INDRAPRASTHA INSTITUTE of
INFORMATION TECHNOLOGY DELHI

# Quantum Computing Simplified

Debajyoti Bera
IIIT-Delhi
www.iiitd.ac.in

ATAL-FDP, Vardhman College of Engg.
Hyderabad, 6th October 2020

# Random Bits


Classical randomized algorithms use classical coins (random bits).

**Classical randomized algorithms are efficient**

Quicksort, Monte-Carlo sampling, …

Satisfiability of 3SAT Boolean formula

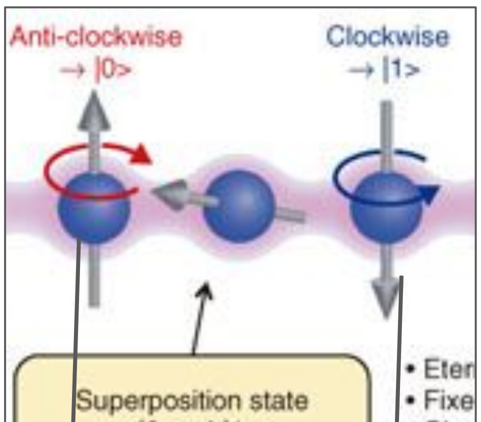
Brute force $O(2^n)$

Deterministic $O(1.439^n)$

Randomized $O(1.321^n)$

# From Bits to Qubits

Photon polarization, Electron spin,
direction of current in Josephson junction, ...



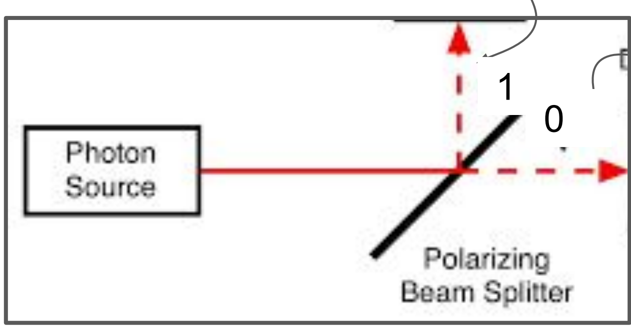

HEAD / 0

TAIL / 1

Quantum operations are inherently probabilistic.

- Spin is observed to be clockwise/anti-clockwise with equal probability.
- Photon can be observed in only one of the paths, with equal probability.

equal chance

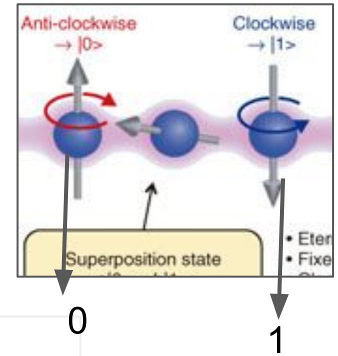



**Quantum ≈ Randomized +++++**
(different rules of randomization)

# Organization for this lecture

1. (Recap) The basic principles of quantum computing
    a. Qubits
    b. Operations
2. Designing quantum algorithms
3. Emerging techniques

# 1 qubit

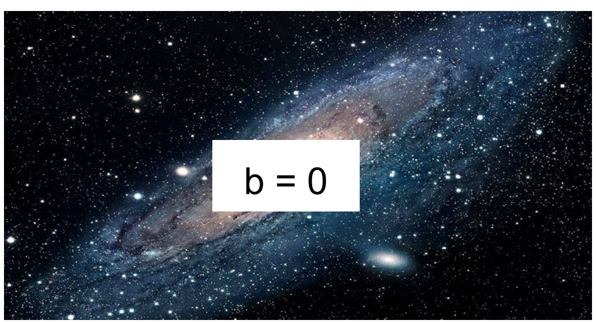



Basic datatype. Can be observed to be in state-0 and state-1.

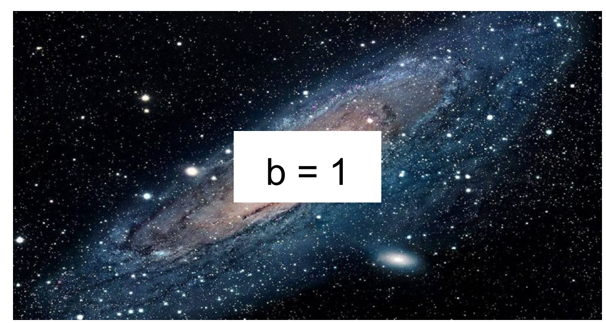
## Behaviour of a random bit

```
1.   var b = 0
2.   b = random(0,1)
     // Q: what is b?
3.   if b=0, print("0")
4.   if b=1, print("1")
```

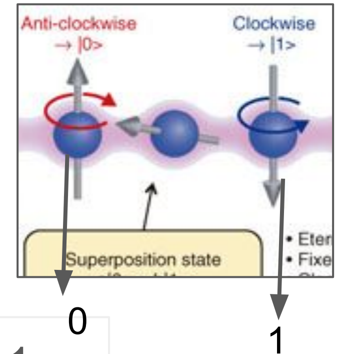
If 0 is printed, b must have been 0.
If 1 is printed, b must have been 1.




OR



# 1 qubit




Basic datatype. Can be observed to be in state-0 and state-1.

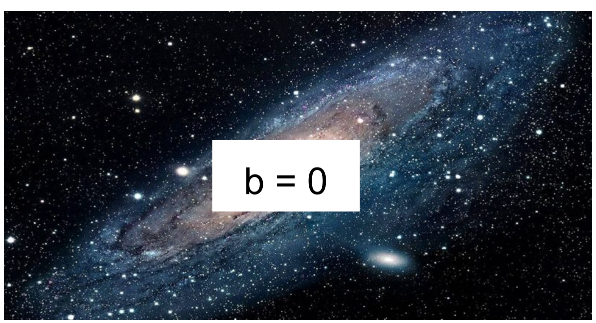
## Behaviour of a qubit

```
1.   var b = 0
2.   b = random(0,1)
     // Q: what is b?
3.   if b=0, print("0")
4.   if b=1, print("1")
```

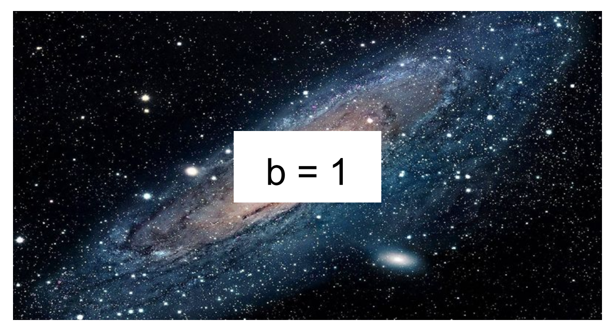
```
1.   qubit |b> = |0>
2.   Apply H on |b>
     // Q: what is state of b?
3.   if b=0, print("0")
4.   if b=1, print("1")
```

**Exercise**

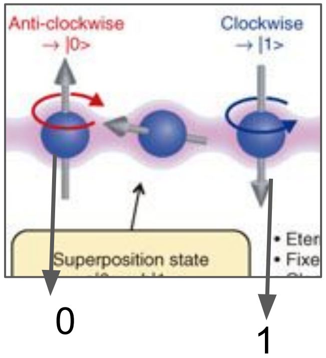
Determine the state |b> by only observing the output of the code.




AND



# First interesting 1-qubit state




Notation for a state of a qubit

Notation for state-0 ≡ vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$$

Vector addition

Amplitude

Notation for state-1≡ vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

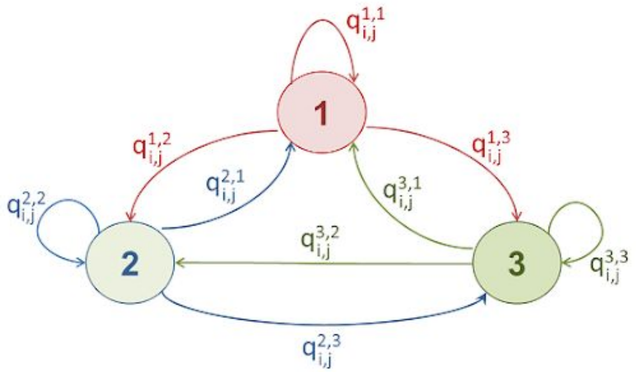
Basis states for a 2-dimensional vector space over complex numbers

2-D Hilbert space

1-qubit state can be mathematically represented as a complex combination of two basis states of a 2-dimensional Hilbert space

2-qubit state can be mathematically represented as a complex combination of four basis states

# Stochastic vector




Classical randomized algorithms use random variables

b = random bit from { 0:½ , 1:½ }

Mathematical representation of b

$$\begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

represents b=0

represents b=1

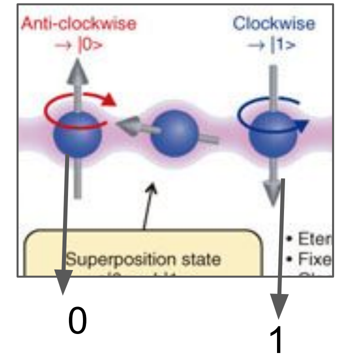
Another representation of b

$$\begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix}$$

not clear what can be be done with alternative representations using a different basis

Algorithms using random bits can be analysed using L1 norm unit vectors over $\Re$

# Return to “First interesting 1-qubit state”




Notation for a state of a qubit

Notation for state-0 ≡ vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$$

Amplitude

Vector addition

Notation for state-1≡ vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Basis states for a 2-dimensional vector space over complex numbers

2-D Hilbert space

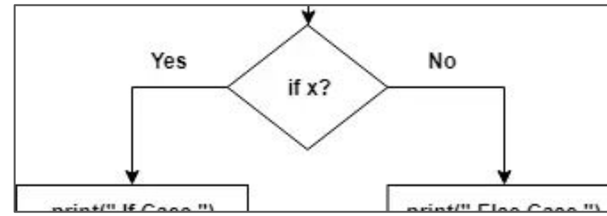
$$|\psi\rangle = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \frac{1+i}{2}\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} + \frac{1-i}{2}\begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$$

$|+\rangle$ $|-\rangle$

Algorithms using qubits can be analysed using L2 norm unit vectors over $\mathfrak{C}$

# “Value” of a qubit (state vector)

Observation/measurement changes the state of a qubit !

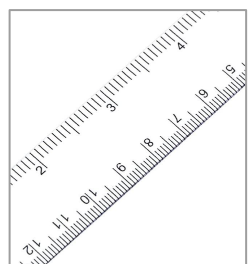



Linear algebraically,
measurement is a projection
onto a set of basis states.

$$|\psi\rangle = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1+i}{2} \underbrace{\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}}_{|+\rangle} + \frac{1-i}{2} \underbrace{\begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}}_{|-\rangle}$$

Measuring in 0/1 basis

- Observe $|0\rangle$ with probability $\left|\frac{1}{\sqrt{2}}\right|^2$
  State changes to $|0\rangle$
- Observe $|1\rangle$ with probability $\left|\frac{1}{\sqrt{2}}\right|^2$
  State changes to $|1\rangle$



Measuring in +/- basis

- Observe $|+\rangle$ with probability $\left|\frac{1+i}{2}\right|^2$
  State changes to $|+\rangle$
- Observe $|-\rangle$ with probability $\left|\frac{1-i}{2}\right|^2$
  State changes to $|-\rangle$

# qubit

Has intrinsic state.
State is a continuum from 2-dimensional numbers.
Observation reveals partial information.
Observation changes (collapses) states.

# Single bit operation

Classical deterministic operations

| f(x) = constant | | f(x) = not(x) | f(x) = x |
|---|---|---|---|
| $0 \to 0$<br>$1 \to 0$ | $0 \to 1$<br>$1 \to 1$ | $0 \to 1$<br>$1 \to 0$ | $0 \to 0$<br>$1 \to 1$ |

Classical randomized operations

If x=0, f(x) =

| value | 0 | 1 |
|---|---|---|
| prob. | ½ | ½ |

If x=1, f(x) =

| value | 0 | 1 |
|---|---|---|
| prob. | ⅓ | ⅔ |

$$\left[f(x)\right] = \begin{bmatrix} 1/2 & 1/3 \\ 1/2 & 2/3 \end{bmatrix} \left[x\right]$$

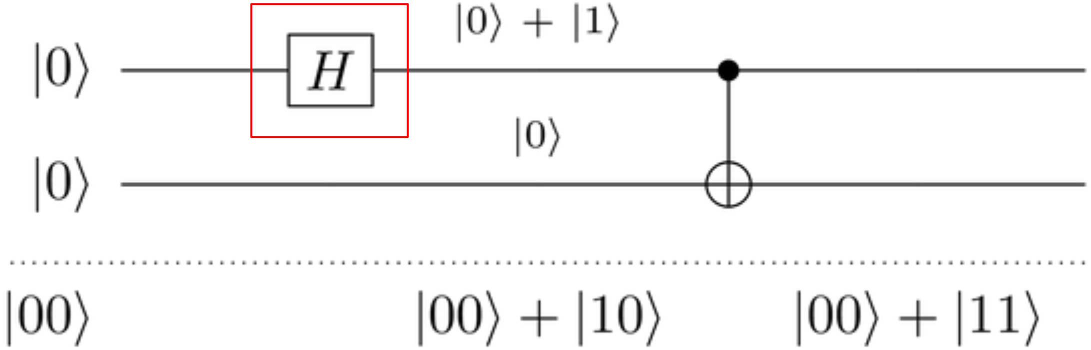
Multiplication by a stochastic matrix

# Qubit operation

Linear operation

Multiplication of state vector by a unitary (L2 length-preserving complex) matrix

```
var b1 = 0
var b2 = 0
var pair1 = (b1,b2)

b1 = randomize (b1)
var pair2 = (b1,b2)
```




$$\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$$

Hadamard gate "H"

$|0\rangle \rightarrow \sqrt{(½)}\,|0\rangle + \sqrt{(½)}\,|1\rangle$

$|1\rangle \rightarrow \sqrt{(½)}\,|0\rangle - \sqrt{(½)}\,|1\rangle$

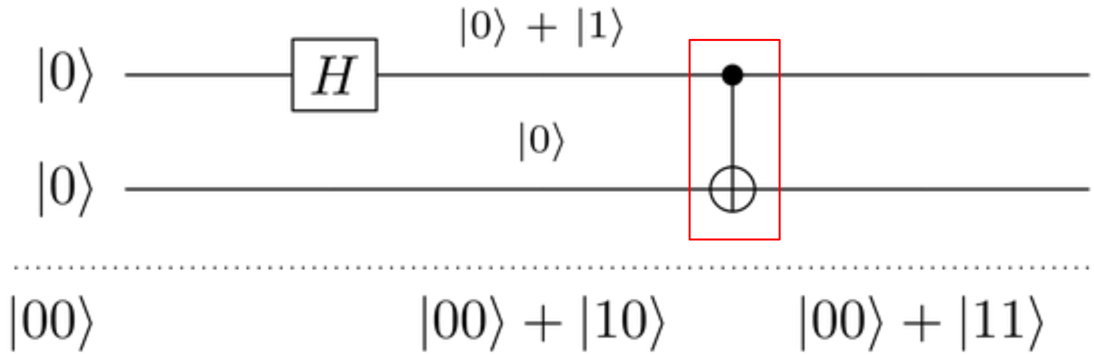
Exercise: $a|0\rangle + b|1\rangle \rightarrow ?$

# Qubit operation

Linear operation

Multiplication of state vector by a unitary (L2 length-preserving complex) matrix

```
var b1 = 0
var b2 = 0
var pair1 = (b1,b2)

b1 = randomize (b1)
var pair2 = (b1,b2)

if b1=0, b2 = b2
if b1=1, b2 = 1-b2
var pair3 = (b1,b2)
```




controlled-NOT gate "CNOT"

$|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$ $\quad$ $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$

$|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$ $\quad$ $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

Exercise: $a|0\rangle|0\rangle + b|1\rangle|0\rangle \rightarrow ?$

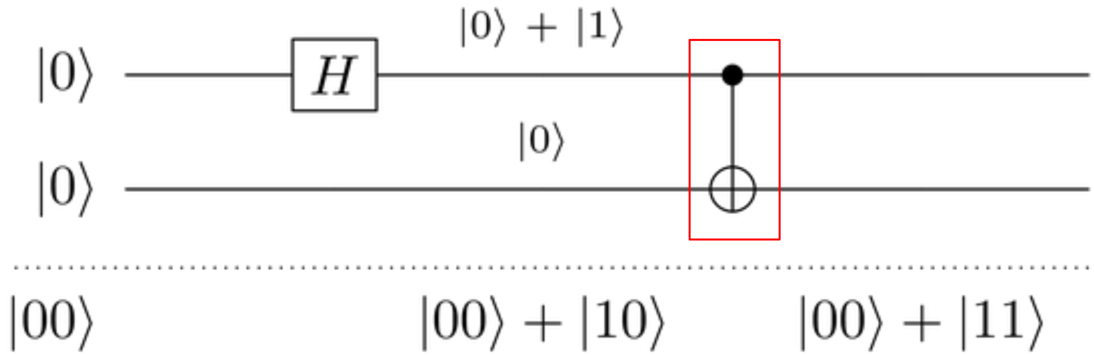
# Qubit operation

Linear operation

Multiplication of state vector by a unitary (L2 length-preserving complex) matrix

```
var b1 = 0
var b2 = 0
var pair1 = (b1,b2)

b1 = randomize (b1)
var pair2 = (b1,b2)

if b1=0, b2 = b2
if b1=1, b2 = 1-b2
var pair3 = (b1,b2)
```

b1 and b2 are always same




$$\frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$$

Measurement outcomes in 00-01-11-10 basis

- $|0\rangle|0\rangle$ with probability ½
- $|1\rangle|1\rangle$ with probability ½

**Entanglement**

1st and 2nd qubits are always same !

# First magic

$$|0\rangle \xrightarrow{H} |+\rangle \xrightarrow{H} |0\rangle$$

$|+\rangle = \sqrt{(½)}\,|0\rangle + \sqrt{(½)}\,|1\rangle$

- Observe $|0\rangle$ with probability ½
- Observe $|1\rangle$ with probability ½

$|+\rangle$ is not $|0\rangle$ **OR** $|1\rangle$

Suppose $|+\rangle$ is a state that is randomly chosen between $|0\rangle$ and $|1\rangle$ with equal probability.

- $|0\rangle$ with prob. ½
  - After 2$^{nd}$ H is applied…
  - $|0\rangle \rightarrow \sqrt{(½)}\,|0\rangle + \sqrt{(½)}\,|1\rangle$
  - Observation will yield
  - $|0\rangle$ and $|1\rangle$ with probability ½
- $|1\rangle$ with prob. ½
  - After 2$^{nd}$ H is applied…
  - $|1\rangle \rightarrow \sqrt{(½)}\,|0\rangle - \sqrt{(½)}\,|1\rangle$
  - Observation will yield
  - $|0\rangle$ and $|1\rangle$ with probability ½
- Overall …
  - Prob. of observing $|0\rangle$ = ¼+¼=½
  - Prob. of observing $|1\rangle$ = ¼+¼=½

$|+\rangle$ is a **superposition** of $|0\rangle$ and $|1\rangle$

# state evolution operation

Specify action only on basis states
Linearly extrapolate on all other states
Unitary, hence reversible

Qubits cannot be copied!

```
is the code that does the bubble sort.
for (int i = ar.length - 1; i > 0; i--) {
    for (int j = 0; j < i; j++) {
        if (ar[j] > ar[j + 1]) {
            temp = ar[j];
            ar[j] = ar[j + 1];
            ar[j + 1] = temp;
        }
```

# Searching

Input is an binary array A of size 100.

Find any index $i$ for which A[$i$] = 1

```
var b = random index from { 1 … 100 }
// b = 1 with prob. 0.01
// b = 2 with prob. 0.01
              …
// b = 100 with prob. 0.01

var c = A[b]
if c = 1:
      print (b)
else:
      print (“not found”)
```

Makes 1 probe to A
Success probability = m/100
Where, m = number of 1s in A

Run the code k times.

Prob. of getting “not found” in all 10 runs = $(1 - m/100)^k$

Prob. of finding good index ≅ k*(m/100) (linear in k)

k = 100/m iterations ensure success

# Quantum searching

Input is an binary array A of size 100.

Find any index $i$ for which A[$i$] = 1

Define UA gate on two “registers”:

1. 10-qubit register 1 to store position
2. 1-qubit register 2 to store value

|p⟩ |0⟩ → |p⟩ | A[p] ⟩
|p⟩ |1⟩ → |p⟩ | 1-A[p] ⟩

1 probe to A

var |b⟩ = 1/10 [ |1⟩ + |2⟩ + … + |100⟩ ]

Apply UA on |b⟩|0⟩ ?

1 probe to A

1/10 [ |1⟩|A[1]⟩ + |2⟩|A[2]⟩ + … ]

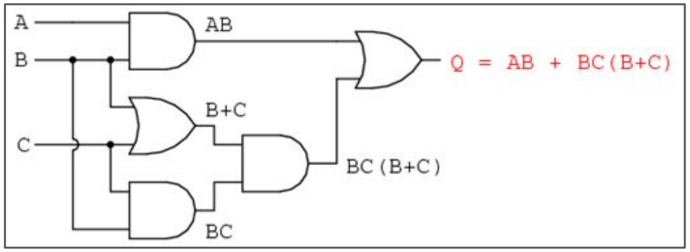
1/10 [ |1⟩|A[1]⟩ + |2⟩|A[2]⟩ + … ]
Observation yields any |b⟩|A[b]⟩ with probability 1/100

Not better than classical

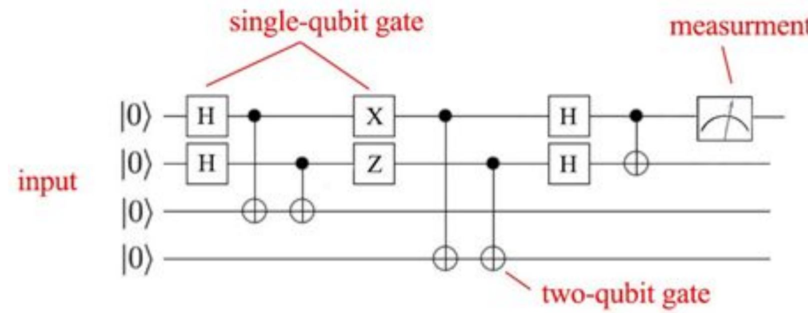
Do not observe (yet).
Run Grover’s search algorithm on this state and then observe.

Constant success probability can be achieved using √(100/m) probes.

# Programming a QC




Early style of designing (efficient) solvers



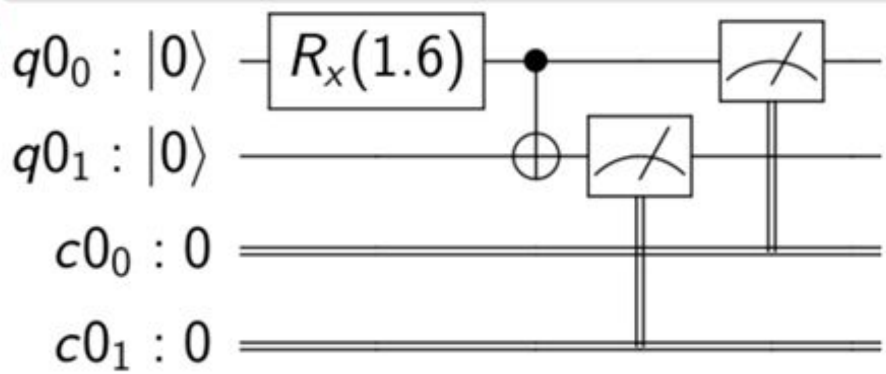

Current style of designing efficient quantum solvers

In [7]:

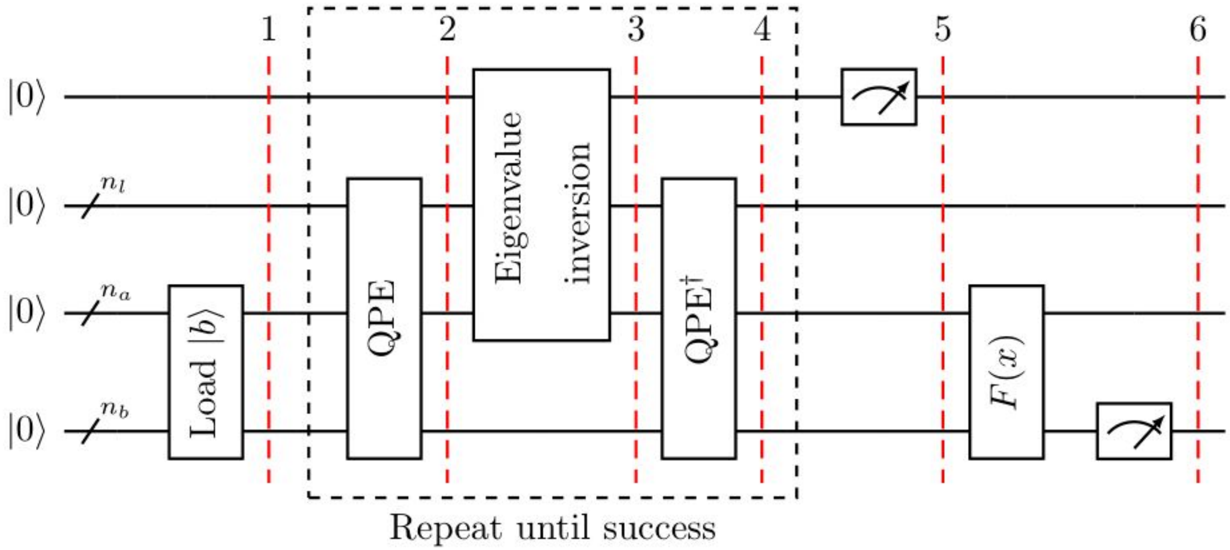
```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from qiskit.tools.visualization import circuit_drawer
import numpy as np

qr = QuantumRegister(2)
cr = ClassicalRegister(2)
qp = QuantumCircuit(qr,cr)

qp.rx( np.pi/2,qr[0])
qp.cx(qr[0],qr[1])

qp.measure(qr,cr)

circuit_drawer(qp)
```

High-level wrapper and subroutines to run Grover's search, etc.

Out[7]:



# Linear system of equations



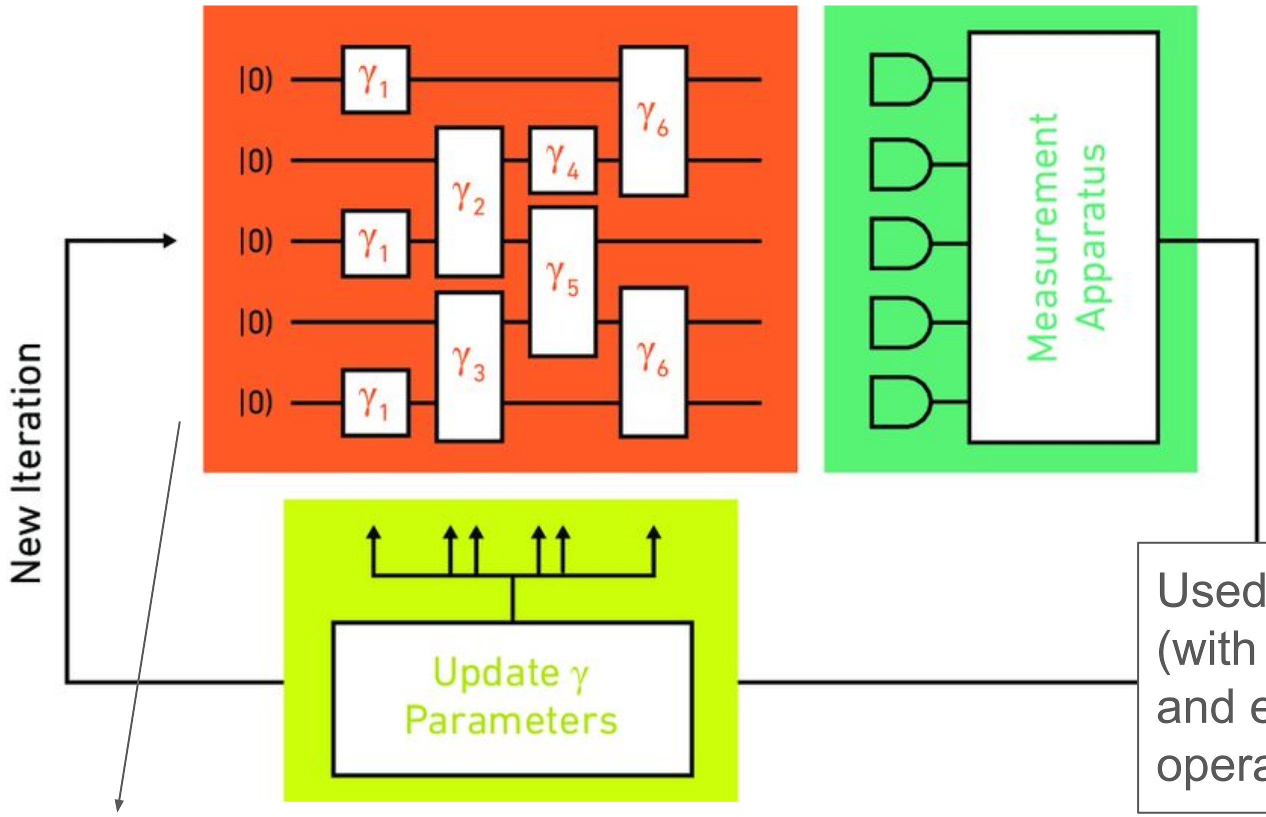

Given N equation in the form of Ax = b.

Gives the solution vector x

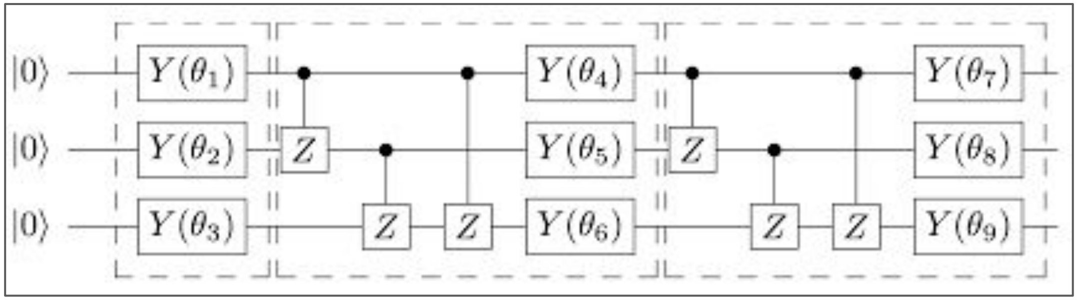
- Classical : conjugate gradient descent ~ O(N)
- Quantum algorithm by HHL : O( log(N))

Gives a random sample from the solution vector x

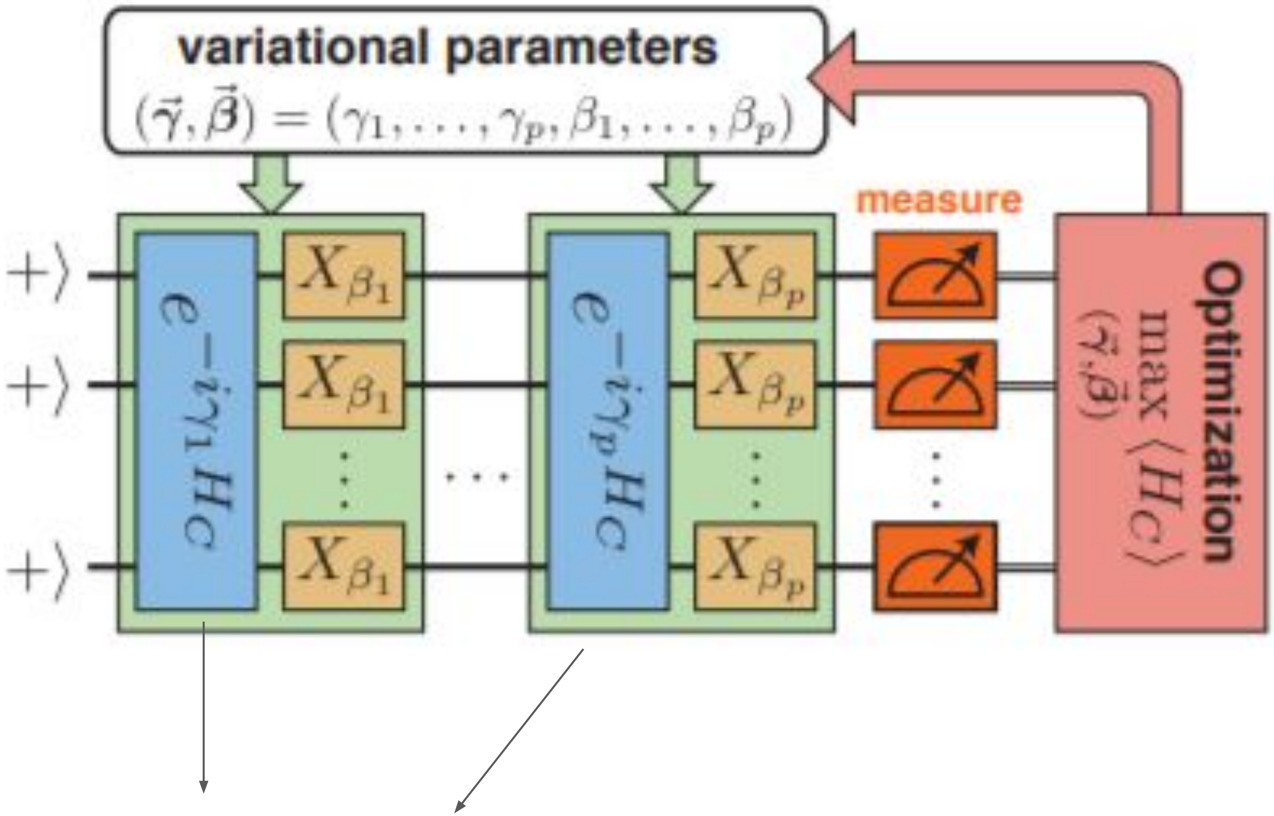
# Variational Quantum Eigensolver




Used to obtain eigenvalue (with lowest absolute value) and eigenvector of an operator

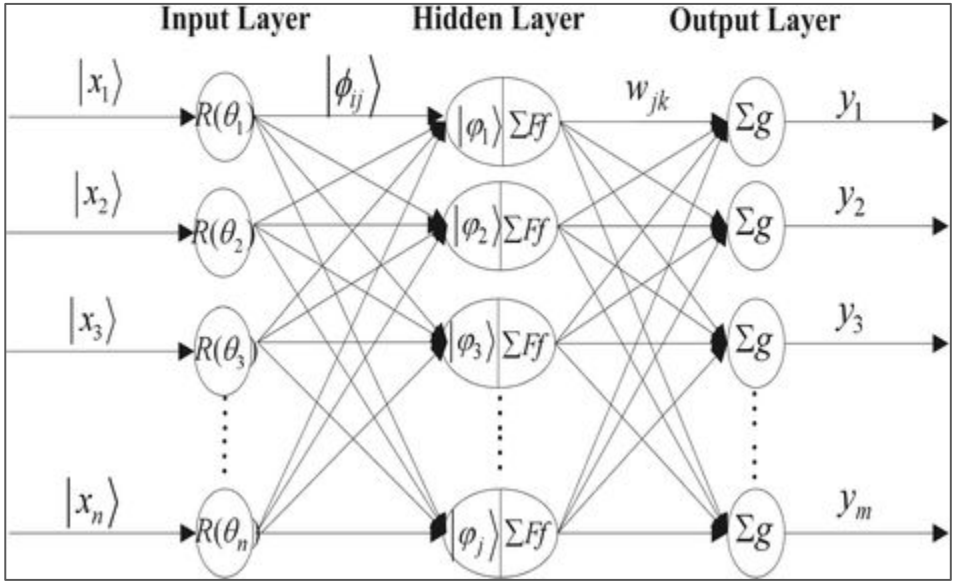


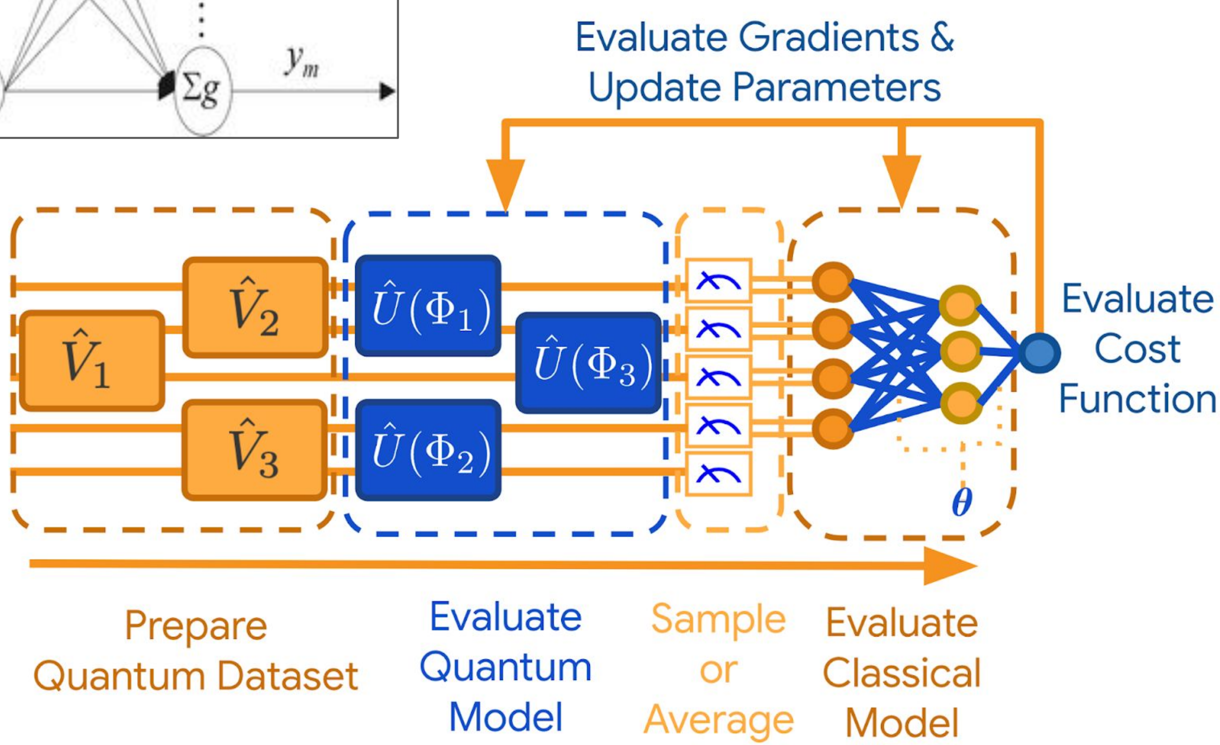
# Quantum Approximate Optimization Algorithms



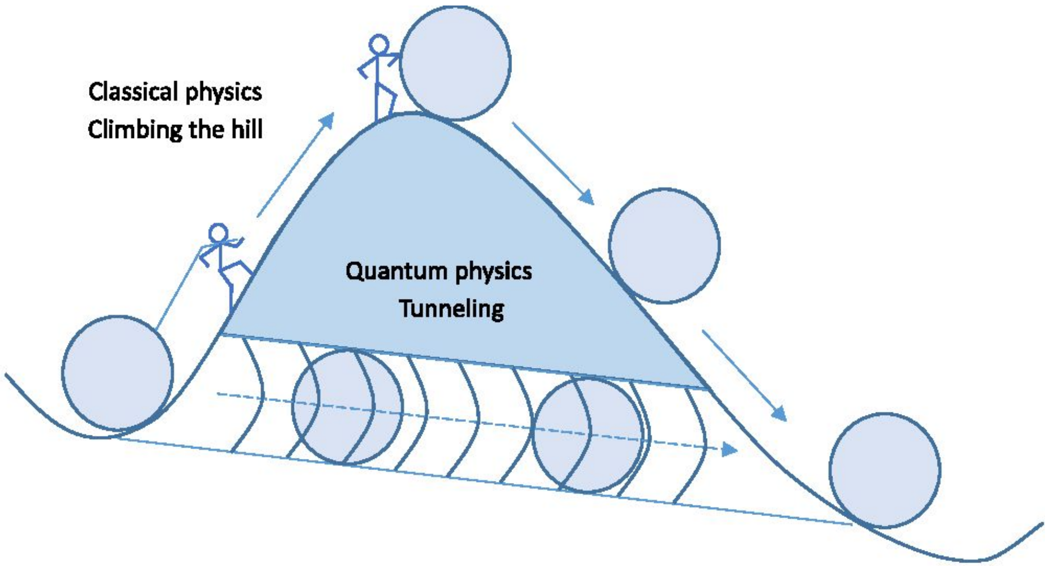

Objective function of the optimization problem

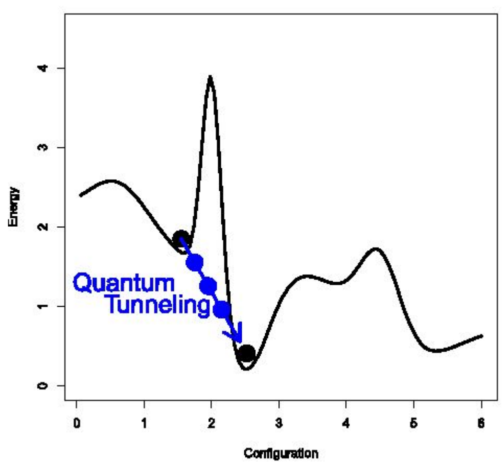
# Quantum neural network

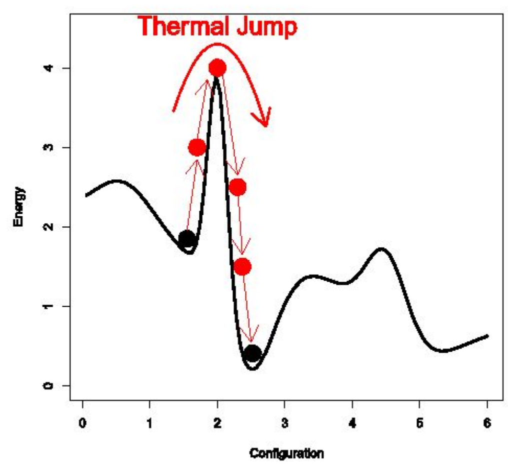


# Quantum annealing



# Promises and Prospects

- **1981** Feynman proposed quantum computer to efficiently simulate many-body quantum systems

moonshot

- **1984** Bennett and Brassard designed quantum protocol for secret key sharing
- **1991** Another QKD protocol by Ekert

BB84 running on 2000KM fiber-optic cable in China

QKD networks : DARPA, Tokyo, Vienna, Japan, …

Q→NU

Quintessence labs

MagiQ

IDQ FROM VISION TO TECHNOLOGY

Qubitekk

QUANTUMXCHANGE

Qasky

# Promises and Prospects

Technology not clear

- **1985** Deutsch proposed a general purpose programmable quantum computer
- **1992** Deutsch and Jozsa solved a (toy) problem in half the time taken by the best classical algorithm

Quantum supremacy race

- **1993** Simon designed algorithm that is efficient on quantum computer but inefficient classically
- **1996** Grover designed algorithm to search in a database of N elements using $\sqrt{N}$ "lookups" (classical best is N/2)

1994 $\sqrt{N}$ is the best

… better-than-classical algorithms for problems on numbers, graphs, geometric objects, strings, statistics, communication, data structures, … but limited speedup

# Promises and Prospects

**2001** 15 factored using $10^{18}$ identical molecules

Requires high-precision

- **1994** Shor designed algorithms to factor n-bit number in $O(n^2)$ time (classical best is $O(\exp(n^{1/3}))$)
- **1995** Shor and Steane designed error-correcting codes

Oops!

- **1998** Gottesman and Knill showed how to **efficiently** simulate certain quantum algorithms **classically**
- **2017** Microsoft releases 40-qubit classical simulator

ODE, PDE, machine learning

Quantum ML

- **2009** Harrow+ designed linear system "solver"

Attacks on cryptography

- **2015** Grassl showed 3000-7000 qubits needed to search AES key using Grover's algorithm
- **2016** Google simulated a Hydrogen molecule with 9 qubits

PORTFOLIO OPTIMIZATION on D-Wave

SIMULATION OF HYDROGEN MOLECULE by Google
(simulation of quantum-mechanical systems was the initial motivation of Richard Feynman to propose a quantum computer)

TRAFFIC OPTIMIZATION & EXPLORE MATERIAL STRUCTURE FOR E-VEHICLE BATTERY by Volkswagen Group and Google

* Maybe you believe in the experiments yet disagree with the meaning

# Summary

Quantum mechanics that drive quantum computing is mysterious

But if you are a believer … (*)

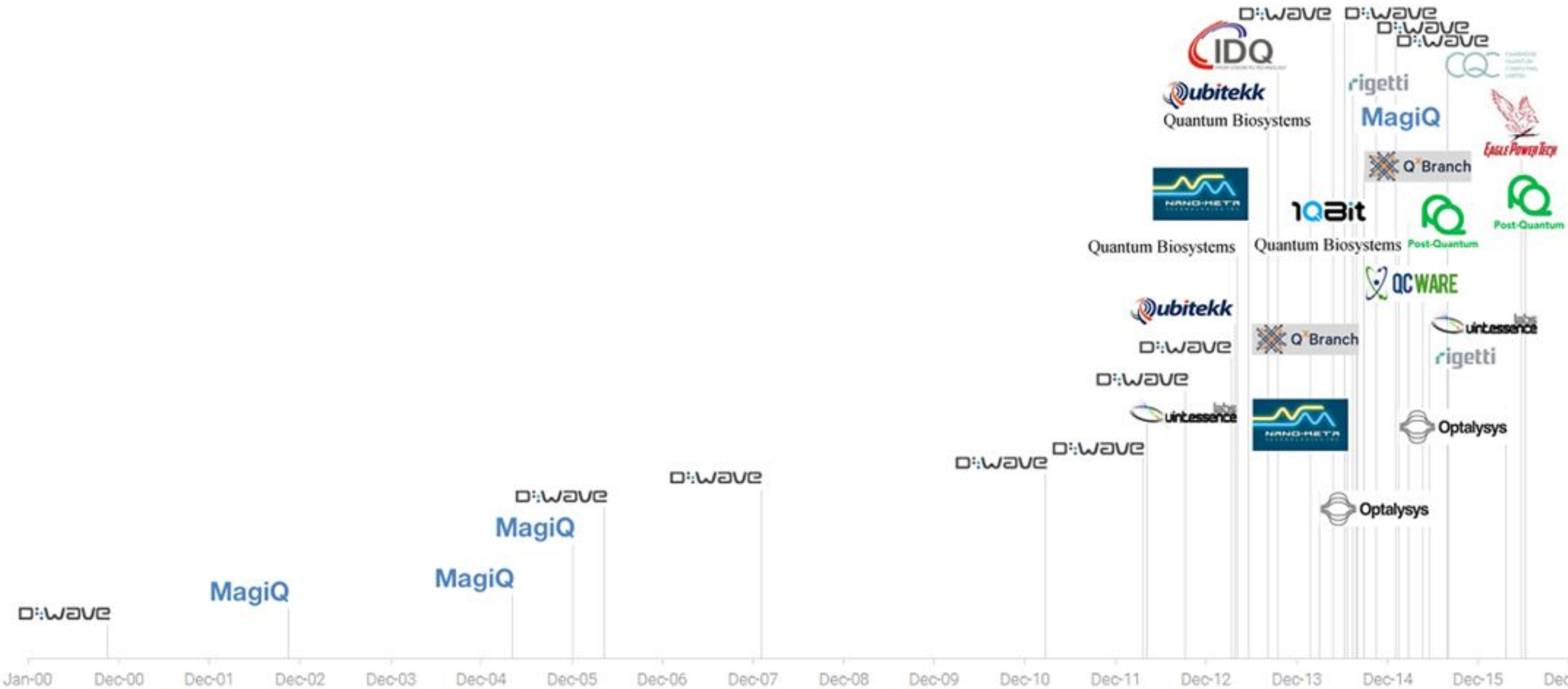
Quantum algorithm design and analysis possible using knowledge of algorithms, probability and linear algebra.

Thanks to physicists, material scientists, engineers, mathematicians, … in universities, R&D labs and corporates ...

These algorithms can be implemented on real quantum computers and experimented with.

Too early to say how and where QC will become useful …

Just the right time to enter the game.




# Thank you for listening.
# Questions?

*dbera@iiitd.ac.in*