

# Introduction to Quantum Computing



Kitty Yeung, Ph.D. in Applied Physics

Creative Technologist + Sr. PM  
Microsoft

[www.artbyphysicistkittyyeung.com](http://www.artbyphysicistkittyyeung.com)



@KittyArtPhysics



@artbyphysicistkittyyeung

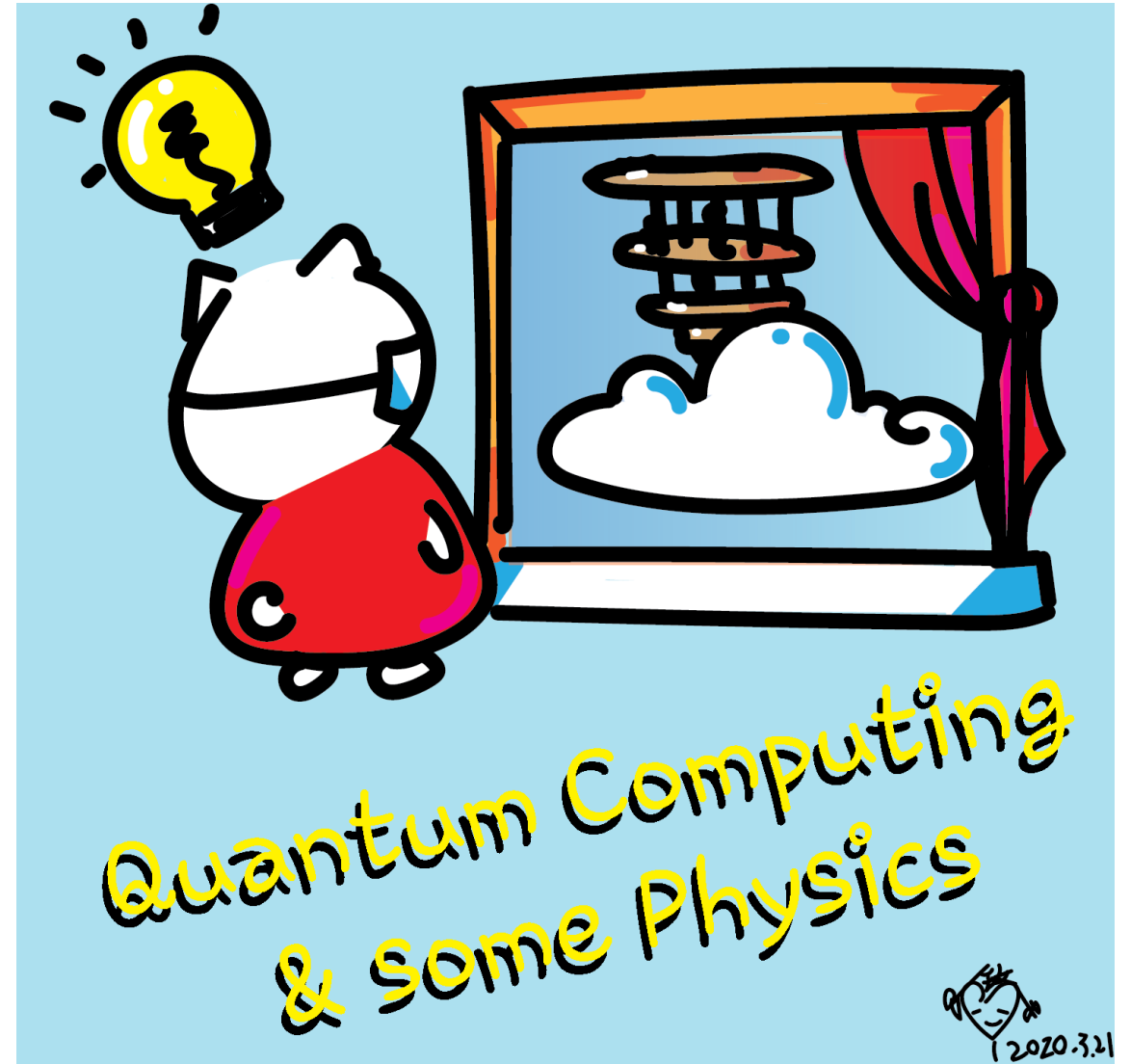
September 6, 2020

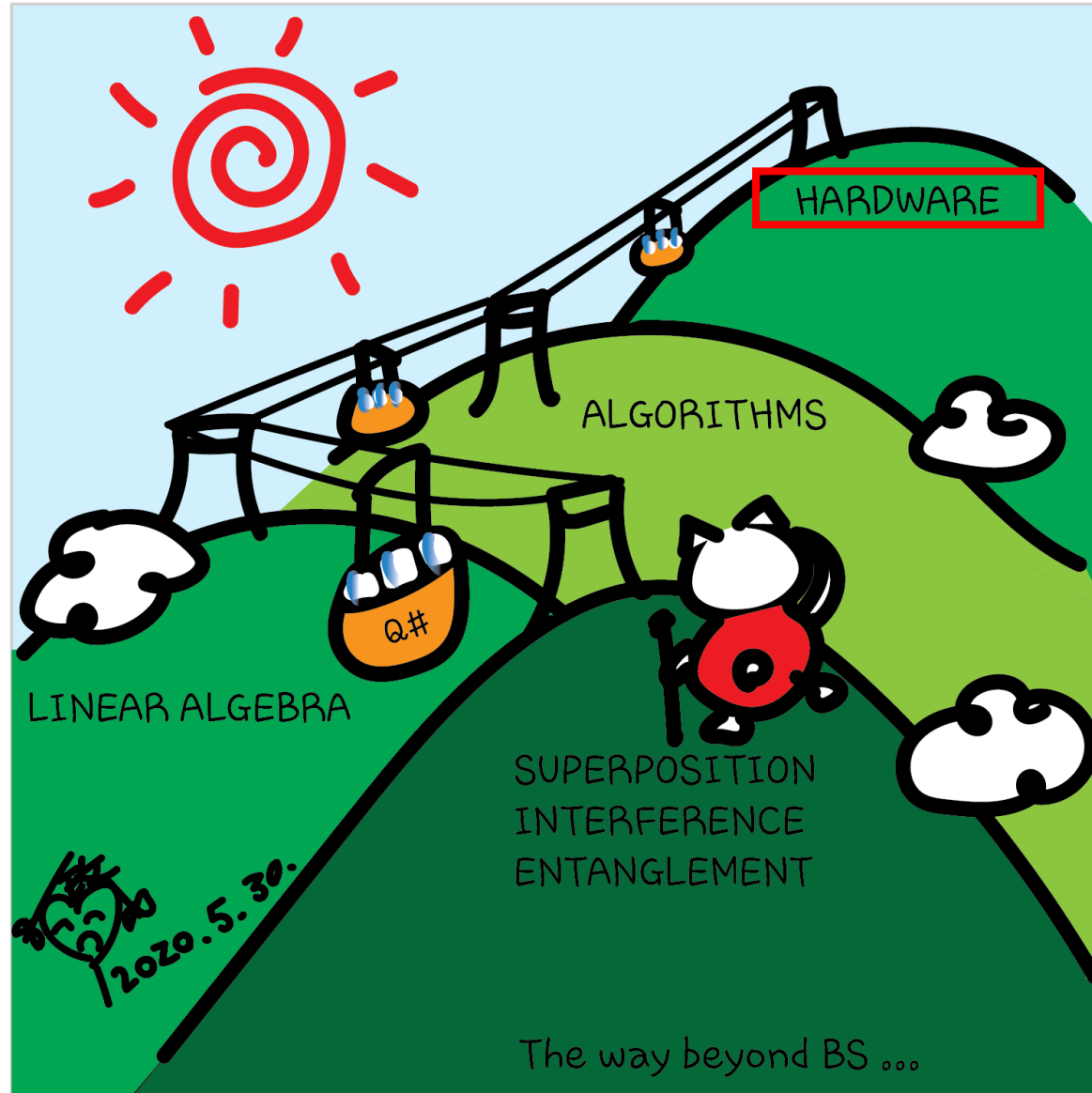
Hackaday, session 20

Other communities, session 12

# Class structure

- [Comics on Hackaday – Quantum Computing through Comics](#) every Sun
- 30 mins – 1 hour every Sun, one concept (theory, hardware, programming), Q&A
- Contribute to Q# documentation  
<http://docs.microsoft.com/quantum>
- Coding through Quantum Katas  
<https://github.com/Microsoft/QuantumKatas/>
- Discuss in Hackaday project comments throughout the week
- Take notes

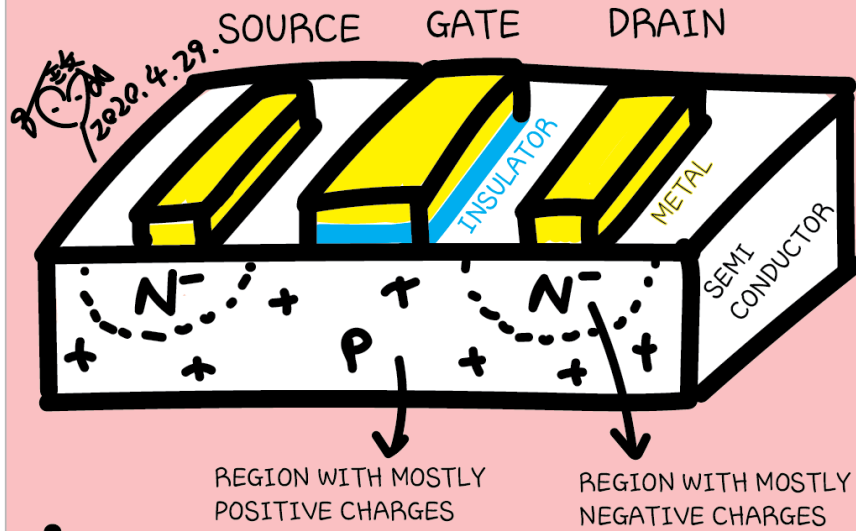




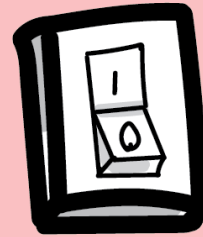
Session 6  
May 3

We leverage various properties of materials to make computing hardware.

25



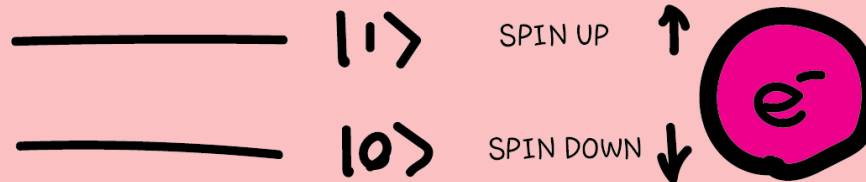
The gate applies voltage to control the electron flow from source to drain of a transistor. At a certain gate voltage level, electrons flow. This is the "on" state which we call "1". When there's no electron flow, we say the transistor is "off", or "0".



Transistors are nano- to micrometer switches.

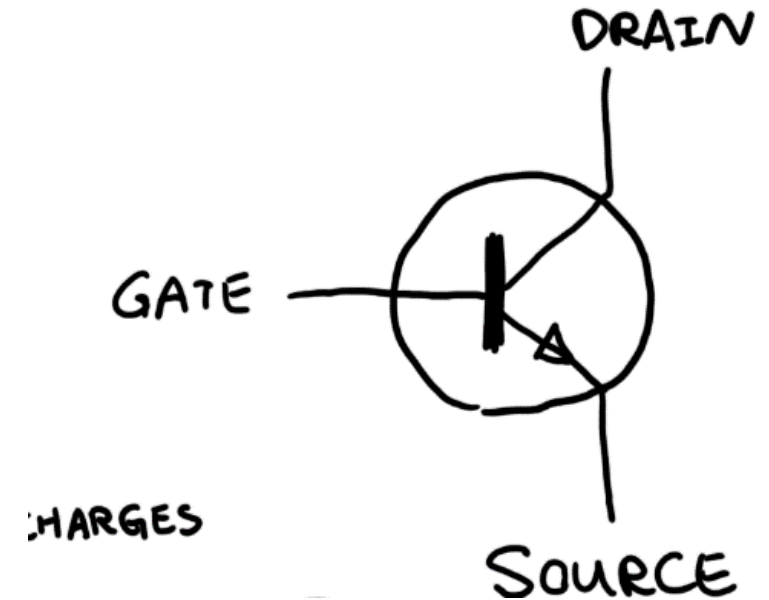
To make a qubit, we need a system with two states that can be in superposition.

The electron spin seems to be a natural candidate.

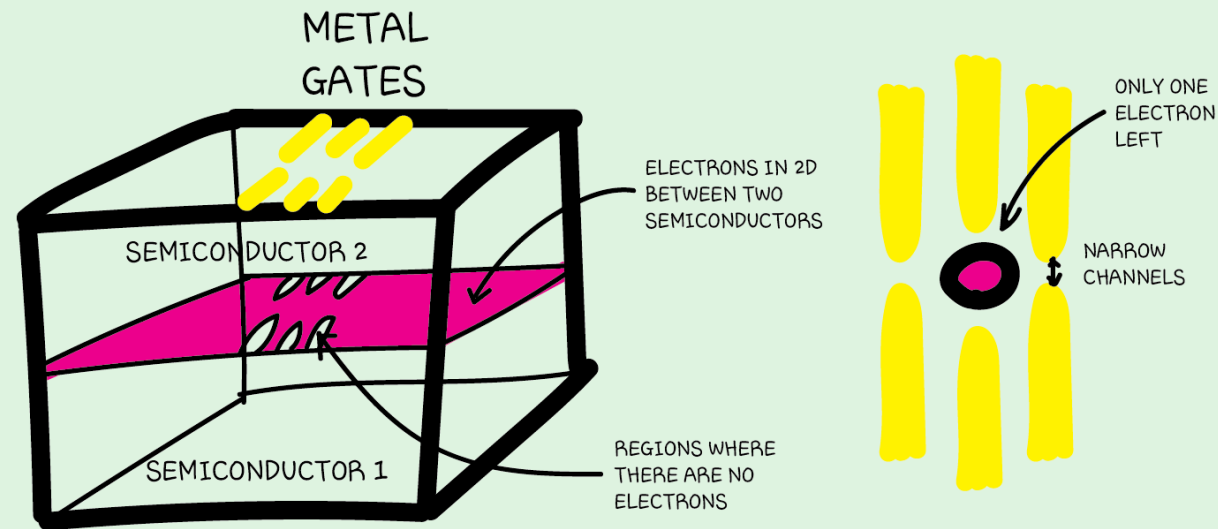


But how do we isolate and control electrons?

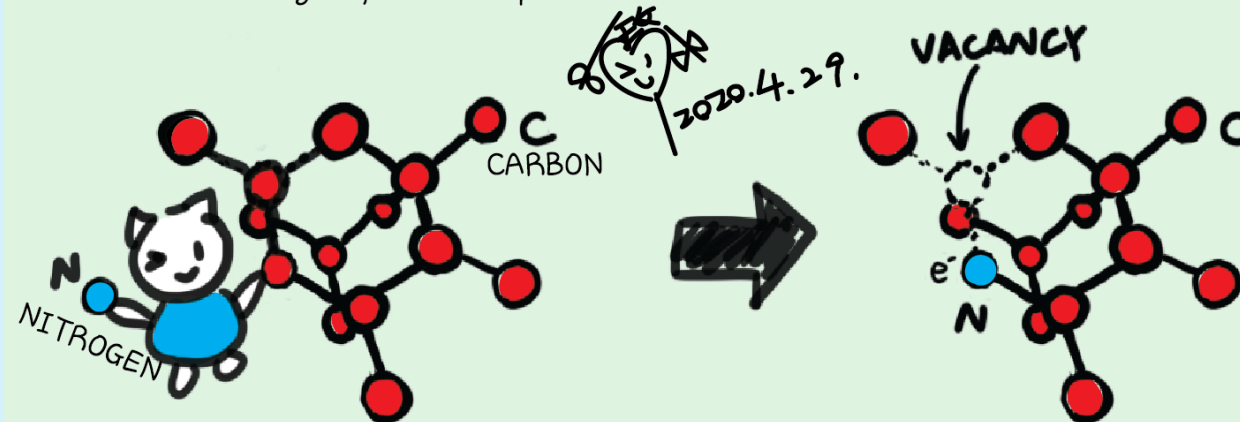
## CIRCUIT SYMBOL





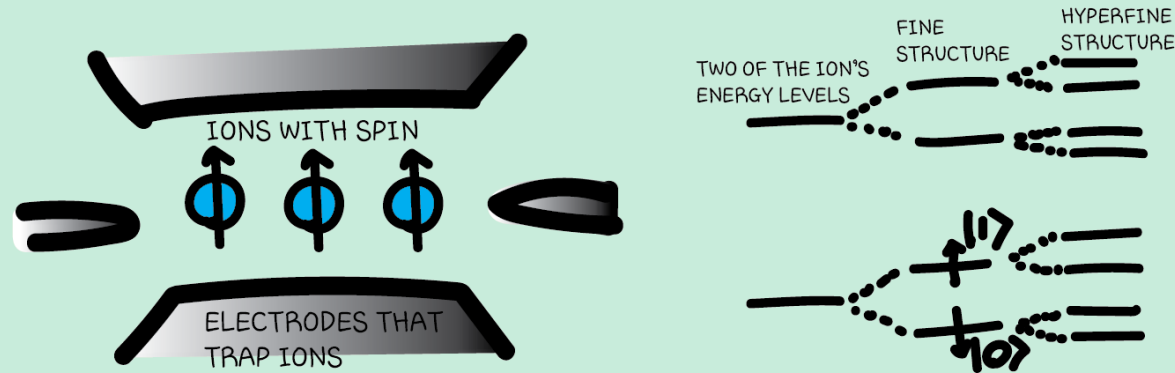


We can create a semiconductor stack. At the interface between the two semiconductors, electrons can be confined in 2D. By applying a gate voltage, the electrons underneath are removed, until there is only one electron left in a small region, called a "quantum dot".

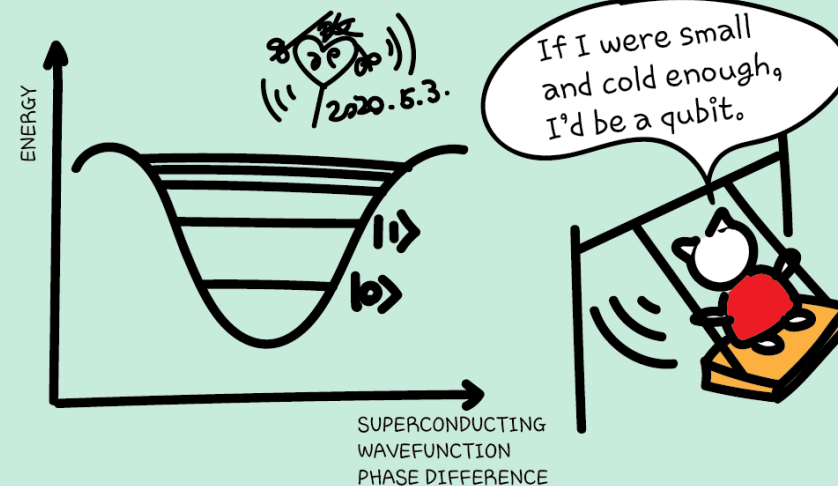
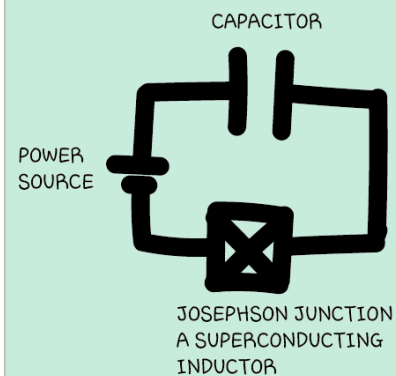


We can also use a crystal lattice, e.g. diamond. We can remove two carbon atoms (each has 6 electrons), replacing them with a single nitrogen atom (which has 7 electrons). The extra electron is bound to the nitrogen-vacancy region.

Although an electron can be conceptually the easiest qubit,  
it doesn't mean it is straightforward to control many electrons.  
There are other systems to explore.

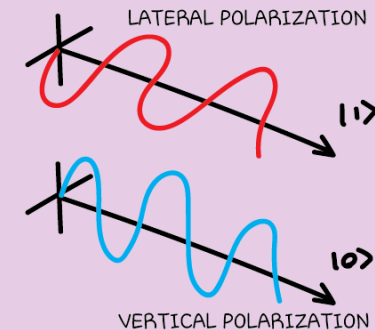
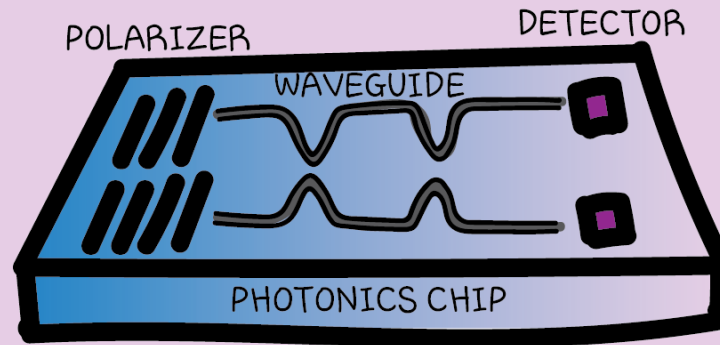


We can use two of an ion's electronic energy levels as the two qubit states. They can be the fine structure due to the ion's electron spins or the hyperfine structure due to electrons' interactions with the ion's nucleus. We can also make an "artificial atom" and use its energy levels as qubit states, e.g. a superconducting circuit. Its oscillation creates a set of discrete energies.

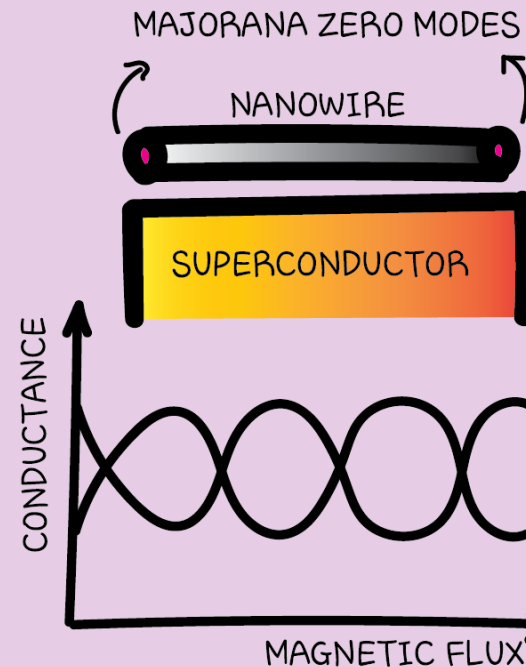


If we can use natural and artificial particles, such as electrons, ions or oscillating circuits, we can also try other types of particles and quasiparticles, and parameters other than energy levels.

28

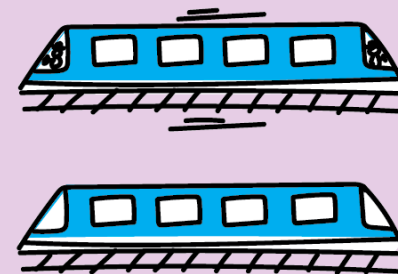
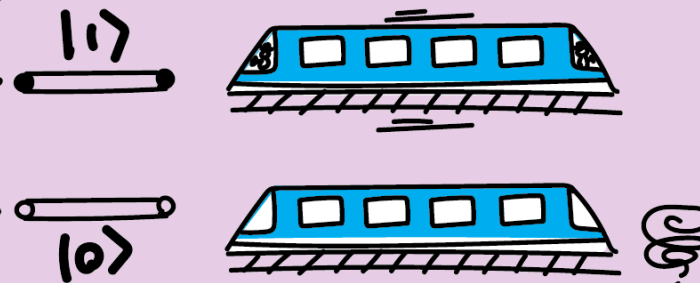


We can use photons' polarizations to encode qubit states.



Or the quasiparticles constructed in a topological material (in this case the ends of a nanowire on a superconductor).

The **zero modes** are locations that can be empty or occupied by an electron, thus giving two qubit states.

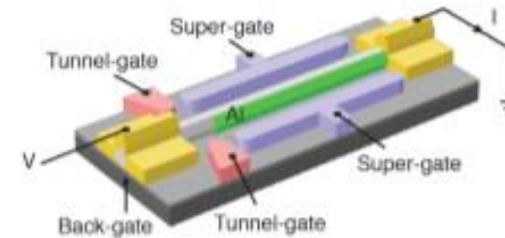
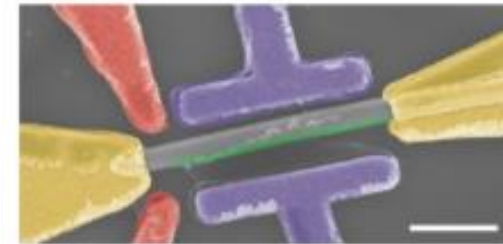


2020.5.3.

# Topological quantum computer

Majorana Fermions – particle equals anti-particle  
Fractional quantum Hall conductance  
Low temperature in magnetic field

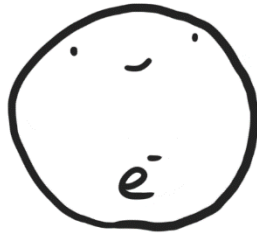
<https://arxiv.org/pdf/cond-mat/0412343.pdf>



Quantized Majorana Conductance

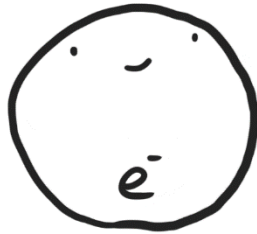
<https://www.nature.com/articles/nature26142>





ELECTRON

annihilation and creation operators  $a_j, a_j^\dagger$



**ELECTRON**



**MAJORANA PAIR**

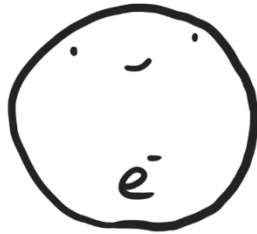
annihilation and creation operators  $a_j, a_j^\dagger$

*Majorana operators*

$$c_{2j-1} = a_j + a_j^\dagger, \quad c_{2j} = \frac{a_j - a_j^\dagger}{i} \quad (j = 1, \dots, N)$$

which satisfy the relations

$$c_m^\dagger = c_m, \quad c_l c_m + c_m c_l = 2\delta_{lm} \quad (l, m = 1, \dots, 2N).$$



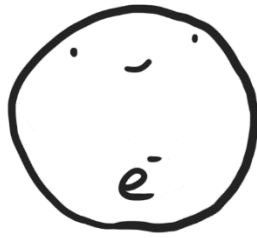
ELECTRON



MAJORANA PAIR



ELECTRONS SITTING IN A SEMICONDUCTOR NANOWIRE



ELECTRON

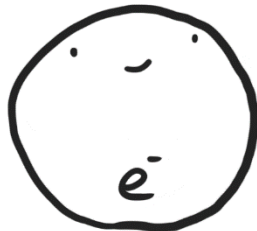


MAJORANA PAIR



ELECTRONS SITTING IN A SEMICONDUCTOR NANOWIRE





ELECTRON



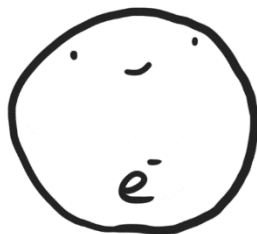
MAJUMDAR PAIR



ELECTRONS SITTING IN A SEMICONDUCTOR NANOWIRE







ELECTRON



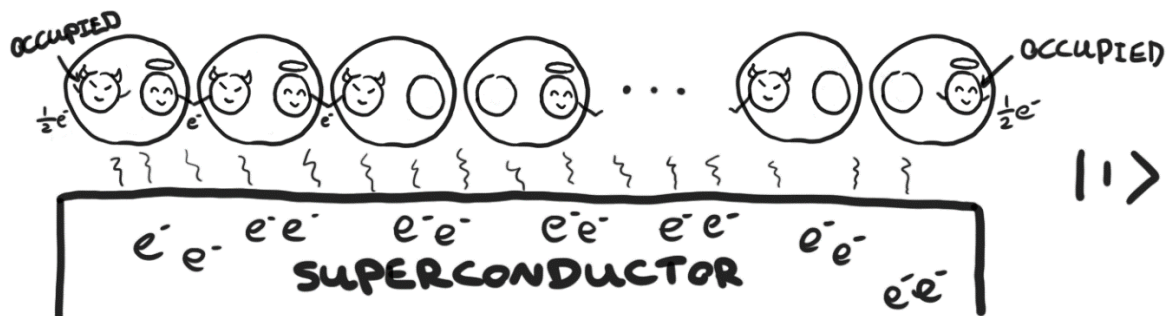
MAJUMDAR PAIR



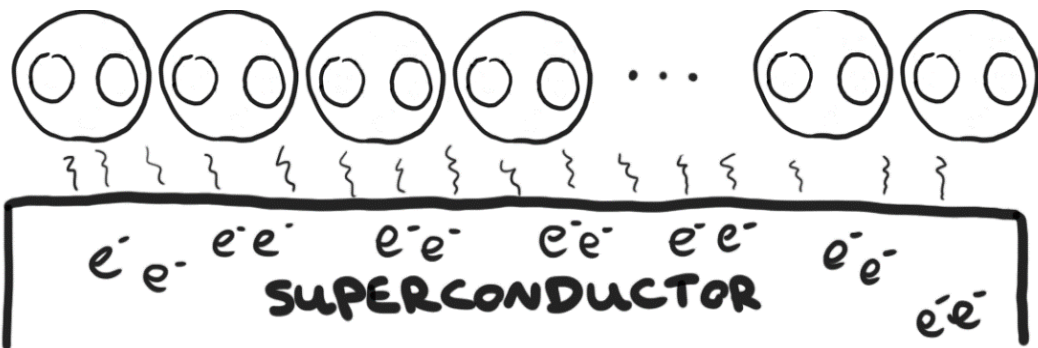
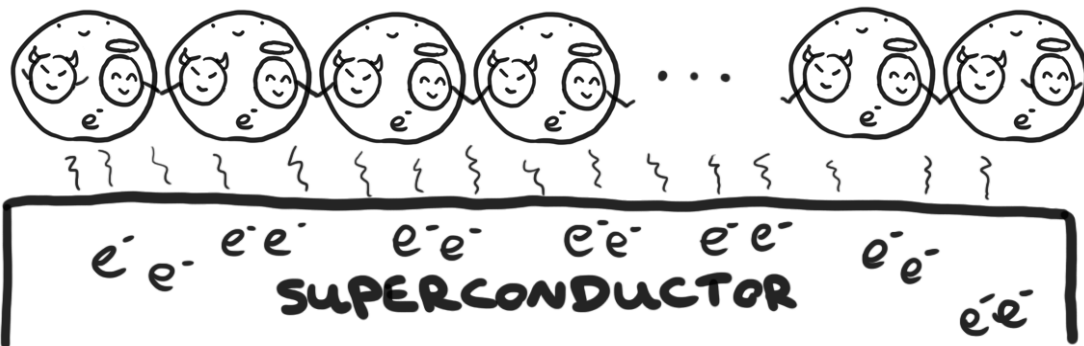
ELECTRONS SITTING IN A SEMICONDUCTOR NANOWIRE

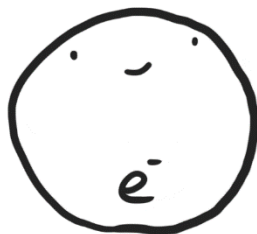


$|0\rangle$



$|1\rangle$





ELECTRON



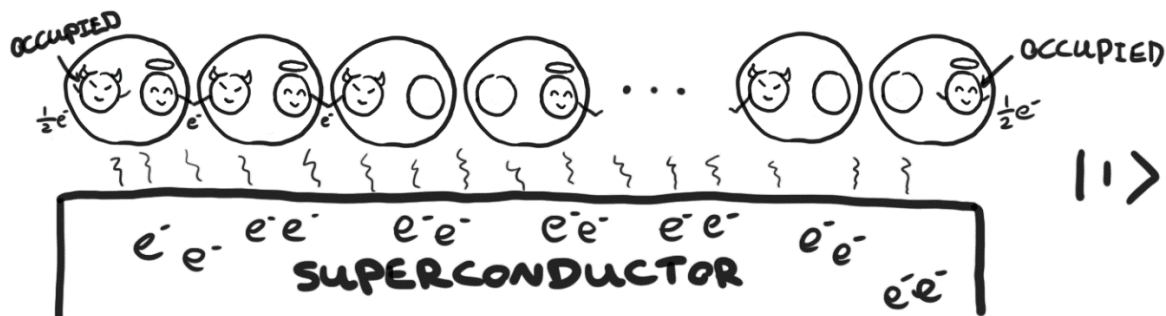
MAJORANA PAIR



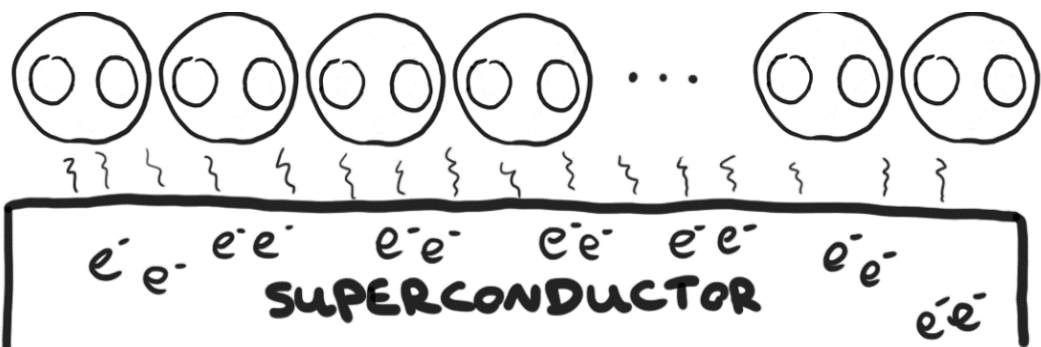
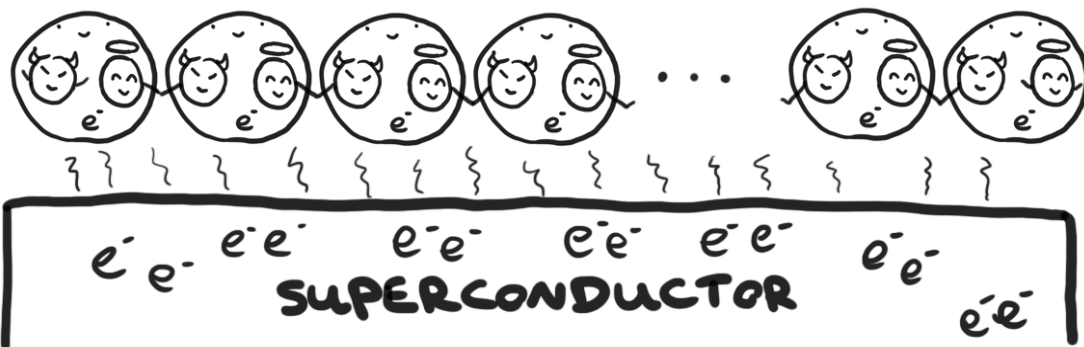
ELECTRONS SITTING IN A SEMICONDUCTOR NANOWIRE



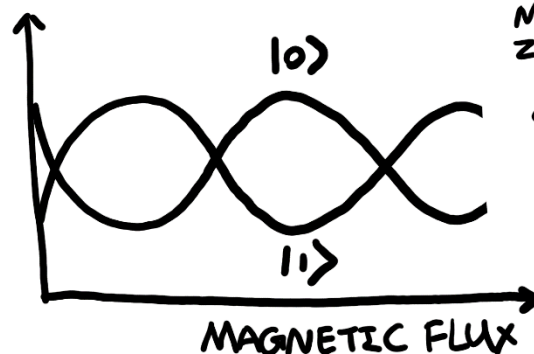
$|0\rangle$



$|1\rangle$



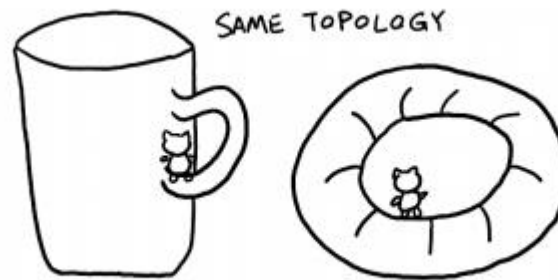
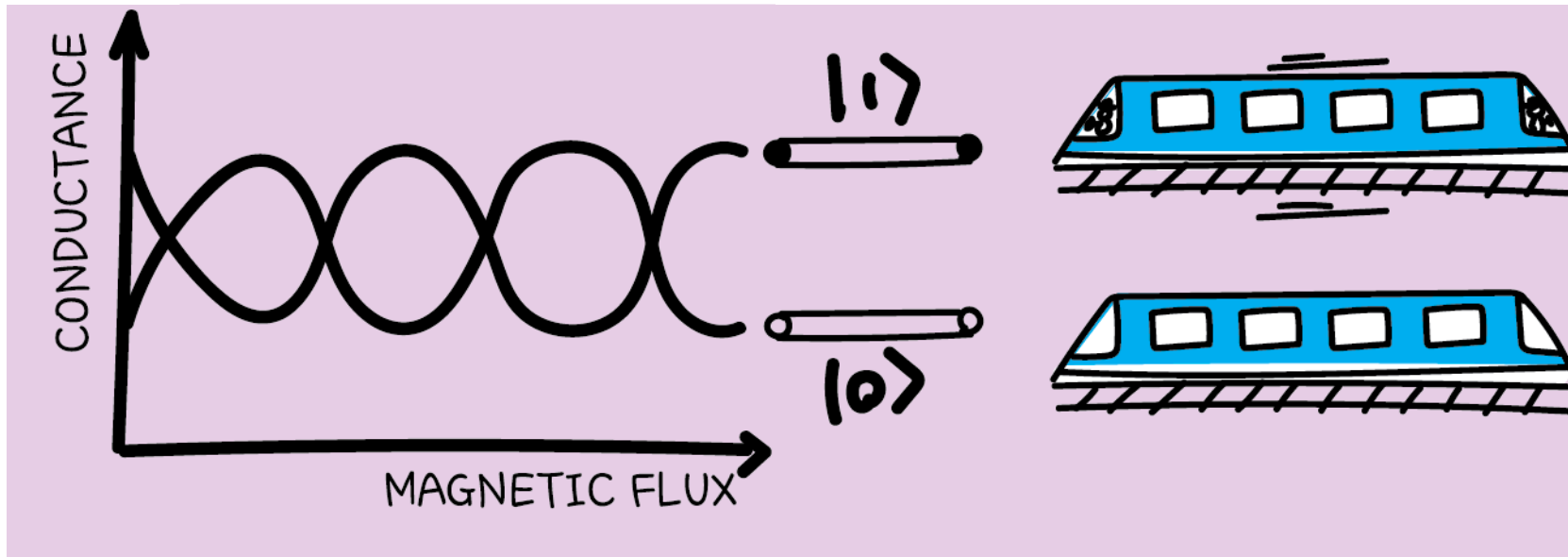
CONDUCTANCE



MAJORANA  
ZERO MODES

— OCCUPIED

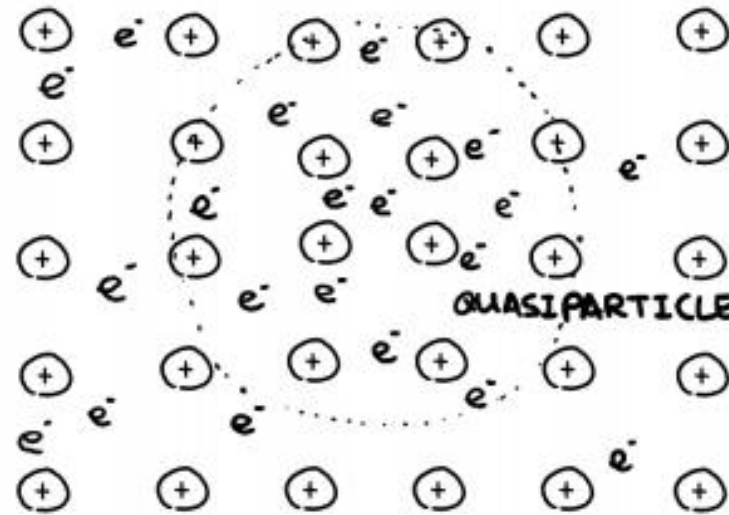
— EMPTY



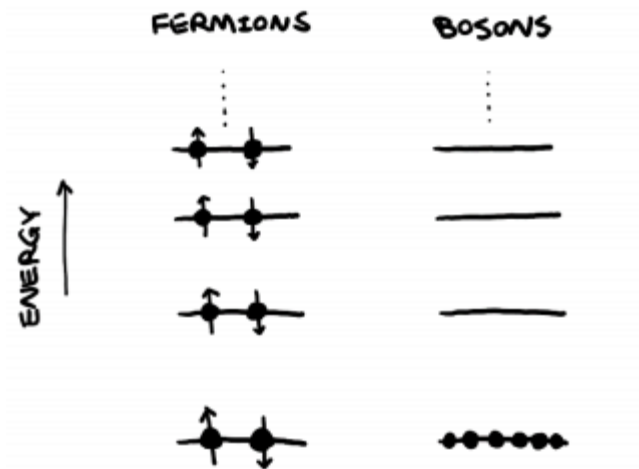
*An often-used analogy: a teacup and a donut have the same topology. Locally we cannot distinguish them.*

# Quasiparticles

- Collective interactions between many microscopic particles
- Approximately be described as an unperturbed particle travelling in free space with an effective mass differing from its original electron mass



# Majorana quasiparticles - anyons



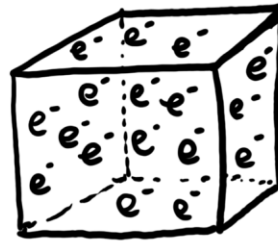
**Fermion** (named after Enrico Fermi): a particle with spins that are integer multiples of  $\frac{1}{2}$ . Fermions with the same spin cannot occupy the same energy level due to Pauli exclusion principle, following Fermi-Dirac statistics. For example, an electron is a fermion as it has spin  $\frac{1}{2}$ . Each energy level can accommodate a maximum of two electrons with opposite spins.

**Boson** (named after Satyendra Nath Bose): a particle with integer spin. For example, a photon with spin 0 is a boson. Bosons can occupy the same energy level, following Bose-Einstein statistics.

**Anyon** (named after Jean Anyon): a quasiparticle that is neither a fermion nor a boson. They only appear in one- or two-dimensional systems. They follow the braiding statistics (see later).



# Dimensionality



3D

IN A METAL



2D

AT THE  
INTERFACE  
BETWEEN  
TWO  
SEMICONDUCTOR  
LAYERS



1D

IN A  
NANOWIRE

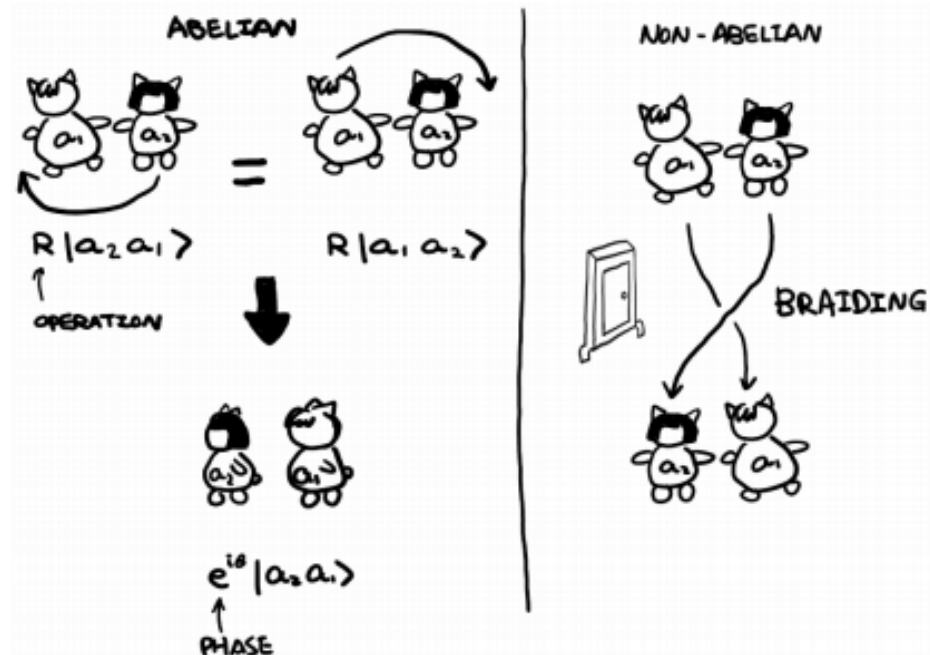


0D

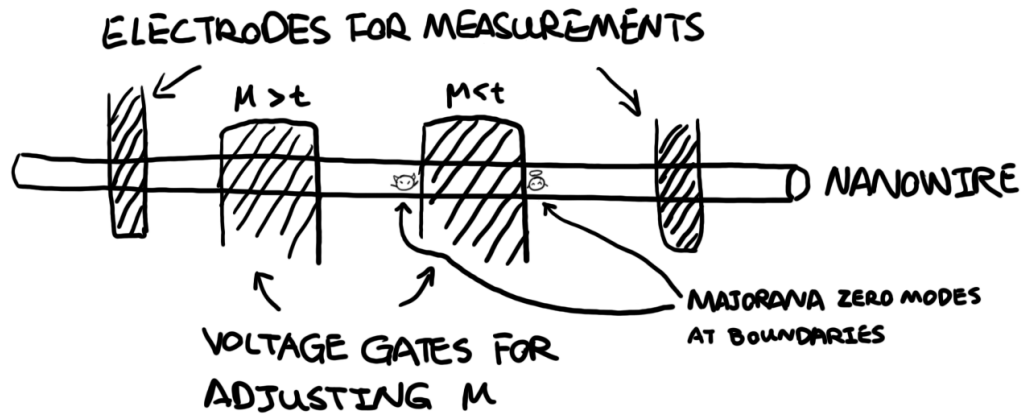
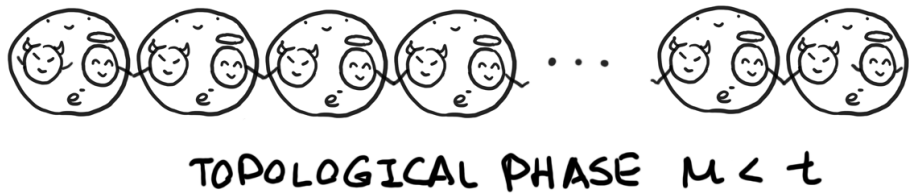
IN AN  
ARTIFICIALLY  
CREATED  
TRAP

# Non-abelian anyons

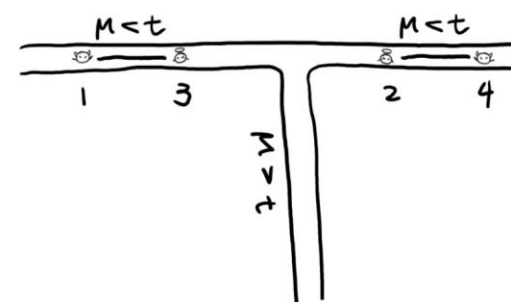
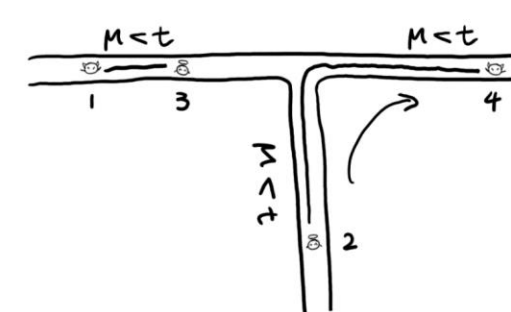
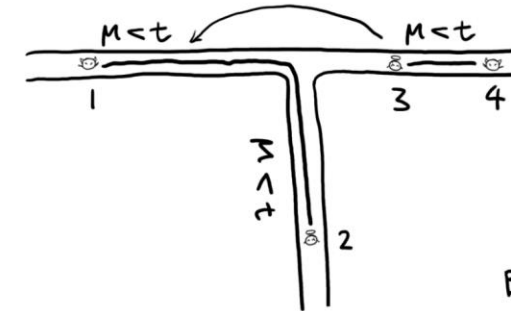
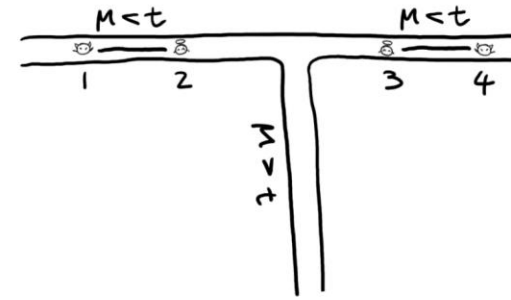
- There are two kinds of anyons, abelian and non-abelian. The terminology comes from group theory, in which an abelian group obeys commutative operations. This means that when applying a group operation to two group elements in an abelian group, the result does not depend on the order in which the elements are written. Interchanging two elements is effectively adding an overall phase (can be complex) to the state, which does not have a measurable effect. A non-abelian group, however, does depend on the order of elements. An operation on a non-abelian group needs to be a unitary matrix. Remember our quantum gates are unitary matrices. Interchanging two non-abelian anyons is equivalent to applying a quantum gate to the state.



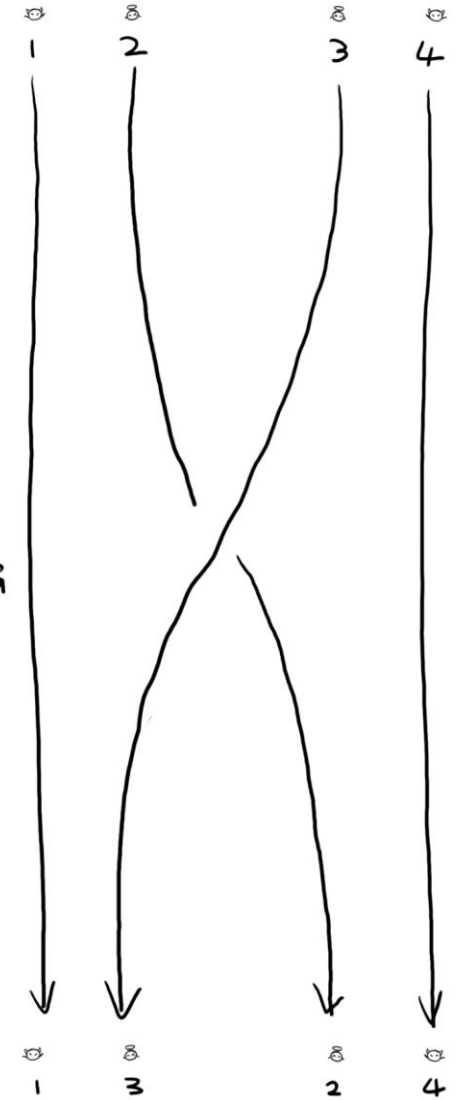
$$H = \underset{M}{\text{CHEMICAL POTENTIAL}} + \underset{t}{\text{KINETIC}}$$



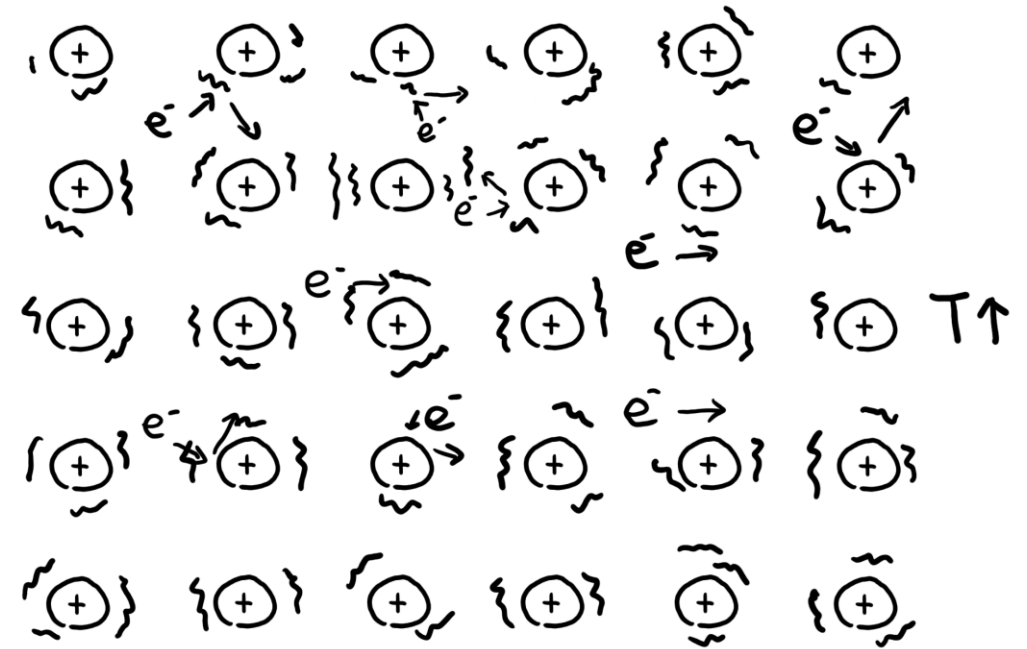
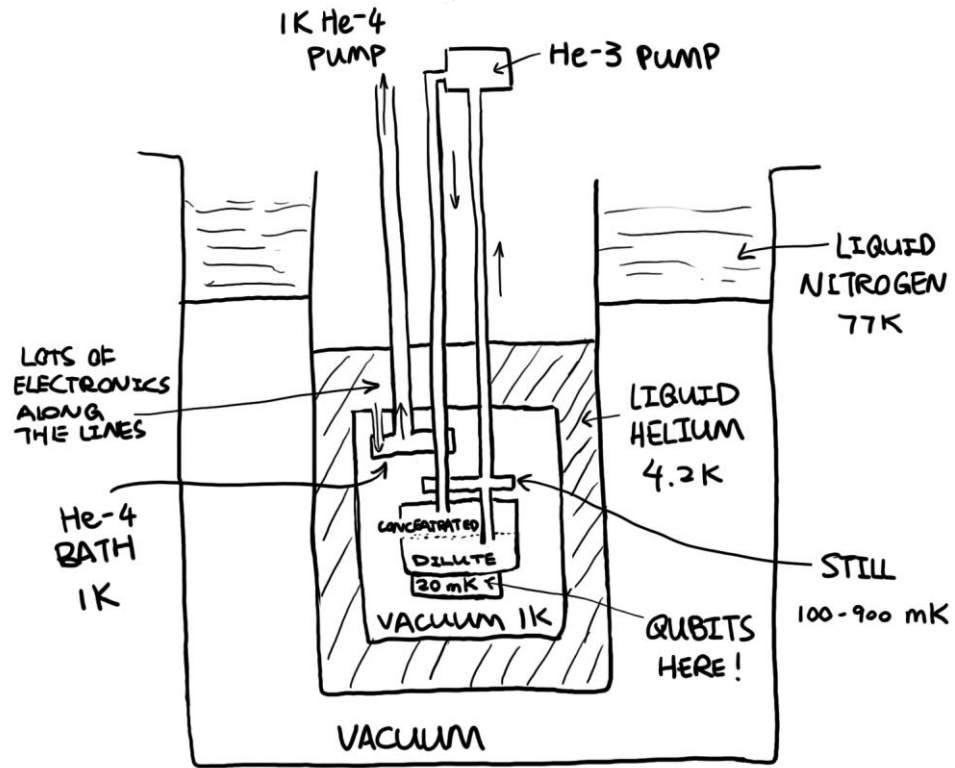
## T-JUNCTION



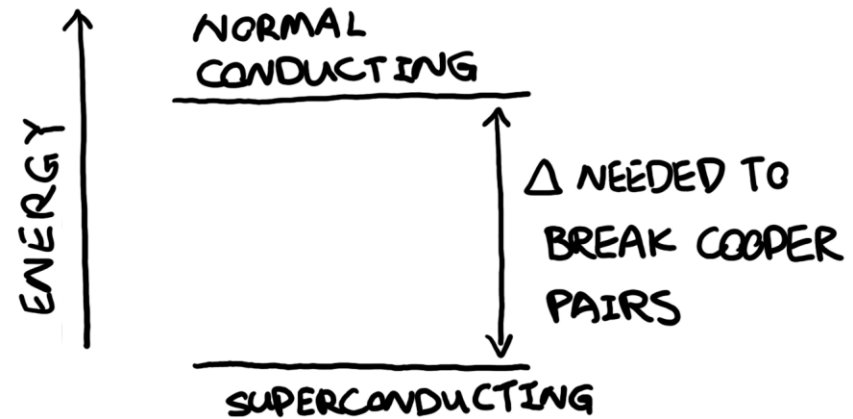
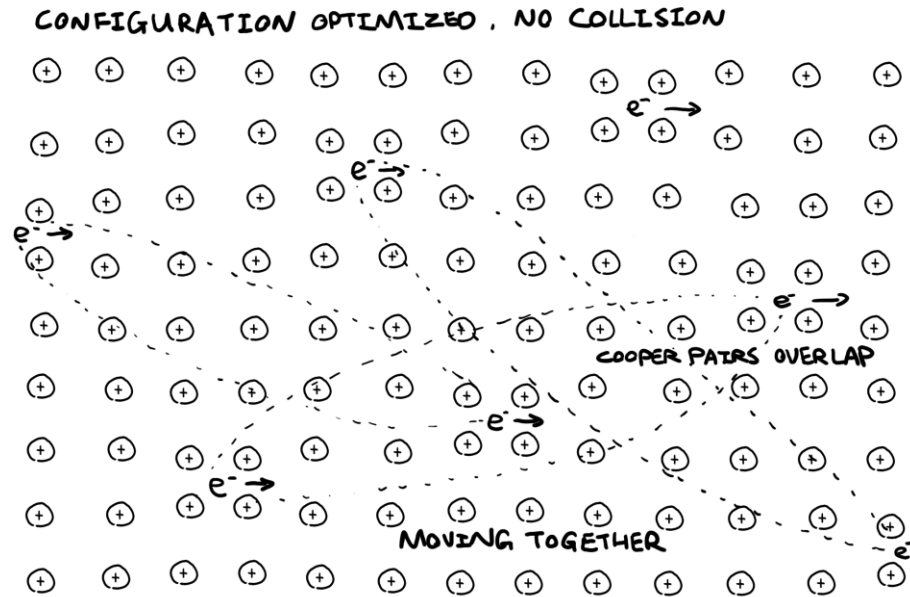
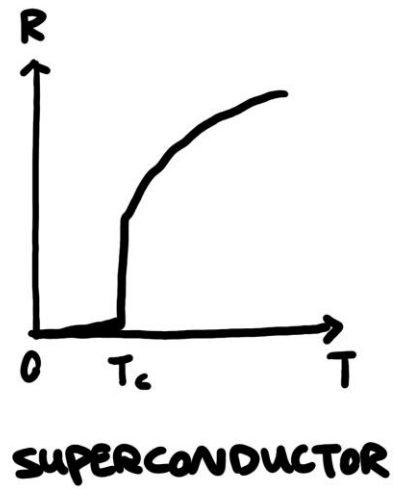
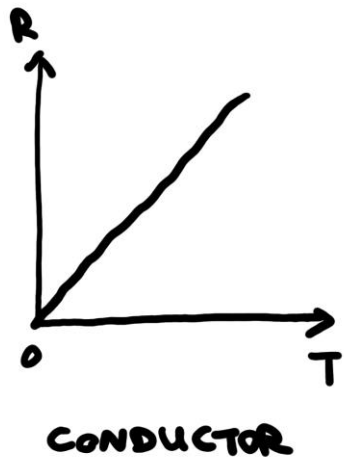
BRAIDING



# Low temperature



# Superconductors








# Reading: Topological Quantum Computer

- <https://scipost.org/SciPostPhys.3.3.021/pdf>
- <https://arxiv.org/abs/cond-mat/0010440>
- <https://arxiv.org/abs/cond-mat/9906453>



# Guest lectures


- Sept 13, **Prof. Terrill Frantz**, Harrisburg University of Science and Technology, **Quantum Cryptography**
  - Sept 20, **Prof. Chris Ferrie**, University of Technology Sydney, **Quantum Tomography** – time-change announcement
- 

# Contribute and demonstrate your skills

- [\*\*microsoft/Quantum\*\*](#): Samples and tools to help get started with the Quantum Development Kit.
- [\*\*microsoft/QuantumLibraries\*\*](#): Standard and domain-specific libraries for the Quantum Development Kit.
- [\*\*microsoft/QuantumKatas\*\*](#): Self-paced programming exercises for learning quantum computing and the Q# programming language.
- [\*\*microsoft/qsharp-compiler\*\*](#): The Q# compiler, Visual Studio extension, and Visual Studio Code extension.
- [\*\*microsoft/qsharp-runtime\*\*](#): Simulation framework, code generation, and simulation target machines for the Quantum Development Kit.
- [\*\*microsoft/iqsharp\*\*](#): Jupyter kernel and Python host functionality for Q#, as well as Docker images for using IQ# in cloud environments.
- [\*\*MicrosoftDocs/quantum-docs-pr\*\*](#): Source code for the documentation published at <https://docs.microsoft.com/quantum>.

# Q# Community

- <https://qsharp.community/>

 **Q# Community**  
Community-Driven Quantum Development

AboutProjectsCommunity BlogResourcesContribute

## Contributing to Q# Community

This community encourages feedback and contributions. Thank you for your interest in making the Q# Community better! There are several ways you can get involved.

### Adding a new repository to the community

Are you someone who has a project they want to contribute to the Q# community organization? There are a few main benefits to doing so:

- Make it easier for new contributors to find your project 📌
- Crowd source maintaining and developing with others familiar with Q#

#### On this page

- Adding a new repository to the community
  - Steps to add a new project
  - Licencing repositories in the community organization
- Reporting issues and suggesting new features
- Finding issues you can help with
- Contributions we accept
- Making changes to the code
  - Style guidelines
  - Testing and Packaging
- Review Process