskill, John (2012-03-26). "Quantum computing and the entanglement frontier". arXiv:1203.5813 [quant-ph].

Sycamore Qubits Geometry



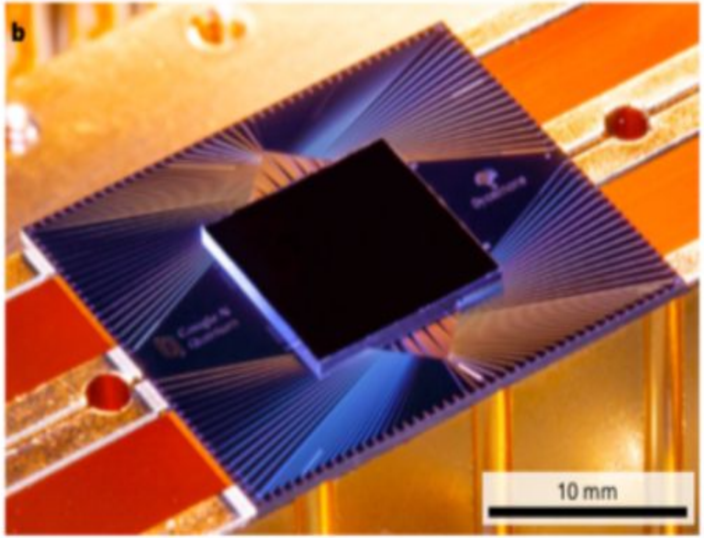
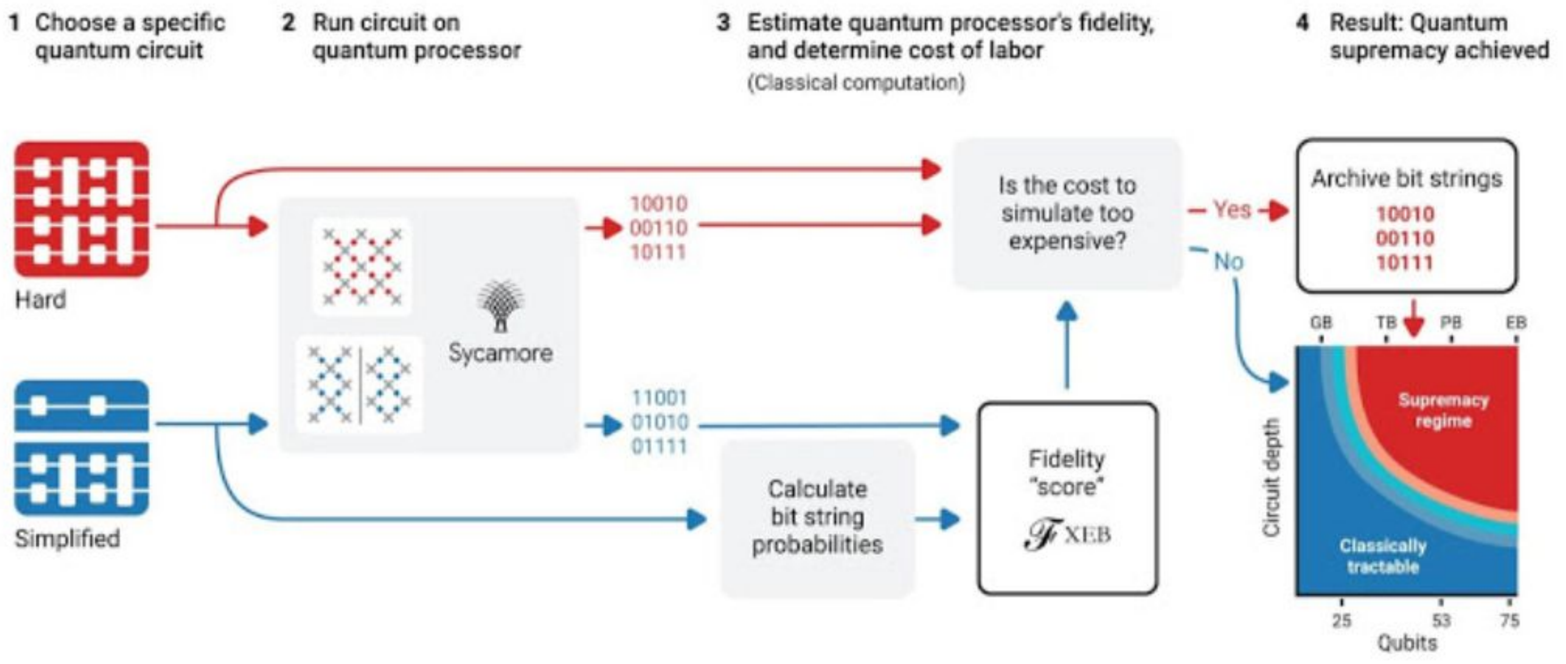


Fig. 1 | The Sycamore processor. a, Layout of processor, showing a rectangular array of 54 qubits (grey), each connected to its four nearest neighbours with couplers (blue). The inoperable qubit is outlined. **b**, Photograph of the Sycamore chip.

Nature | Vol574 | 24OCTOBER2019

Google's process for demonstrating quantum supremacy



Process for demonstrating quantum supremacy.

Control operations for the quantum supremacy circuits

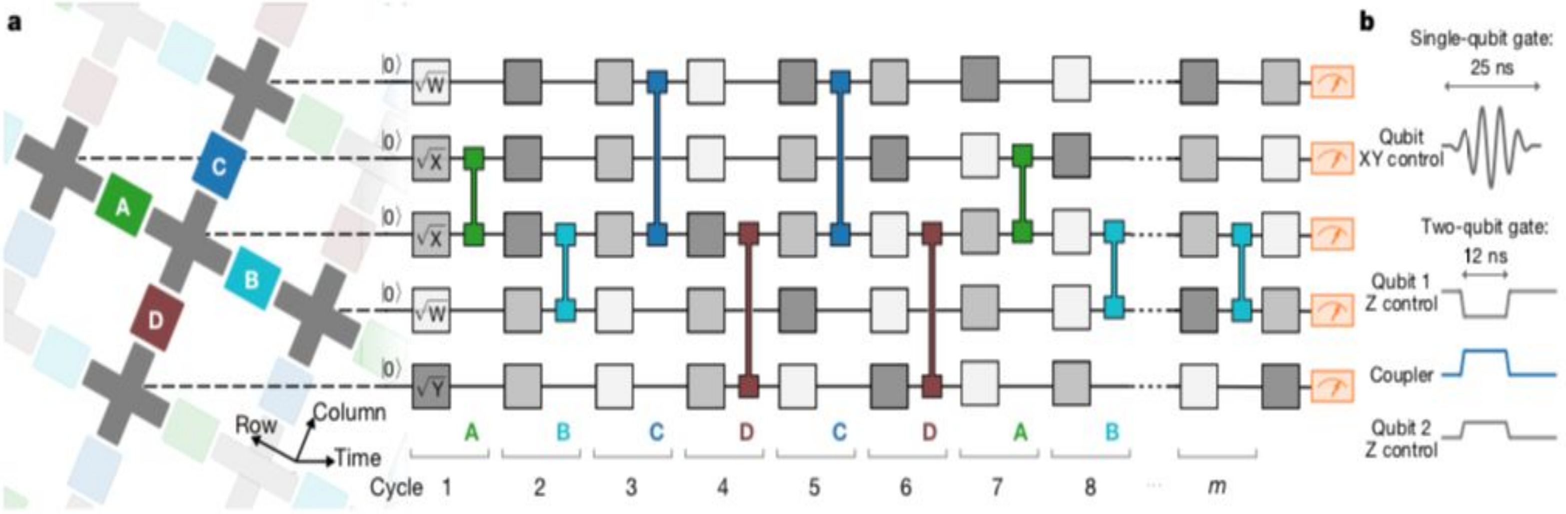
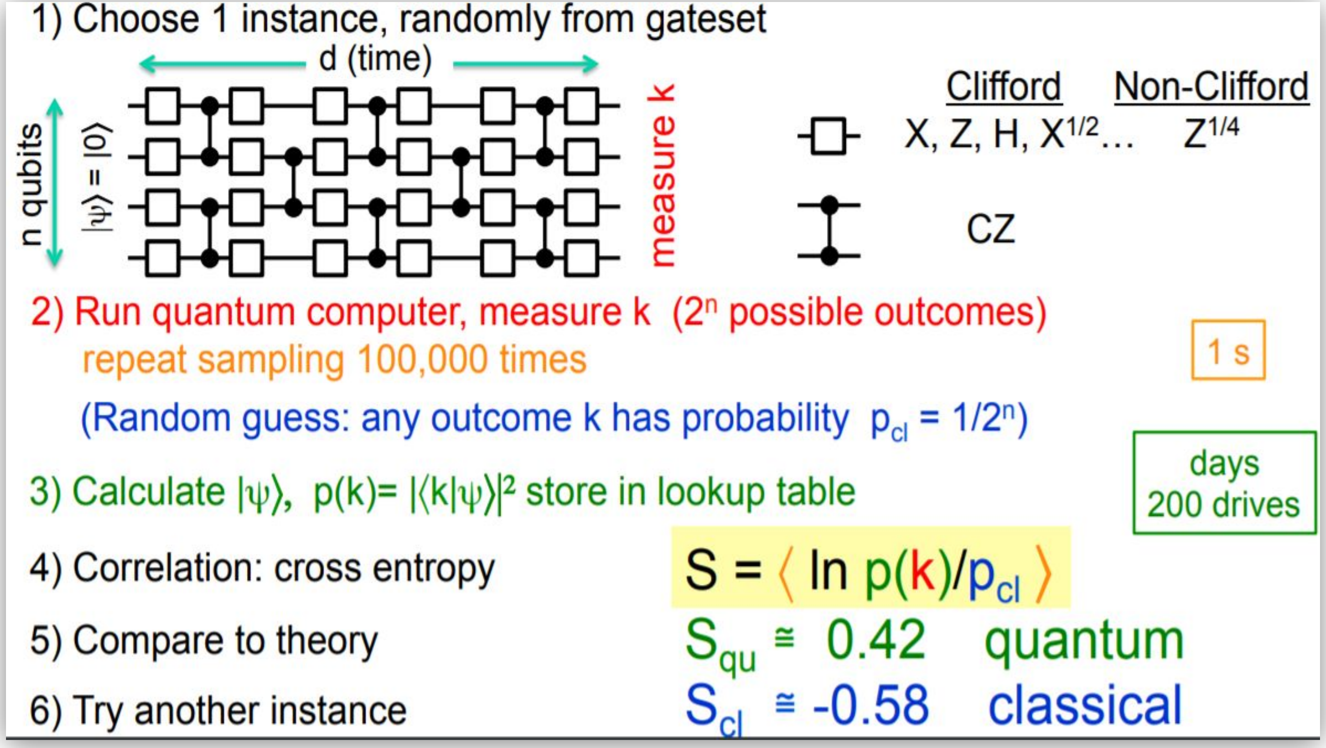


Fig. 3 | Control operations for the quantum supremacy circuits. a, Example quantum circuit instance used in our experiment. Every cycle includes a layer each of single- and two-qubit gates. The single-qubit gates are chosen randomly from $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$, where $W = (X+Y)/\sqrt{2}$ and gates do not repeat sequentially. The sequence of two-qubit gates is chosen according to a tiling pattern, coupling each qubit sequentially to its four nearest-neighbour qubits. The couplers are divided into four subsets (ABCD), each of which is executed simultaneously across the entire array corresponding to shaded colours. Here we show an intractable sequence (repeat ABCDCDAB); we also use different coupler subsets along with a simplifiable sequence (repeat EFGHEFGH, not shown) that can be simulated on a classical computer. **b**, Waveform of control signals for single- and two-qubit gates.

Quantum Supremacy Algorithm: Qubit Speckle



Controversy on Google's Quantum Supremacy:

The recently reported experimental results claiming "quantum supremacy" achieved by Google quantum device are critically discussed. The Google team constructed a quantum chaotic system based on Josephson junction technology which cannot be reliably simulated by the present day supercomputers.

However, the similar "supremacy" can be realized for properly designed micromechanical devices, like periodically forced Duffing oscillator, using the available technology of quartz clocks. It is also reminded that classical and quantum chaotic systems behave in a similar way.

Reference Updated by Google Jan. 2020

On the meaning of "quantum supremacy" experiments. Robert Alicki* International Centre for Theory of Quantum Technologies (ICTQT), University of Gda´nsk, 80-308 Gda´nsk, Poland (Dated: January 6, 2020)

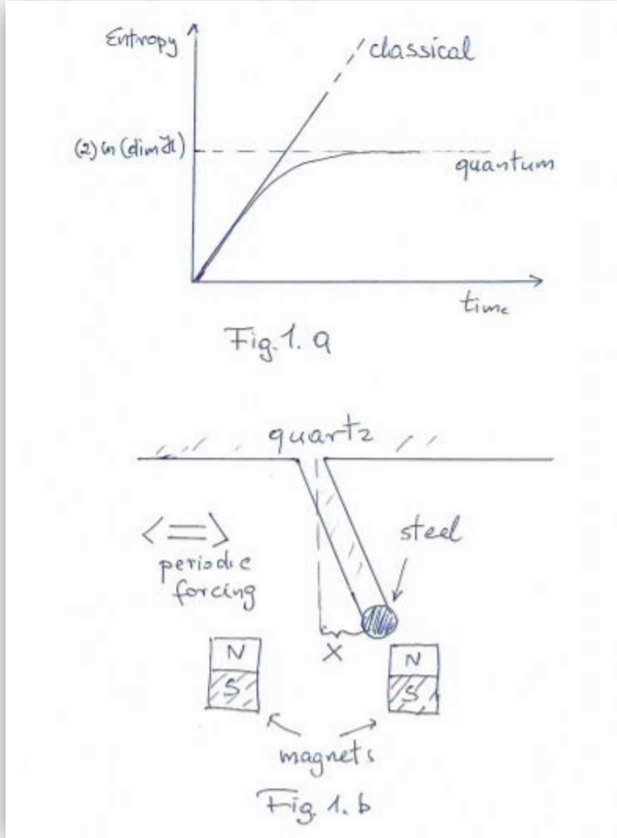
Methods for Supremacy

Fig a) Schematic plots of a dynamical entropy as

a function of time for a chaotic classical system and its quantum counterpart.

Fig b) Proposal of a physical implementation of the

periodically forced Duffing oscillator.



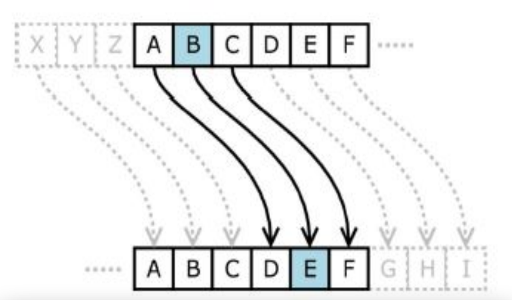
Impact Area: Cryptography

The ancient art of secret communication

→ Julius Caesar encrypted his letters by shifting the alphabet (easy to break)

→ The nazis encrypted their messages using fancy \Enigma" machines with secret settings that changed every day (broken by Turing and others using the rst computers)

→ Since the 1970s: more systematic, mathematical study
→ Two branches: codemakers and codebreakers



Conclusion

Barriers in Quantum Architectures

There are four important challenges in Quantum Architectural Progress,

- **Qubit Coherence Time:** No clarity about the threshold, but current time of 1/10000 is not enough to run serious algorithms and hybrids of quantum and classical operations.

- **Qubit Count:** True potential of Quantum computing can be experienced with the Qubit counts, as per today's world of data processing, requires not hundreds of bits, it needs more than thousands.

- **Connectivity of the Qubit:** A qubit must be possible to connect any other qubit, currently the implementations are not fully connected, this leads to some constraints in the Algorithms

- **Quantum Algorithms:** Classical solves with the numbers and maths logic that we know, quantum algorithm deals with the Quantum physics, and classical computing cannot provide an abstraction for this.

Lack of symmetry in Quantum Computers

A key for manageable development of algorithms and codes on a classical computer is because of the Symmetry in resources. Resources such as Registers, Memory, Storage, File System, everything will behave the same or at least those which are of the same type.

But in Quantum Computers currently doesn't have Registers, Memory, Storage, File System.

Before bringing the entanglement, all the qubits are same in superposition state. Some qubits can be either a *source* or a *target* for entanglement. Some can be only a source. Some can only be a target. And with specific other qubits at that.

The connectivity map will differ between 5, 8, 20, 49, 50, 64, 72, 128, and 2048-qubit quantum computers

All these leads redesign of algorithm for each type of machines with its different unpredictable architecture behaviour

Do we need Memory ?

- More memory means more data can be processed.

- Modern classical computers support *virtual memory* so that each program sees a vast, linear, symmetric *address space* for memory. That's the critical requirement for supporting the more familiar concepts of variables, nested function calls, arrays, lists, tables, etc. which are critical to the rapid development of classical computer programs.

- In Quantum computers no memory or registers other than relatively few Qubits 5, 7, 20, 53, 72, 128, 2048. But issue is not the count of Qubit, it is the lack of organization, consistent, symmetry which is the key to facilitate the development of algorithm.

Is a Quantum Coprocessor Good Enough?

- It's more of an approximation than the true Universal Quantum Computer.

- The current hybrid mode of having a GPU or Floating point co processor and having a quantum coprocessor have a big difference going back to **Algorithms**.

    - All of the operations performed by a floating-point coprocessor were integrated cleanly into existing algebraic expressions, with no visible algorithmic differences.
    - Whereas,the use of quantum computing requires explicit algorithmic differences

- GPU is partly an abstract from the CPU, with support for integers, floating point, algebraic expressions and classical control flow and data structures, but in Quantum coprocessor we cannot have that abstract to develop on.

- Executing relatively short snippets of quantum code distributed with classical logic and data processing as well should be a very viable model for the future quantum computation.

Summary

Raw Quantum physics vs Intellectual power of Turing machine, mathematics, logic, and rich data types

References

[1]R. Vignesh, P. G. Poonacha, "Quantum computer architectures: an idea whose time is not far away", IEEE International Conference on Computers Communications and Systems (ICCCS), pp. 6-13, Nov. 2015.
Online:https://ieeexplore-ieee-org.aurarialibrary.idm.oclc.org/document/7100621/

[2]A. M. Steane, “A Tutorial on Quantum Error Correction”. In G. Casati, D. L. Shepelyansky, and P. Zoller, editors, Proc. Int. Sch. Phys. “Enrico Fermi”, course CLXII “Quantum Comput. Algorithms Chaos”, pages 1–32. IOS Press, Amsterdam 2006, 2006.
Online: https://www2.physics.ox.ac.uk/sites/default/files/ErrorCorrectionSteane06.pdf

[3]Y. S. Weinstein, “Syndrome measurement strategies for the [[7,1,3]] code”, Quantum Inf. Process. 14(6), 1841-1854 (2015).
Online: https://arxiv.org/pdf/1306.0498.pdf

[4]D. Kielpinski, C. R. Monroe, and D. J. Wineland, “Architecture for a large-scale ion-trap quantum computer”, Nature (London) 417, 709 (2002).
Online: http://iontrap.umd.edu/wp-content/uploads/2012/12/Architecture-for-a-large-scale-ion-trap-quantum-computer.pdf

[5]Rainer Blatt, David Wineland, “Entangled States of Trapped Atomic Ions”, Nature Vol 453,19 June 2008, doi:10.1038/nature07125
Online: https://www.nature.com/articles/nature07125

References

[6] M. F. Brandl, “A Quantum von Neumann Architecture for Large-Scale Quantum Computing”, arXiv:1702.02583 (2017).

[7] X. Fu, M. A. Rol, C. C. Bultink, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, and K. Bertels, “An experimental microarchitecture for a superconducting quantum processor,” in Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, ser. MICRO-50 ’17. New York, NY, USA: ACM, 2017.
Online: http://doi.acm.org/10.1145/3123939.3123952

[8] Quantum Algorithm Implementations for Beginners. https://arxiv.org/pdf/1804.03719.pdf

[9] Instruction Set Architectures for Quantum Processing Units. https://arxiv.org/pdf/1707.06202.pdf

[10]Optimization of Circuits for IBM’s five-qubit Quantum Computers. https://arxiv.org/pdf/1810.00129.pdf

[11] Fault tolerance in the IBM Q Experience. https://pdfs.semanticscholar.org/c5bc/28e8b9f20cd1aa11004d3e4871ab7c1cb999.pdf

[12] Evaluating IBM’s Quantum Compiler and Quantum Computer Architectures As They Pertain to Quantum Walk Simulation Algorithms.https://math.ucsd.edu/_files/undergraduate/honors-program/honors-program-presentations/2018-2019/Clara_Woods_Honors_Thesis.pdf

References

[13]From the Quantum Moore's Law toward Silicon Based Universal Quantum Computing

[14]https://ieeexplore-ieee-org.aurarialibrary.idm.oclc.org/document/7237090

[15]https://medium.com/@jonathan_hui/qc-how-to-build-a-quantum-computer-with-superconducting-circuit-4c30b1b296cd

[16]https://www.techrepublic.com/article/d-waves-pegasus-topology-most-connected-quantum-computer-that-will-improve-machine-learning/https://arxiv.org/pdf/1707.06202.pdf

[17]https://medium.com/@jackkrupansky/what-is-a-universal-quantum-computer-db183fd1f15a

[18]https://www.theguardian.com/technology/2016/may/22/age-of-quantum-computing-d-wave

[19]https://www.techrepublic.com/article/d-wave-quantum-computers-the-smart-persons-guide/

[20]https://www.dwavesys.com/sites/default/files/ttt_arxiv_v1.pdf

References

[21]https://support.dwavesys.com/hc/en-us/articles/360003695354-What-is-the-Chimera-Topology-

[22]https://hackernoon.com/how-does-rsa-work-f44918df914b

[23]https://docs.dwavesys.com/docs/latest/c_gs_4.html#connectivity

[24]Rachel Courtland”Google aims for quantum computing supremacy [News]” https://ieeexplore.ieee.org/abstract/document/7934217/

[25]Masoud MOHSENIHartmut Neven”Chips including classical and quantum computing processors” https://patents.google.com/patent/US20170177534A1/en

[26] Michael CouryGeordie RoseJeremy Hilton“Quantum computing integrated development environment” https://patents.google.com/patent/US20030169041A1/en

[27]Tad Hogg“Adiabatic quantum computing for random satisfiability problems”https://journals.aps.org/pra/abstract/10.1103/PhysRevA.67.022314

[28]T. Pellizzari, S. A. Gardiner, J. I. Cirac, and P. Zoller “Decoherence, Continuous Observation, and Quantum Computing: A Cavity QED Model”https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.75.3788

[29]C. Negrevergne, T. S. Mahesh, “ Benchmarking Quantum Control Methods on a 12-Qubit System” https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.96.170501

[30] Michael Freedman, Chetan Nayak “Quantum computing devices with majorana hexon qubits” https://patents.google.com/patent/US20200098821A1/en