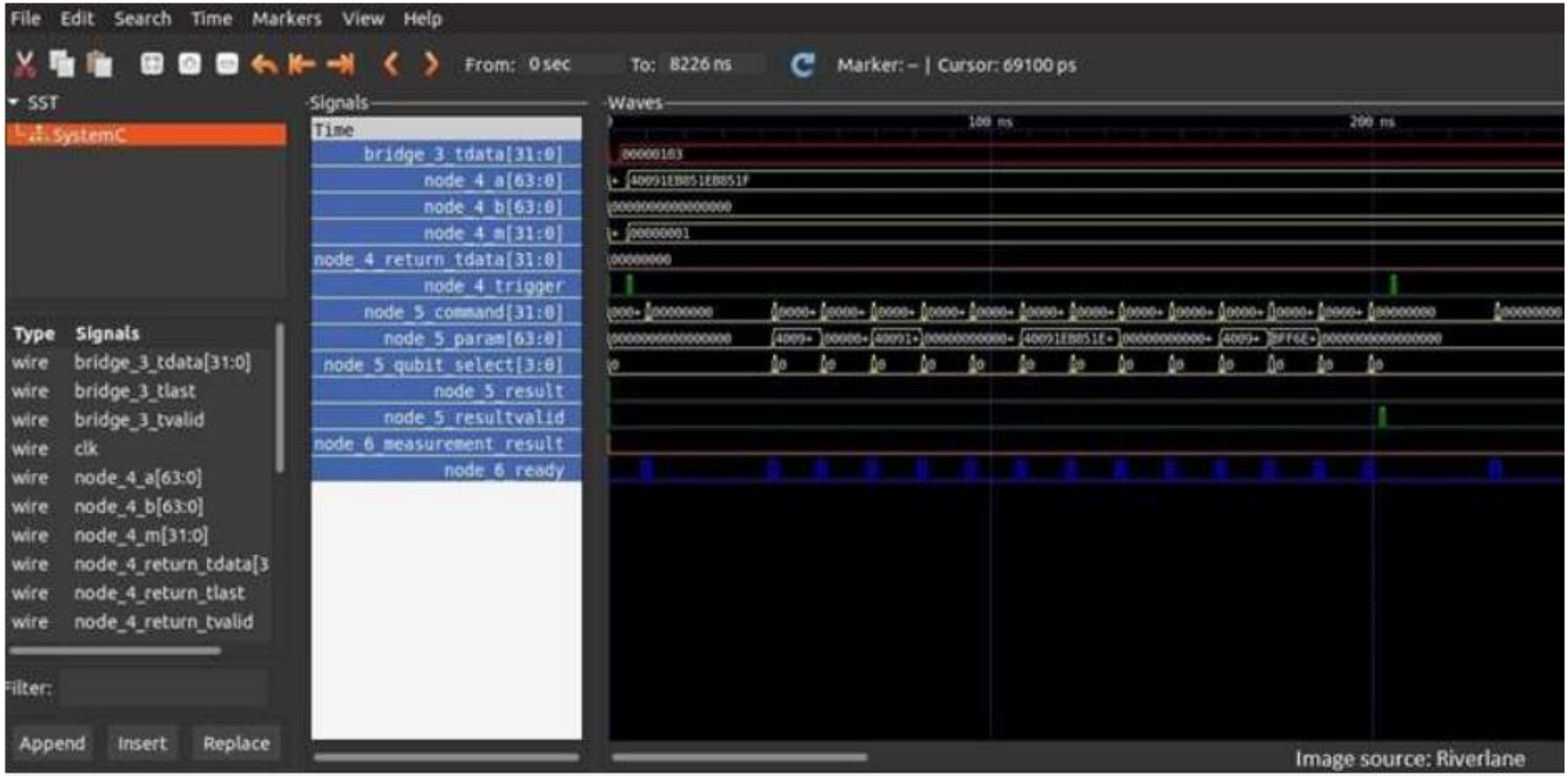
Part 3

Try this !

- https://medium.com/rigetti/how-to-write-a-quantum-program-in-10-lines-of-code-for-beginners-540224ac6b45

Quantum Operating System ?

- Riverlane was announced as the lead partner in a consortium which was awarded a £7.6m grant to build a radically new operating system for quantum computers.
- The first successful trials of Deltaflow.OS took place in September, using quantum hardware belonging to leading trapped-ion company, Oxford Ionics.



An early Deltaflow.OS dashboard

Top Quantum Programming Languages

- Quantum programming languages are the foundations to interpret ideas into instructions to be carried out by quantum computers.
- Here are the top quantum programming languages to know for 2022.

- **QCL**
- Quantum computing language is one of the first implemented quantum programming languages that **resembles C language** in regards to syntax and data types.
- It is usually used for writing programs for quantum computers.
- As every quantum machine has to be controlled by classical devices, the pre-existing quantum programming languages incorporate classical control structures like **loops and conditional execution** and allow them to operate on classical and quantum data.

- **QMASM**
- Quantum macro assembler was published in 2016.
- It is a kind **of low-level language** that is specially used for quantum annealing.
- The significance of QMASM **relieves the programmer from having to know system-specific hardware** details while still allowing programs to be expressed at a low level of abstraction.

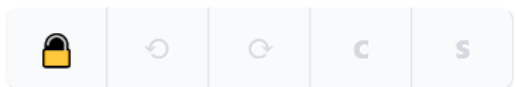
- **Silq**
- Silq was originally published in 2020.
- It is a high-level programming language when compared to the QCL and QMASM.
- It is written in **D language** which has 482 stars and 10 contributions on github and is regularly updated too.

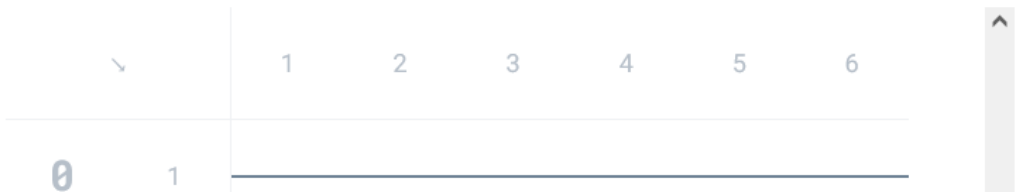
Quantum Simulators

QUANTUM JAVASCRIPT

Q is a quantum circuit simulator, drag-and-drop circuit editor, and powerful JavaScript library that runs right here in your web browser. There's nothing to install and nothing to configure, so jump right in and experiment. (Q recently celebrated our one-year anniversary. You can read the corresponding post on Medium, the discussion on Hacker News, and the thread on Reddit.)

Here's your first quantum circuit—a Bell state. It uses superposition and entanglement to calculate. (And here's how to make one yourself.) Tap and drag the tiles around to get a feel for the Q editor. It's easy to use on both desktop and mobile devices. Made a mistake? Just tap the Undo button.





qsim

Optimized quantum circuit simulators

qsim

qsim is a full wave function simulator written in C++. It uses gate fusion, AVX/FMA vectorized instructions and multi-threading using OpenMP to achieve state of the art simulations of quantum circuits. qsim is integrated with Cirq and can be used to run simulations of up to 40 qubits on a 90 core Intel Xeon workstation.

Get started with qsim on Cirq GitHub repository

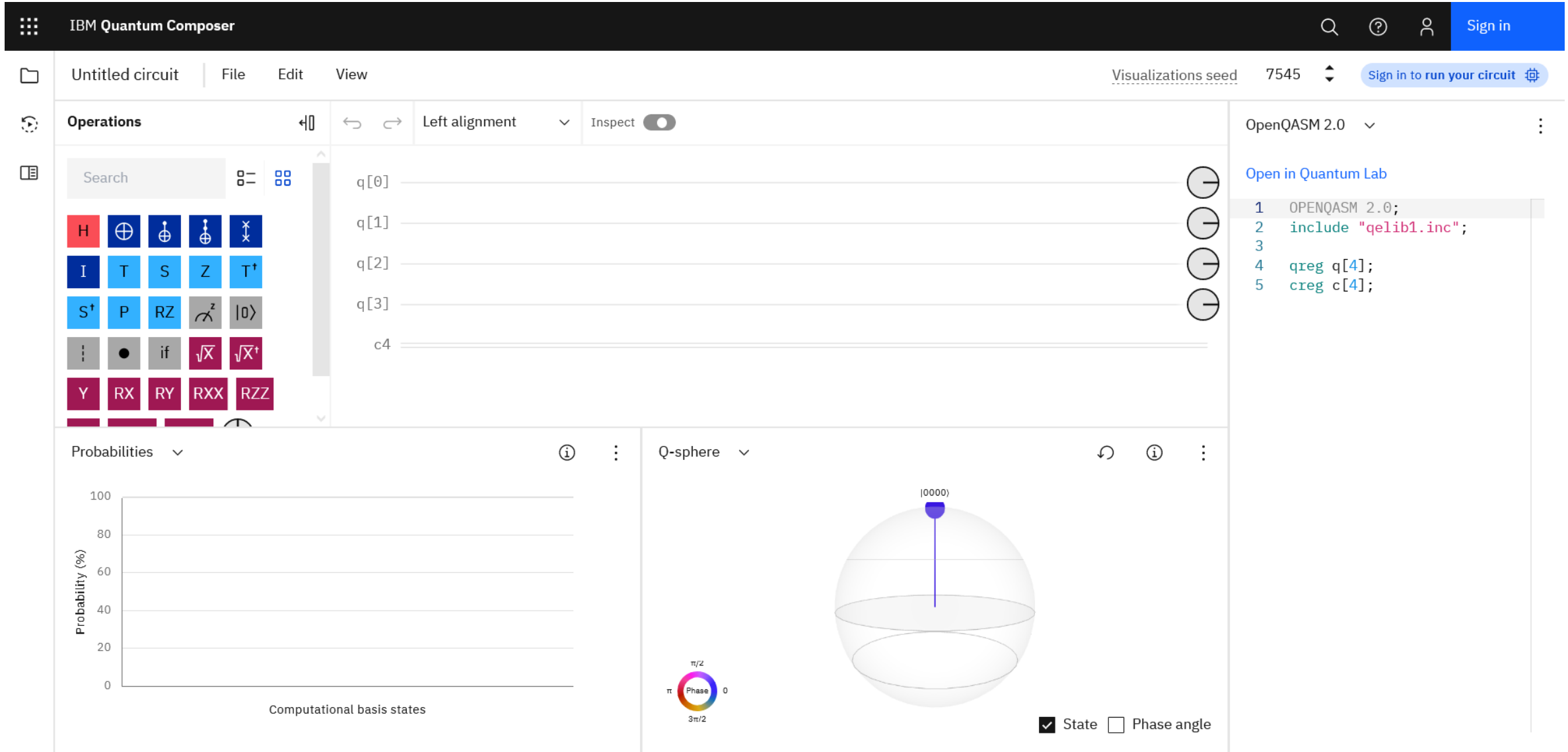
```
import cirq
import qsimcirq

# Pick up to ~25 qubits to simulate (requires ~256MB of RAM)
qubits = [cirq.GridQubit(i,j) for i in range(5) for j in range(5)]

# Define a circuit to run
# (Example is from the 2019 "Quantum Supremacy" experiement)
circuit = (cirq.experiments.
    random_rotations_between_grid_interaction_layers_circuit(
    qubits=qubits, depth=16))

# Measure qubits at the end of the circuit
circuit.append(cirq.measure(*qubits, key='all_qubits'))

# Simulate the circuit with qsim and return just the measurement values
```


IBM Quantum Composer
Sign in
Untitled circuit
File
Edit
View
Visualizations seed
7545
Sign in to run your circuit
Operations
Search
H
I
T
S
Z
T†
S†
P
RZ
|0⟩
if
√X
√X†
Y
RX
RY
RXX
RZZ
Left alignment
Inspect
q[0]
q[1]
q[2]
q[3]
c4
OpenQASM 2.0
Open in Quantum Lab
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[4];
5 creg c[4];
Probabilities
100
80
60
40
20
0
Probability (%)
Computational basis states
Q-sphere
|0000⟩
π/2
π
Phase
0
3π/2
State
Phase angle

- https://quantumai.google/qsim
- https://quantum-computing.ibm.com/composer/files/new
- https://quantumjavascript.app/

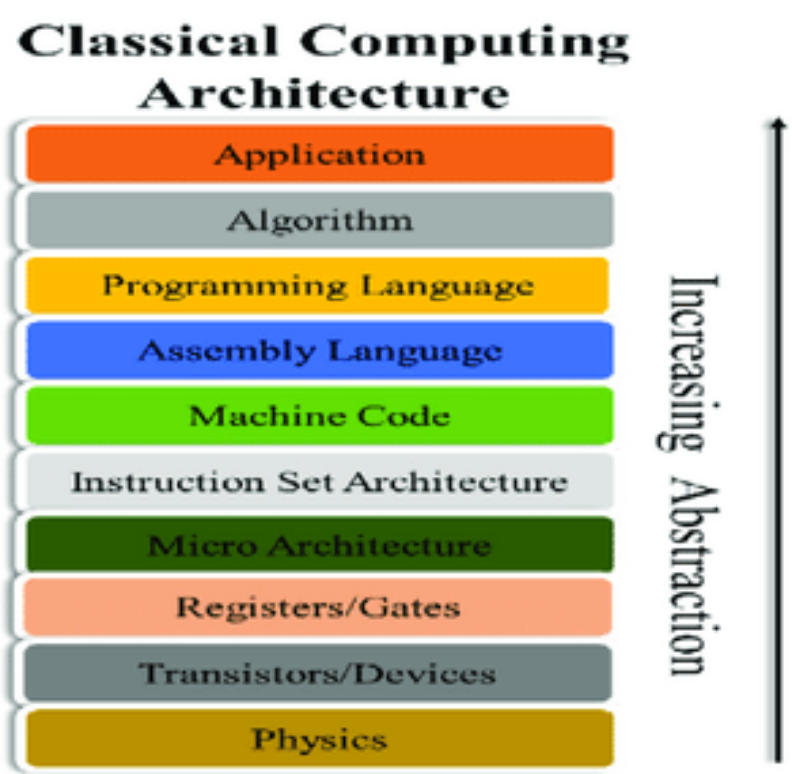
A Comparison of Classical and Quantum Computing

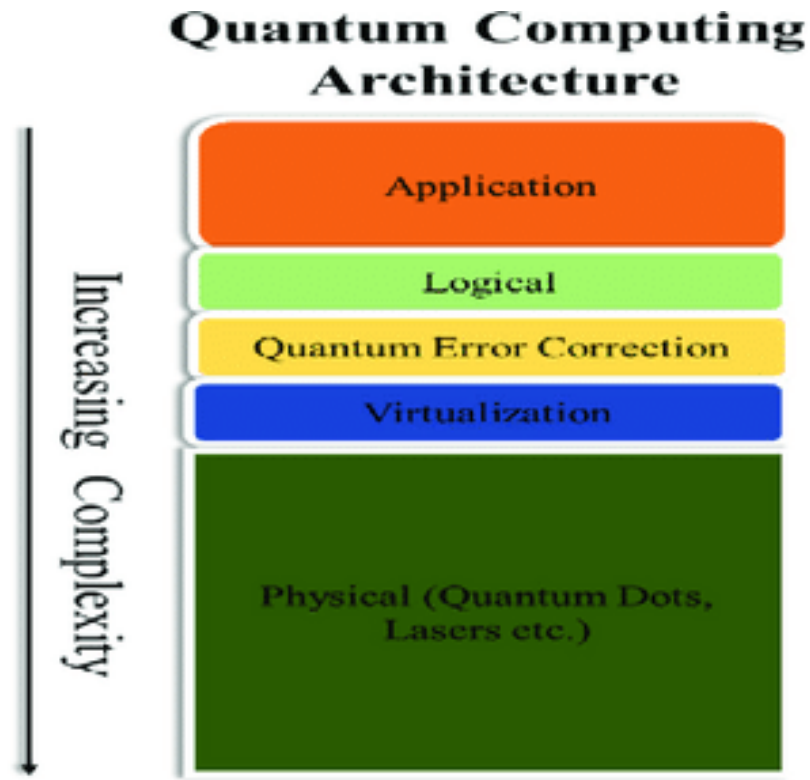
- Classical computing relies, at its ultimate level, on principles expressed by **Boolean algebra**.
- Data must be processed in an **exclusive binary state** at any point in time or bits.

Classical computing vs. quantum computing

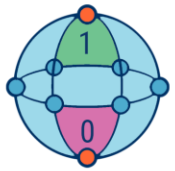
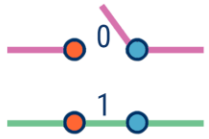
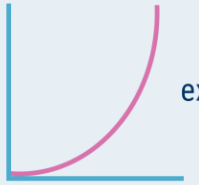
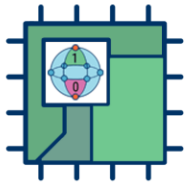
Classical computing	Quantum computing
Uses logic based on digital logic	Is based on quantum theory
Sends digital signals using bits	Sends data through the use of particles or photons
Uses circuits with complementary metal oxide semiconductors	Operates in extreme cold environments
Encryption is based on mathematical algorithms	Encryption is based on quantum properties

TechTarget


Classical Computing Architecture
Application
Algorithm
Programming Language
Assembly Language
Machine Code
Instruction Set Architecture
Micro Architecture
Registers/Gates
Transistors/Devices
Physics
Increasing Abstraction


Quantum Computing Architecture
Application
Logical
Quantum Error Correction
Virtualization
Physical (Quantum Dots, Lasers etc.)
Increasing Complexity

Quantum Computing Vs. Classical Computing

Quantum Computing	Classical Computing
 Calculates with qubits, which can represent 0 and 1 at the same time	Calculates with transistors, which can represent either 0 or 1 
 Power increases exponentially in proportion to the number of qubits	Power increases in a 1:1 relationship with the number of transistors 
 Quantum computers have high error rates and need to be kept ultracold	Classical computers have low error rates and can operate at room temp 
 Well suited for tasks like optimization problems, data analysis, and simulations	Most everyday processing is best handled by classical computers 

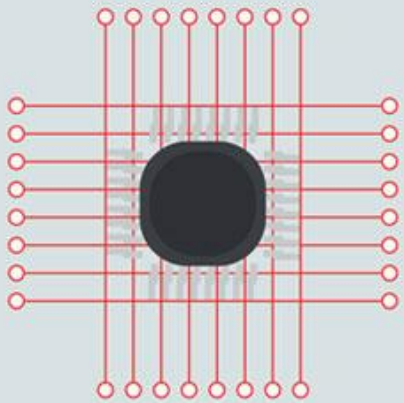
Classical computer

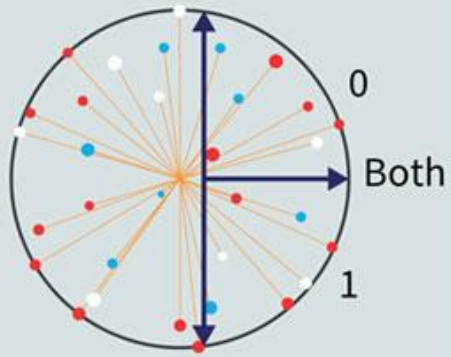
0
1

Classical bit

Classical bits can only be 0 or 1

Quantum computer





Qubit

Qubits follow the superposition principle and can exist as 0 and 1 at the same time

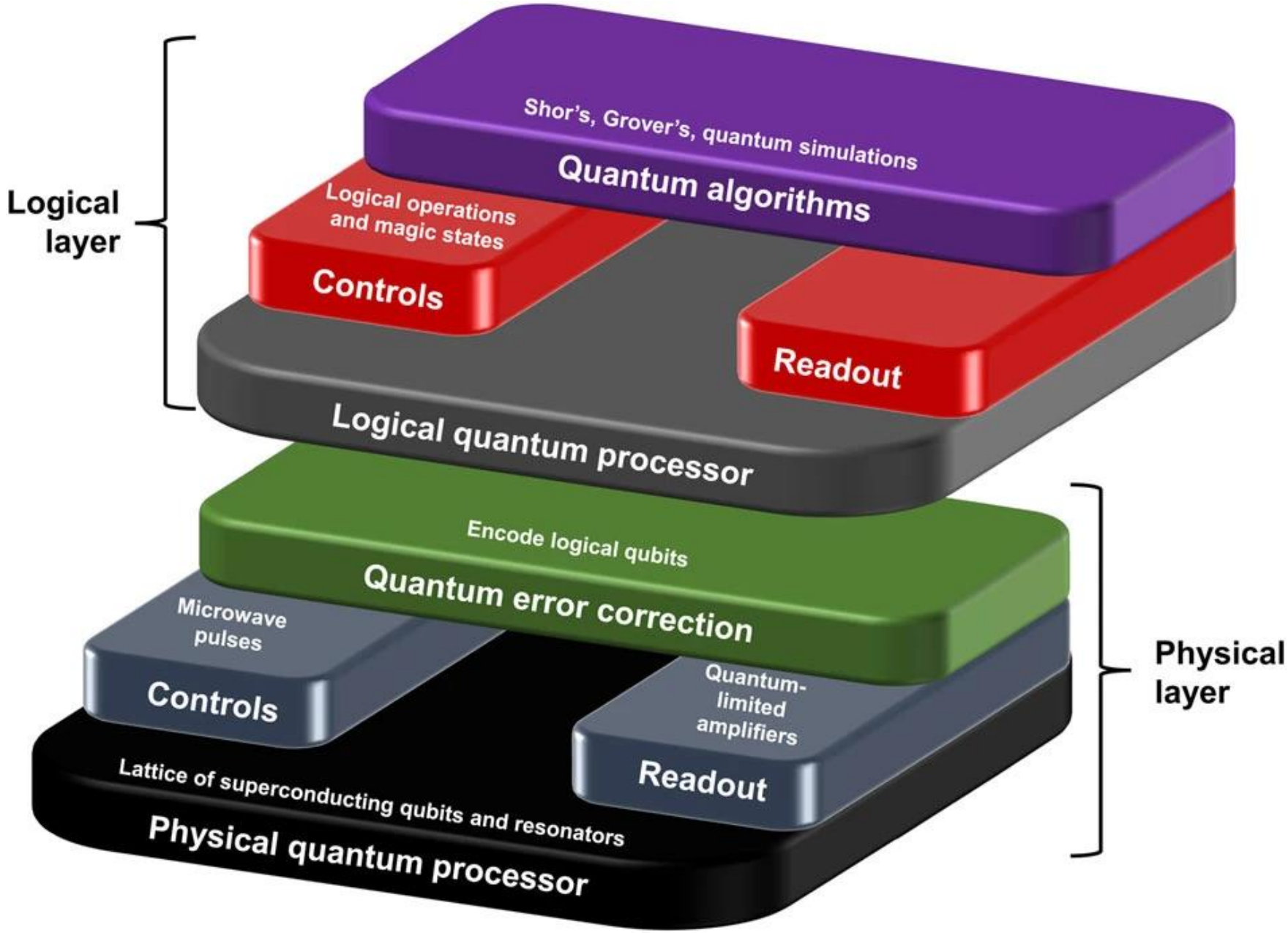
Hardware

- Physical qubit
- Processor
- Control circuits
- Interface
- Detectors, chips for trapping ions, vacuum, laser, microwave or optical system (ion-trapped)
- Wirings
- Refrigeration
- Housing (package)
- Signal shielding

Software

- Error correction
- Control processing languages
- Specification
- Simulation
- Debugging
- Programming languages
- Compilers
- Schedulers
- Optimisation (eg, resource estimators)
- Verification or validation
- Function libraries

Quantum Computing Model


Logical layer
Shor's, Grover's, quantum simulations
Quantum algorithms
Logical operations and magic states
Controls
Readout
Logical quantum processor
Encode logical qubits
Quantum error correction
Microwave pulses
Controls
Quantum-limited amplifiers
Readout
Lattice of superconducting qubits and resonators
Physical quantum processor
Physical layer

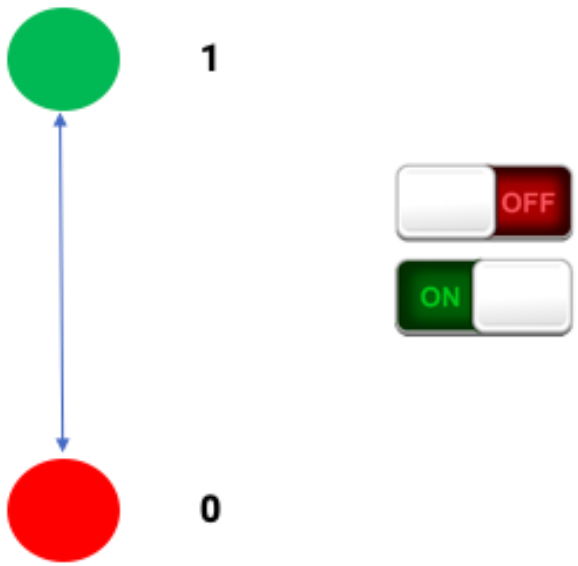
Now …let's talk about Bits and Qubits

- While the time that each transistor or capacitor need be either in 0 or 1 before switching states is now measurable in billionths of a second, there is still a limit as to how quickly these devices can be made to switch state.
- As we progress to smaller and faster circuits, we begin to reach the **physical limits** of materials and the threshold for **classical laws** of physics to apply.
- Beyond this, the **quantum worl**d takes over.

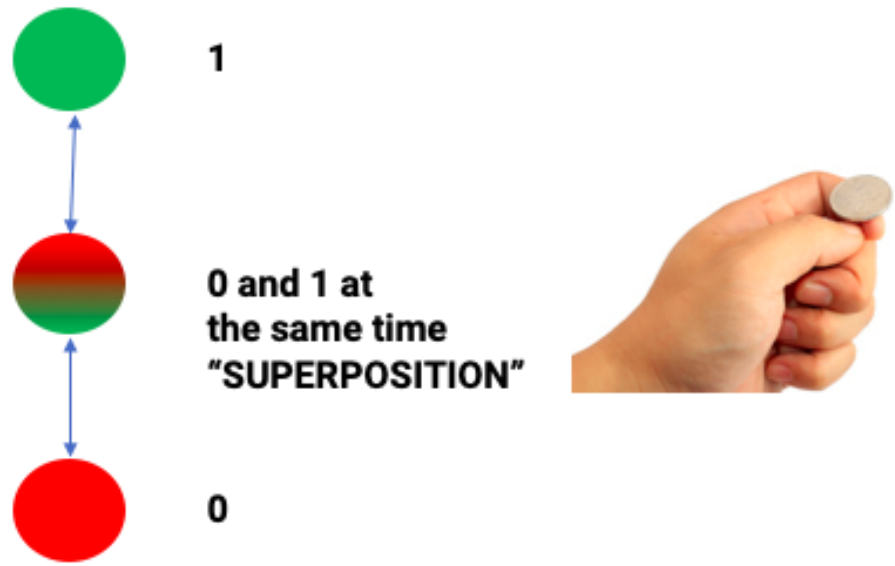
- In a quantum computer, a number of elemental particles such as electrons or photons can be used with either their **charge or polarization** acting as a representation of 0 and/or 1.
- Each of these particles is known as a **quantum bit, or qubit**, the nature and behavior of these particles form the basis of quantum computing.

BITS vs QBITS

Classical Computer – Operations on BITS



Quantum Computer – Operations on Quantum BITS



Quibits can take same value simultaneously. This characteristic expands the possibility of parallel calculations

The difference between bit and qubit

BIT

Is the smallest unit of information in current computers. It represents only one of two values, **0** or **1**

QUBIT

Quantum analogue of the classical bit. It can take on two values simultaneously, 0 and 1. This characteristic expands possibility of producing parallel calculations

Quantum Superposition and Entanglement

- The **two most relevant aspects** of quantum physics are the principles of **superposition and entanglement**.

Superposition: Think of a qubit as an electron in a magnetic field.

- The electron's spin may be either in alignment with the field, which is known as a spin-up state, or opposite to the field, which is known as a spin-down state.
- According to quantum law, the particle enters a superposition of states, in which it behaves as if it were in both states simultaneously. Each qubit utilized could take a superposition of both 0 and 1.

***Entanglement*:** Particles that have interacted at some point retain a type of connection and can be entangled with each other in pairs, in a process known as *correlation*.

- Knowing the spin state of one entangled particle - up or down - allows one to know that the spin of its mate is in the opposite direction.
- Quantum entanglement allows qubits that are separated by incredible distances to interact with each other instantaneously (**not limited to the speed of light**).
- No matter how great the distance between the correlated particles, they will remain entangled as long as they are isolated.

- Taken together, quantum superposition and entanglement create an enormously enhanced computing power.
- Where a 2-bit register in an ordinary computer can store only one of four binary configurations (00, 01, 10, or 11) at any given time, a 2-qubit register in a quantum computer can store all four numbers simultaneously, because each qubit represents two values. If more qubits are added, the increased capacity is expanded exponentially.

- To offer an idea of the scale, 500 qubits can represent the same information as 2^500 normal bits. While a typical computer would need millions of years to find all the prime factors of a 2,048-bit number (a number with 617 digits), a quantum computer can do the job in minutes.

Difficulties with Quantum Computers

- **Interference** - During the computation phase of a quantum calculation, the slightest disturbance in a quantum system (**say a stray photon or wave of EM radiation**) causes the quantum computation to collapse, a process known as **de-coherence**. A quantum computer must be **totally isolated** from all external interference during the computation phase.
- **Error correction** - Given the nature of quantum computing, error correction is ultra critical - even a single error in a calculation can cause the validity of the entire computation to collapse.
- **Output observance** - Closely related to the above two, retrieving output data after a quantum calculation is complete risks corrupting the data.