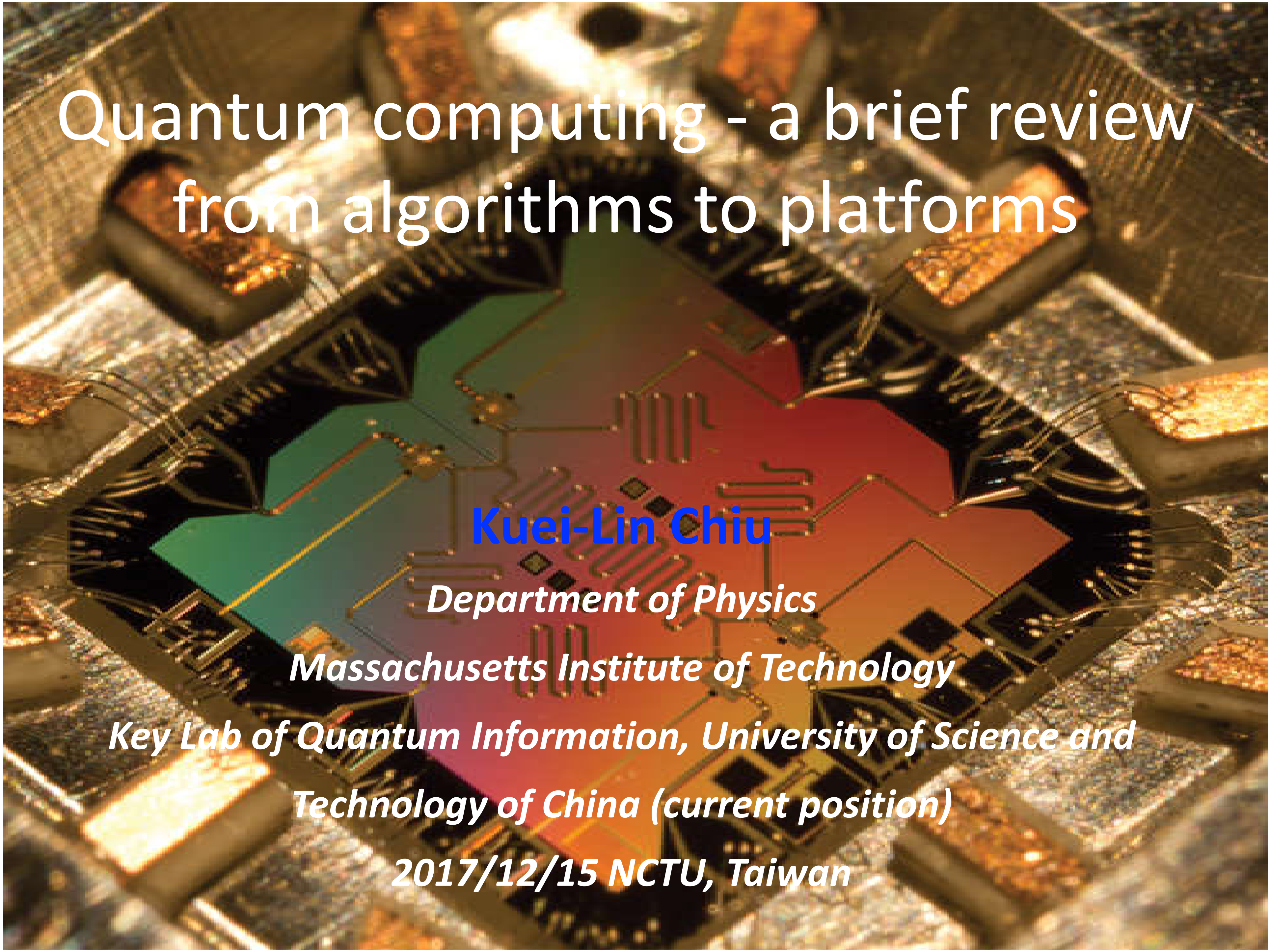


Quantum computing - a brief review from algorithms to platforms

Kuei-Lin Chiu

Department of Physics

Massachusetts Institute of Technology

Key Lab of Quantum Information, University of Science and Technology of China (current position)

2017/12/15 NCTU, Taiwan

Outline

- Introduction to quantum computing

- The Algorithms
  - Deutsch-Jozsa algorithm (judging)
  - Grover’s algorithm (searching)
  - Shor’s algorithm (factoring)

- The platforms
  - Superconducting circuits
  - Semiconductor quantum dots

- Progress and prospect

What is quantum computing?

- Quantum computing uses the properties of quantum mechanics to design hardware and algorithms, and to perform certain calculations which are usually difficult for classical computers to complete
- The unit of quantum computing is quantum bits (“qubits”), in comparison with the “bits” used in classical computing

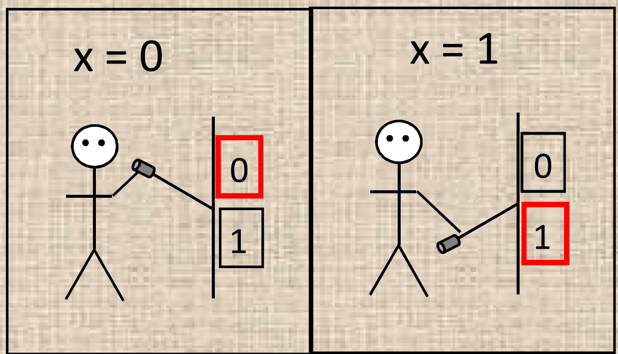
Classical Computation

Data unit: bit

● = ‘1’ ○ = ‘0’

Valid states:

x = ‘0’ or ‘1’



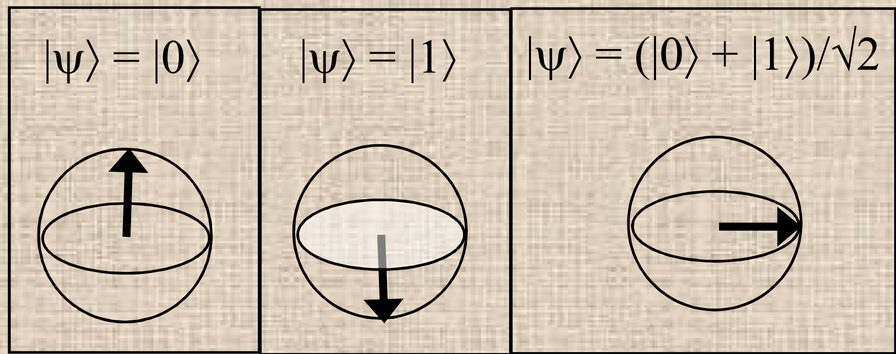
Quantum Computation

Data unit: qubit

⇧ $=|1\rangle$ ⇩ $=|0\rangle$

Valid states:

$$|\psi\rangle = c_1|0\rangle + c_2|1\rangle$$



When we measure a quantum state, it can be quite different

Classical Computation	Quantum Computation
Measurement: deterministic	**Measurement: stochastic**

Classical Computation — Measurement: deterministic

State	Result of measurement
x = '0'	'0'
x = '1'	'1'

Quantum Computation — Measurement: stochastic

State	Result of measurement
$\lvert\psi\rangle = \lvert 0\rangle$	'0'
$\lvert\psi\rangle = \lvert 1\rangle$	'1'
$\lvert\psi\rangle = \frac{\lvert 0\rangle - \lvert 1\rangle}{\sqrt{2}}$	'0' 50% '1' 50%

Single qubit $\lvert 0\rangle, \lvert 1\rangle$

Hilbert space

$$\mathcal{H}_2 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Arbitrary state

$$\lvert\psi\rangle = c_1\lvert 0\rangle + c_2\lvert 1\rangle = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

Operator

$$U\lvert\psi\rangle = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

Two qubits $\lvert 00\rangle, \lvert 01\rangle, \lvert 10\rangle, \lvert 11\rangle$

$$\mathcal{H}_2^{\otimes 2} = \mathcal{H}_2 \otimes \mathcal{H}_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

$$\lvert\Psi\rangle = \begin{matrix} c_1\lvert 00\rangle + c_2\lvert 01\rangle + \\ c_3\lvert 10\rangle + c_4\lvert 11\rangle \end{matrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}$$

$$U\lvert\Psi\rangle = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}$$

Operation of qubits is through quantum gates

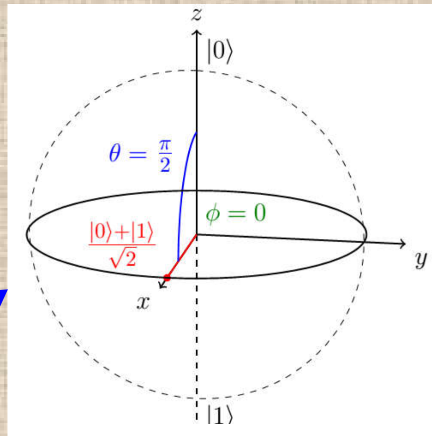
one-qubit gate

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

Hadamard gate (rotate state around y by π/2)

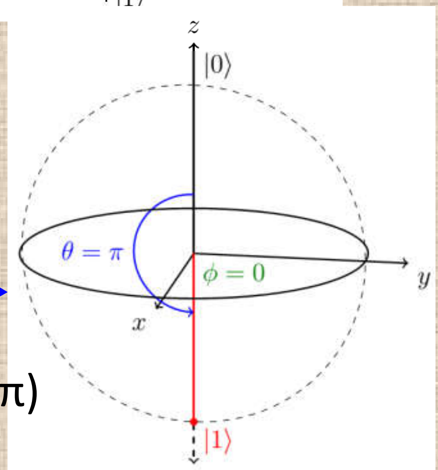
$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$



$$\overset{1}{\alpha|0\rangle} + \overset{0}{\beta|1\rangle} \;—\boxed{X}—\; \overset{0}{\beta|0\rangle} + \overset{1}{\alpha|1\rangle}$$

Pauli-X gate (rotate state around x by π)

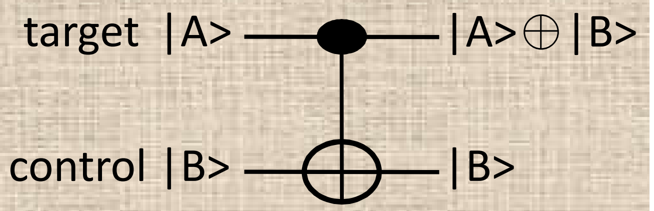


(a) Pauli-X

$$\overset{1}{\alpha|0\rangle} + \overset{1}{\beta|1\rangle}\ \overset{|x>}{} \;—\boxed{Z}—\; \overset{1}{\alpha|0\rangle} - \overset{1}{\beta|1\rangle}\ \overset{|-x>}{}$$

Pauli-Z gate (rotate state around z by π)

two-qubit gate



Controlled-NOT gate

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

Input		Output	
A	B	A'	B'
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Arbitrary quantum logic gate can be decomposed into one-qubit and two qubit gates

Quantum algorithm

- The difference between polynomial-time and exponential-time algorithm

If the dimension of a question is ***n***, and the steps (or said the time) required to solve this question is ***T(n)***, which can be polynomial or exponential function of *n*

	n=10	n=20	n=30	n=40	n=50	n=60
n	10^{-5}s	2×10^{-5}s	3×10^{-5}s	4×10^{-5}s	5×10^{-5}s	6×10^{-5}s
n^2	10^{-4}s	4×10^{-4}s	9×10^{-4}s	16×10^{-4}s	25×10^{-4}s	36×10^{-4}s
n^3	10^{-3}s	8×10^{-3}s	27×10^{-3}s	64×10^{-3}s	1.25×10^{-1}s	2.16×10^{-1}s
2^n	$1024 * 10^{-6}$s ~10^{-3}s	~1s	~1000s	12.7 days	35.7 years	366 centuries
3^n	5.9×10^{-2}s	58 mins	6.5 years	3855 centuries	2×10^8 centuries	1.3×10^{13} centuries

Grow slowly

*The age of earth is roughly $4.5 * 10^7$ centuries

Grow insanely fast

In math, if the complexity of a problem grow exponentially with its input dimension, we refer this problem to **NP** (non-deterministic polynomial)

Deutsch-Jozsa Algorithm

- Used to determine whether a function is "constant" or "balanced"
- For n-bit input $x=\{0,1\}^n$ (meaning (0 or 1, 0 or 1…,0 or 1), in total 2^n possible combinations):

➢ f(x)=0 (or 1) for all x, it is called "constant"

➢ f(x)=0 for half of x and f(x)=1 for the other half, it is called "balanced"

- In classical algorithm it takes $T(2^n)$ steps to verify while in D-J algorithm it only takes T(n)

A simplified example for classical algorithm: if x=1,2,3…,8 (i.e. n=3); you need to try f(1), f(2)…each by each. Let's say you tried the first half input and found f(1)=f(2)=f(3)=f(4)=0; then you need to try the 5th input; if f(5)=0 then f(x) is "constant", however if f(5)=1 then f(x) is "balanced". So the maximal times of tries is 5 if you are unlucky. In general, for a n-bit input, you need to try $(2^n/2)+1$ times.

A simplified example of D-J Algorithm

- For a 1-bit (x=0, 1) input: if f(0)=f(1), f(x) is constant; if f(0)≠f(1), f(x) is balanced. In classical algorithm, you need to try 2 times to find out.
- See how quantum algorithm works differently. If we define an "Oracle" (applying on a 2-qubit state, however the 2^{nd} qubit is an ancilla and will be disregarded in the end) **U_f: |x>|y> → |x>|y⊕f(x)>** and let the input |y> be a superposition state (|0>—|1>):

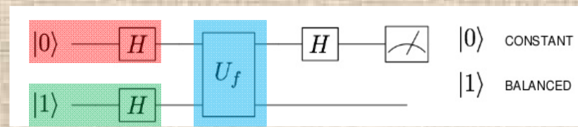
  U_f: |x>(|0>—|1>) → |x>(|0⊕f(x)>—|1⊕f(x)>)= $(-1)^{f(x)}$ |x>(|0>—|1>)

  if f(x)=0 →|x> (|0>-|1>) ; if f(x)=1 → |x> (|1>-|0>)

The effect of Oracle is simply adding a phase factor related with f(x)

- Let |x>=|0>+|1>, and operate Oracle U_f: (|0>+|1>)(|0>—|1>) → [$(-1)^{f(0)}$ |0> +$(-1)^{f(1)}$ |1>](|0>—|1>)

  First qubit



- Now project the first qubit onto the basis of |±>=(|0>±|1>):

  If we get |+>, meaning f(0)=f(1)=0 (or 1), the function is constant

  If we get |->, meaning f(0)=0, f(1)=1, the function is balanced

➢ We only need to perform Oracle 1 time to determine the function

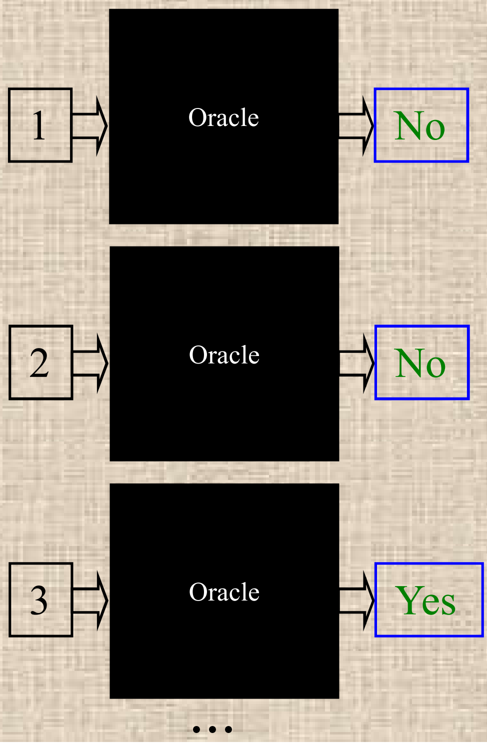
➢ D-J algorithm is not useful in practical application (i.e., for 2-qubit case, if f(00)=0, f(01)=f(10)=f(11)=1, then f(x) is neither constant nor balanced). However, it serves as a good example to see how quantum algorithm operates differently than classical one

Grover's Search Algorithm

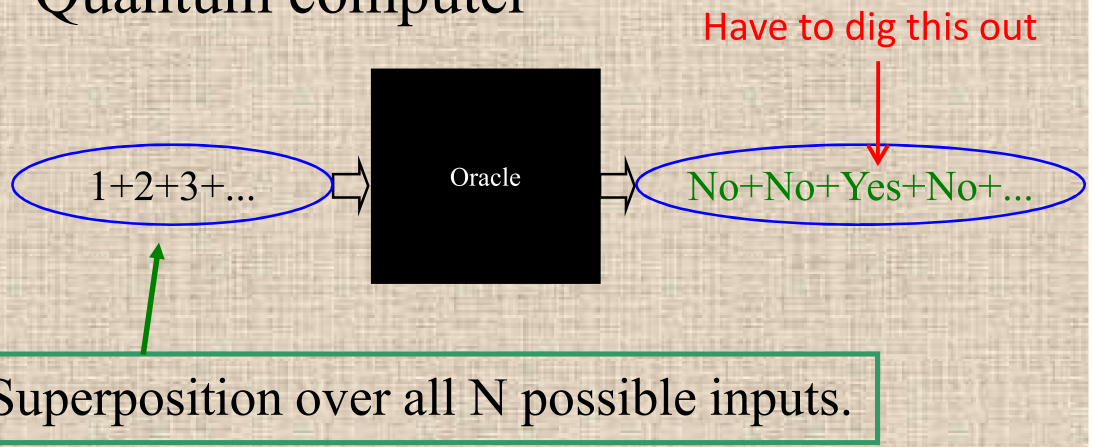
- Imagine we are looking for the solution to a problem with *N* possible inputs. We have a black box (or "oracle") that can check whether a given answer is correct.

Question: I'm thinking of a number between 1 and 100. What is it?

Classical computer

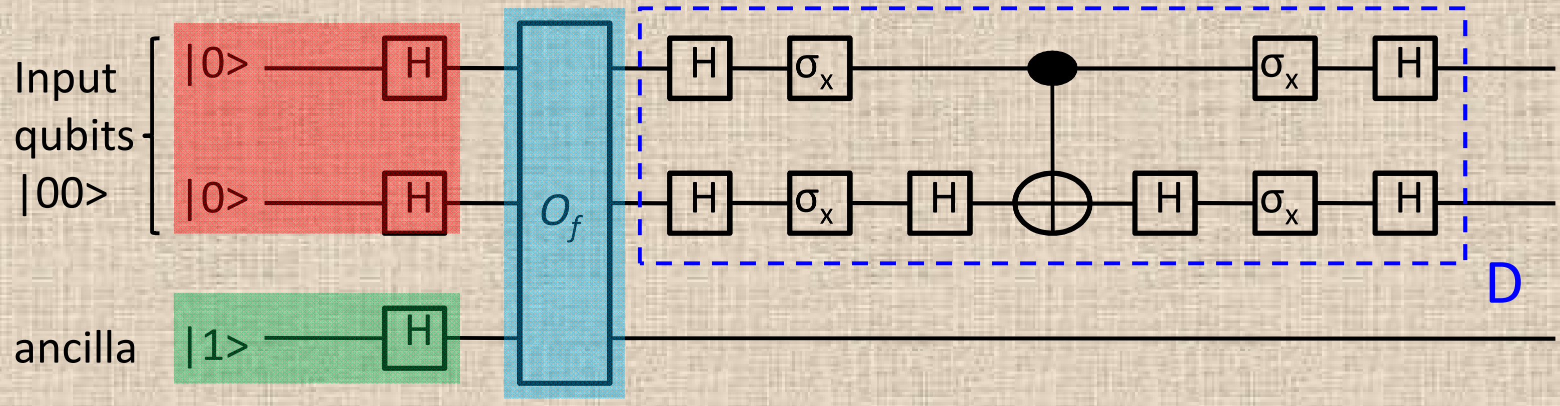


Quantum computer



Using Grover's algorithm, a quantum computer can find the answer in √N queries!

A simple example (search 1 out of 4)



- For a 2-qubit input, we have |00>, |01>, |10>, |11>. Let's say |10> is the answer, so f(00)=f(01)=f(11)=0 but f(10)=1.
- H-gate prepare |0> → 1/√2(|0>+|1>) and |1> → 1/√2(|0>-|1>) , so the total wavefunction before the oracle is
- $$\underbrace{\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)}_{\text{input}}\frac{1}{\sqrt{2}}\underbrace{(|0\rangle - |1\rangle)}_{\text{ancilla}}$$
- Then the whole wavefunction goes through oracle, because of the $(-1)^{f(x)}$ phase, there will be a minus sign in front of the answer state:

$$\frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

When you measure the four basis, they still give you same probability. So the key is to transform the phase difference in |10> into an amplitude difference for us to measure

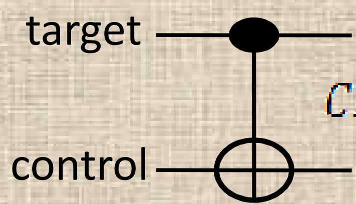
This can be simply done by a matrix D composed of different quantum gates

$$D = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \Rightarrow D\frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Apply on wave function

|00>
|01>
|10> ....the answer
|11>

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

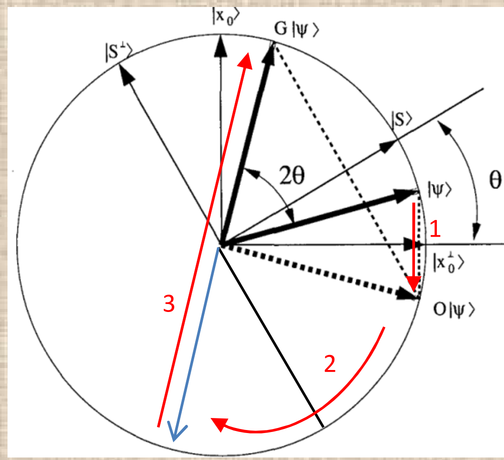


$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boxed{\sigma_x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$ Pauli-X gate

In classical search:

$$\bar{N} = \sum_{i=1}^{N} \frac{1}{N} i = \frac{1}{N}\sum_{i=1}^{N} i = \frac{N+1}{2} \approx \frac{N}{2}$$



$|X_0>$: answer state
$|\Psi>$: initial state
|S>: the H|00> state
($|\Psi>$= |S> in our case)
1→2→3: a Grover's iteration rotates $|\Psi>$ by 2θ (1 -> oracle, 2&3 -> matrix D), θ~1/√N

In Grover's search, the average times of queries is √N

Shor's algorithm

To factor an odd integer *N*

(let's say 21)

- Integer factorization is a NP, and forms the basis of RSA cryptosystem (Given $N=pq$, find *p* and *q*).
- 1. Pick up an integer *a* (said 2), $1<a<N$. Define a function $f(x)=a^x \bmod N$
- 2. Find the periodicity *r* of *f(x)* (2^x mod 21 is 1,2,4,8,16,11,1,2..; so *r* for 2^x is 6)
- 3. If *r* is odd, go back to step1 and choose another *a*. If not, compute *f(r/2)* (*f(3)=8*)
- 4. $\gcd(a^{r/2} + 1, N)$ and $\gcd(a^{r/2} - 1, N)$ are both nontrivial factors of N. We are done. (gcd of 9 and 21 is 3, gcd of 7 and 21 is 7; 3 and 7 are the prime factors of 21)

Only step 2 is performed by quantum computing, the rest of the steps are still classical

- The quantum part of Shor algorithm is a bit complicated. However, let's get a feeling on how to search period in a given function using quantum algorithm
- F(x)= ½(cos(πx)+1), x could be the states span by 3 qubits (x=|0>,|1>,..,|7>). When x is even, f(x)=1; when x is odd, f(x)=0. The period of f(x) is 2. Our mission is to confirm that

Consider two registers $\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle = \frac{1}{\sqrt{8}}\left[|0\rangle\underbrace{|f(0)\rangle}_{1} + |1\rangle\underbrace{|f(1)\rangle}_{0} + |2\rangle\underbrace{|f(2)\rangle}_{1} + \cdots + |7\rangle\underbrace{|f(7)\rangle}_{0}\right]$ (n=3)

- Measure the second register, ex: if we get |0>, then the wavefunction collapses to |φ>=½(|1>+|3>+|5>+|7>)|0>
- Using the quantum Fourier transform(QFT), we expand the function:

$$|x\rangle \to \frac{1}{\sqrt{8}}\sum_{k=0}^{7} e^{2\pi i k x/8}|k\rangle$$

QFT

$$\begin{aligned}|\varphi\rangle = &\left(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle + e^{\frac{i2\pi}{4}}|2\rangle + \cdots + e^{\frac{i7\pi}{4}}|7\rangle\right) \quad \longleftarrow X=1\\ &+\left(|0\rangle + e^{\frac{i3\pi}{4}}|1\rangle + e^{\frac{i6\pi}{4}}|2\rangle + \cdots + e^{\frac{i21\pi}{4}}|7\rangle\right) \quad \longleftarrow X=3\\ &+\left(|0\rangle + e^{\frac{i5\pi}{4}}|1\rangle + e^{\frac{i10\pi}{4}}|2\rangle + \cdots + e^{\frac{i35\pi}{4}}|7\rangle\right) \quad \longleftarrow X=5\\ &+\left(|0\rangle + e^{\frac{i7\pi}{4}}|1\rangle + e^{\frac{i14\pi}{4}}|2\rangle + \cdots + e^{\frac{i49\pi}{4}}|7\rangle\right) \quad \longleftarrow X=7\end{aligned}$$

Only |0> and |4> survive

Where QM plays a role: Cancel out

$$|\varphi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |4\rangle)$$

We have equal probability to measure |0> or |4>. If we get 0> we can't find the period, but if we get |4>, we can.

$$|\varphi\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} \exp\left(\frac{2\pi i x k}{r}\right)|k\frac{N}{r}\rangle$$

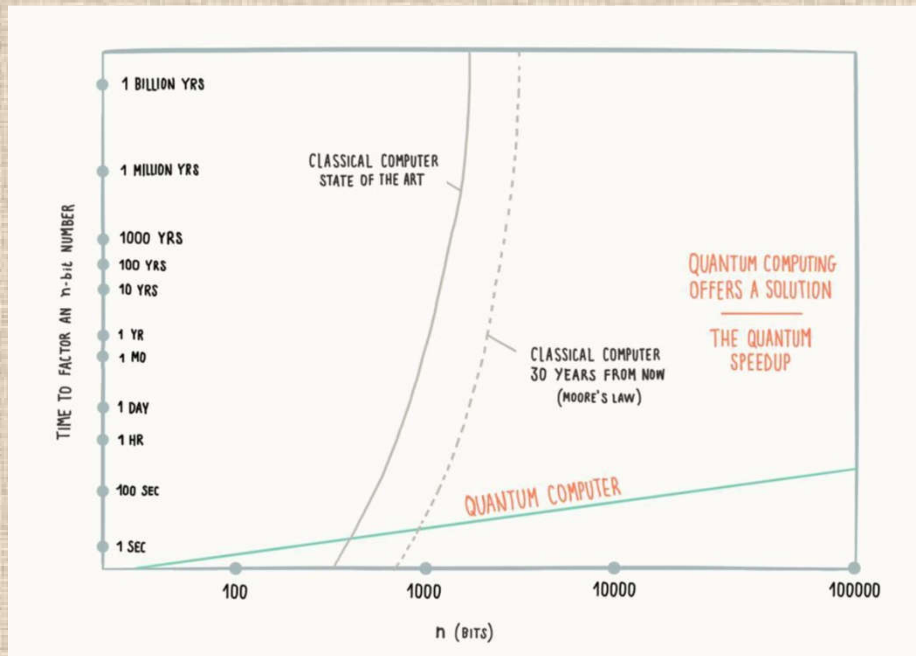
Compare with the strictly derived final state after QFT (which we skip), 4=*k(=1)N(=8)/r*, *r=2 (done)*

The power of Shor's algorithm is to utilize the superposition property of QM, which makes the "**unnecessary**" information interfere **destructively** and the "**useful**" information interfere **constructively** in FT

Factoring an integer with **n-bits**

Classical algorithm takes time **O(exp($n^{1/3}$))**

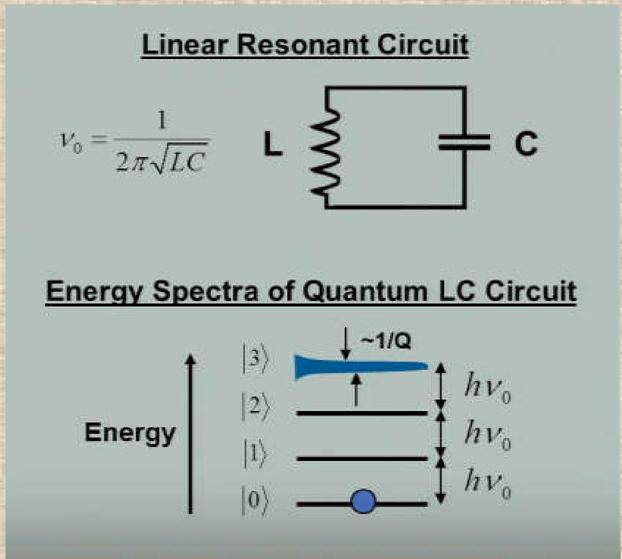
Shor's quantum algorithm takes time **O(n^2log*n*)**

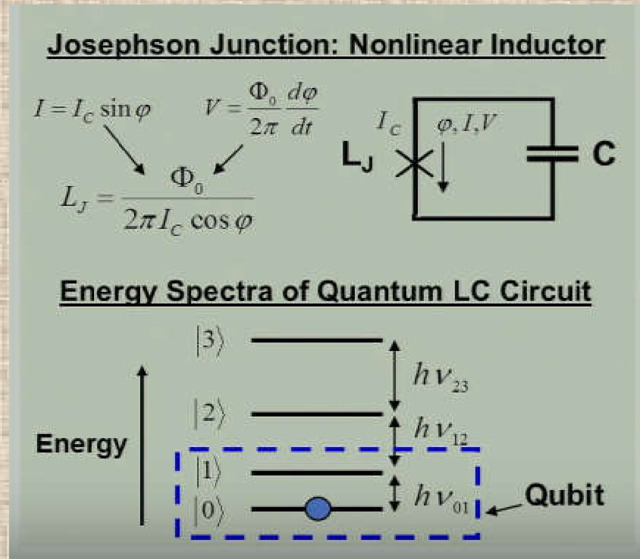


Source: New Enterprise Associates

Superconducting qubit

- A quantum LC resonator using Josephson junction as an inductor provide anharmonic states for a **two-level system**





Quantized electrical harmonic oscillator

$$H = \frac{\hat{q}^2}{2C} + \frac{\hat{\phi}^2}{2L}$$

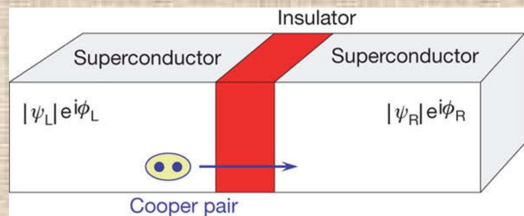
Capacitive energy

inductive energy

q → capacitor charge (momentum p)

ϕ → inductor flux/phase (position x)

Josephson junction



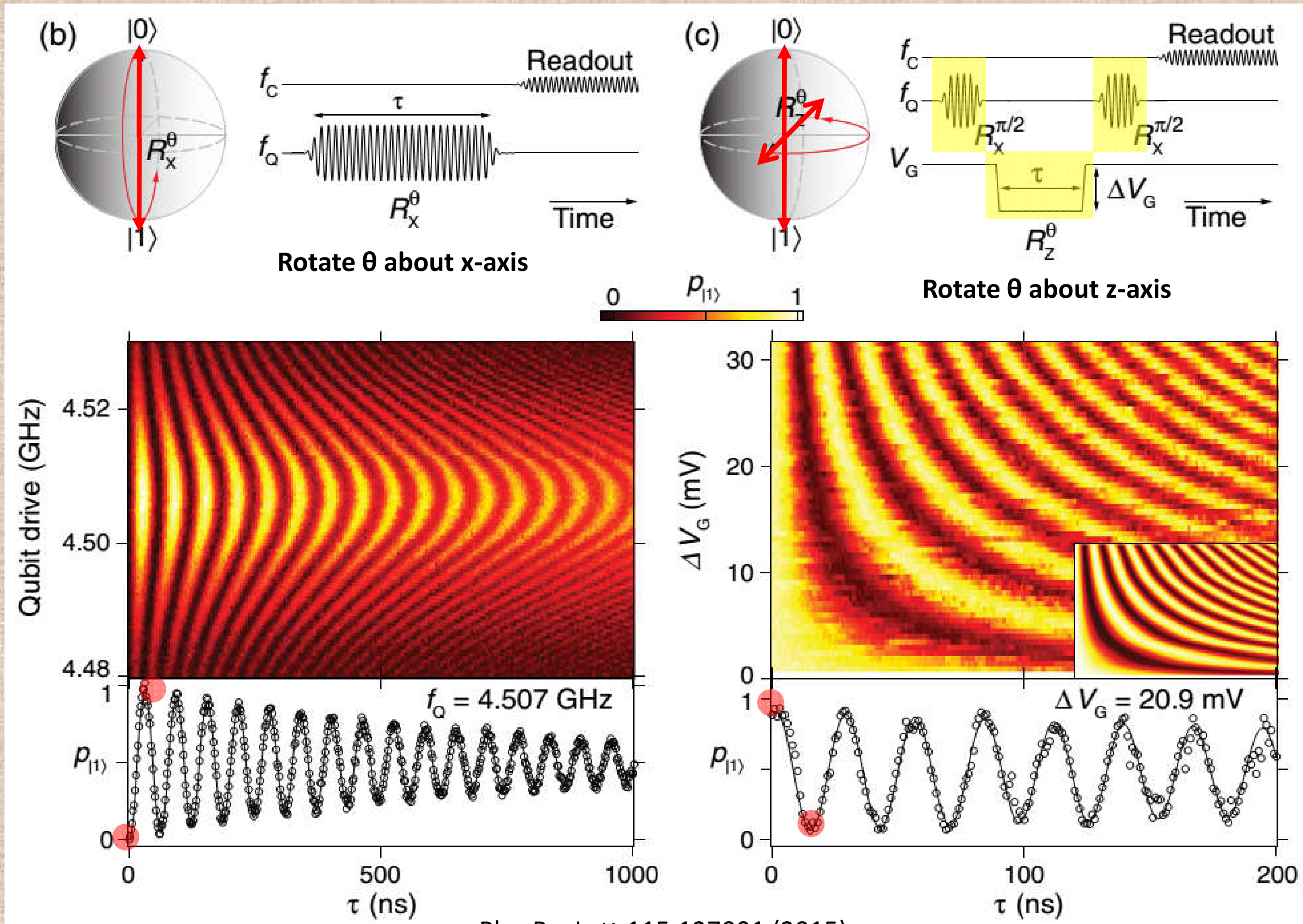
Current: $I = I_C \sin\phi$

Voltage: $V = (\Phi_0/2\pi)(d\phi/dt)$

Inductance: $V = L_J\, dI/dt$

Transmon qubit

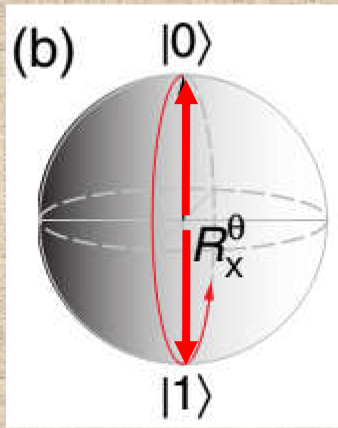
Single qubit gate operation with state-dependent cavity readout



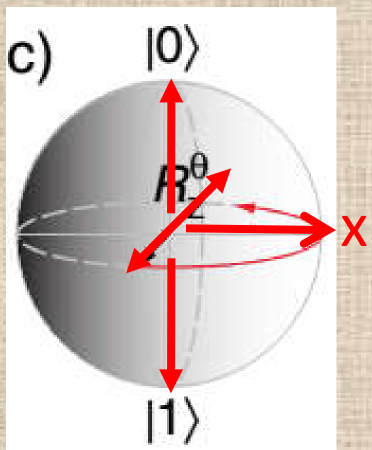
PhysRevLett.115.127001 (2015)

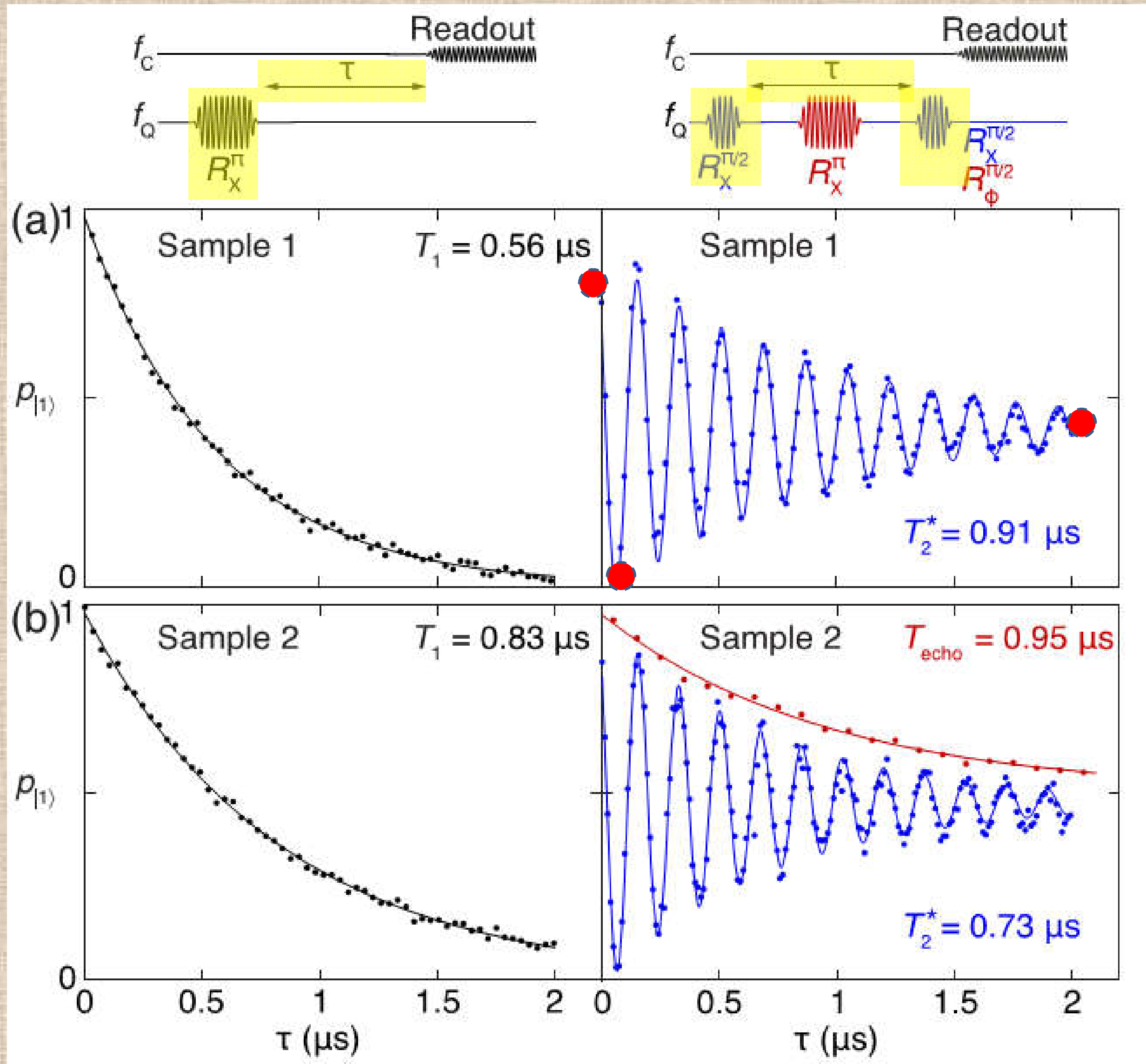
Relaxation and decoherence time

T_1



T_2*

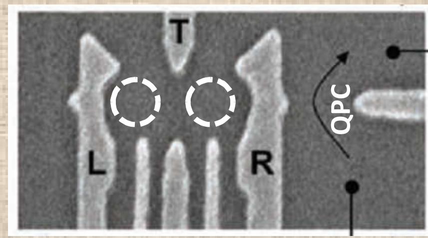




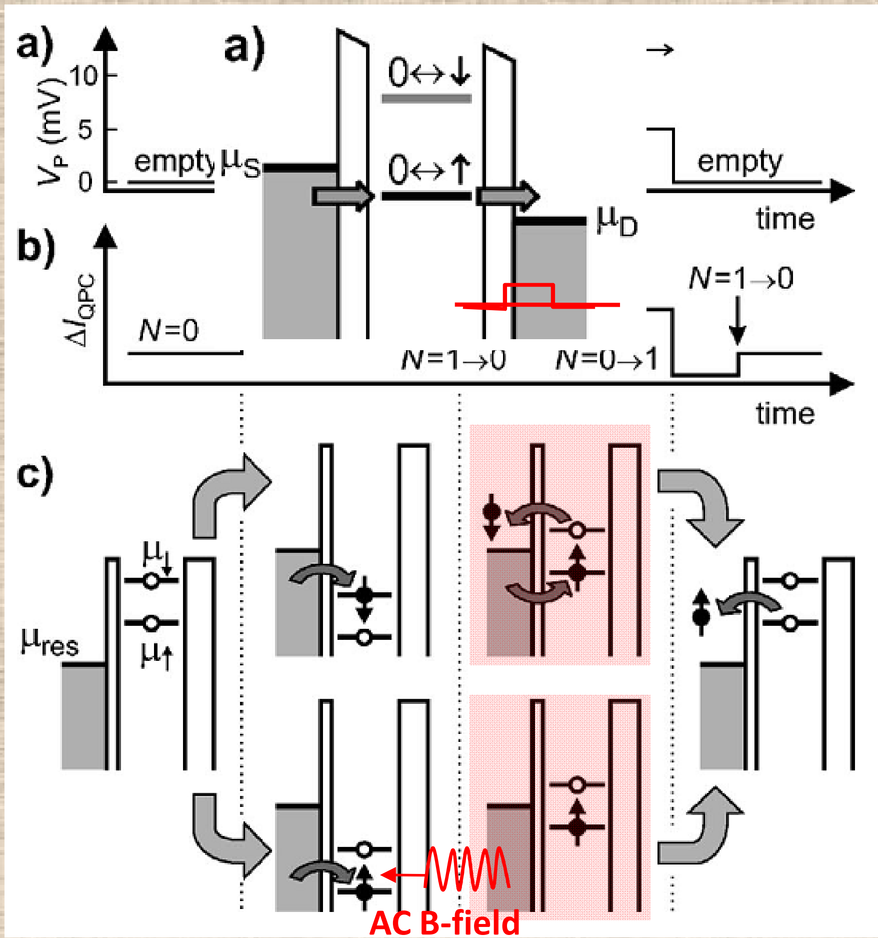
PhysRevLett.115.127001 (2015)

Decoherence time/ gate operation time= 1 µs/ 15 ns

Quantum dot qubits



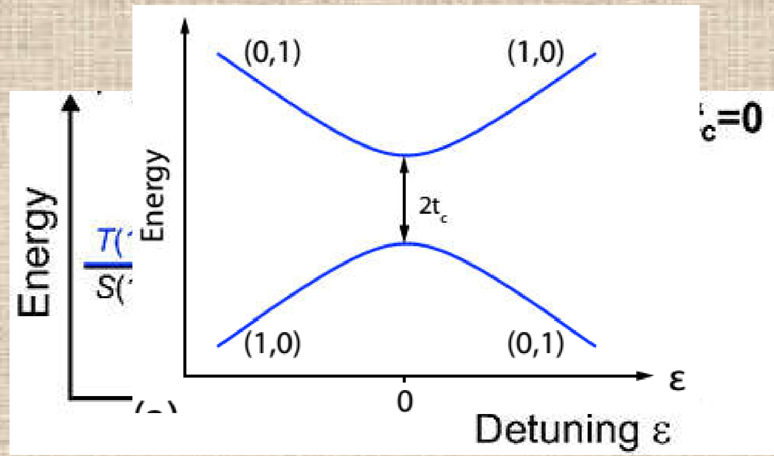
Single quantum dots

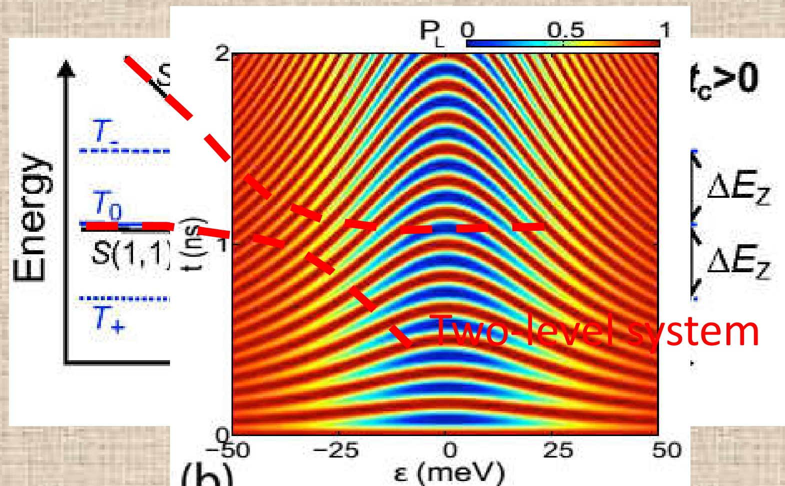


Drawback: large static B-field is required

RevModPhys.79.1217

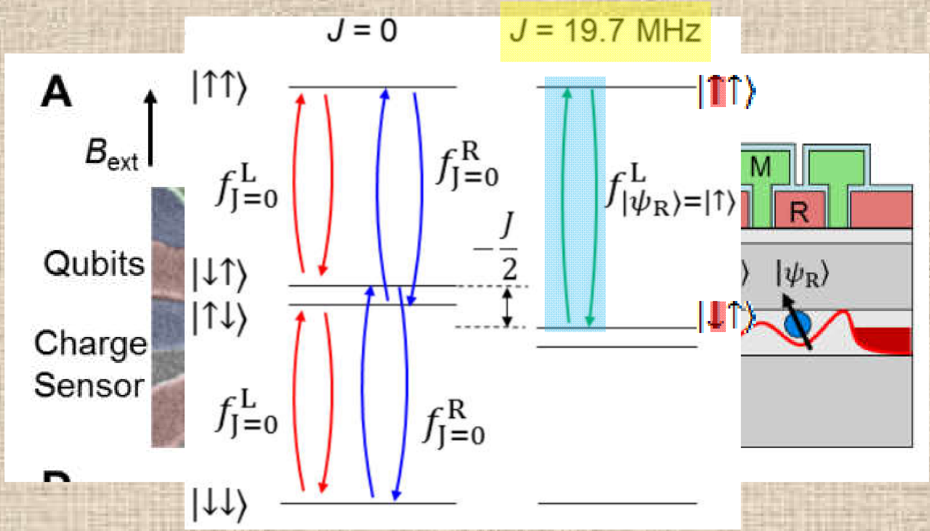
Double quantum dots

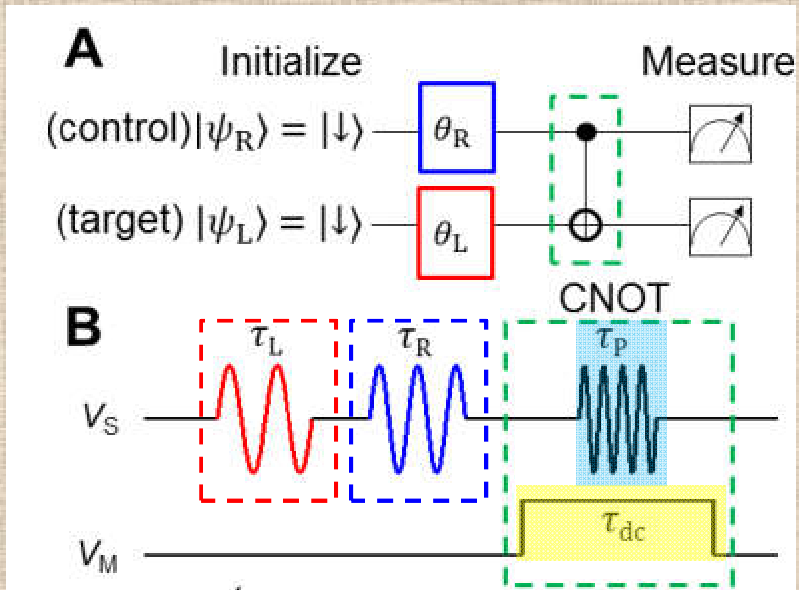




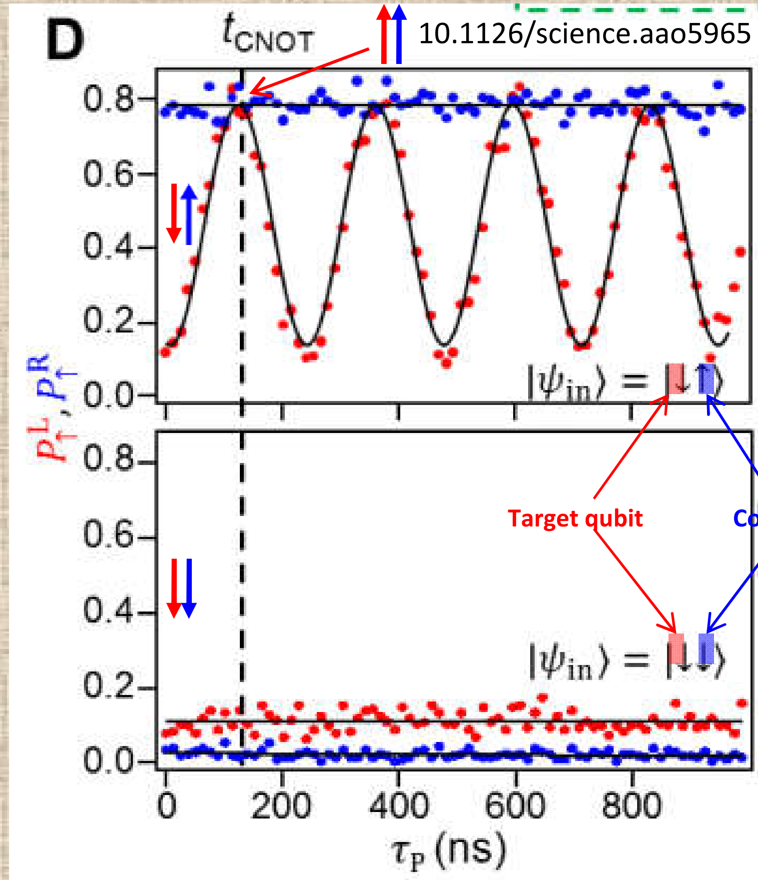
Rabi oscillation

2 qubits CNOT gate in Si QDs





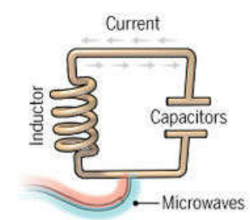
$$H(t) = J(t)\left(\boldsymbol{S}_L \cdot \boldsymbol{S}_R - \frac{1}{4}\right) + \boldsymbol{S}_L \cdot \boldsymbol{B}_L + \boldsymbol{S}_R \cdot \boldsymbol{B}_R$$



A	B	A'	B'
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Progress and prospect

Superconducting loops



A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into super-position states.

Longevity (seconds)	**0.00005**
Logic success rate	**99.4%**
Number entangled	**9** ← IBM 20 (2017)

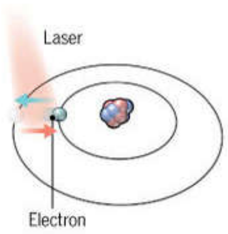
Company support
Google, IBM, Quantum Circuits

Pros
Fast working. Build on existing semiconductor industry.

Cons
Collapse easily and must be kept cold.

Gate operation time ~ ns

Trapped ions



Electrically charged atoms, or ions, have quantum energies that depend on the location of electrons. Tuned lasers cool and trap the ions, and put them in super-position states.

Longevity (seconds)	**>1000**
Logic success rate	**99.9%**
Number entangled	**14**

Company support
ionQ

Pros
Very stable. Highest achieved gate fidelities.

Cons
Slow operation. Many lasers are needed.

Gate operation time ~ 100 ms

Silicon quantum dots

These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.

Longevity (seconds)	**0.03**
Logic success rate	**~99%**
Number entangled	**2**

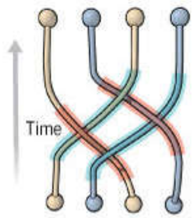
Company support
Intel

Pros
Stable. Build on existing semiconductor industry.

Cons
Only a few entangled. Must be kept cold.

Gate operation time ~ 100 ns

Topological qubits



Quasiparticles can be seen in the behavior of electrons channeled through semi-conductor structures. Their braided paths can encode quantum information.

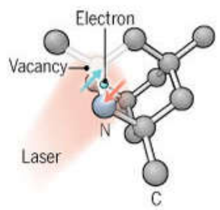
Longevity (seconds)	**N/A** ← Expect to be very long!
Logic success rate	**N/A**
Number entangled	**N/A**

Company support
Microsoft, Bell Labs

Pros
Greatly reduce errors.

Cons
Existence not yet confirmed.

Diamond vacancies



A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.

Longevity (seconds)	**10**
Logic success rate	**99.2%**
Number entangled	**6**

Company support
Quantum Diamond Technologies

Pros
Can operate at room temperature.

Cons
Difficult to entangle.

Note: Longevity is the record coherence time for a single qubit superposition state, logic success rate is the highest reported gate fidelity for logic operations on two qubits, and number entangled is the maximum number of qubits entangled and capable of performing two-qubit operations.

Thank you for your attention!

http://www.sciencemag.org/news/2016/12/scientists-are-close-building-quantum-computer-can-beat-conventional-one