

## Introduction to Quantum Computing



Prof. dr. Egon Pavlica  
University of Nova Gorica

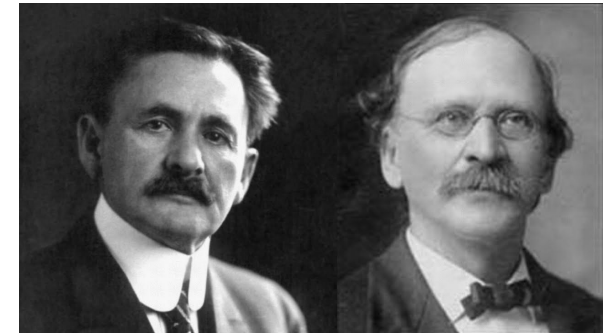
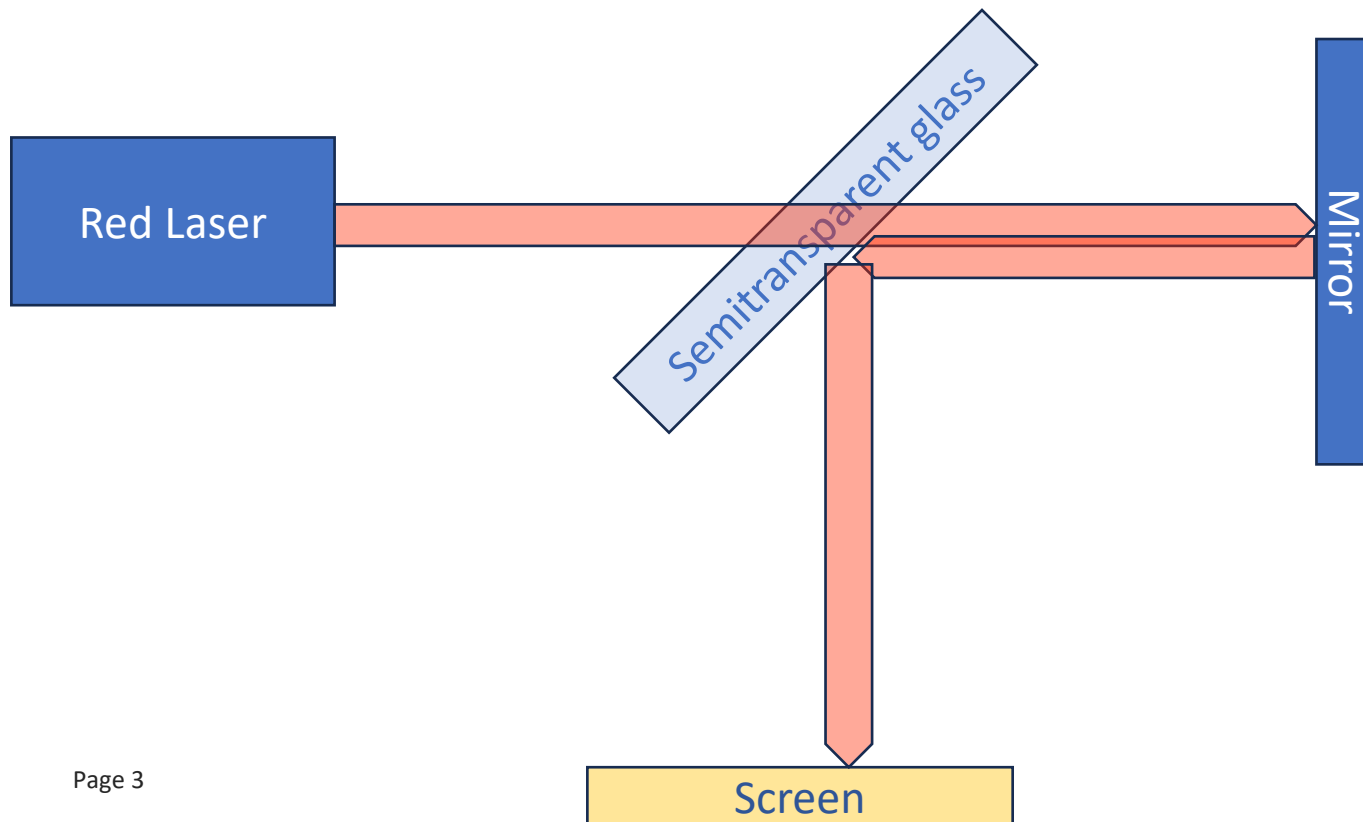
Supported by



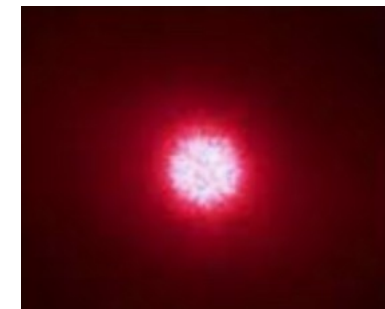
# Outline

- Michelson-Morley experiment
- Two-slits experiment
- Classical computer
- Introduction to qubit
- Two beam—splitters' experiment
- Mathematical description of two beam-splitters' experiment
- Tutorial on quantum computer (Qiskit)

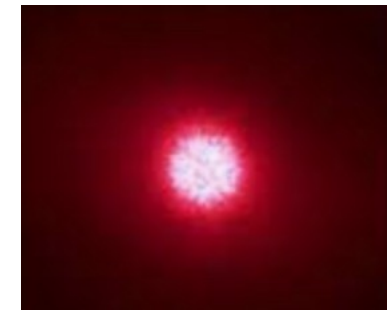
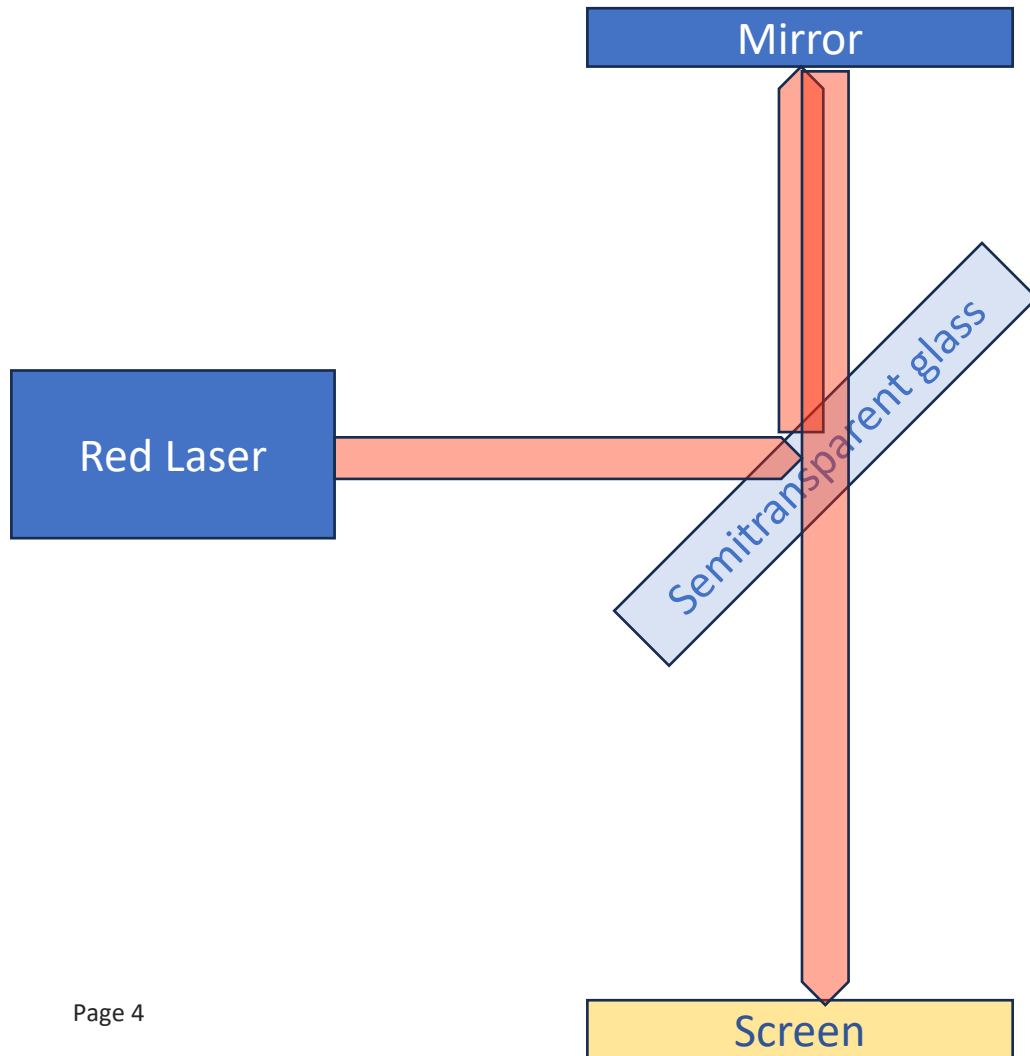
# Michelson – Morley experiment



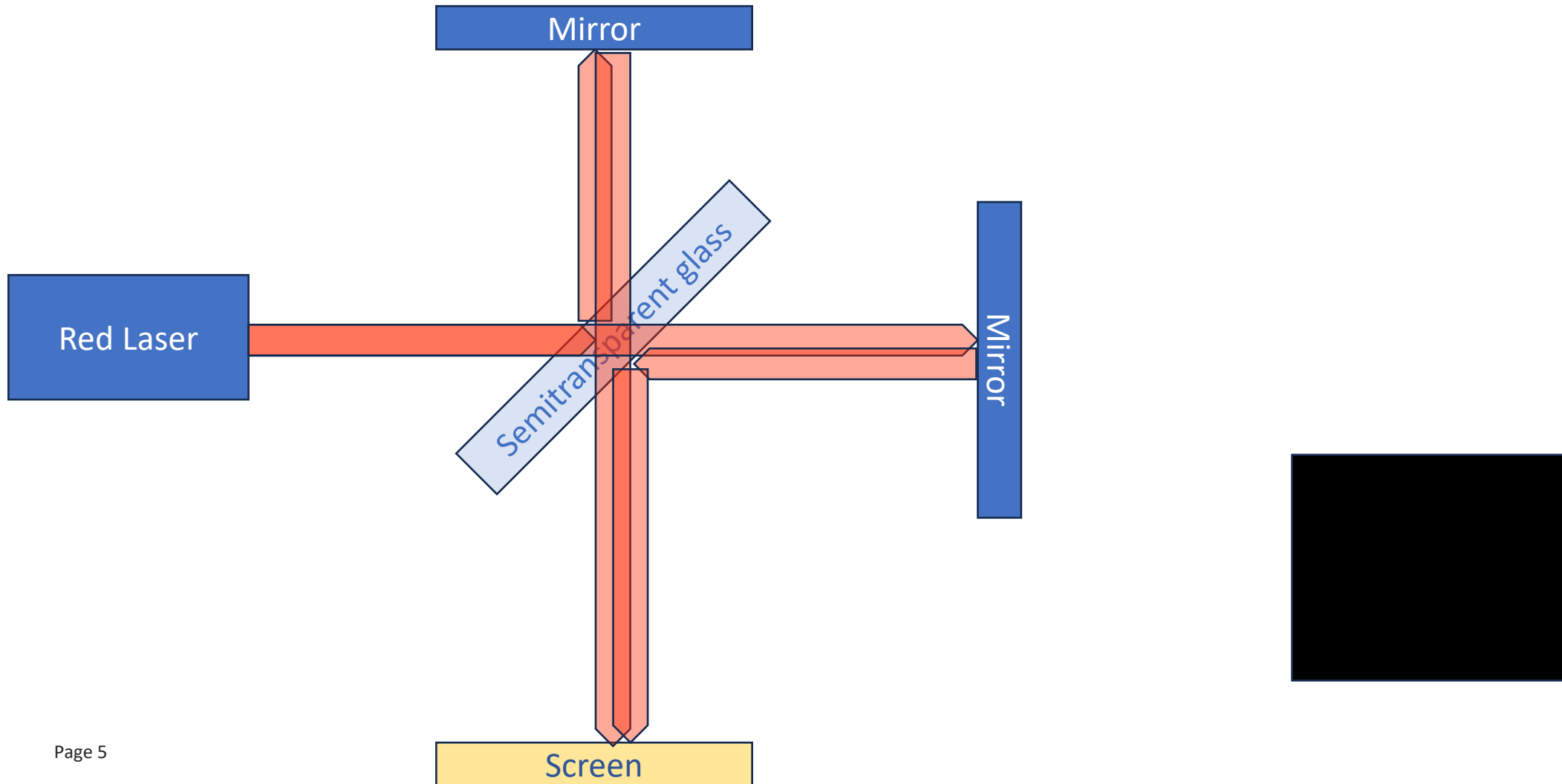
between April and July 1887



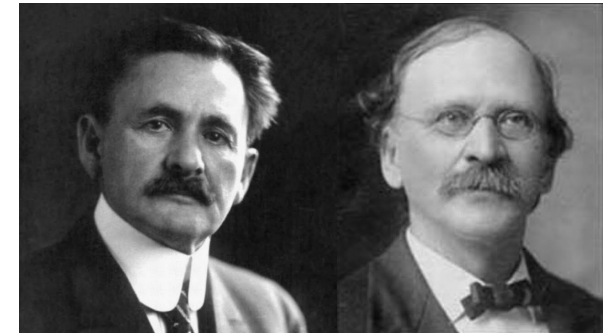
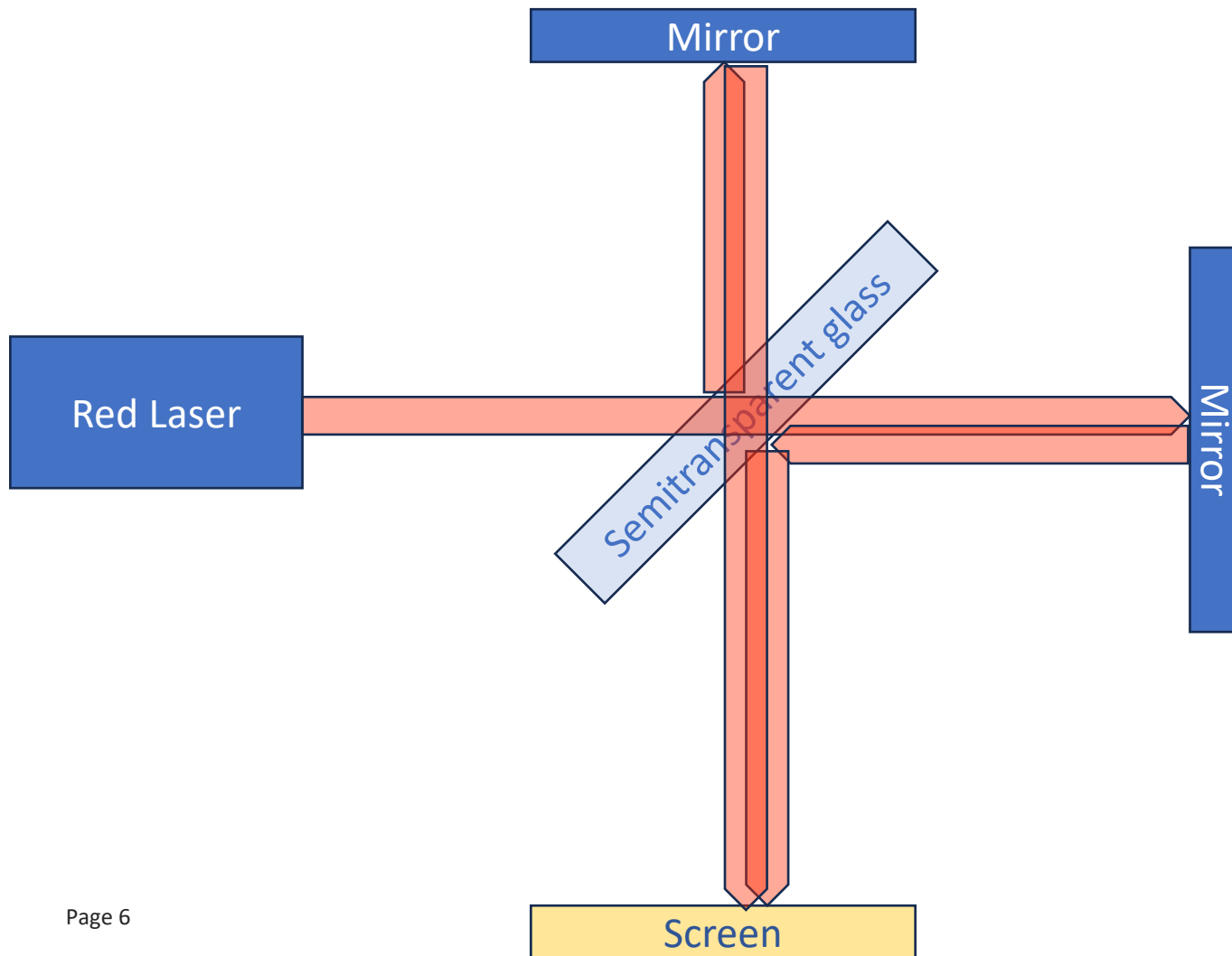
# Michelson – Morley experiment



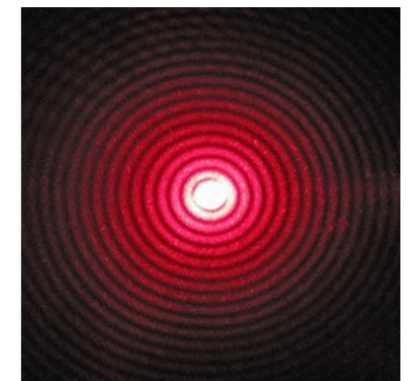
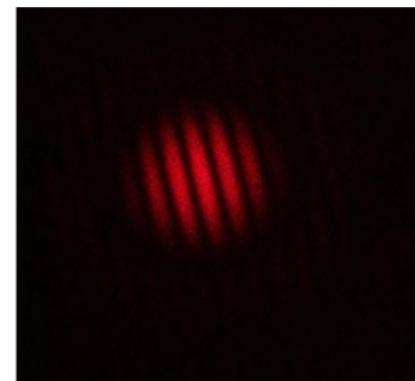
# Michelson – Morley experiment



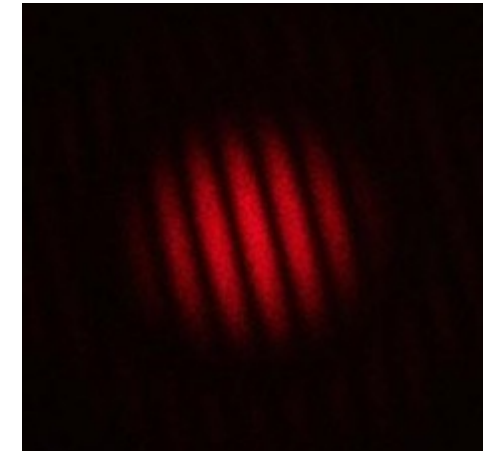
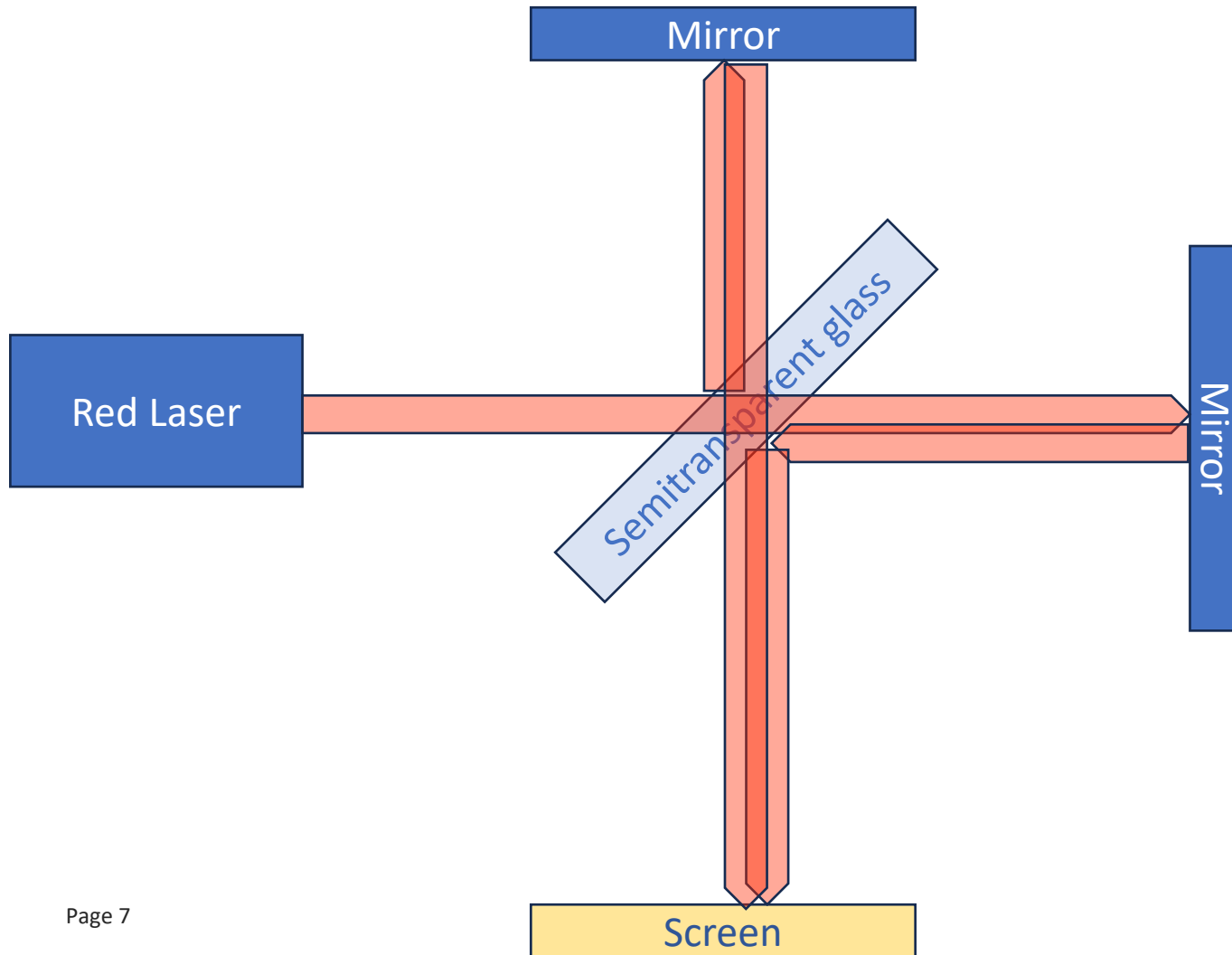
# Michelson – Morley experiment



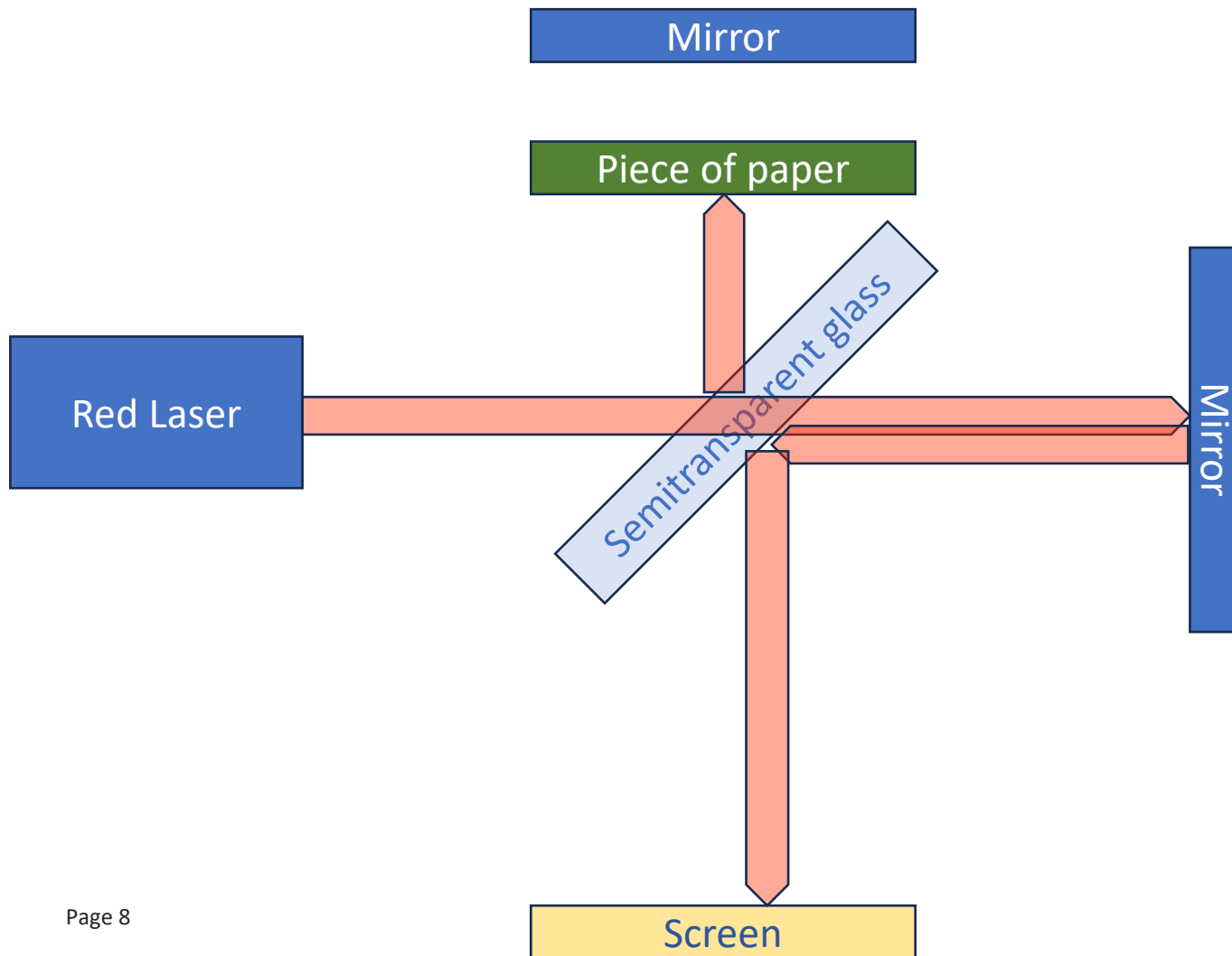
between April and July 1887



# Michelson – Morley experiment

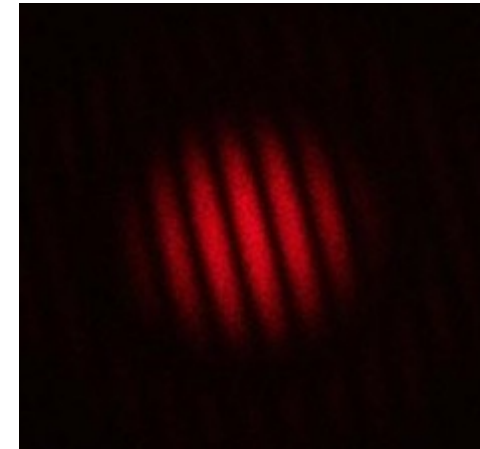
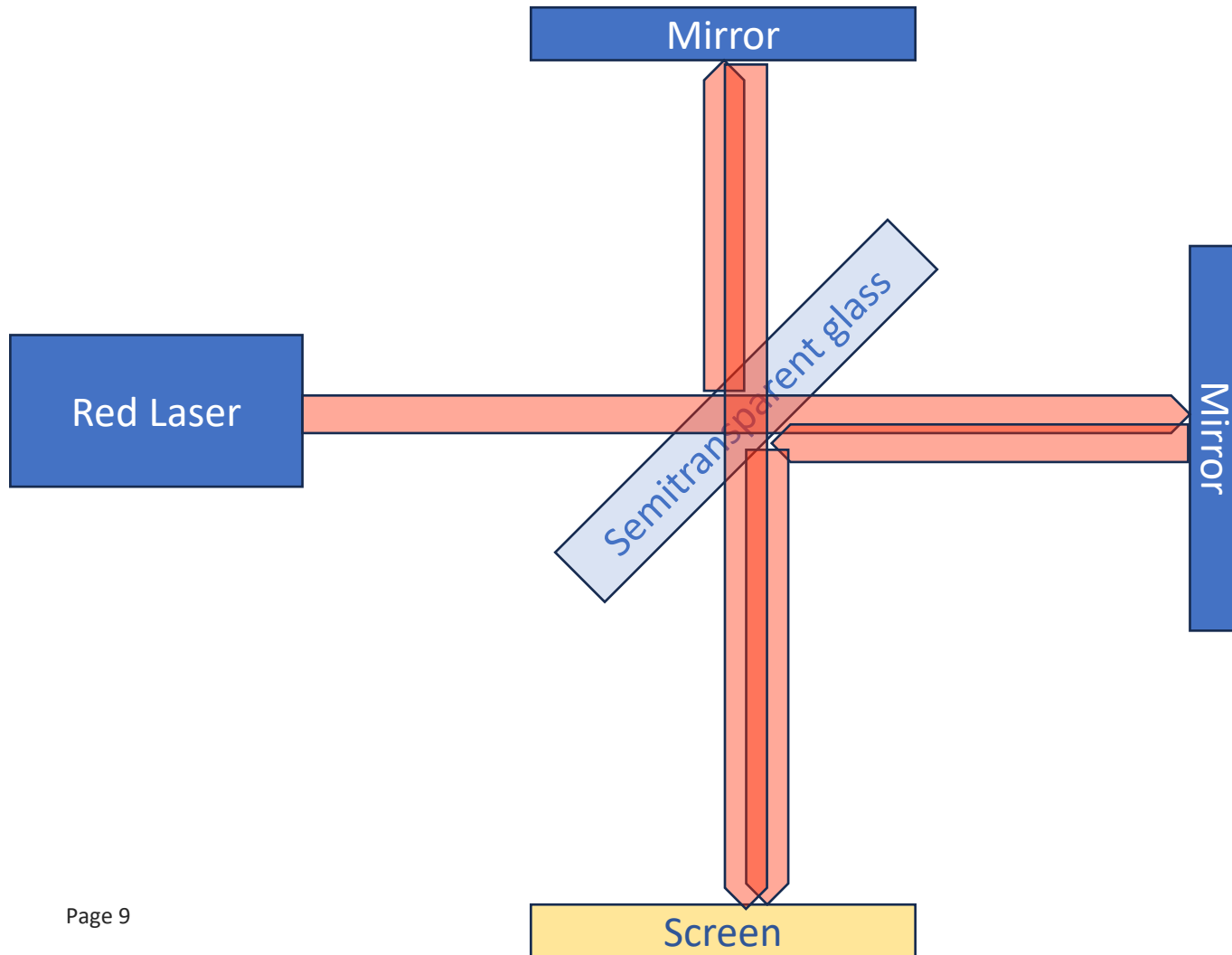


# Michelson – Morley experiment

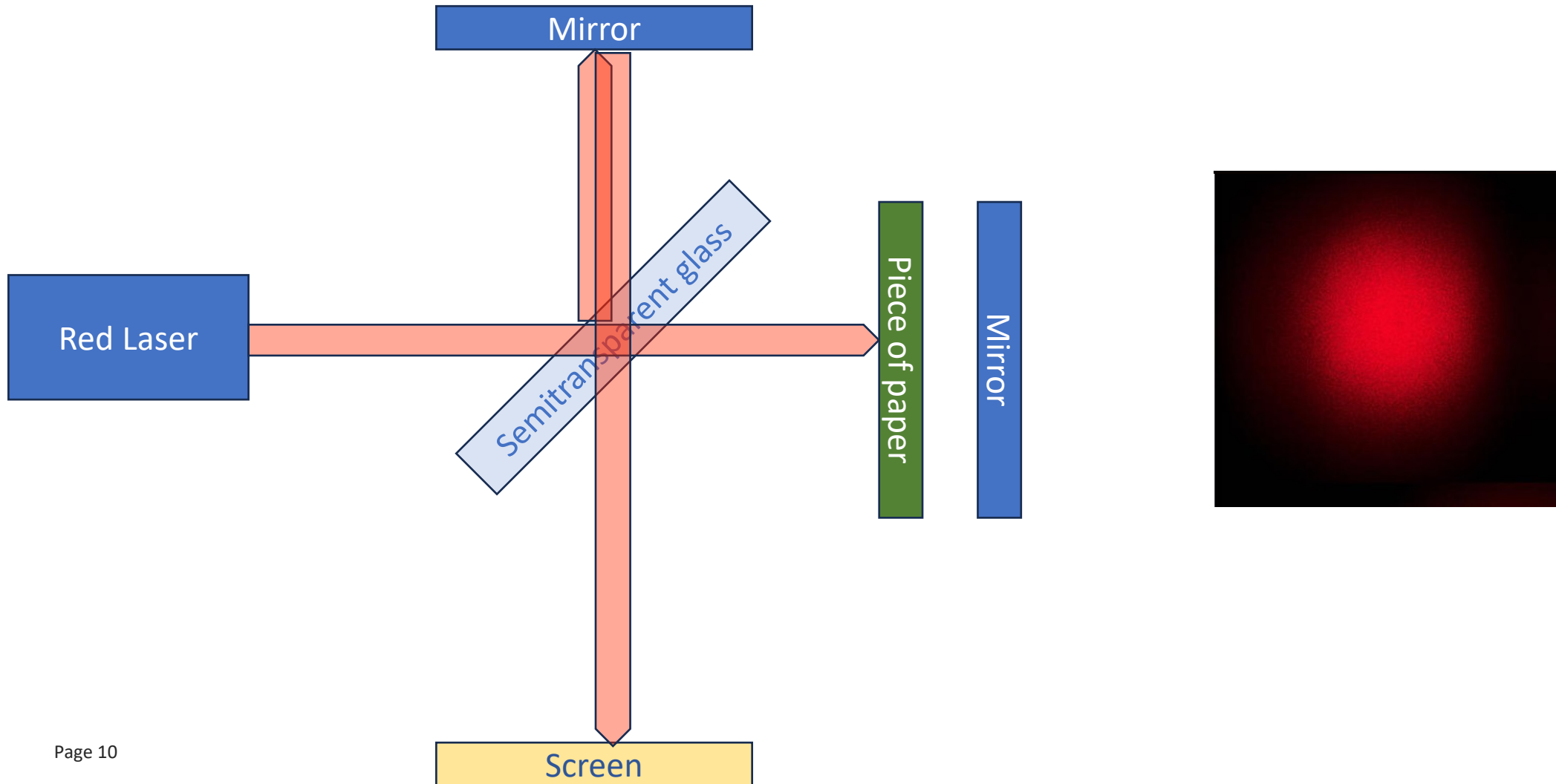




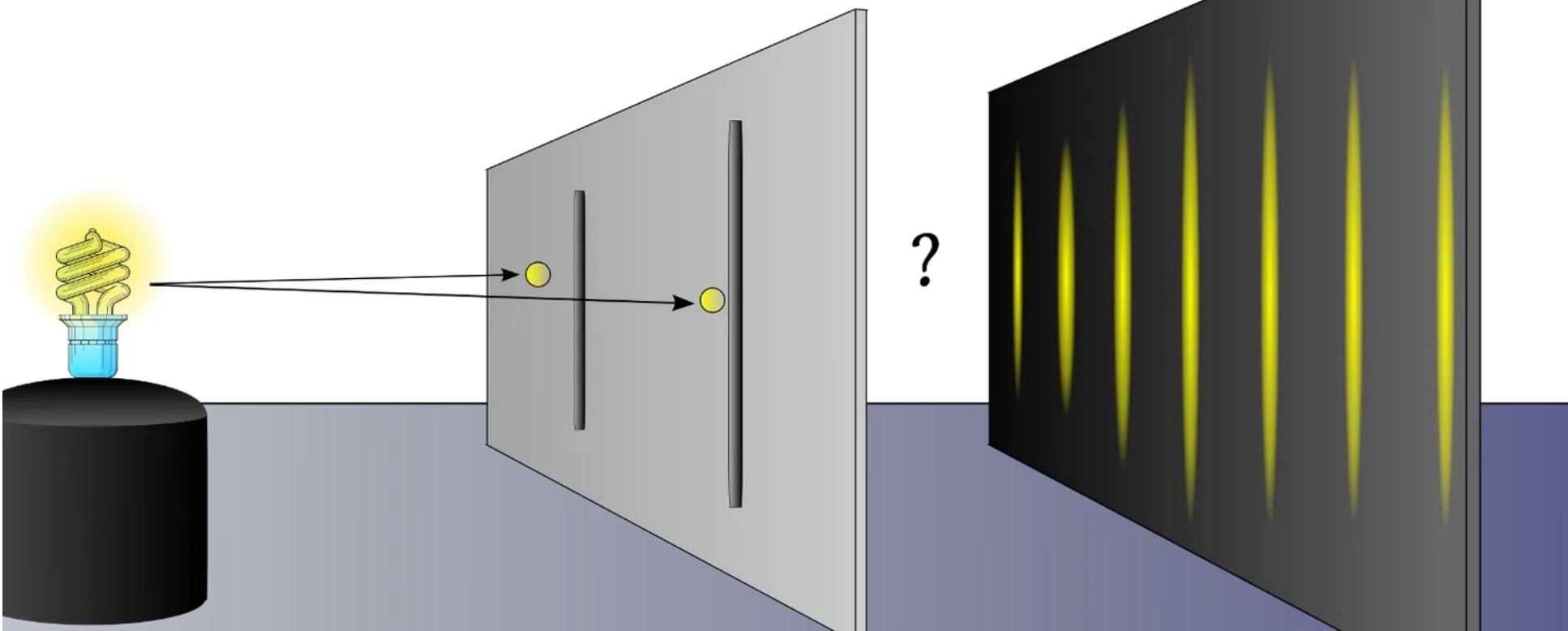
# Michelson – Morley experiment



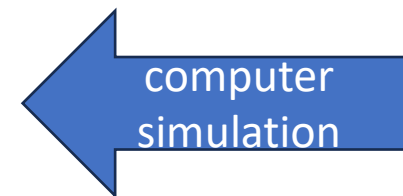
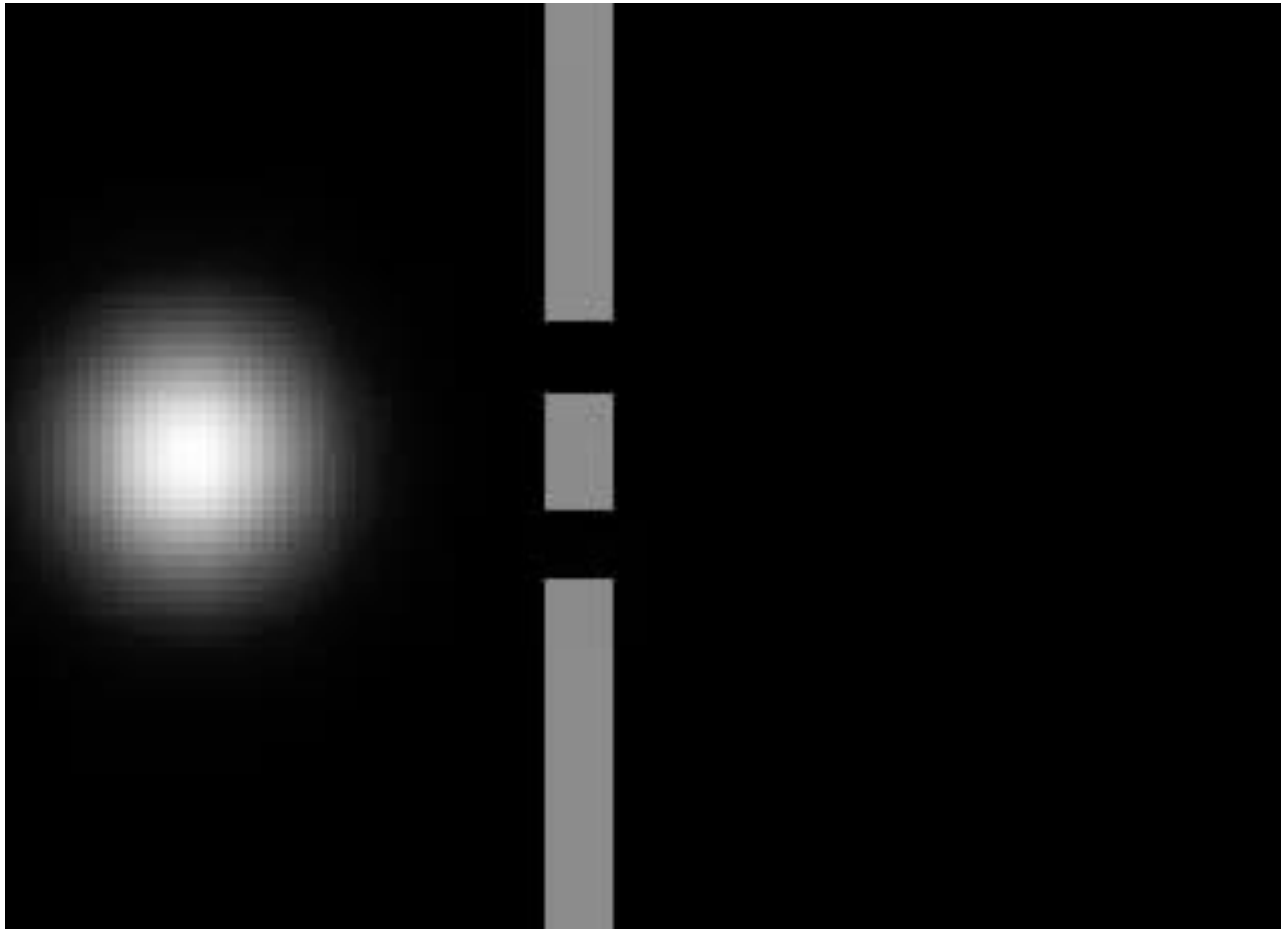
# Michelson – Morley experiment



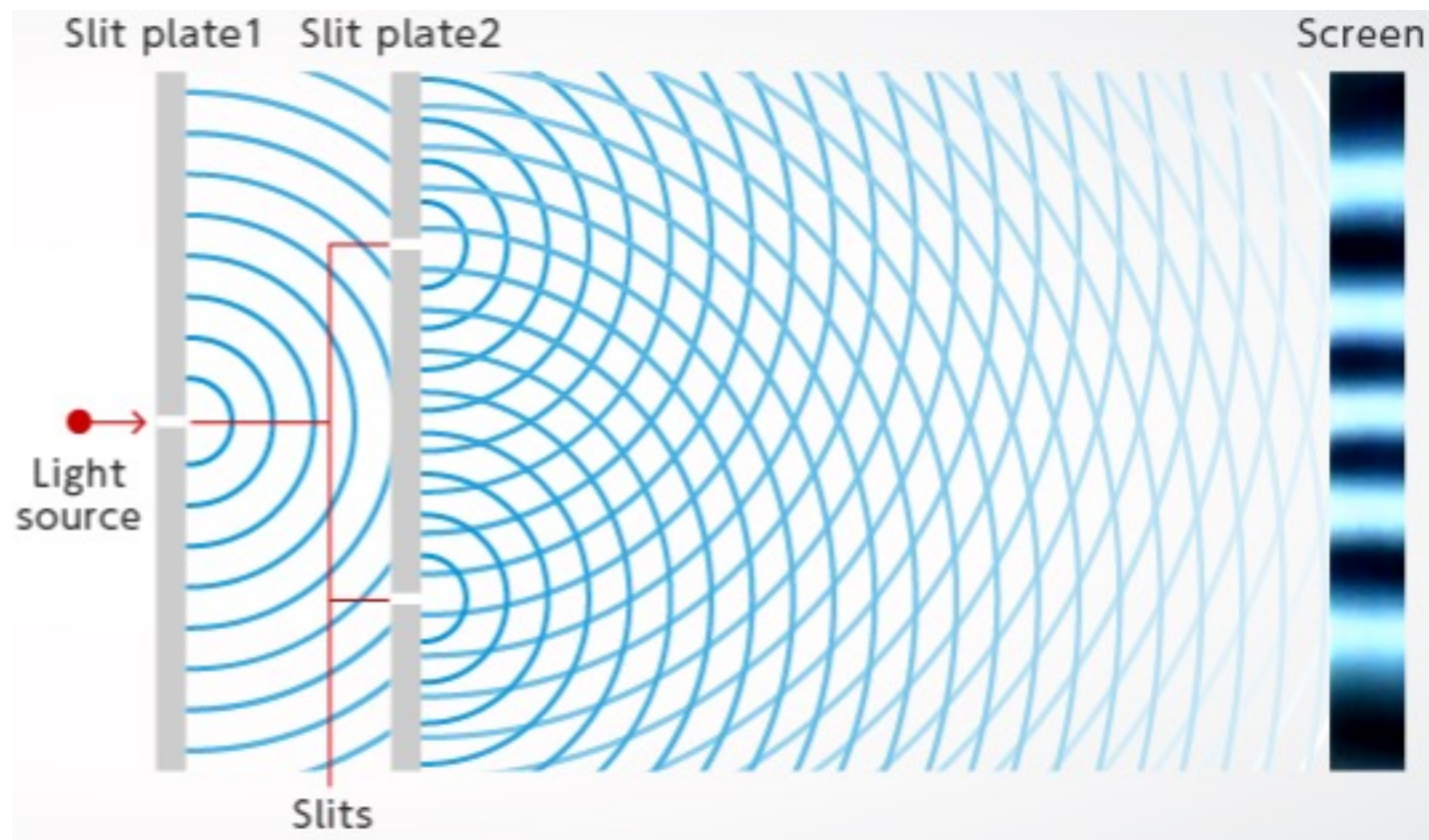
# Two-slits experiment setup



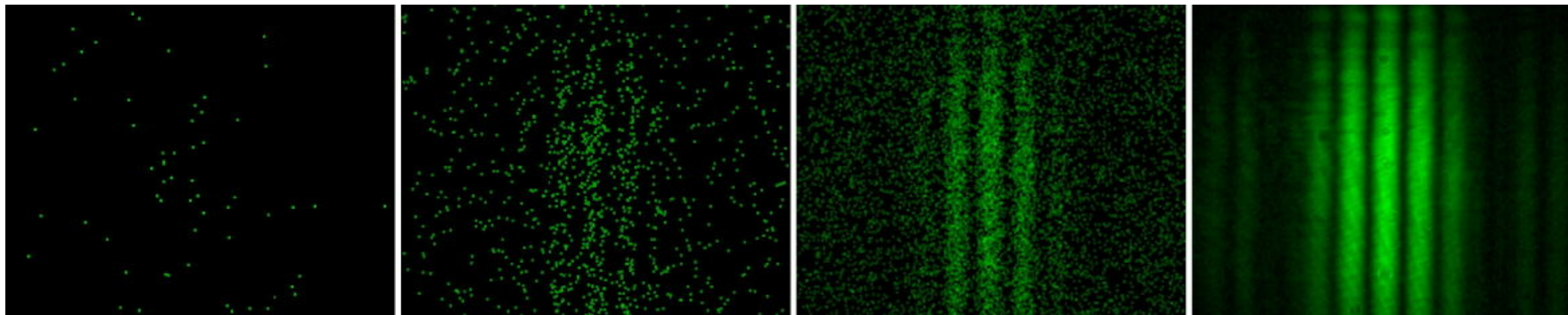
# Two-slits experiment computer simulation



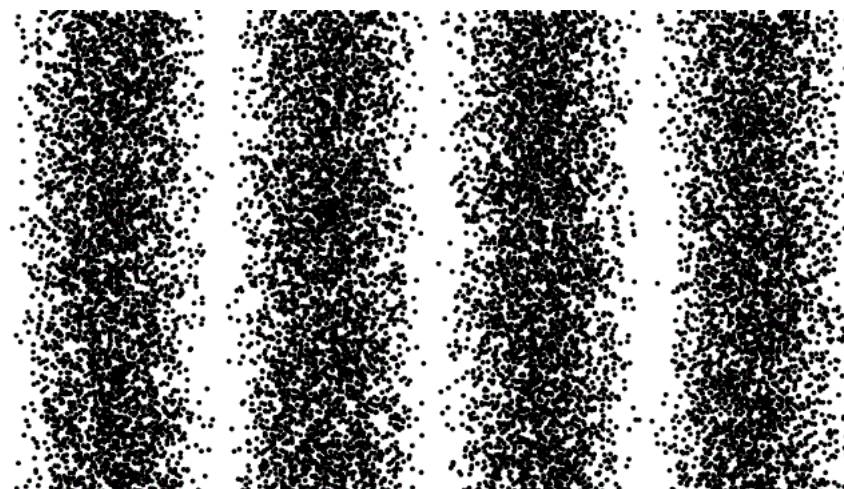
# Two-slits experiment



# Two-slits experiment with low light intensity

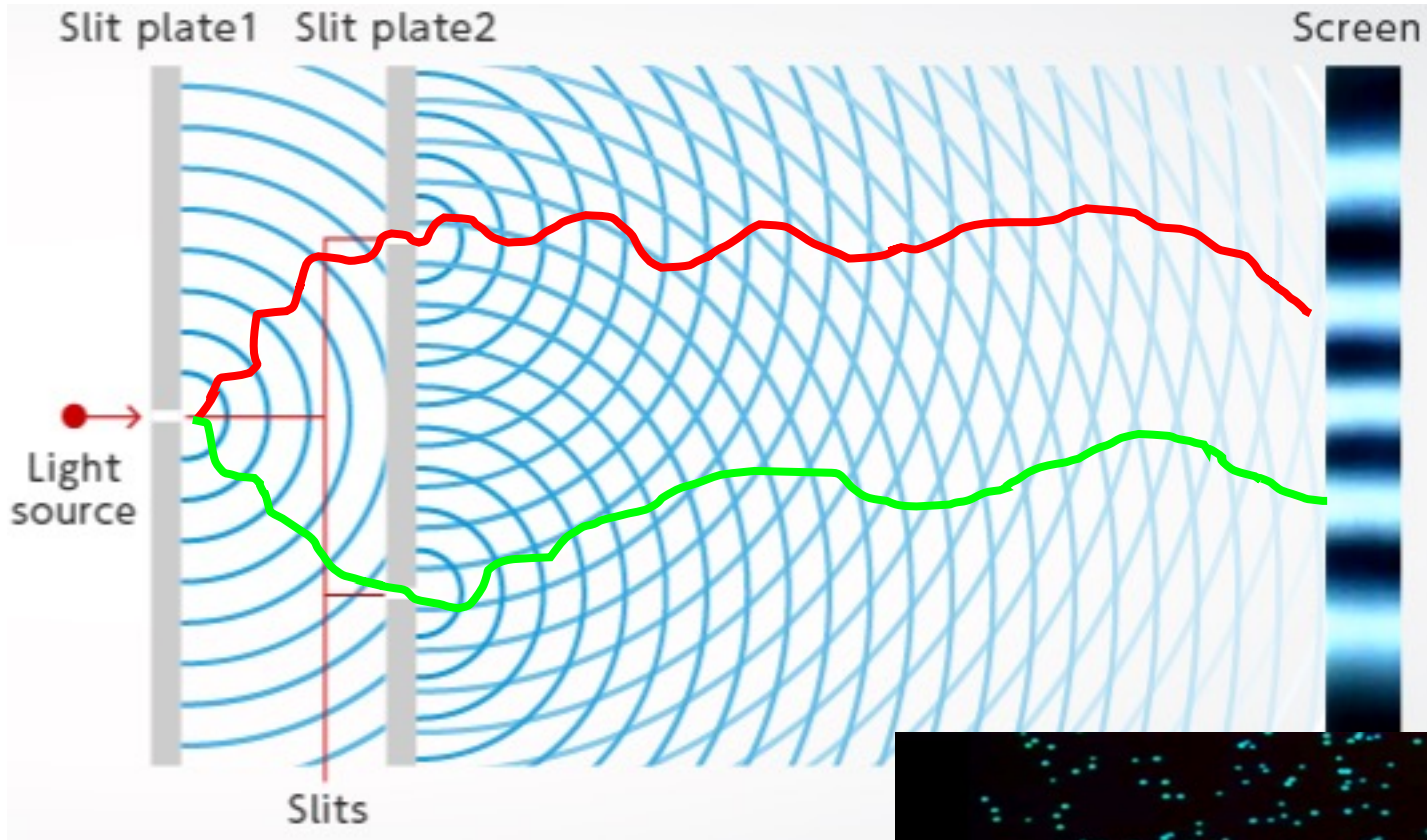


number of photons



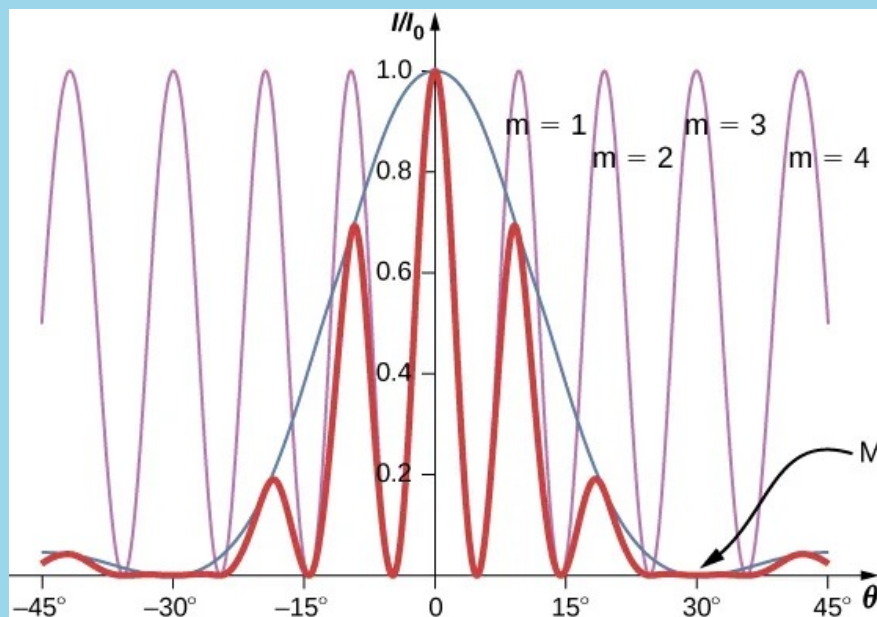
computer  
simulation

# We don't know where a single photon will travel!!!



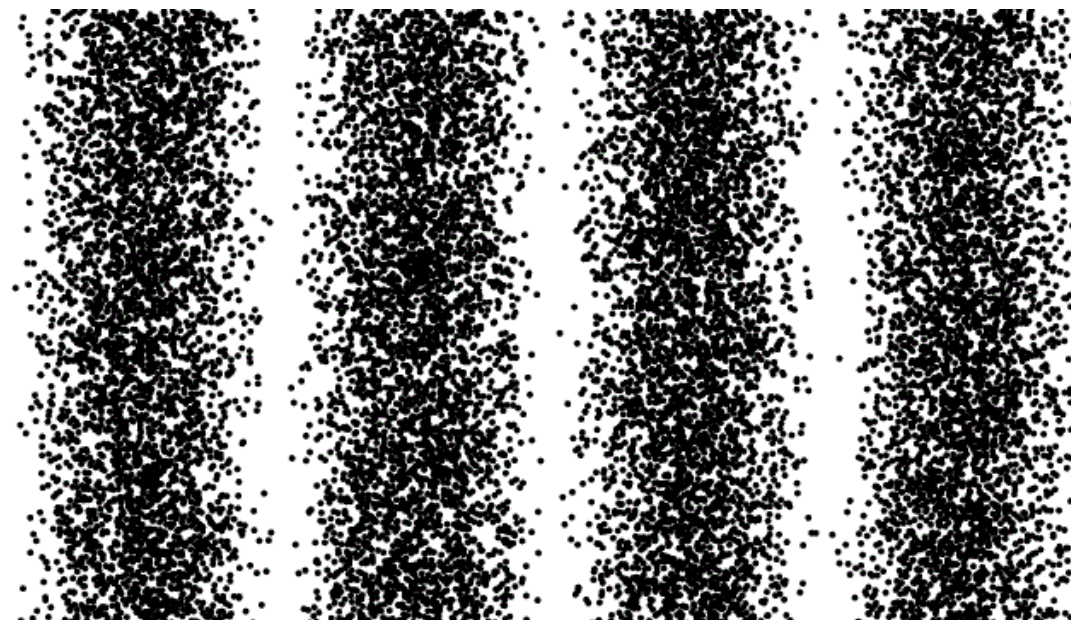
# Mathematics

- We don't know where a single photon will travel -> we know **the probability**
- We know where bunch of photons will travel – interference pattern



# Photons

- Single photon knows where to go
- Photons interfere with each-other
- Photon obeys **quantum mechanics**

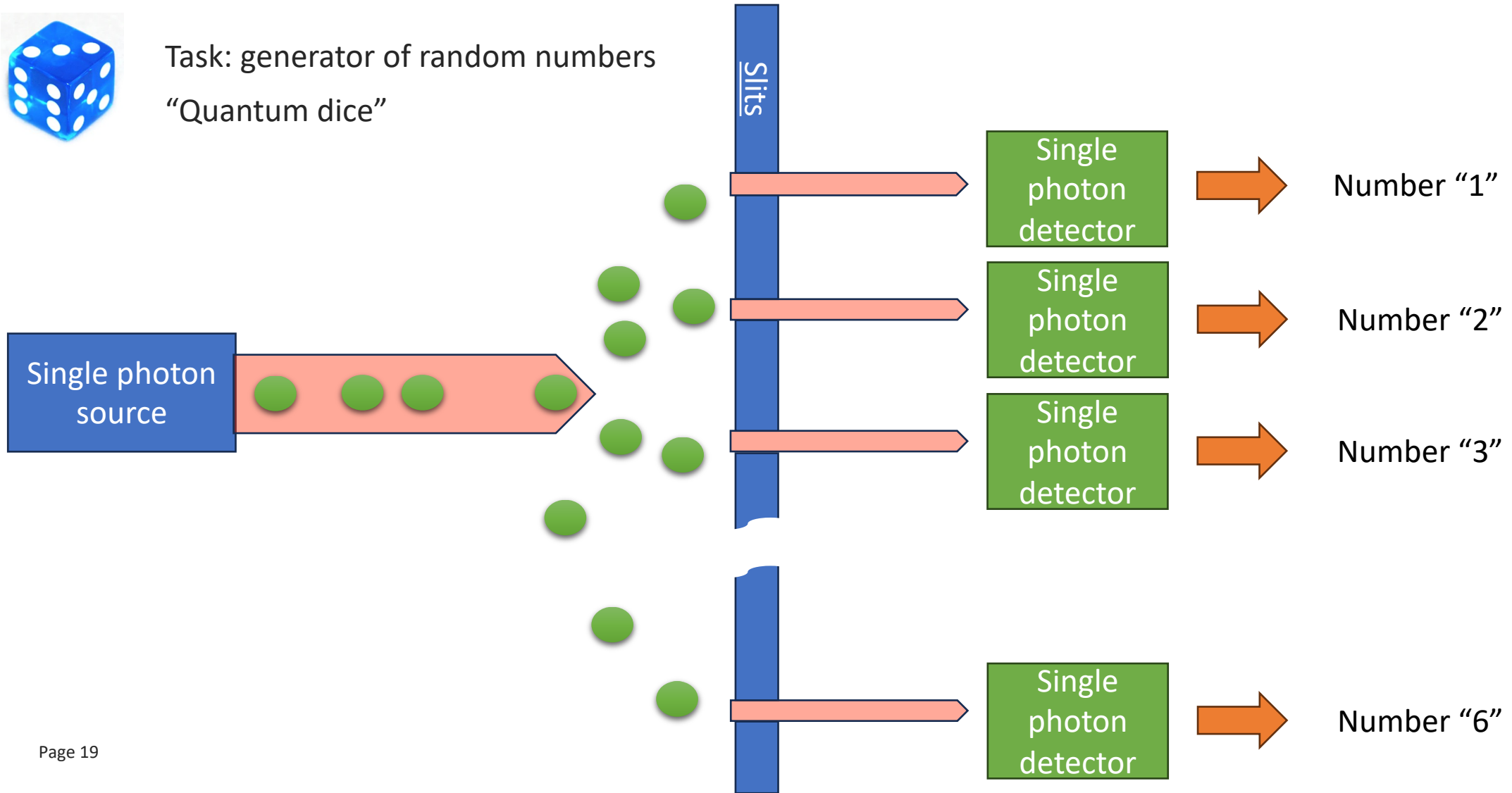




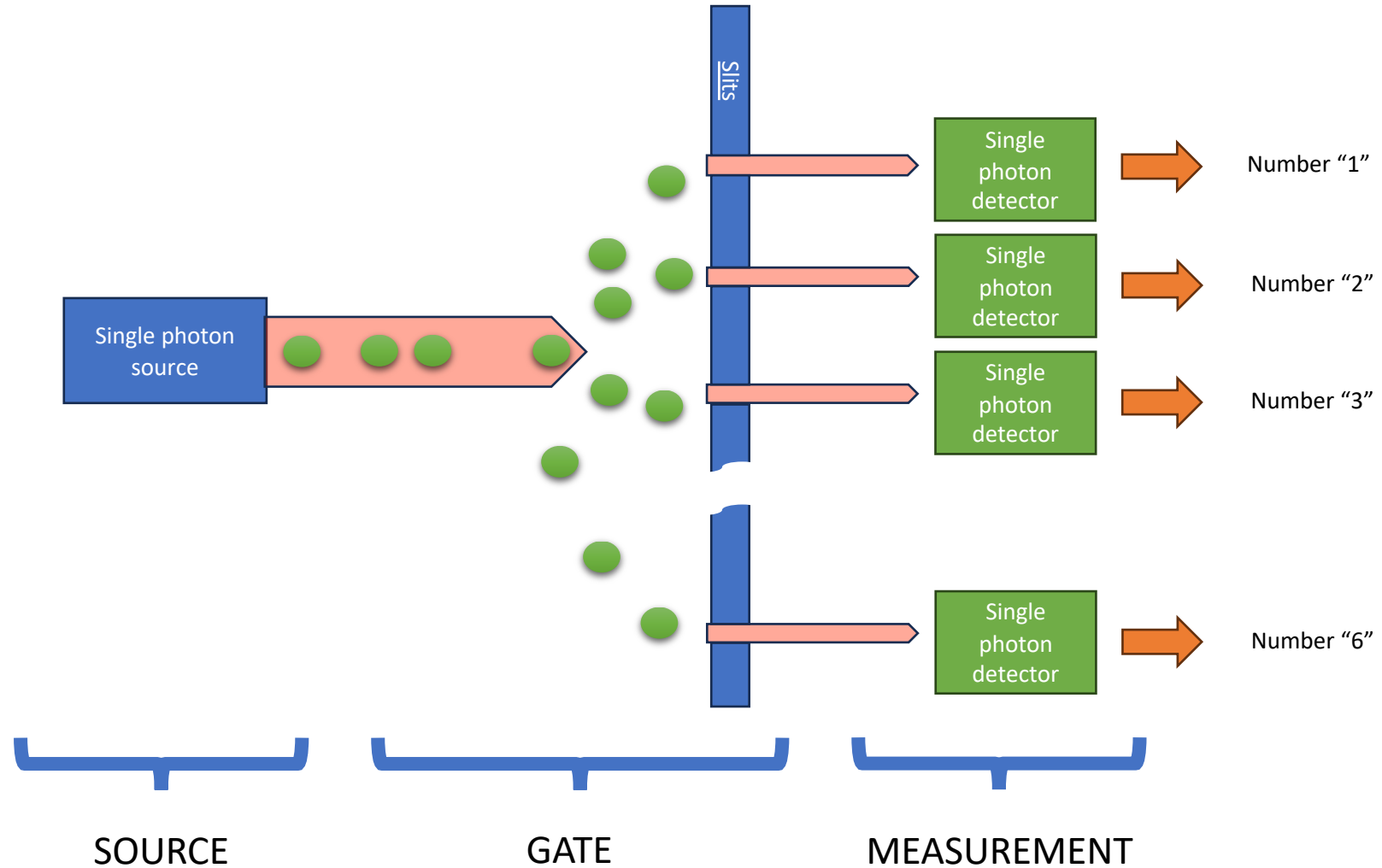
# A Simple quantum computer



Task: generator of random numbers  
"Quantum dice"



# Basic blocks of quantum computer

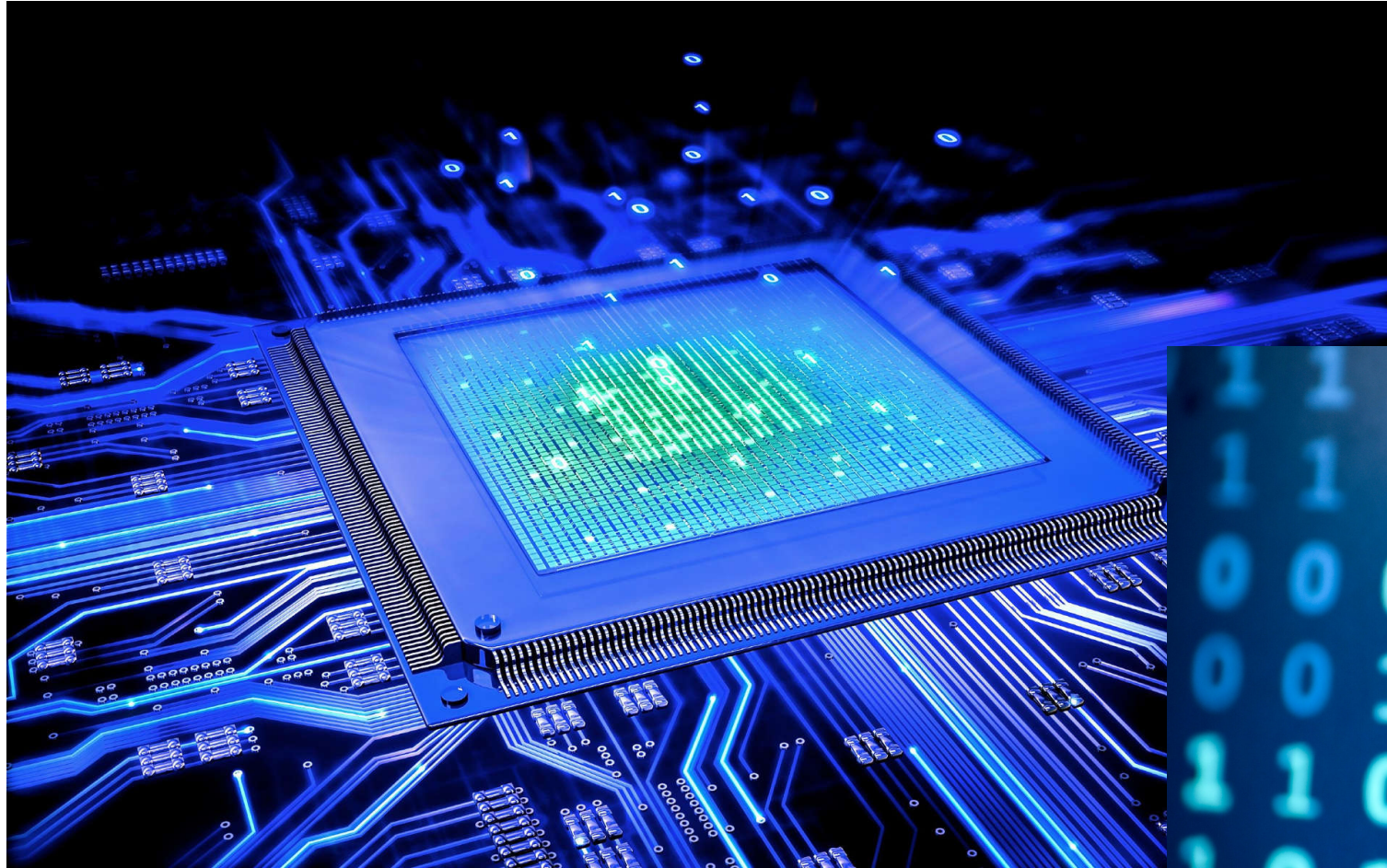


# Classical computers

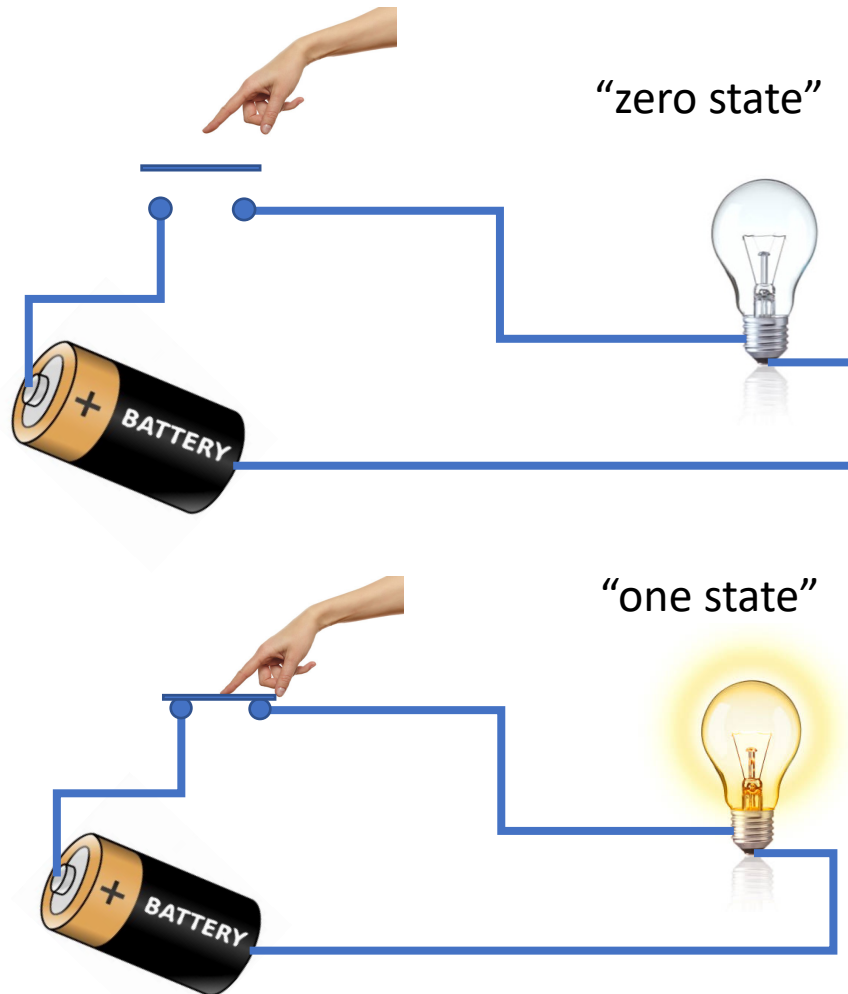


Table with 7 columns and multiple rows of data. The text is rotated 90 degrees counter-clockwise. The data appears to be a list of identifiers or codes.

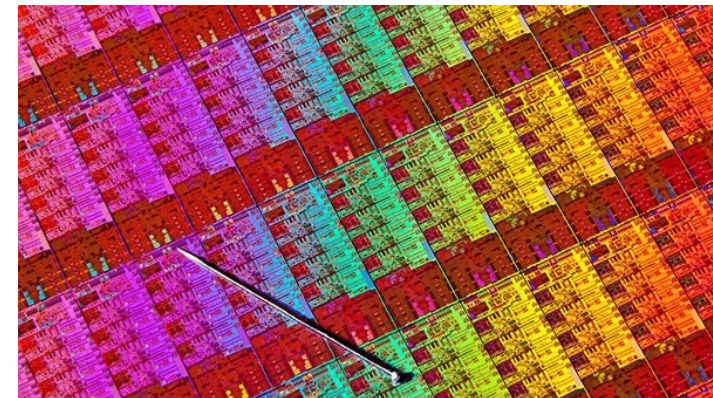
# Classical computation is about 0 and 1



# Classical computer – a box of switches

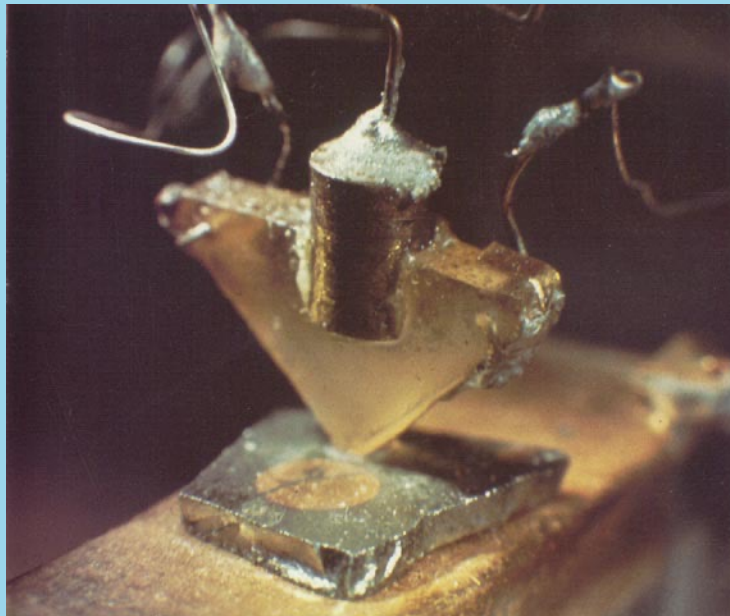


- Switch in computer is realized by a transistor
- A modern CPU has billions of transistors: e.g. Apple M2 Max - 67 billion transistors

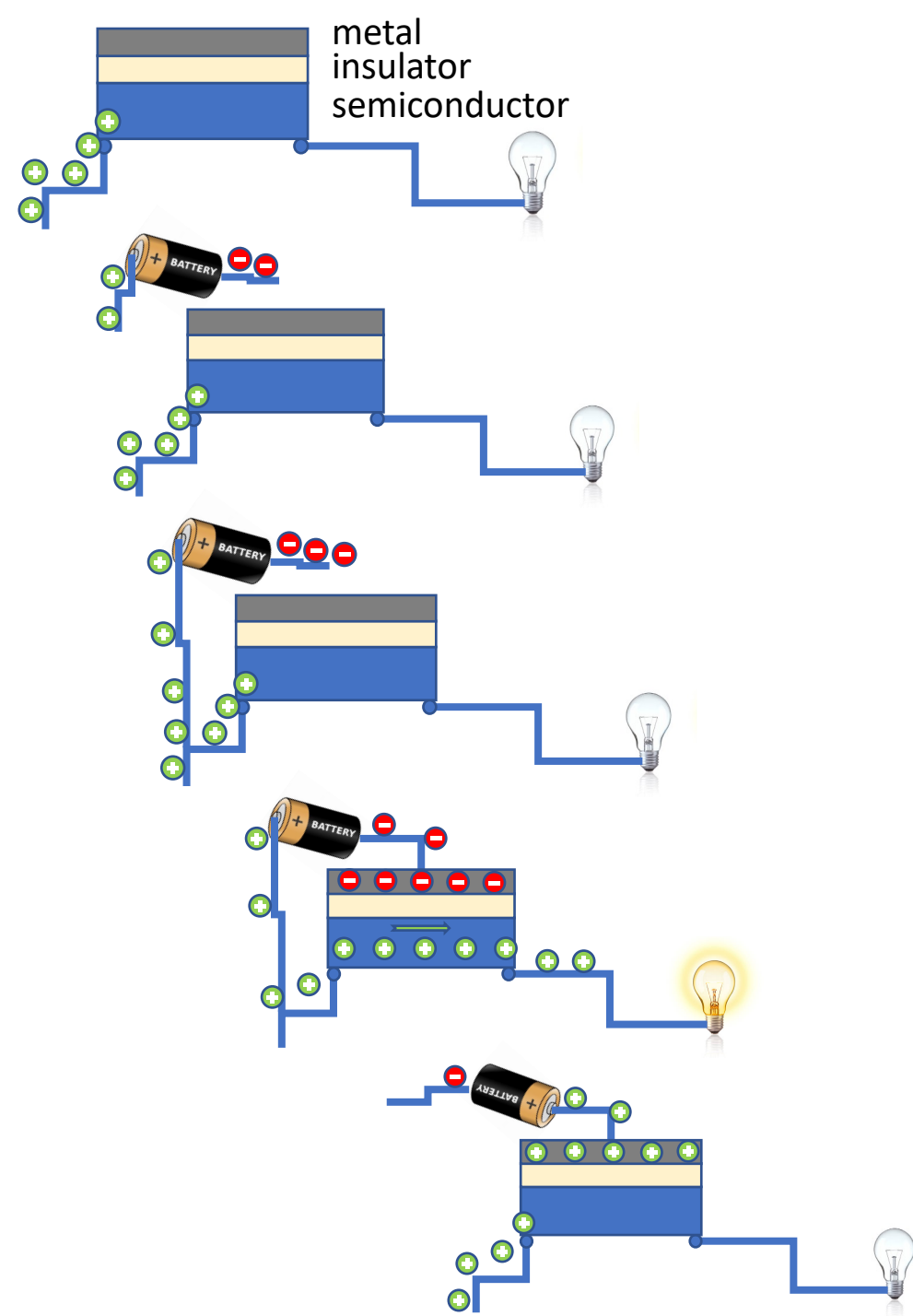


# Transistor – electronic switch

- Semiconducting material enabled minituarzation of electric switches



*J. Bardeen, W. Brattain 1947*

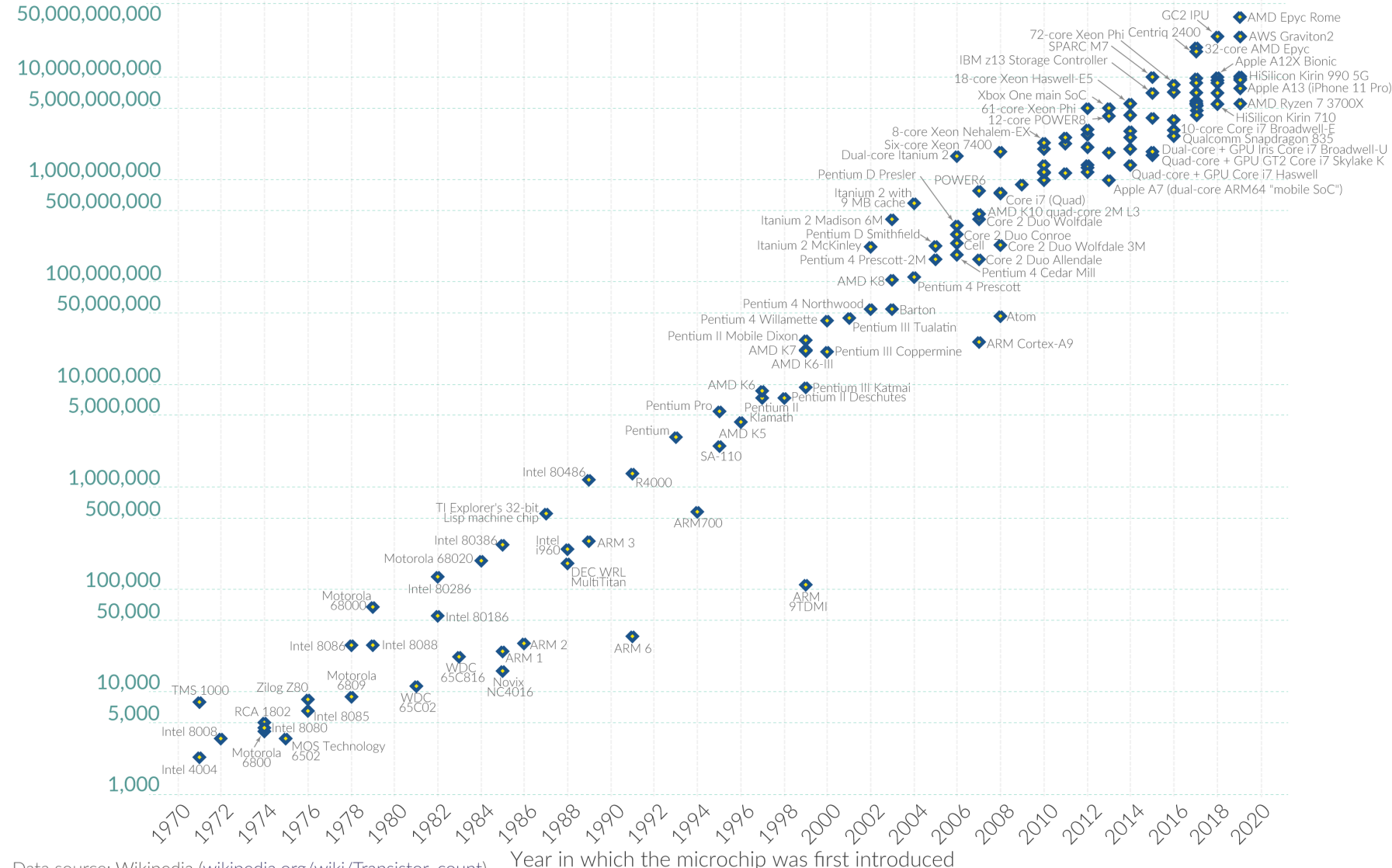


# Moore's Law: The number of transistors on microchips doubles every two years



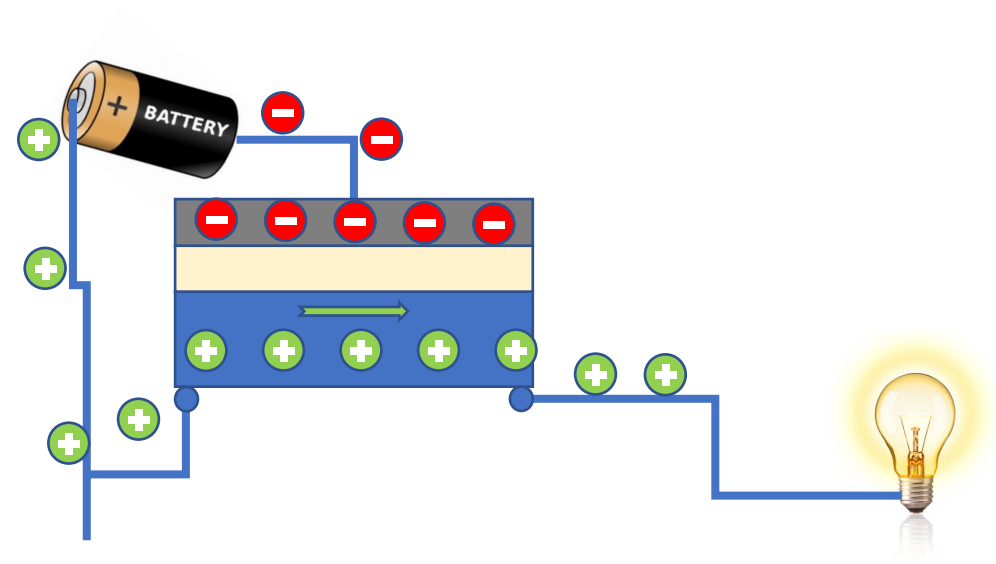
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

## Transistor count



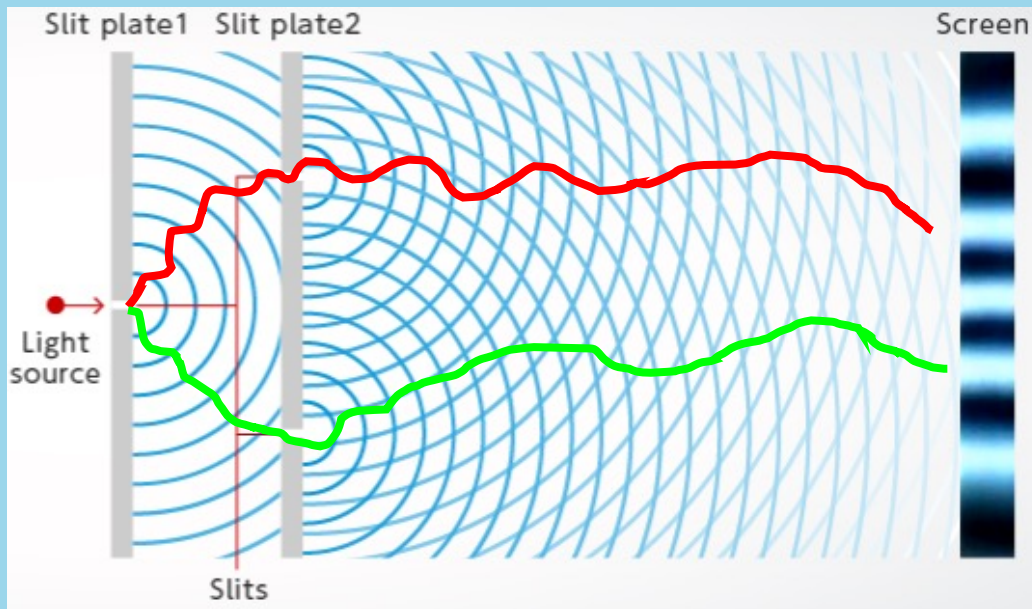


Minituarization reduced insulator thickness to 2 -5 nm in 2024.  
Still number of electrons is large  
as a result the quantum phenomena averages out



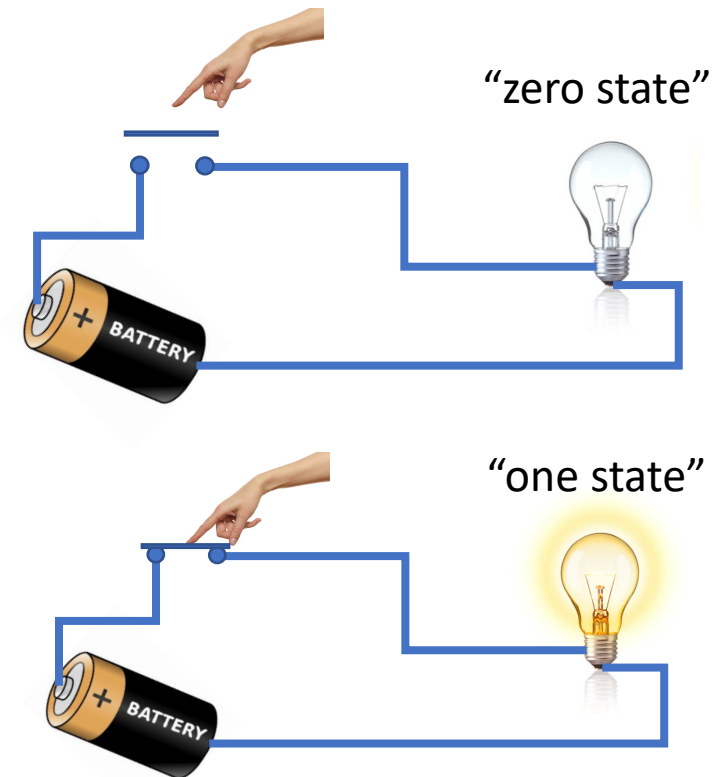
# Quantum state

- Single photon can hold the information
- Single photon hold more than just „0“ or „1“



# Classical state

- Semiconductor can be conducting or non-conducting
- semiconductor can hold the information of „0“ or „1“.



## Introduction of quantum bit - qubit

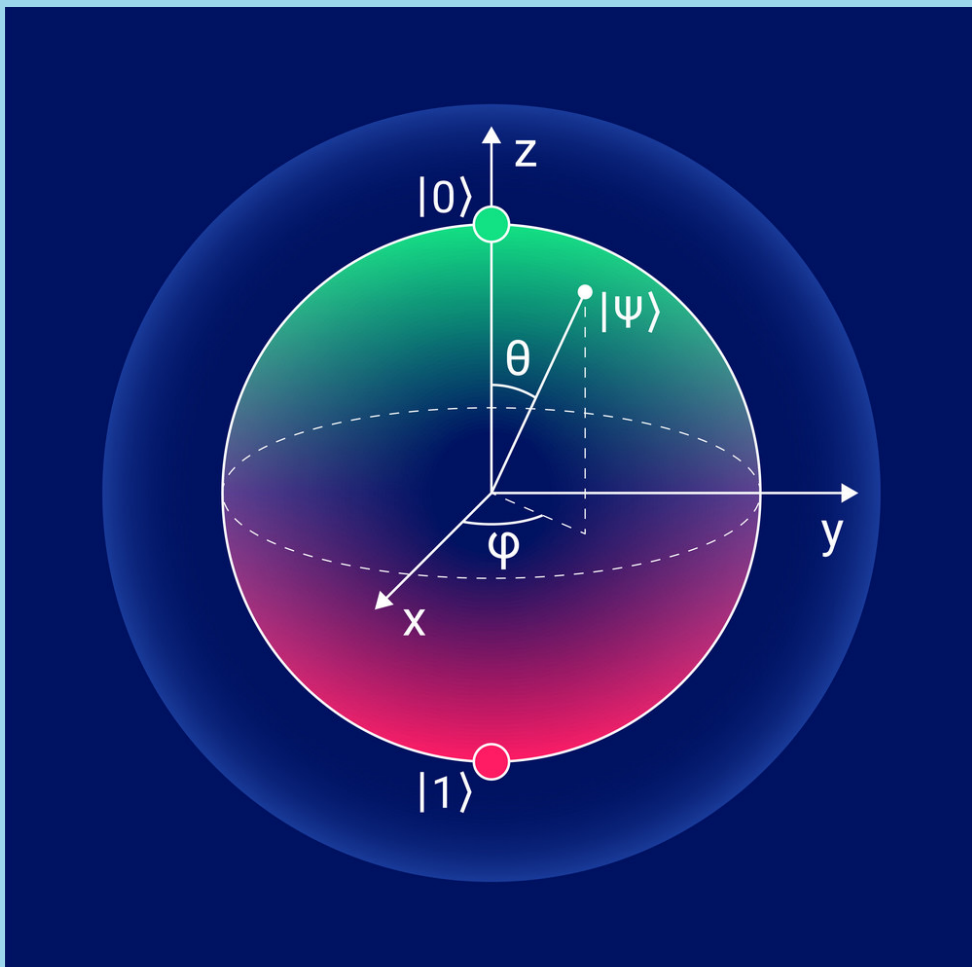
Classical bit is “0” or “1”

$$BIT = |0\rangle \text{ or } BIT = |1\rangle$$

Quantum bit – **superposition** of both states – “0” and “1”

$$QuBIT = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$$





Bloch sphere representation of qubit

$$|\Psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$$

$$\alpha = \cos \frac{\theta}{2} \quad \beta = e^{i\varphi} \sin \frac{\theta}{2}$$

## Probability

to be in “0” and “1” must be 1

$$QuBIT = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$$

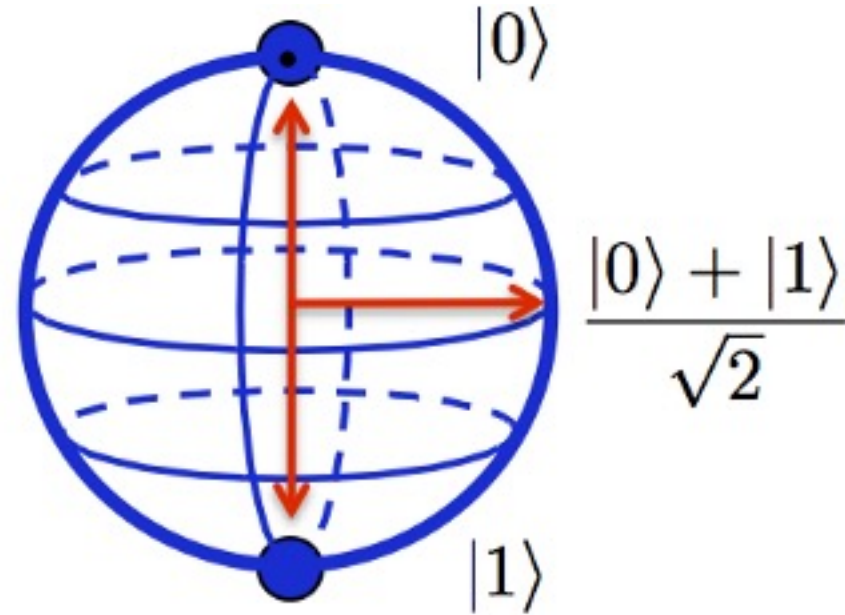
$$P(QuBIT) = \alpha^2 + \beta^2 = 1$$

# Classical vs Quantum bit

● 0

● 1

**Classical Bit**



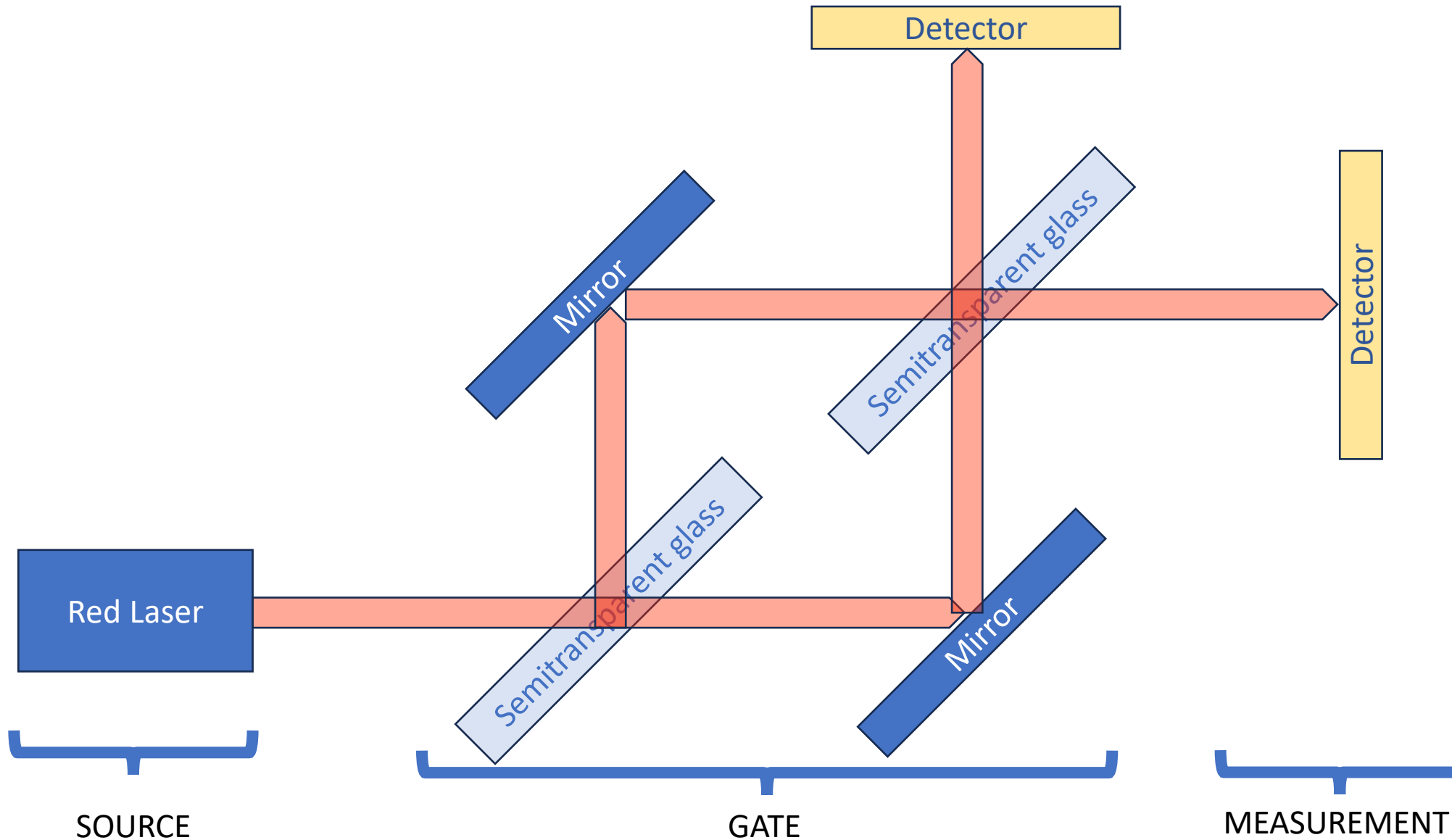
**Qubit**



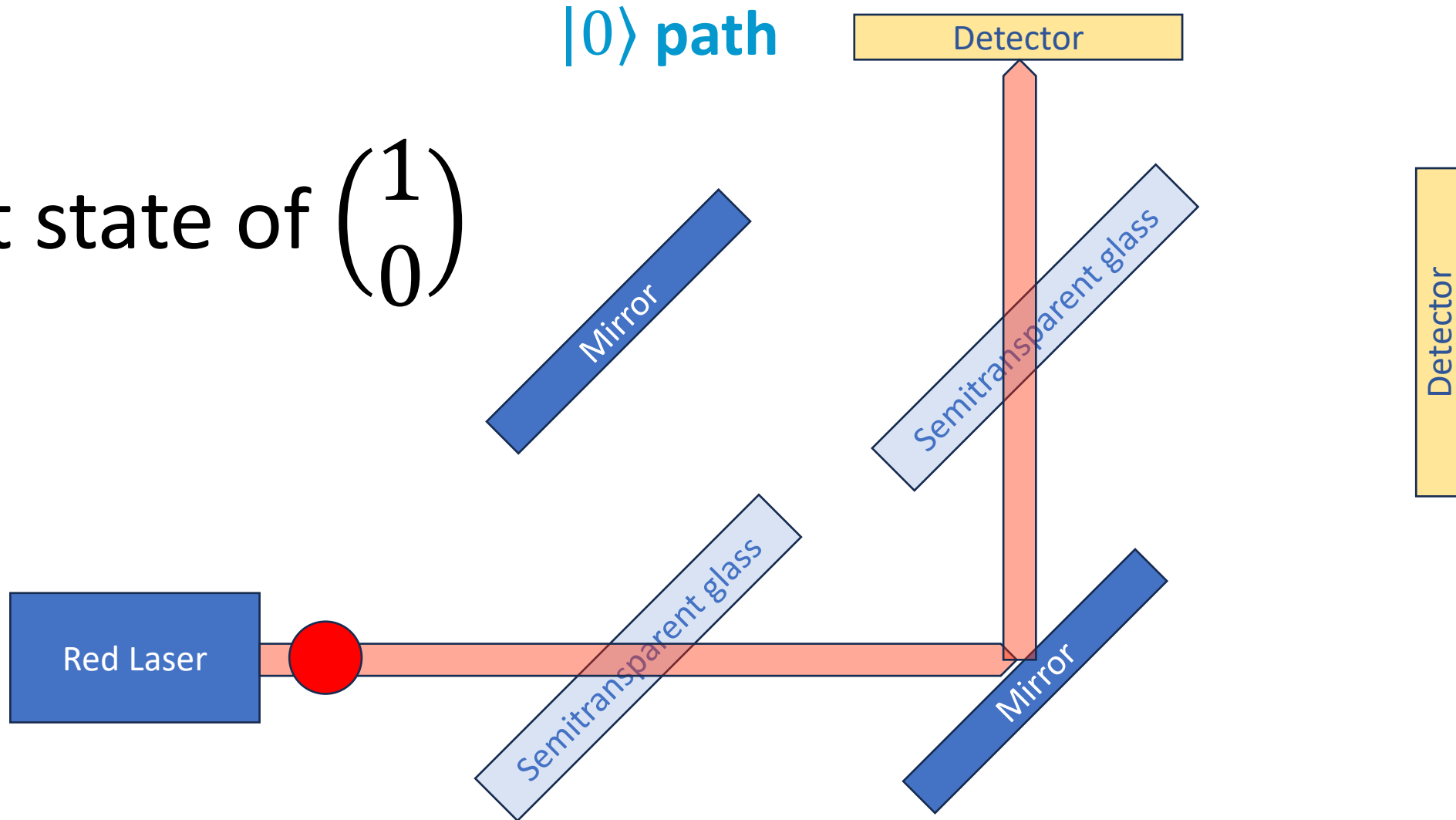
Qubit systems

Physical support	Name	Information support	$ 0\rangle$	$ 1\rangle$
Photon	Polarization encoding	Polarization of light	Horizontal	Vertical
	Number of photons	Fock state	Vacuum	Single photon state
	Time-bin encoding	Time of arrival	Early	Late
Coherent state of light	Squeezed light	Quadrature	Amplitude-squeezed state	Phase-squeezed state
Electrons	Electronic spin	Spin	Up	Down
	Electron number	Charge	No electron	One electron
Nucleus	Nuclear spin addressed through NMR	Spin	Up	Down
Optical lattices	Atomic spin	Spin	Up	Down
Josephson junction	Superconducting charge qubit	Charge	Uncharged superconducting island ( $Q=0$ )	Charged superconducting island ( $Q=2e$ , one extra Cooper pair)
	Superconducting flux qubit	Current	Clockwise current	Counterclockwise current
	Superconducting phase qubit	Energy	Ground state	First excited state
Singly charged quantum dot pair	Electron localization	Charge	Electron on left dot	Electron on right dot
Quantum dot	Dot spin	Spin	Down	Up
Gapped topological system	Non-abelian anyons	Braiding of Excitations	Depends on specific topological system	Depends on specific topological system
Vibrational qubit <sup>[15]</sup>	Vibrational states	Phonon/vibron	$ 01\rangle$ superposition	$ 10\rangle$ superposition
van der Waals heterostructure <sup>[16]</sup>	Electron localization	Charge	Electron on bottom sheet	Electron on top sheet

# Mach-Zehnder interferometer

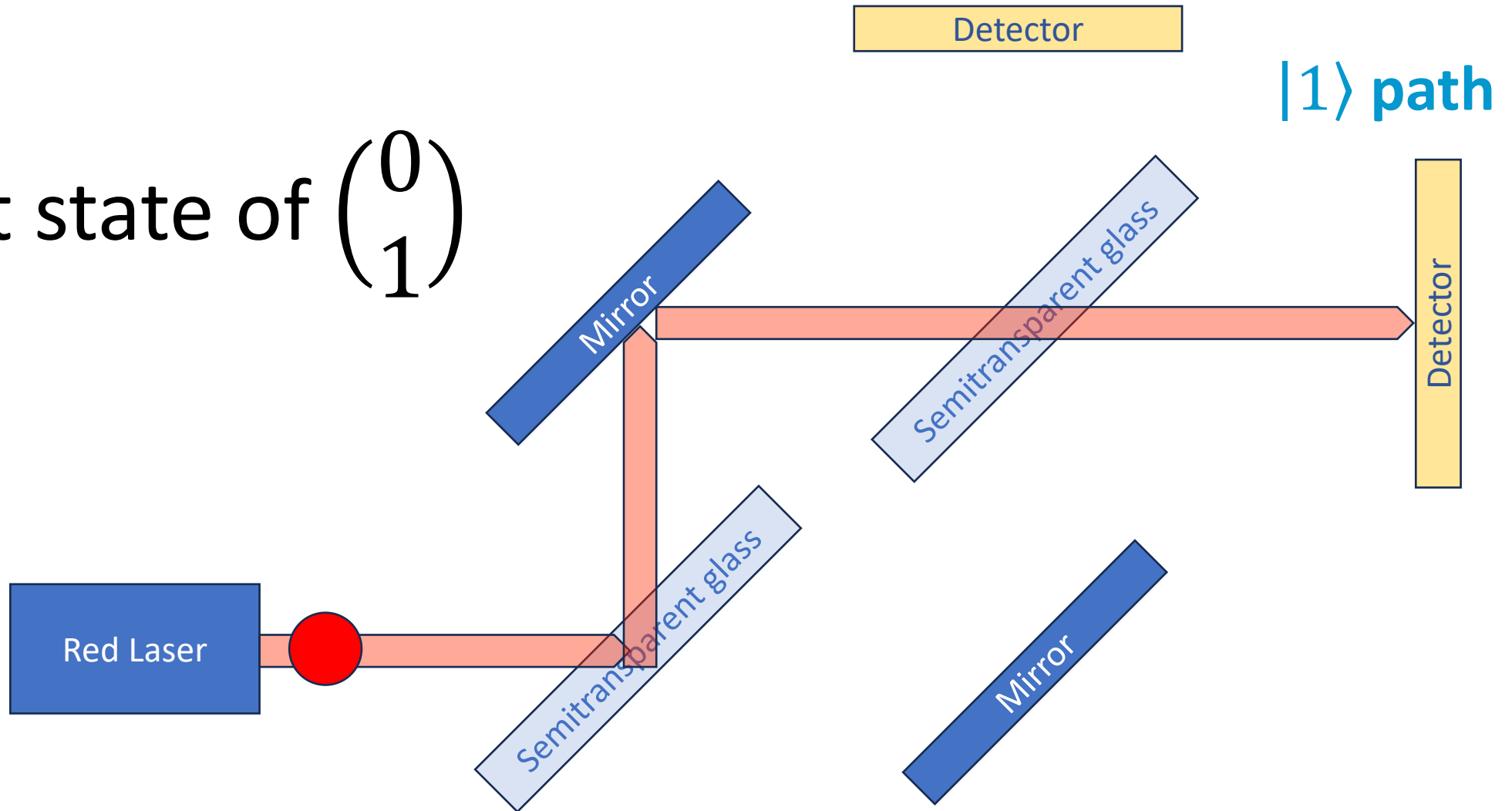


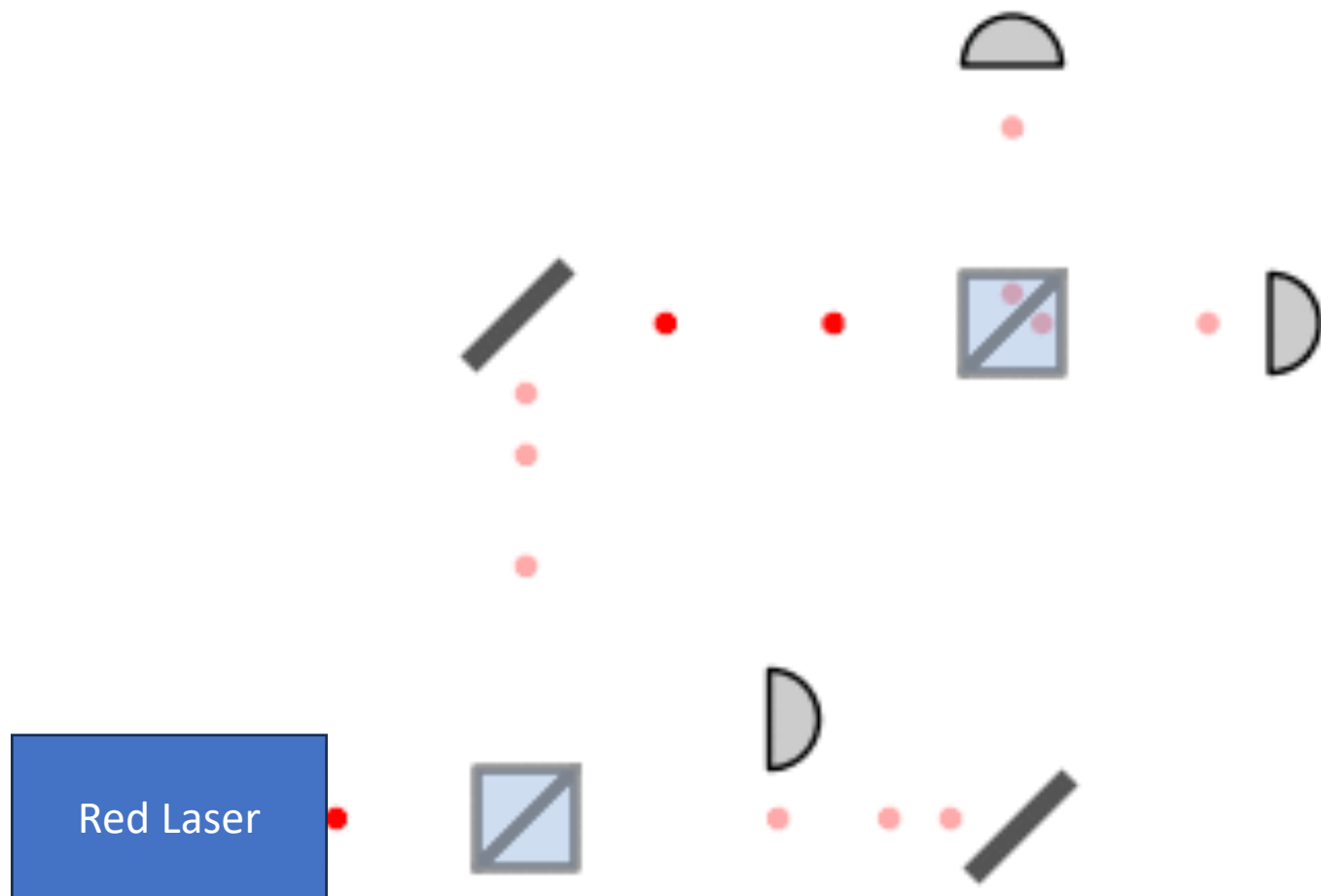
Qubit state of  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$





Qubit state of  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$





Experiment shows that photons travel only to path  $|1\rangle$

# Mathematical description

Qubit at the exit of the laser:  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

After BS:  $\alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

Beam-splitter operation:  $A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$

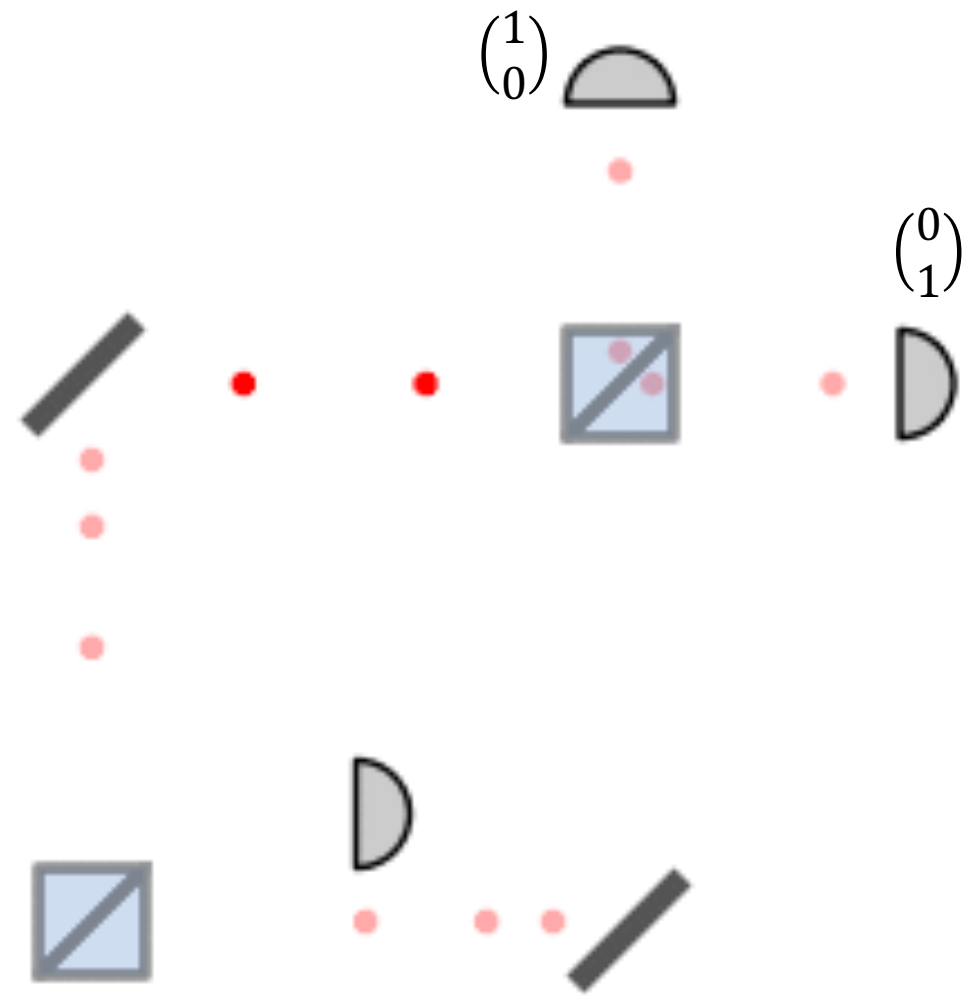
After first beam-splitter:

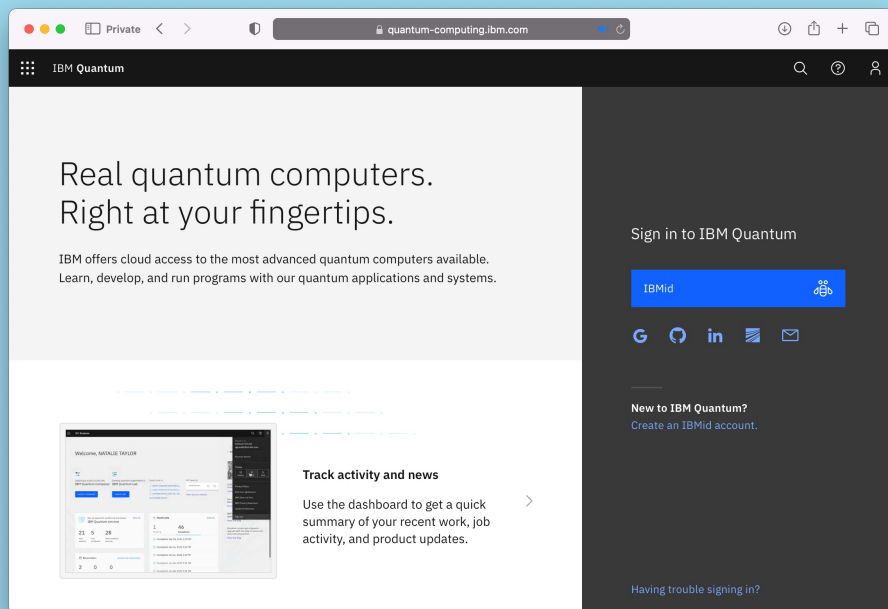
$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}$$

After second beam-splitter:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} = i \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

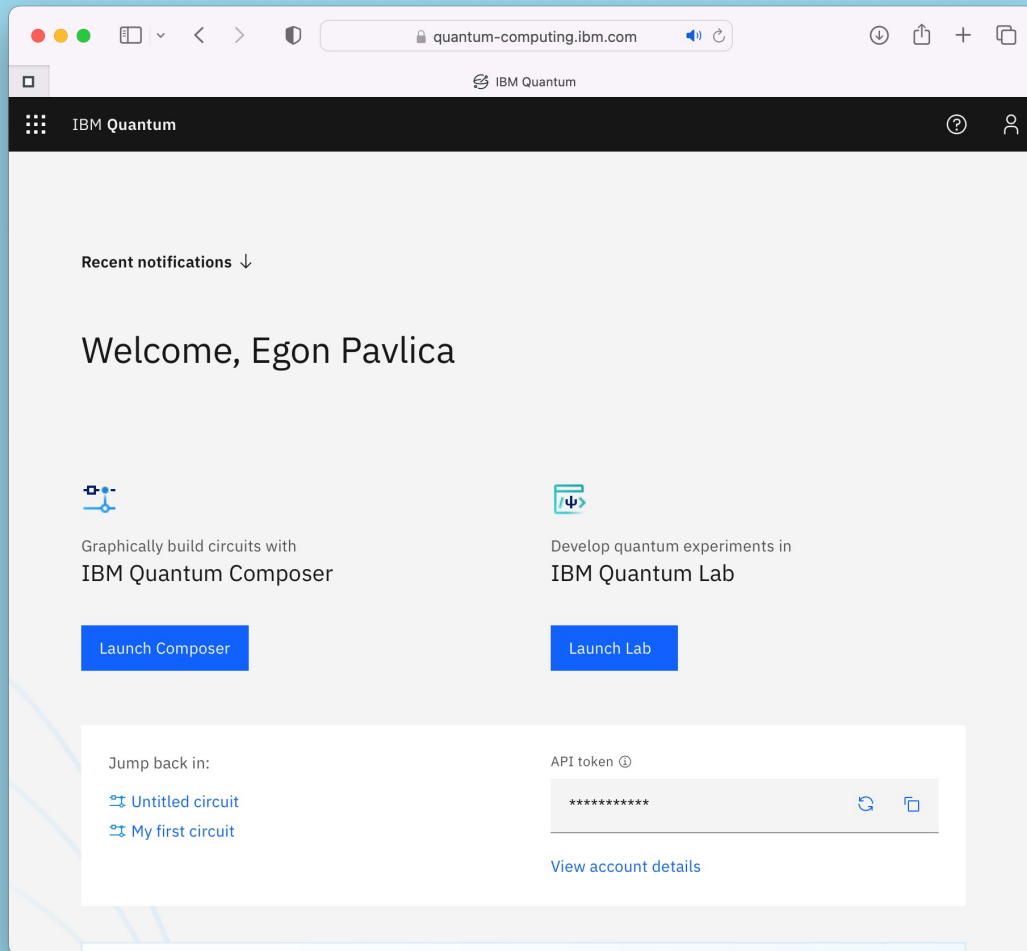
Red Laser





1. Create an account:

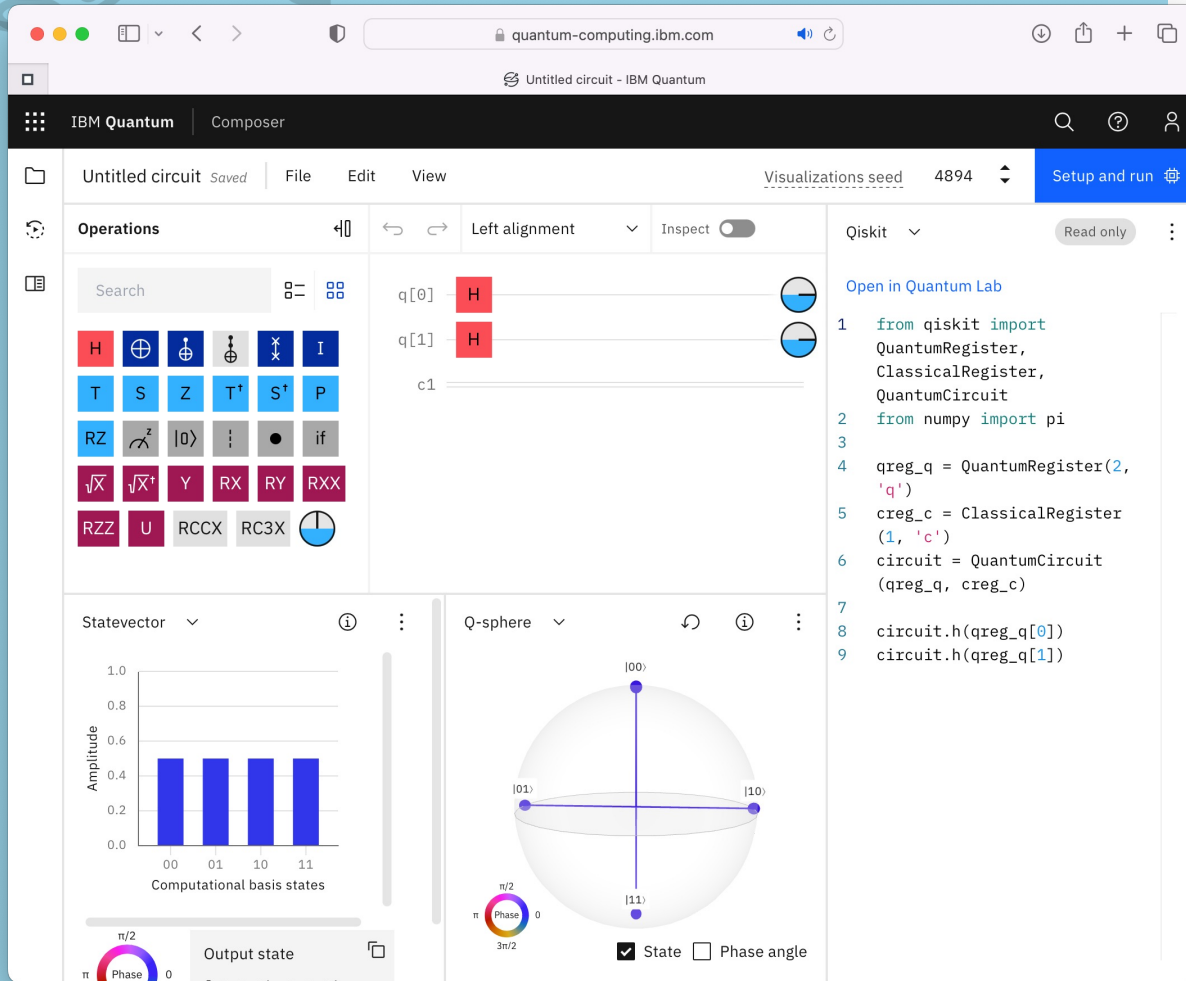
<https://quantum-computing.ibm.com>



1. Create an account:

<https://quantum-computing.ibm.com>

2. Launch IBM Quantum Composer



The screenshot shows the IBM Quantum Composer interface. The top navigation bar includes the IBM Quantum logo and the word "Composer". The main workspace is divided into several sections:

- Operations Panel:** A grid of quantum gates including Hadamard (H), CNOT, T, S, Z, T†, S†, P, RZ, |0⟩, and various rotation gates (RX, RY, RXX, RZZ, U, RCCX, RC3X).
- Circuit Diagram:** Shows two qubits, q[0] and q[1], each with a Hadamard gate. A classical register, c1, is also present.
- Statevector:** A bar chart showing the amplitudes for computational basis states 00, 01, 10, and 11, all having an amplitude of approximately 0.5.
- Q-sphere:** A Bloch sphere visualization with points for states |00⟩, |01⟩, |10⟩, and |11⟩. The state |00⟩ is highlighted.
- Code Editor:** Contains the following Qiskit code:

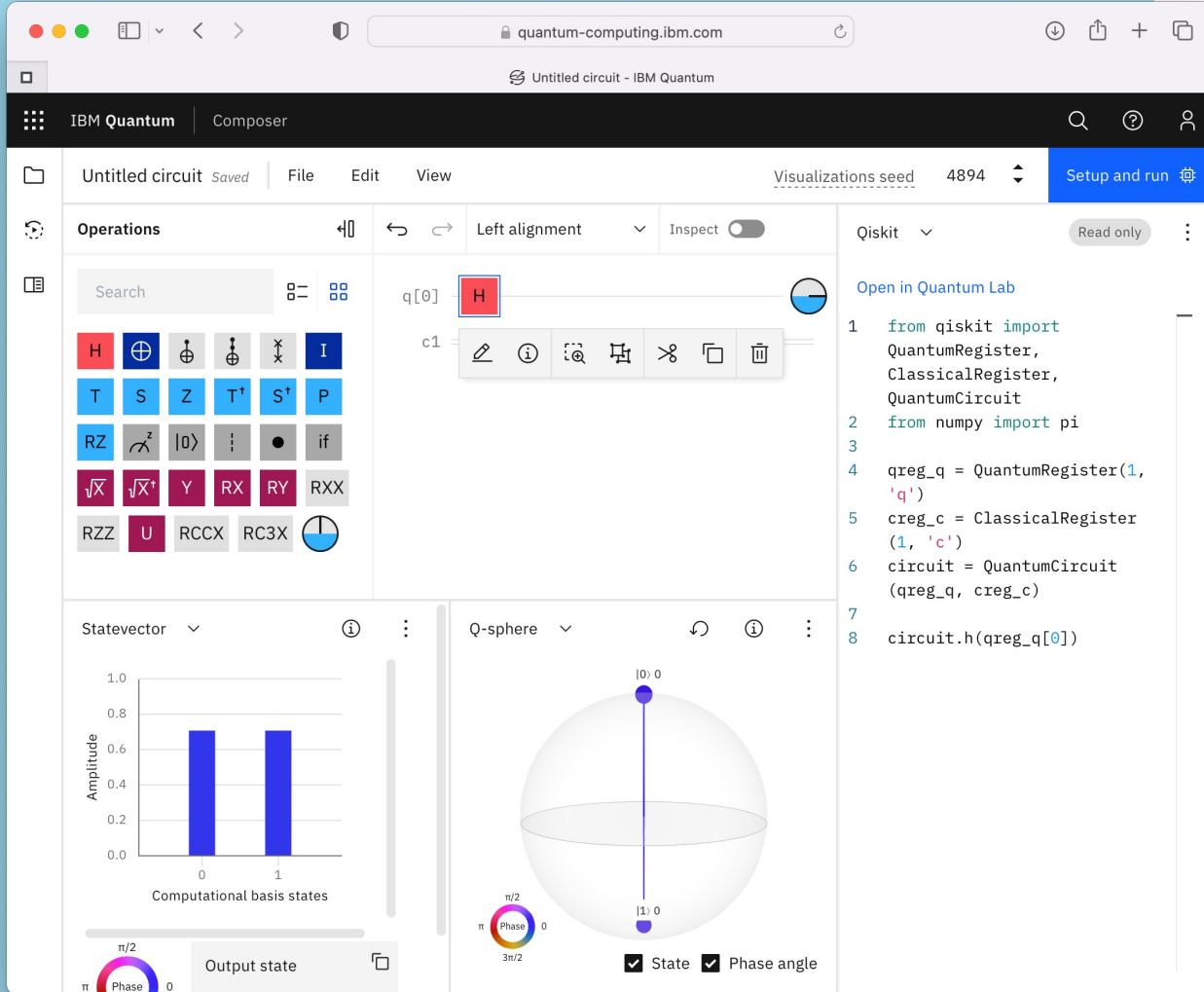
```
1 from qiskit import
2 QuantumRegister,
3 ClassicalRegister,
4 QuantumCircuit
5 from numpy import pi
6
7 qreg_q = QuantumRegister(2,
8 'q')
9 creg_c = ClassicalRegister
10 (1, 'c')
11 circuit = QuantumCircuit
12 (qreg_q, creg_c)
13
14 circuit.h(qreg_q[0])
15 circuit.h(qreg_q[1])
```

1. Create an account:

<https://quantum-computing.ibm.com>

2. Launch IBM Quantum Composer

3. Modify to have one, two or three qubits -> [study the changes](#)



1. Create an account:

<https://quantum-computing.ibm.com>

2. Launch IBM Quantum Composer

3. Modify to have one, two or three qubits -> **study the changes**

4. Leave only one qubit, and study **H and S operations**

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

Phase change:  $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$

## Setup Beam-splitter gate

**1st Beam-splitter:**  $A \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$

$$\begin{aligned} S \cdot H \cdot S \begin{bmatrix} 1 \\ 0 \end{bmatrix} &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \cdot \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} \end{aligned}$$

**1st+2nd Beam-splitter:**  $A \cdot A \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix}$

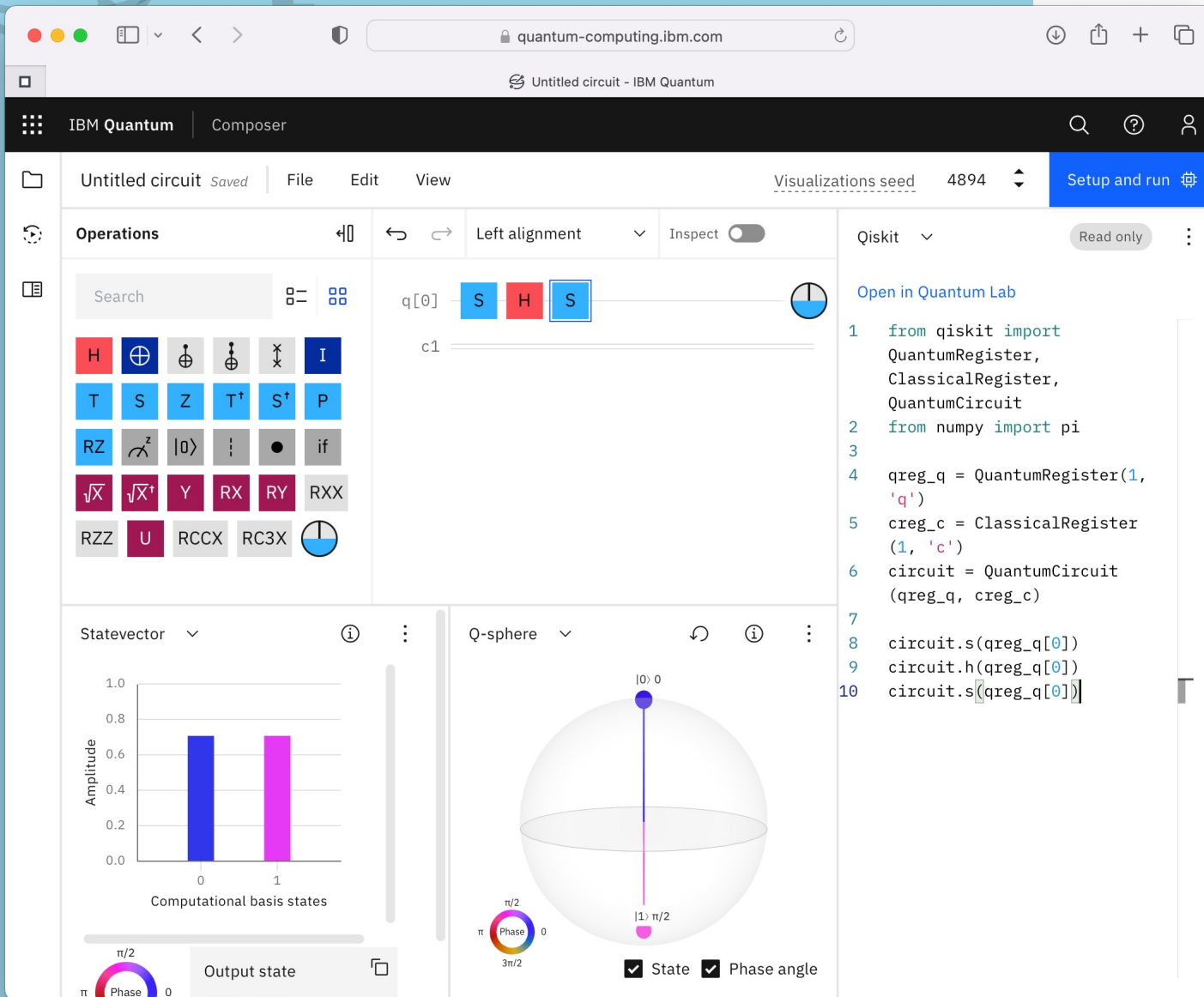
$$(S \cdot H \cdot S) \cdot (S \cdot H \cdot S) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix}$$



# 1st Beam splitter

$$S \cdot H \cdot S \begin{bmatrix} 1 \\ 0 \end{bmatrix} =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$$



The screenshot shows the IBM Quantum Composer interface. The circuit consists of three gates on qubit q[0]: an S gate (blue), an H gate (red), and another S gate (blue). The code editor on the right contains the following Python code:

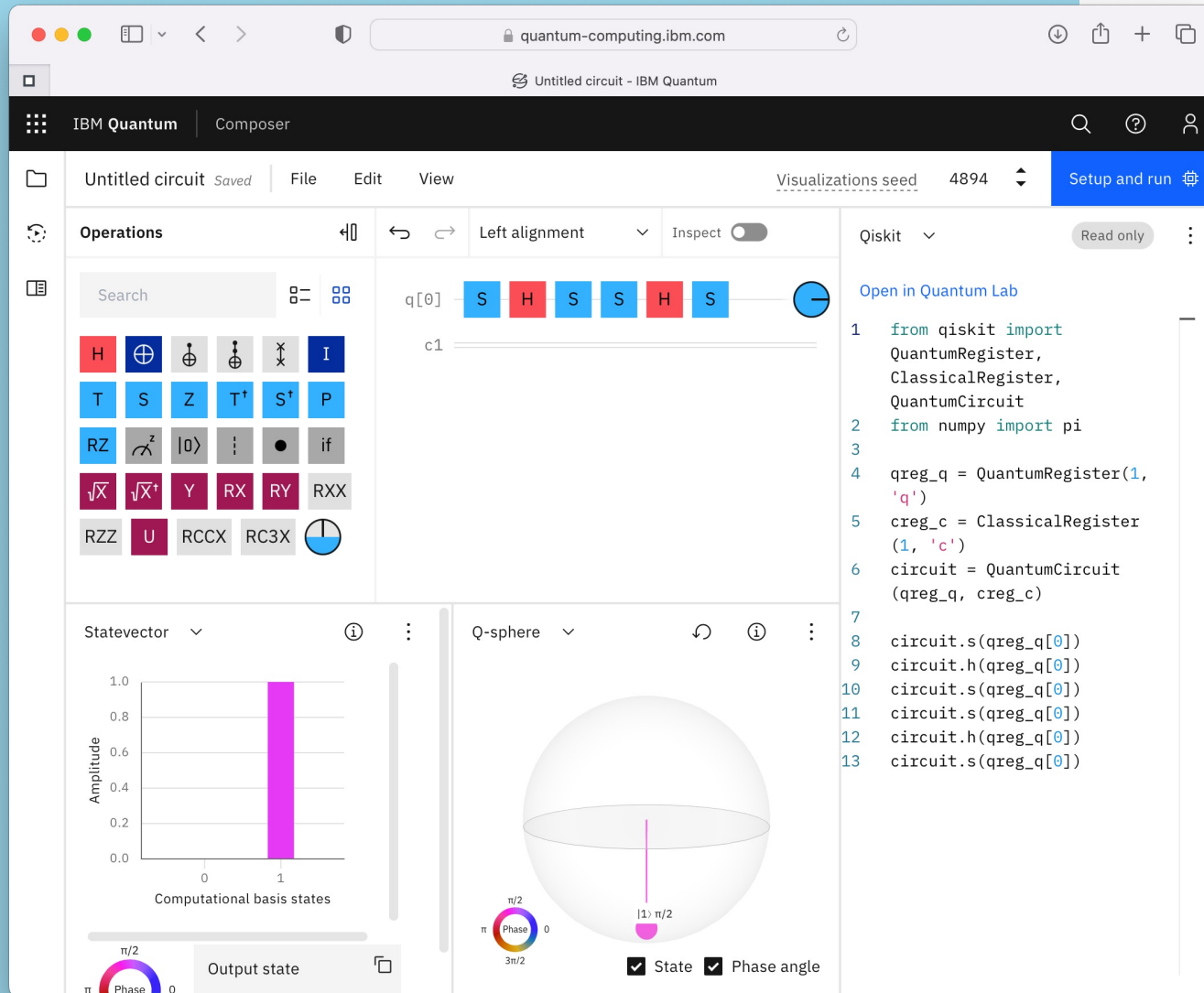
```

1 from qiskit import
  QuantumRegister,
  ClassicalRegister,
  QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(1,
  'q')
5 creg_c = ClassicalRegister
  (1, 'c')
6 circuit = QuantumCircuit
  (qreg_q, creg_c)
7
8 circuit.s(qreg_q[0])
9 circuit.h(qreg_q[0])
10 circuit.s(qreg_q[0])
  
```

At the bottom, the Statevector panel shows a bar chart with two bars of equal height (approximately 0.7) for computational basis states 0 and 1. The Q-sphere panel shows a Bloch sphere with a point at the top pole labeled  $|0\rangle, 0$  and a point at the bottom pole labeled  $|1\rangle, \pi/2$ . The phase angle is indicated as  $\pi/2$ .

## 2nd Beam splitter

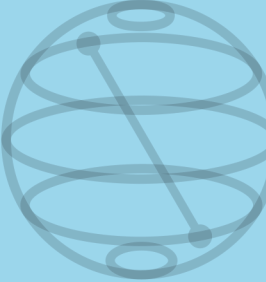
$$(S \cdot H \cdot S) \cdot (S \cdot H \cdot S) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix}$$



The screenshot shows the IBM Quantum Composer interface. The circuit consists of a single qubit q[0] with the following sequence of operations: S, H, S, S, H, S. The operations are visualized as colored blocks: S (blue), H (red), S (blue), S (blue), H (red), S (blue). The interface includes a toolbar with various quantum gates, a statevector plot showing the amplitude of computational basis states (0 and 1), and a Q-sphere visualization showing the state of the qubit. The statevector plot shows a single bar at state 1 with an amplitude of 1.0. The Q-sphere shows the state vector pointing to the south pole, corresponding to state |1>.

```

1 from qiskit import
  QuantumRegister,
  ClassicalRegister,
  QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(1,
  'q')
5 creg_c = ClassicalRegister
  (1, 'c')
6 circuit = QuantumCircuit
  (qreg_q, creg_c)
7
8 circuit.s(qreg_q[0])
9 circuit.h(qreg_q[0])
10 circuit.s(qreg_q[0])
11 circuit.s(qreg_q[0])
12 circuit.h(qreg_q[0])
13 circuit.s(qreg_q[0])
  
```



Compute resource

ibmq\_quito

### Status timeline

Queued ^

- Created: Jul 06, 2023 9:20 PM
- In queue
- Running  
quantum computation time was 0ms
- Completed

### Details

Sent from

[double beam-splitter](#)

Created on

Jul 06, 2023 9:20 PM

Instance

ibmq-q/open/main

Program

circuit-runner

# of shots

4096

# of circuits

1

# Add a measure and run the quantum program!



### Set up and run your circuit

Step 1

Choose a system or simulator

Search by system or simulator

**ibmq\_quito** [See details](#)

System status ● Online  
Total pending jobs 39

5 Qubits 16 QV 2.5K CLOPS

**ibmq\_belem** [See details](#)

System status ● Online  
Total pending jobs 37

5 Qubits 16 QV 2.5K CLOPS

**ibmq\_lima** [See details](#)

System status ● Online  
Total pending jobs 43

5 Qubits 16 QV 2.5K CLOPS

Step 2

Choose your settings

Instance

ibmq-q/open/main

Shots \*

4096

Job limit: 5 remaining

Tags (optional)

Add tags

Close

Run on ibmq\_quito

double beam-splitter Saved | File Edit View

Visualizations seed 4894 Setup and run

Operations

Search

Qiskit Read only

Open in Quantum Lab

```

1 from qiskit import
  QuantumRegister,
  ClassicalRegister,
  QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(1,
  'q')
5 creg_c = ClassicalRegister
  (1, 'c')
6 circuit = QuantumCircuit
  (qreg_q, creg_c)
7
8 circuit.s(qreg_q[0])
9 circuit.h(qreg_q[0])
10 circuit.s(qreg_q[0])
11 circuit.s(qreg_q[0])
12 circuit.h(qreg_q[0])
13 circuit.s(qreg_q[0])
14 circuit.measure(qreg_q[0],
  creg_c[0])

```

Statevector Q-sphere

Amplitude

tational basis State Phase angle

Measurement outcome

Frequency

Outcome 0

Frequency: 245

Check the number of  $|0\rangle$  and  $|1\rangle$

Note: *Real quantum computers have also errors!*

- ✓ Created: Jul 06, 2023 9:59 PM
- ✓ In queue: 29m 44s
- ✓ Running: Jul 06, 2023 10:29 PM  
quantum computation time was 1s
- ✓ Completed: Jul 06, 2023 10:29 PM

# REGINNA<sup>4.0</sup>

Want more? Take the **red** pill ;)

Egon Pavlica (<mailto:egon.pavlica@ung.si>)



Supported by



Funded by the  
European Union



[www.reginna4-0.eu](http://www.reginna4-0.eu)