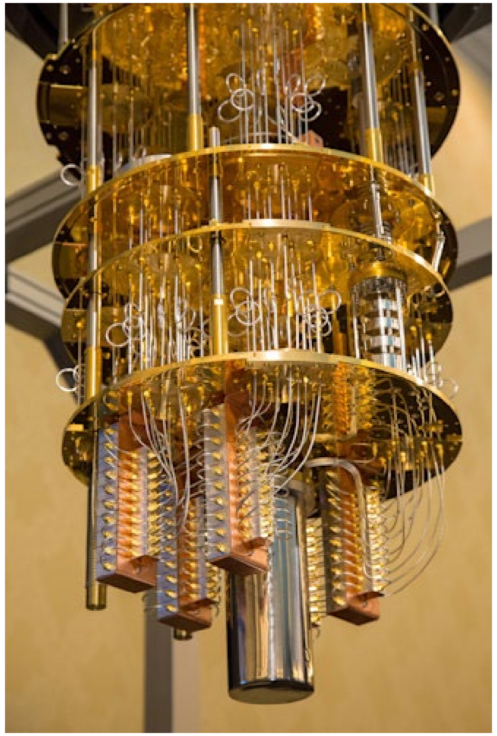
16. Quantum computers and quantum gates



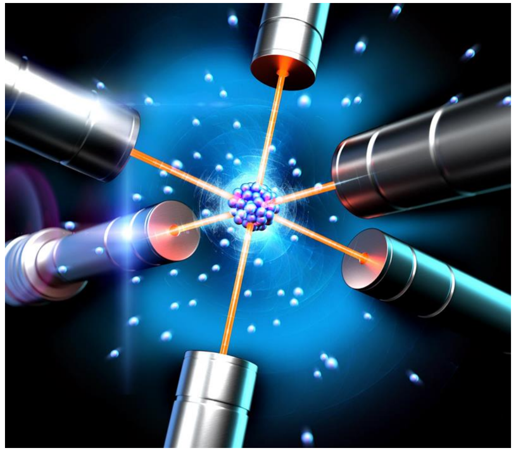
Controllable quantum systems

For many decades physicists examined quantum systems mainly under natural evolution

$$|\psi(t)\rangle = \exp(-\frac{it}{\hbar}H)|\psi(0)\rangle$$

The Hamiltonian is given "naturally" and we just passively observed its effects.

But with improving technology, it has become possible for people to take an active role in engineering quantum systems.

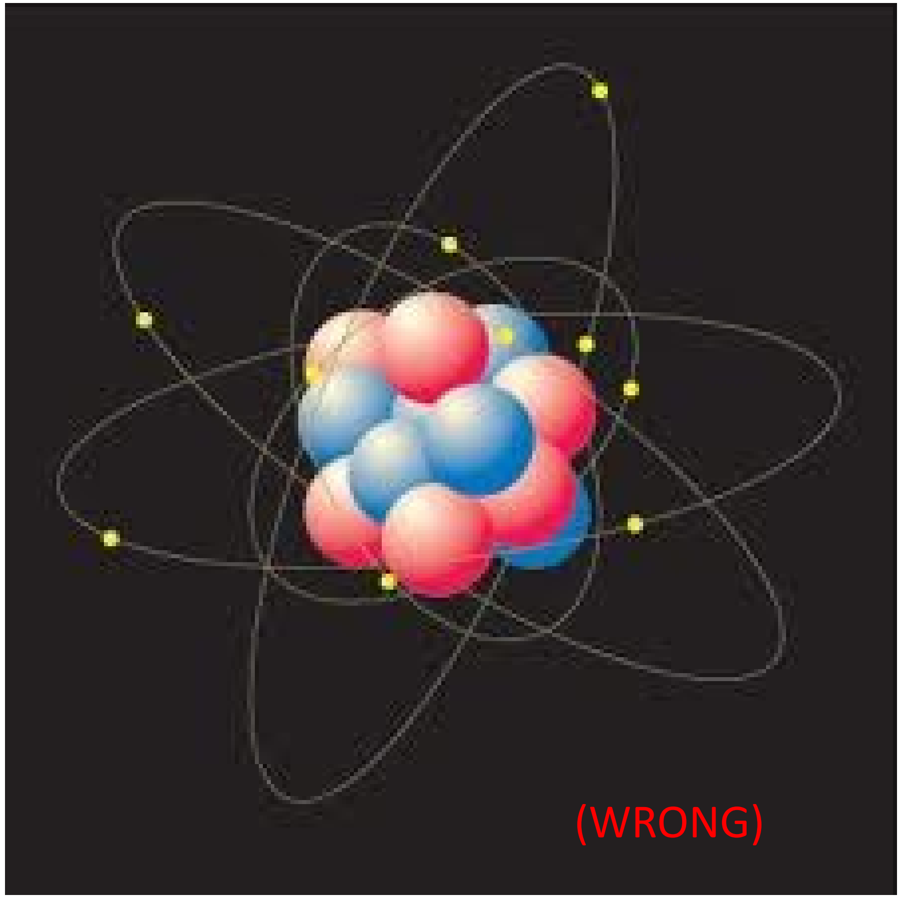


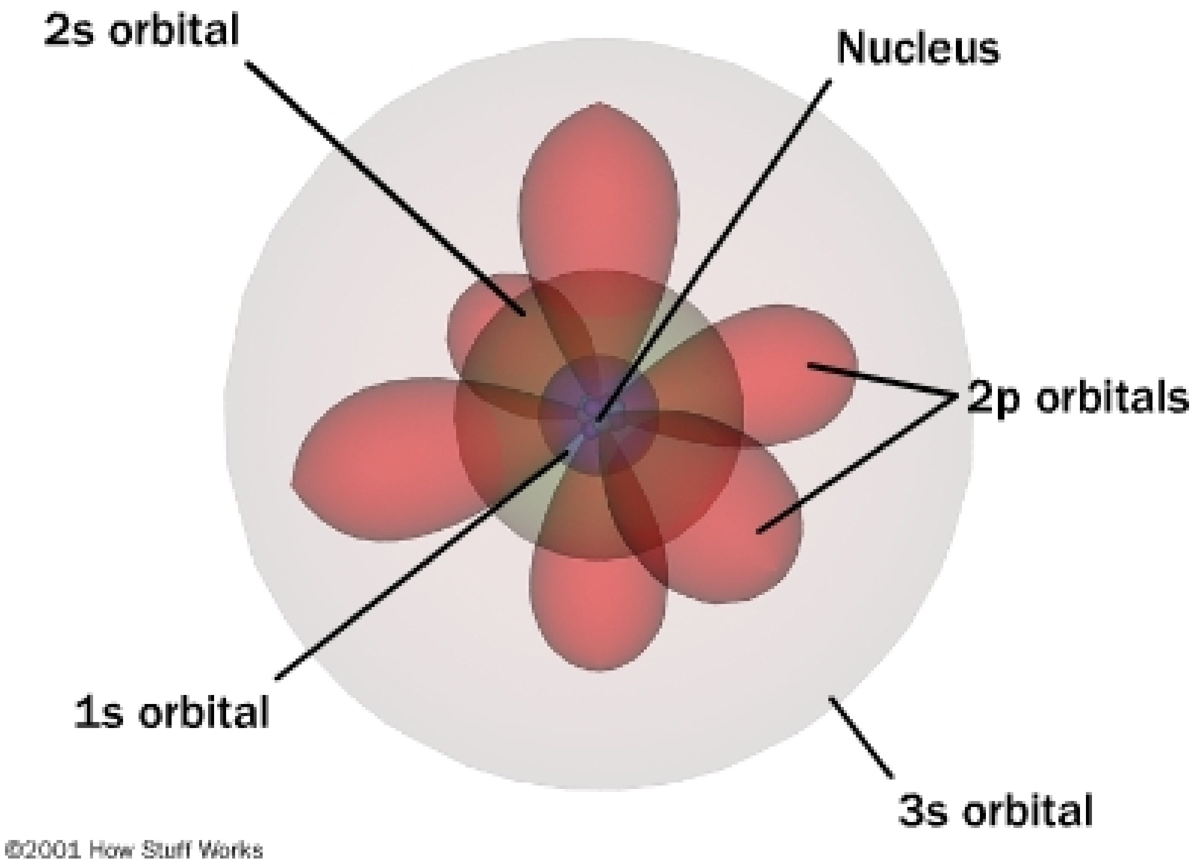
Instead of the Hamiltonian being just given naturally, we can apply man-made Hamiltonians.   Also we can control the time that they are applied for.

This in turn means we can apply unitary operations and make all kinds of quantum states.

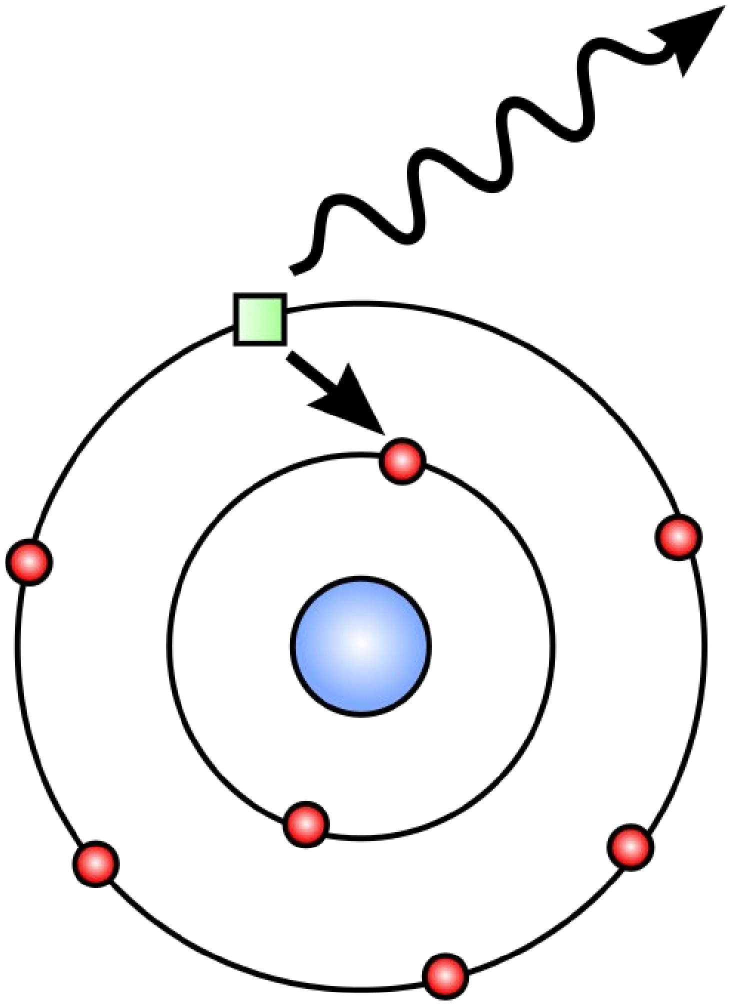
Things described by quantum mechanics

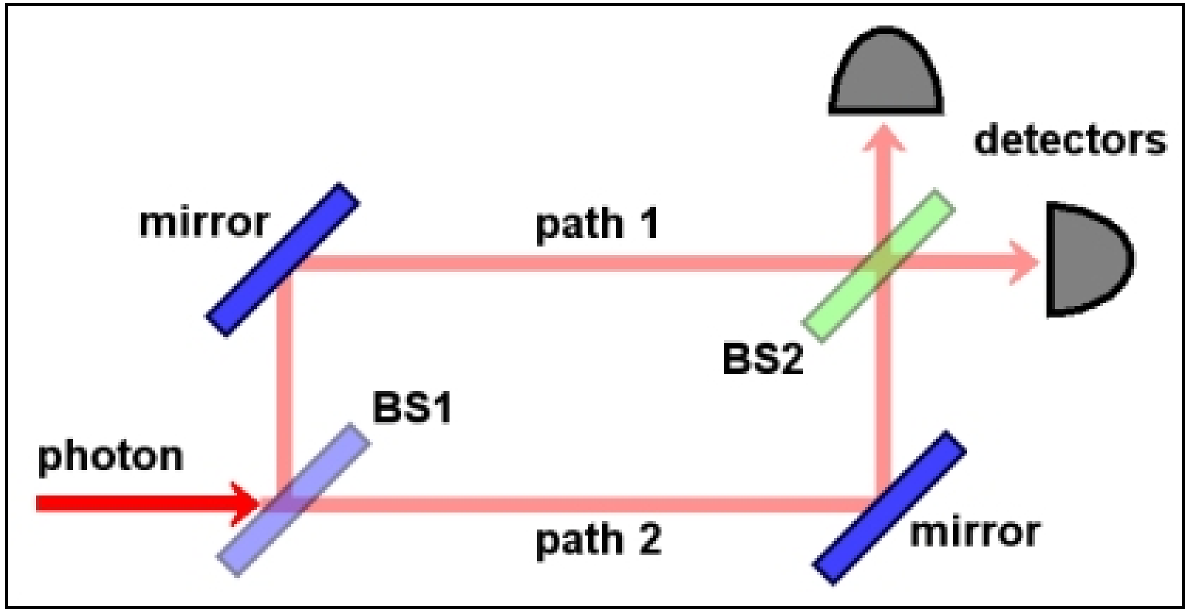
Internal electronic structure of atoms



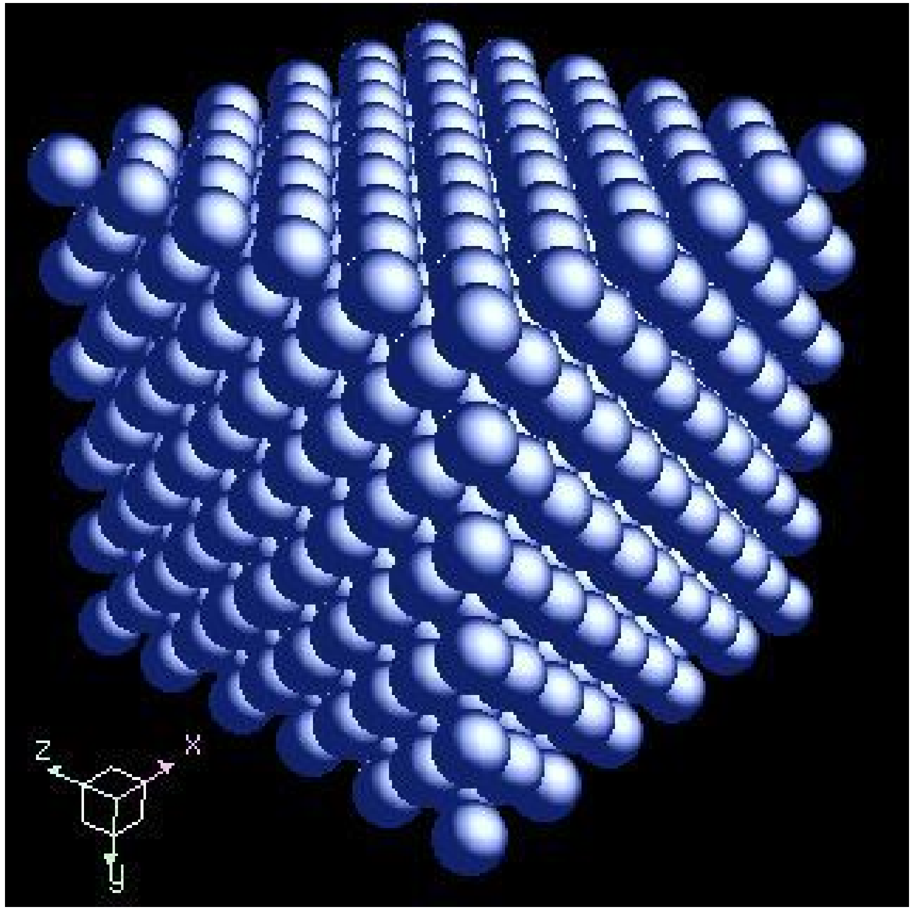


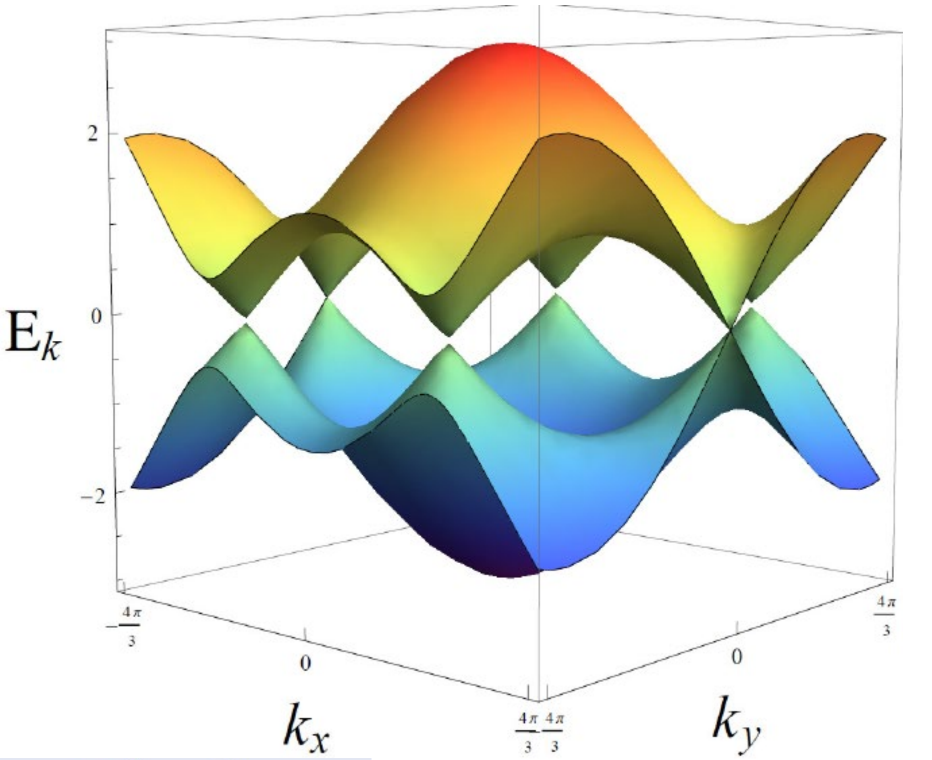
Photons (light)

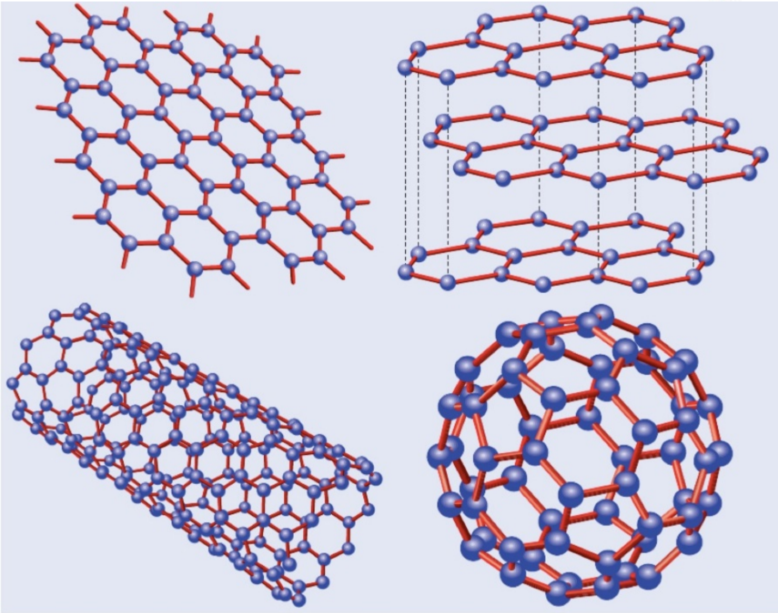




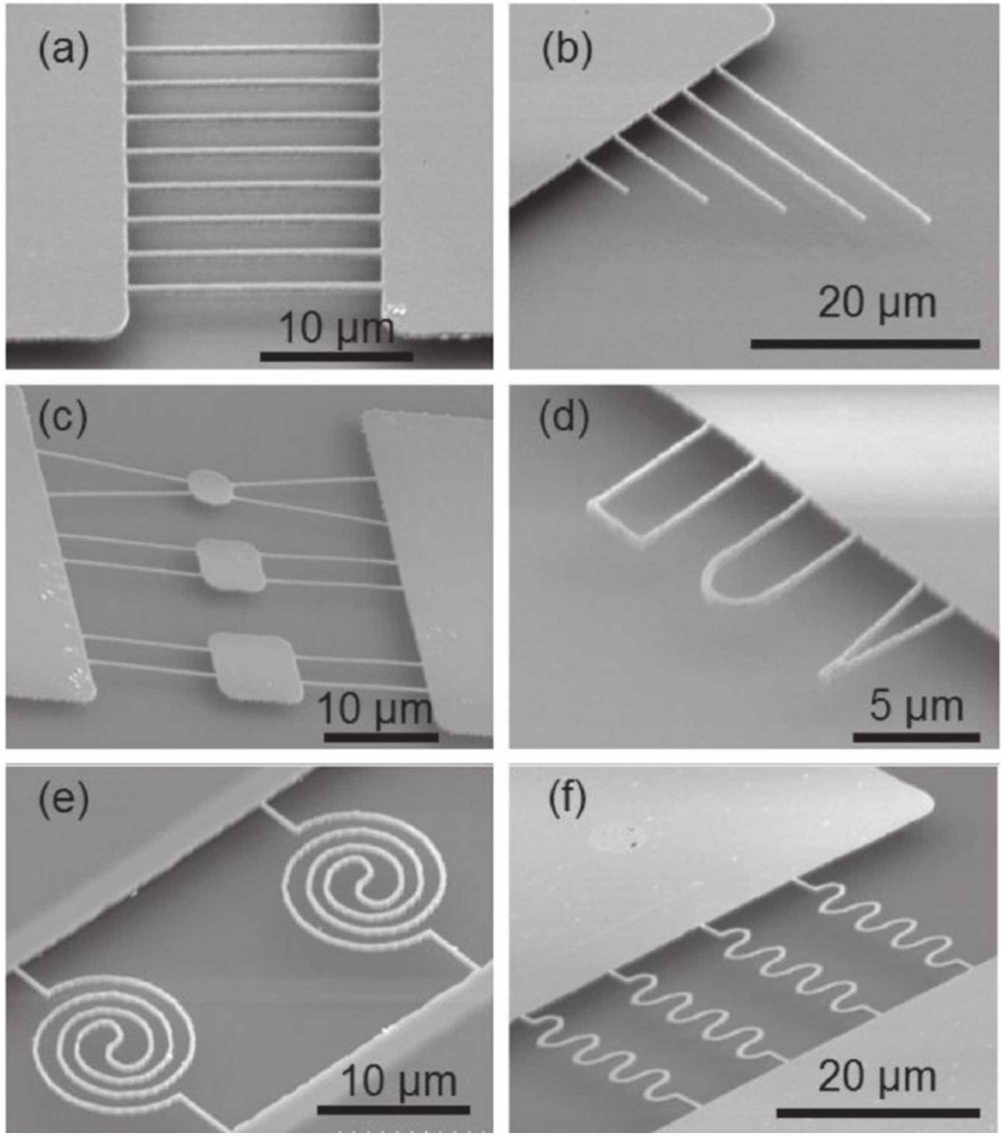
Solid materials



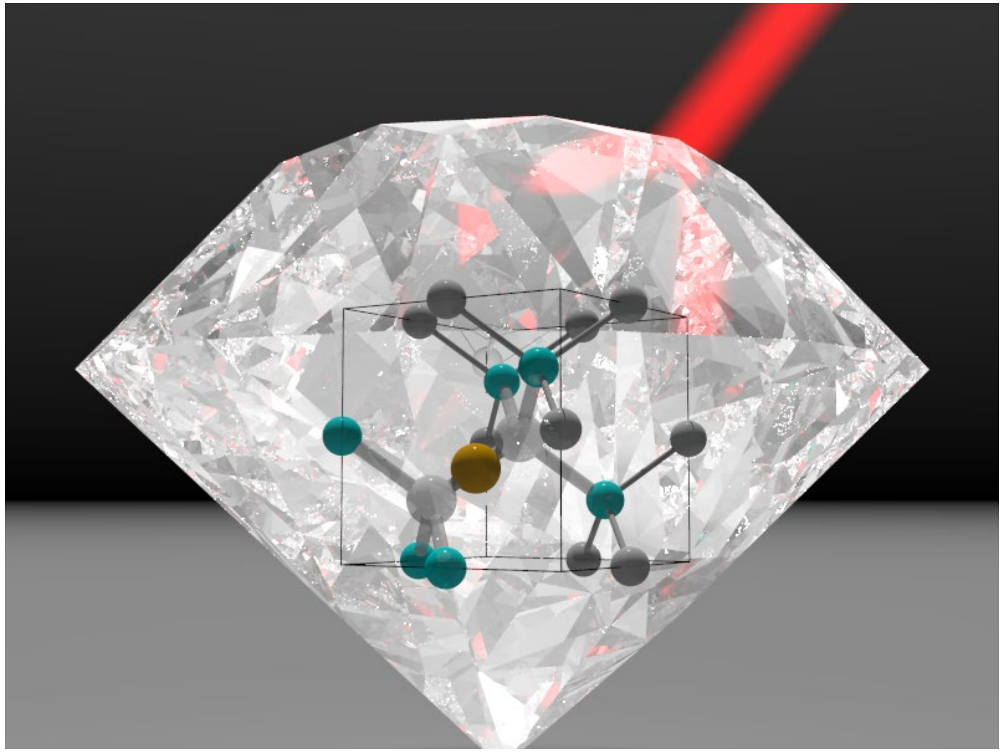


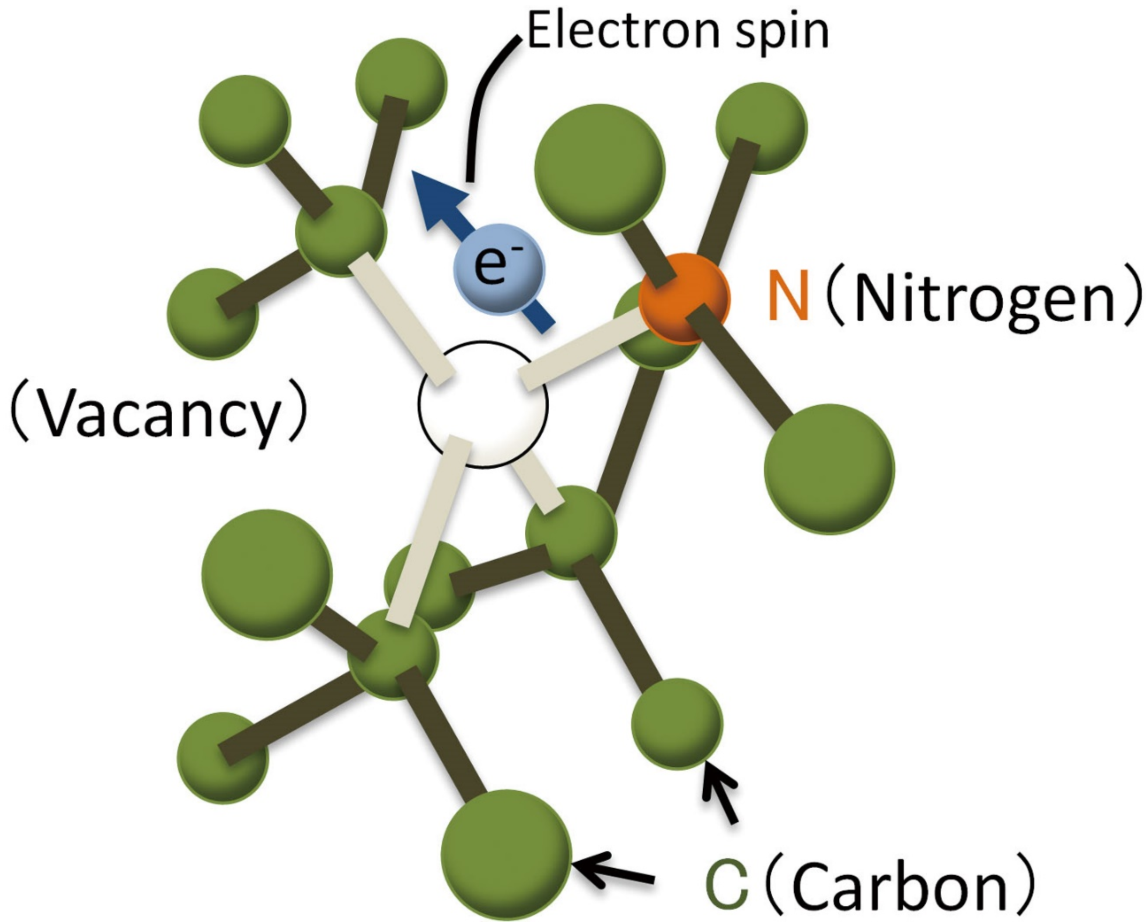


Micro and nanomechanical resonators

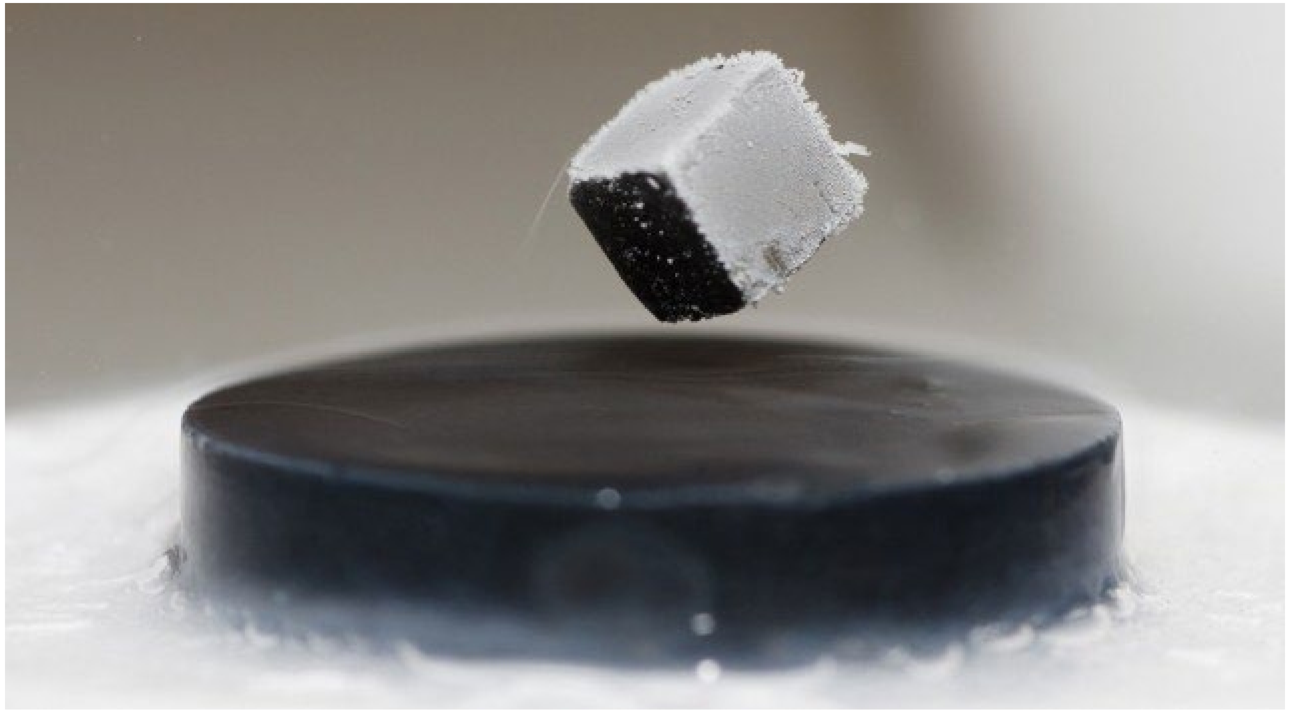


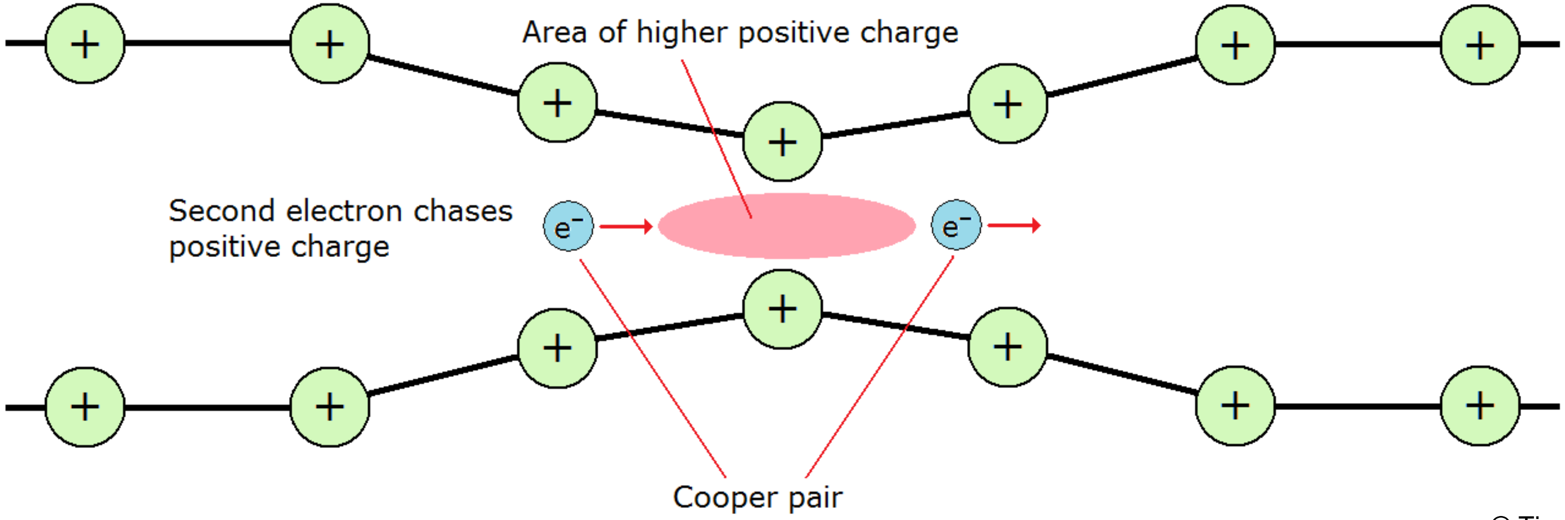
N-V centers



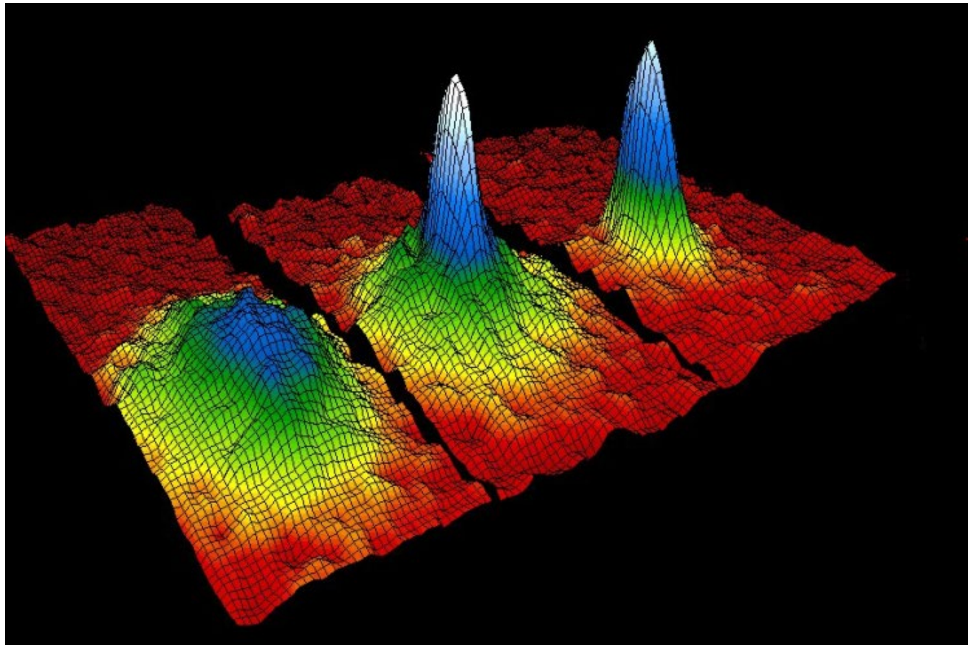


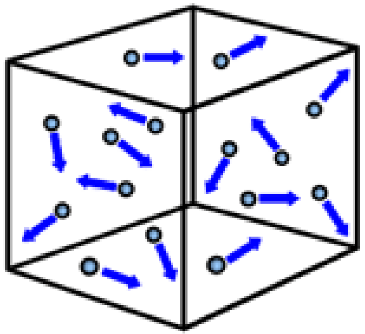
Superconductors

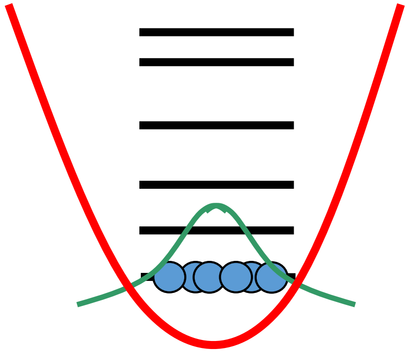


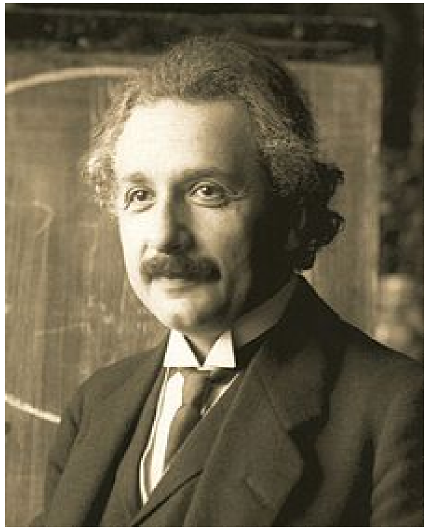


Bose-Einstein condensates





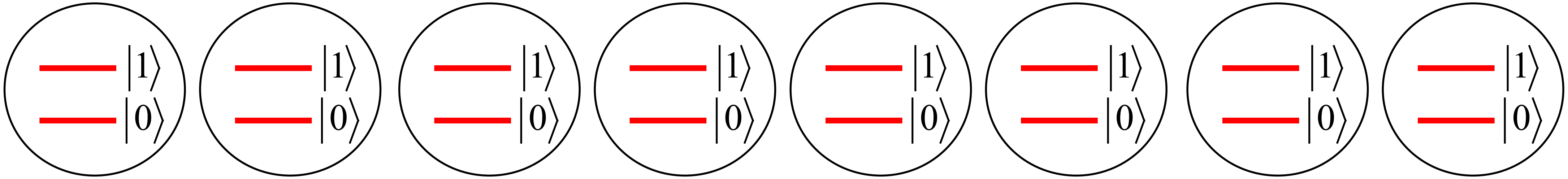




Quantum computers

Quantum computers are the ultimate in controllable quantum systems.

This consists of a large number of qubits, creating a single controllable quantum system.



For a quantum computer with N qubits, there are a total of 2^N different states:

e.g. for $N = 8$

Decimal representation

$$|0\rangle = |00000000\rangle \qquad |252\rangle = |11111100\rangle$$
$$|1\rangle = |00000001\rangle \qquad |253\rangle = |11111101\rangle$$
$$|2\rangle = |00000010\rangle \quad \cdots\cdots \quad |254\rangle = |11111110\rangle$$
$$|3\rangle = |00000011\rangle \qquad |255\rangle = |11111111\rangle$$

Binary representation

A general quantum state can be written as a superposition of these 2^N quantum states.

(Binary representation)

$$|\psi\rangle = \sum_{n_1=0}^{1} \sum_{n_2=0}^{1} \dots \sum_{n_N=0}^{1} a_{n_1 n_2 \dots n_N} |n_1 n_2 \dots n_N\rangle$$

Or we could equally write this as

(Decimal representation)

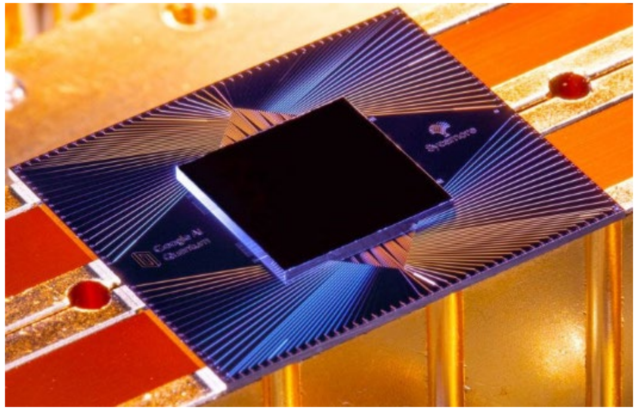
$$|\psi\rangle = \sum_{m=0}^{2^N-1} a_m |m\rangle$$

The quantum computer also allows us to control this state in a completely arbitrary way:

$$|\psi_{\text{output}}\rangle = U_{\text{algorithm}} |\psi_{\text{input}}\rangle$$

Quantum input data

$|\psi_{\text{input}}\rangle$

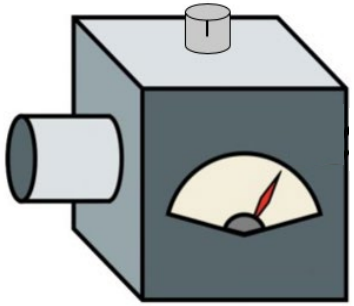


Processing (i.e. quantum algorithm)

Quantum output

$|\psi_{\text{output}}\rangle$

Measurement



Outcomes

$|00000000\rangle$

$|00000001\rangle$

$|00000010\rangle$

$\vdots$

$|11111111\rangle$

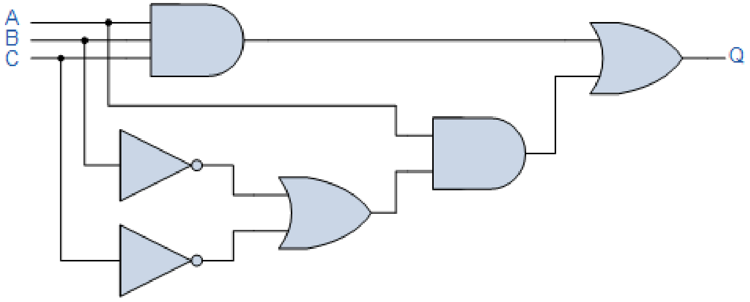
Elementary gates

One of the requirements of a quantum computer is the ability to make an arbitrary unitary evolution

$$|\psi_{\text{output}}\rangle = U_{\text{algorithm}}|\psi_{\text{input}}\rangle \qquad \begin{pmatrix} \\ \psi_{\text{output}} \\ \\ \end{pmatrix} = \underbrace{\begin{pmatrix} & & \\ & U_{\text{algorithm}} & \\ & & \end{pmatrix}}_{2^N} \begin{pmatrix} \\ \psi_{\text{input}} \\ \\ \end{pmatrix} \quad 2^N$$

How can we make such a huge, general, complicated matrix?

➡ The same way as we deal with constructing a general classical algorithm

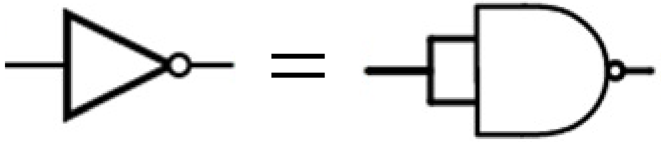


We construct a complicated algorithm out of simpler elementary gates, like AND, OR, NOT, NAND, NOR, XOR, etc.

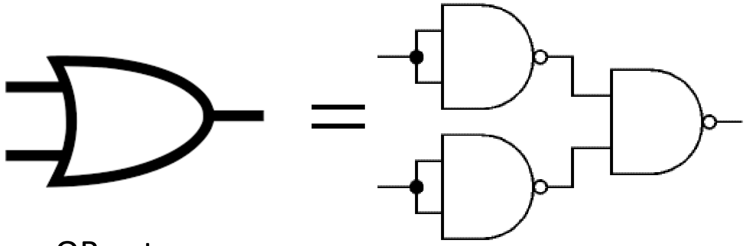
Universality

What is the simplest set of gates such that you can build up an arbitrary algorithm?

In classical logic, a famous result is that you can make any other gate in terms of a NAND gate



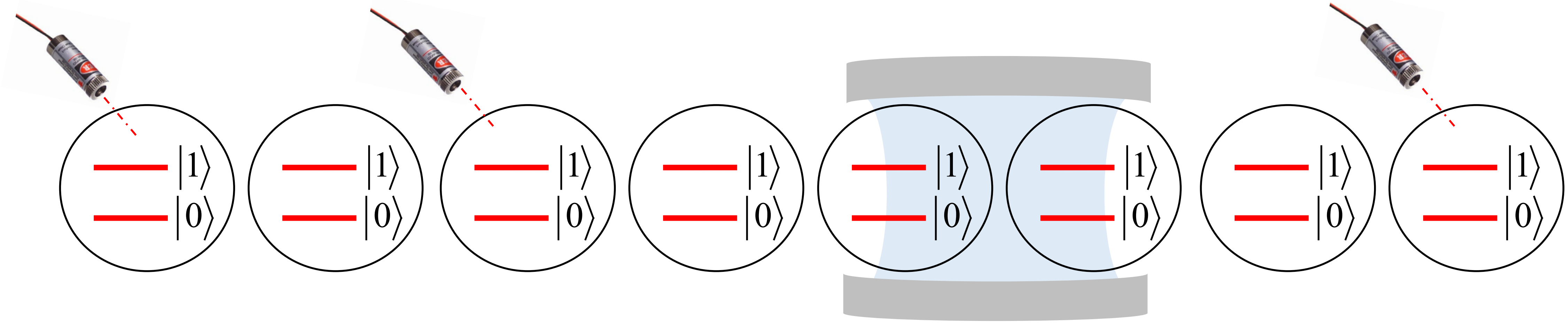
NOT gate



OR gate

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

In quantum computing, the analogous result is that an arbitrary unitary $U_{\text{algorithm}}$ can be made from just one and two qubit gates.



Classical one bit gates

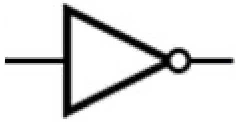
Before introducing one qubit gates, let's list all the possible classical one bit gates.

There are two commonly used one bit gates

1) Wire (do nothing)

input	output
0	0
1	1

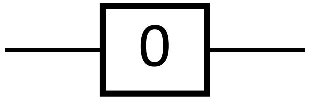
1) NOT



input	output
0	1
1	0

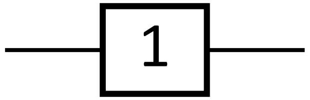
There are also two other possible gates

3) Reset to 0



input	output
0	0
1	0

4) Reset to 1



input	output
0	1
1	1

Quantum versions of one bit gates

Since the quantum gates should be written in terms of a unitary matrix, we can write the analogous gates for the first two as

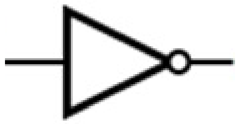
1) Wire (do nothing)

input	output
0	0
1	1

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

$$I|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

2) NOT

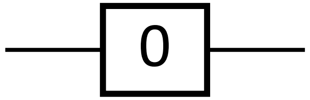


input	output
0	1
1	0

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

$$X|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

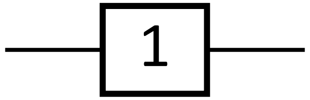
3) Reset to 0



input	output
0	0
1	0

Not reversible

4) Reset to 1



input	output
0	1
1	1

Not reversible

Quantum versions of one bit gates

Since the quantum gates should be written in terms of a unitary matrix, we can write the analogous gates for the first two as

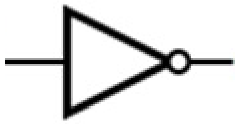
1) Wire (do nothing)

input	output
0	0
1	1

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

$$I|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

2) NOT

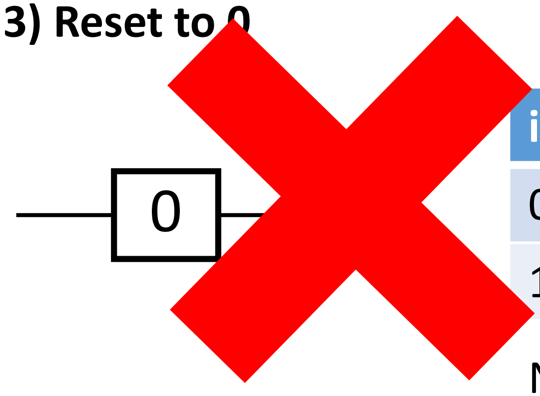


input	output
0	1
1	0

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

$$X|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

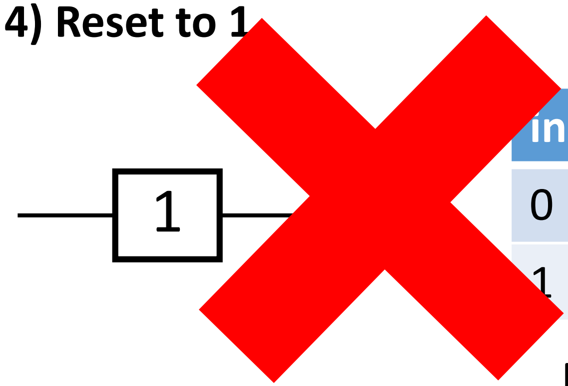
3) Reset to 0



input	output
0	0
1	0

Not reversible

4) Reset to 1



input	output
0	1
1	1

Not reversible

These are not reversible, so a unitary gate cannot be made for these.

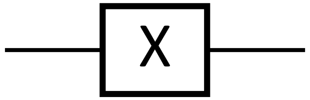
One qubit gates

The quantum notation for the 2 possible one qubit gates are

1) Wire (do nothing)

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

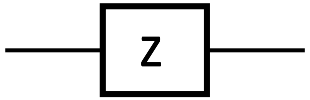
2) NOT



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

In addition to the classical counterparts, there are more gates that can be done with quantum mechanics.

3) Phase flip gate



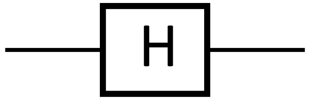
$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$Z|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$$

This is reversible because

$$Z^+ Z = Z^2 = I$$

4) Hadamard gate



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

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$H^+ H = H^2 = I$$

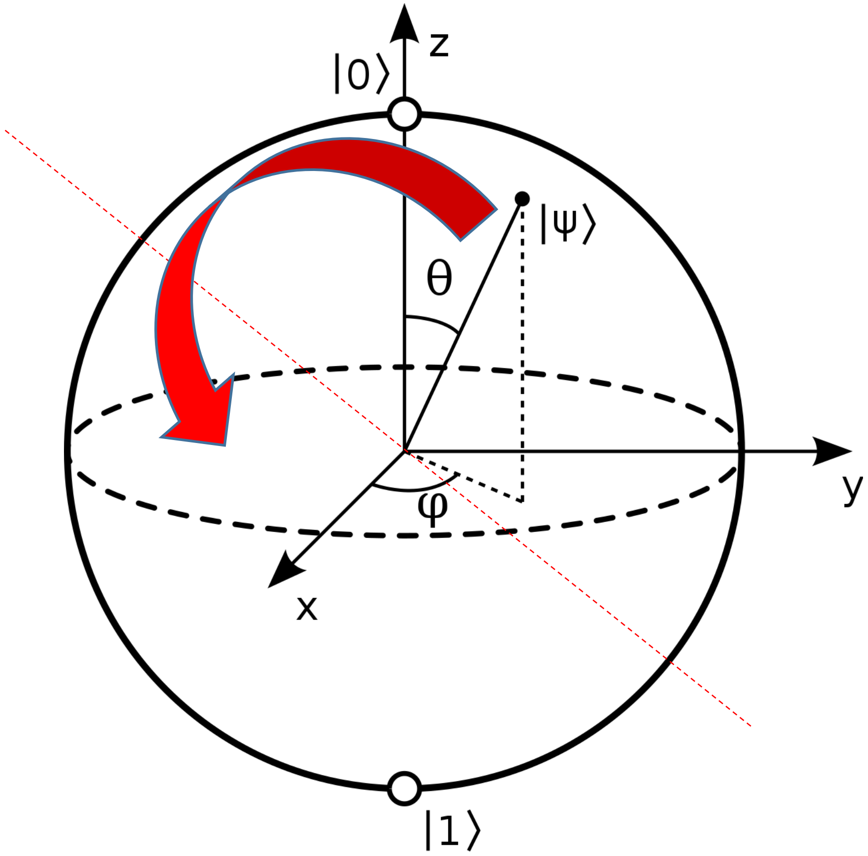
Most general one qubit gate

Actually there are many more gates that are possible, but the previous page are the main ones that are often used in quantum algorithms.

The most general one qubit gate is

$$e^{i\frac{\phi}{2}\left(n_x X + n_y Y + n_z Z\right)} \qquad n_x^2 + n_y^2 + n_z^2 = 1$$

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

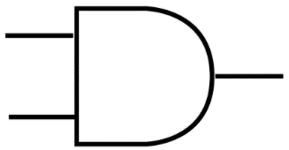


This can be visualized by a rotation by an angle ϕ around the axis specified by $\left(n_x, n_y, n_z\right)$

Classical two bit gates

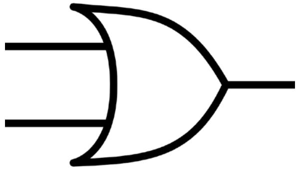
What about two qubit gates? Again let's start with the classical case for two bit gates.   There are many possible types but here are the common ones

1) AND



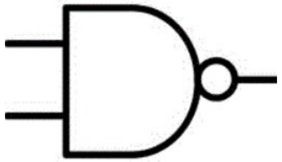
input 1	input 2	output
0	0	0
0	1	0
1	0	0
1	1	1

2) OR



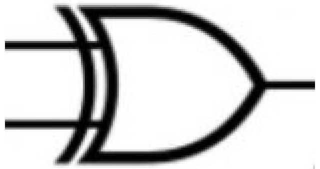
input 1	input 2	output
0	0	0
0	1	1
1	0	1
1	1	1

3) NAND



input 1	input 2	output
0	0	1
0	1	1
1	0	1
1	1	0

4) XOR



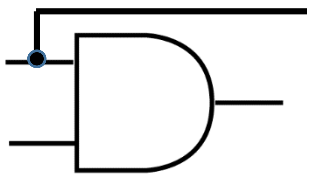
input 1	input 2	output
0	0	0
0	1	1
1	0	1
1	1	0

None are reversible because there is only one output and two inputs!

Tweaked classical two bit gates

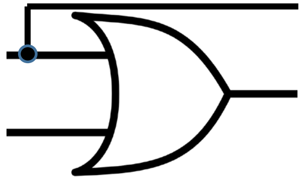
Ok we can get around the one output problem by just copying one of the inputs to the outputs. Let's use input 1:

1) AND



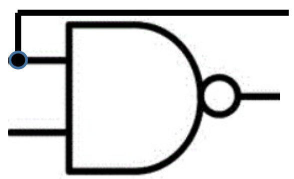
in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	1

2) OR



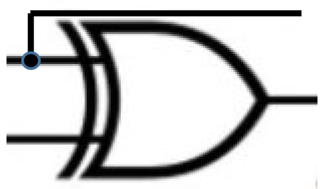
in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	1

3) NAND



in 1	in 2	in 1 copy	out
0	0	0	1
0	1	0	1
1	0	1	1
1	1	1	0

4) XOR



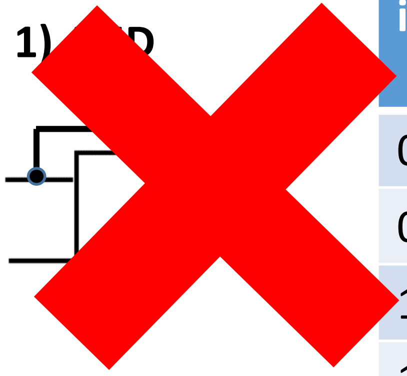
in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

If any of the output combinations are repeated, it is not reversible.

Tweaked classical two bit gates

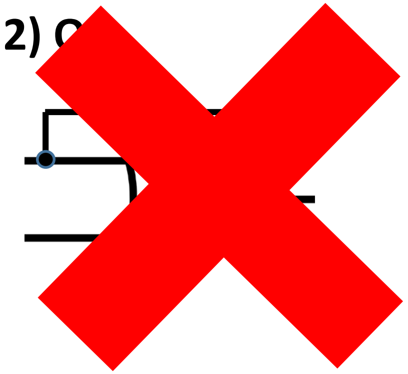
Ok we can get around the one output problem by just copying one of the inputs to the outputs. Let's use input 1:

1) AND



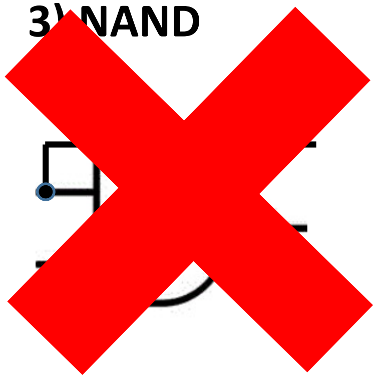
in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	1

2) OR



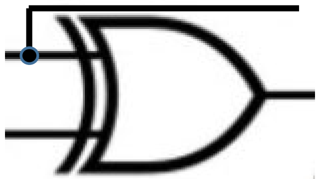
in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	1

3) NAND



in 1	in 2	in 1 copy	out
0	0	0	1
0	1	0	1
1	0	1	1
1	1	1	0

4) XOR



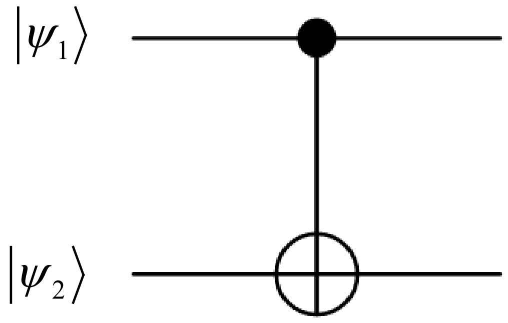
in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Only the XOR is compatible with a unitary evolution

The CNOT gate

The XOR is a gate we can make a quantum version out of, since it is reversible.

In the quantum context it is called the CNOT gate, which stands for "controlled-NOT"



$$U_{CNOT} = \begin{array}{cc} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \end{array}$$

in 1	in 2	in 1 copy	out
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

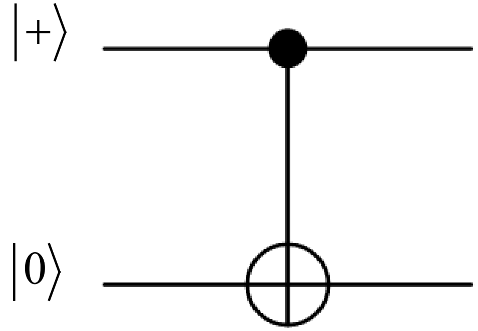
If in 1 =0, then do nothing to in 2

If in 1 = 1, then apply NOT to in 2

$$U_{CNOT} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \\ b \\ d \\ c \end{pmatrix}$$

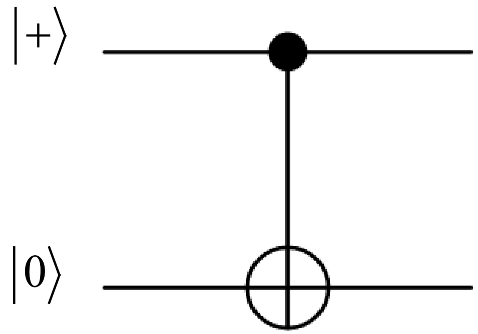
Question: CNOT gate

Apply the CNOT gate to the state $|+\rangle|0\rangle$. What is the final state?



Question: CNOT gate

Apply the CNOT gate to the state $|+\rangle|0\rangle$. What is the final state?



Matrix method

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

$$\frac{U_{CNOT}}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\0\\1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

Bra-ket method

$$U_{CNOT}|+\rangle|0\rangle = U_{CNOT}\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|0\rangle\right)$$

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