

Wprowadzenie do komputera kwantowego – tutorial

Fizyka dla informatyków. Wykład QI - część I .

Literatura do wykładu QI:

<https://qiskit.org/textbook/preface.html>

dr Gustaw Szawioła, doc. dydak.

Politechnika Poznańska

Wydział Inżynierii Materiałowej i Fizyki Technicznej

Instytut Badań Materiałowych I Inżynierii Kwantowej

0. HARDWARE KWANTOWY I BIBLIOTEKA QISKIT W NOTATNIKU JUPYTER – – PODSTAWOWE INFORMACJE

Możliwości kwantowego eksperymentowania

IBM Q

The screenshot displays the IBM Q website in a web browser. The browser's address bar shows the URL <https://www.research.ibm.com/ibm-q/>. The page features the IBM logo and a navigation bar with links to "Marketplace", "IBM Q Network", "Learn", "Experiment", and "GitHub". A large hero image shows a person walking through a quantum computing laboratory filled with complex machinery and server racks. On the left side of the hero image, there is a large stylized "Q" logo and the text: "IBM Q is an industry-first initiative to build commercially available universal quantum computers for business and science." Below this text is a video player with a play button icon and the text "Watch video (04:46)". A blue button on the right side of the hero image says "Contact an expert". The browser's taskbar at the bottom shows various application icons and the system clock indicating 08:36 on 20.12.2019.

Nadprzewodnictwo – zjawisko Meissnera

Lewitacja magnesu nad nadprzewodnikiem



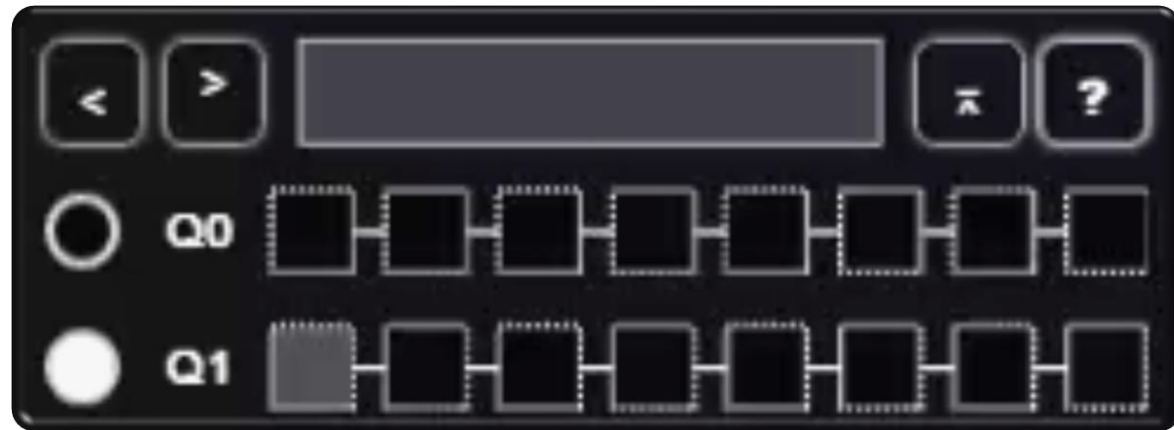
https://www.youtube.com/watch?v=L01q4lghrml&list=RDCMUC3j3w-oUtlAm_KI857ydvUA&index=1

Kwantowy kalkulator ...



<https://qiskit.org/modelq/>

....i jego użycie



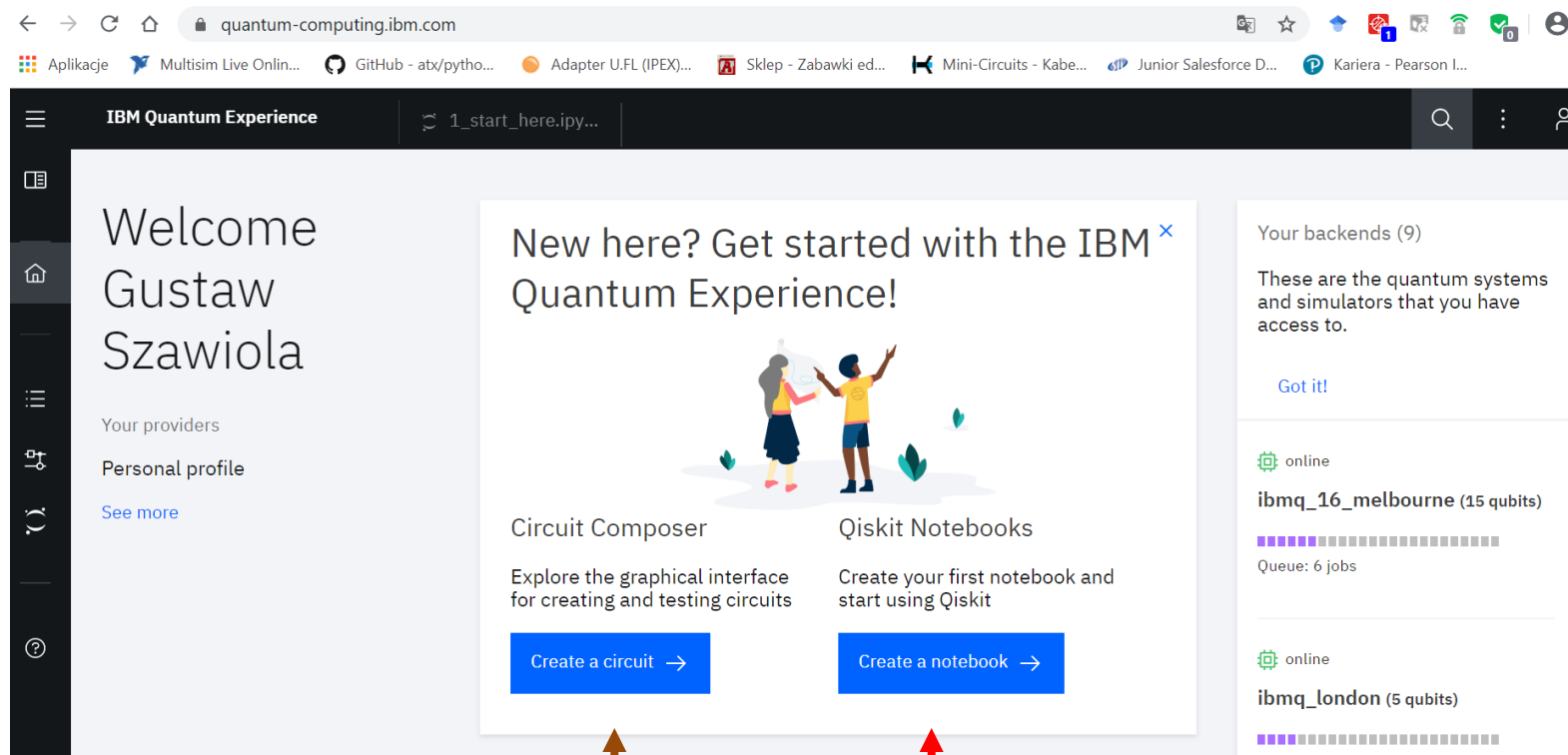
<https://qiskit.org/modelq/>

Logowanie na platformie IBM

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

The screenshot shows the IBM Quantum Experience website. The main header reads "Real quantum computers. Right at your fingertips." Below this, it states "IBM offers access to the most advanced quantum computers for you to do real work. Learn, develop, and run quantum programs on our systems with the IBM Quantum Experience cloud platform." A "Learn More" link is provided. On the left, there is a sidebar with navigation icons. Below the main text, there is a section titled "Write quantum programs" with a subtext "Easily program with Qiskit software integrated into the platform - no installation required." This section includes an image of a laptop displaying a Qiskit circuit and a code editor. On the right, there is a "Sign in to get started" section with a blue "IBMID" button and social media icons for Google, GitHub, LinkedIn, Twitter, and Email. Below this, it says "New to IBM Quantum Experience? Create an IBMID account."

Uruchamianie środowiska *Jupyter*



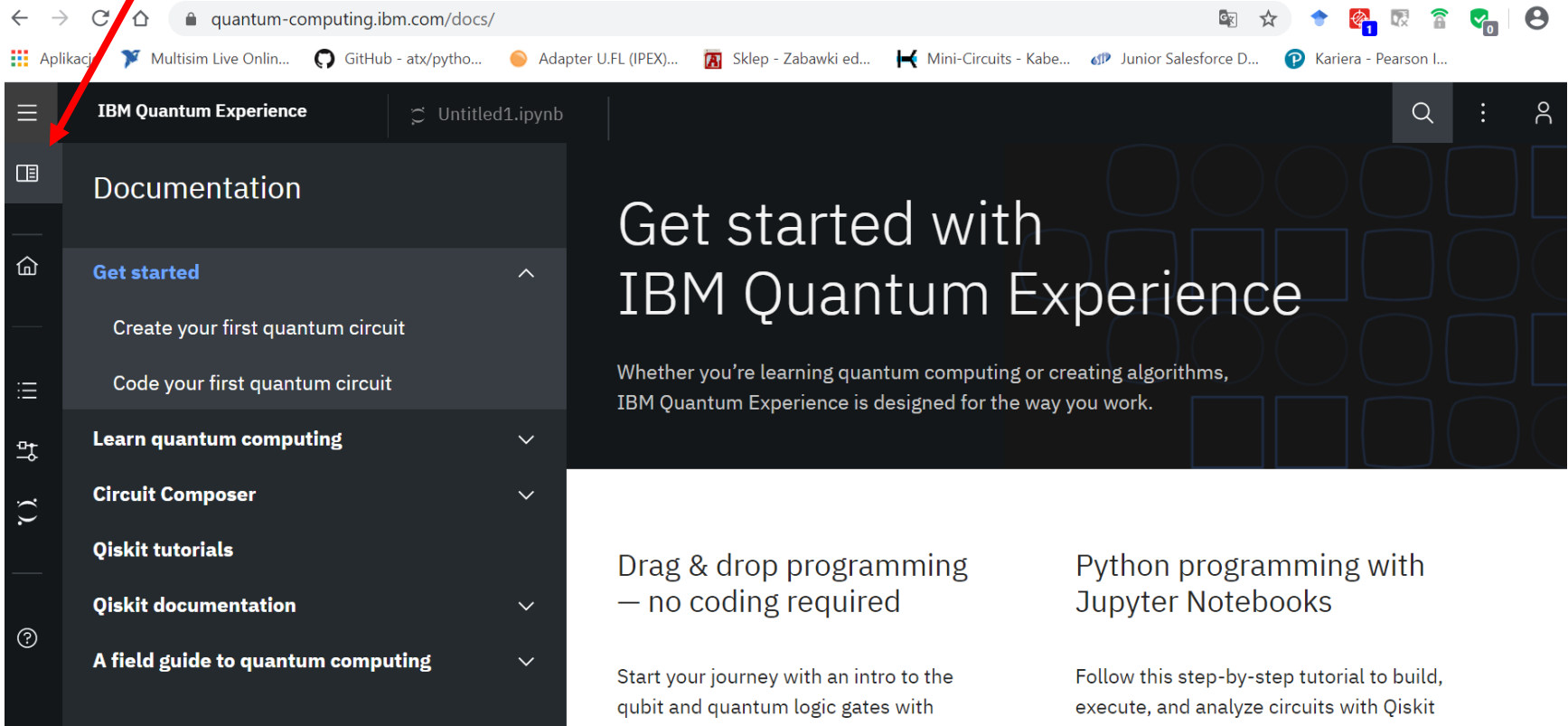
Przycisk pozwalający na przejście do trybu pracy z programowaniem w trybie graficznym

Przycisk pozwalający na przejście do trybu pracy z programowaniem w języku Python

Nawigacja do przewodników

Przycisk dokumentacji i tutoriali

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



The screenshot shows the IBM Quantum Experience website. A red arrow points to the 'Documentation' link in the left-hand navigation menu. The menu is open, showing options like 'Get started', 'Learn quantum computing', 'Circuit Composer', 'Qiskit tutorials', 'Qiskit documentation', and 'A field guide to quantum computing'. The main content area features the heading 'Get started with IBM Quantum Experience' and a description: 'Whether you're learning quantum computing or creating algorithms, IBM Quantum Experience is designed for the way you work.' Below this, there are two columns of text: 'Drag & drop programming — no coding required' and 'Python programming with Jupyter Notebooks'. The bottom of the page shows the text 'Start your journey with an intro to the qubit and quantum logic gates with' and 'Follow this step-by-step tutorial to build, execute, and analyze circuits with Qiskit'.

IBM Quantum Experience

Documentation

Get started

Create your first quantum circuit

Code your first quantum circuit

Learn quantum computing

Circuit Composer

Qiskit tutorials

Qiskit documentation

A field guide to quantum computing

Get started with IBM Quantum Experience

Whether you're learning quantum computing or creating algorithms, IBM Quantum Experience is designed for the way you work.

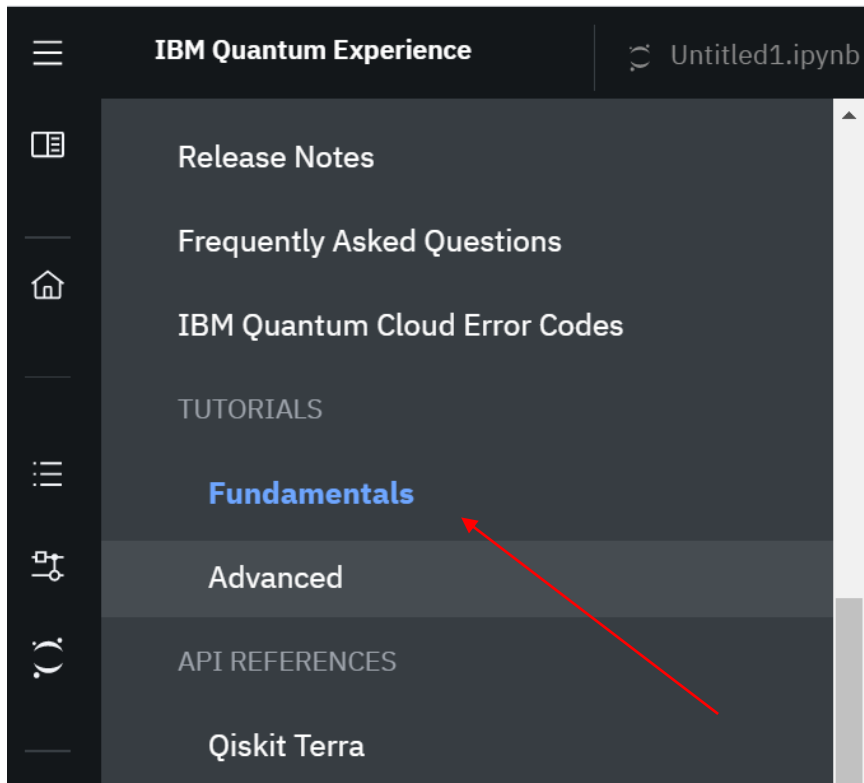
Drag & drop programming — no coding required

Python programming with Jupyter Notebooks

Start your journey with an intro to the qubit and quantum logic gates with

Follow this step-by-step tutorial to build, execute, and analyze circuits with Qiskit

Wprowadzający przewodnik



The screenshot shows the IBM Quantum Experience web interface. The top bar includes the 'IBM Quantum Experience' logo and a file name 'Untitled1.ipynb'. The left sidebar contains a navigation menu with the following items: 'Release Notes', 'Frequently Asked Questions', 'IBM Quantum Cloud Error Codes', 'TUTORIALS', 'Fundamentals' (highlighted in blue), 'Advanced', 'API REFERENCES', and 'Qiskit Terra'. A red arrow points from the bottom right towards the 'Fundamentals' link in the sidebar. The main content area on the right displays the title 'Fundamental Tutorials' and a list of four tutorial topics.

IBM Quantum Experience | Untitled1.ipynb

Release Notes

Frequently Asked Questions

IBM Quantum Cloud Error Codes

TUTORIALS

Fundamentals

Advanced

API REFERENCES

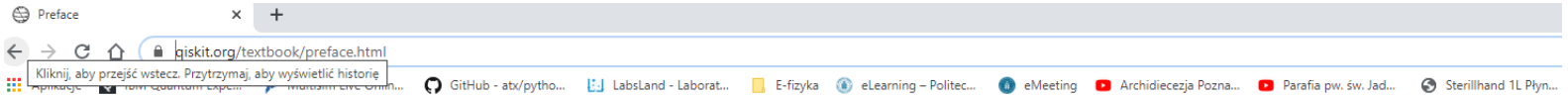
Qiskit Terra

Fundamental Tutorials

- [Getting Started with Qiskit](#)
- [Plotting Data in Qiskit](#)
- [Quantum Circuit Properties](#)
- [Summary of Quantum Operations](#)

Podręcznik

<https://qiskit.org/textbook/preface.html>



Qiskit Textbook

Preface

Using the Textbook

Interactivity Index

About

0. Prerequisites

1. Quantum States and Qubits

1.1 Introduction

1.2 The Atoms of Computation

1.3 Representing Qubit States

1.4 Single Qubit Gates

2. Multiple Qubits and Entanglement

2.1 Introduction

2.2 Multiple Qubits and Entangled States

2.3 Phase Kickback

2.4 Proving Universality

2.5 More Circuit Identities

3. Quantum Protocols and Quantum Algorithms

3.1 Defining Quantum Circuits

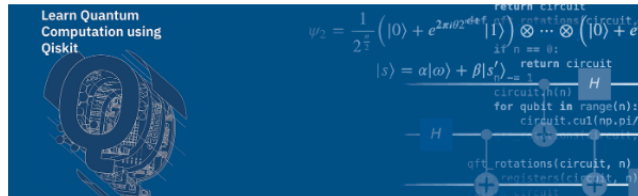
3.2 Quantum Teleportation

3.3 Superdense Coding



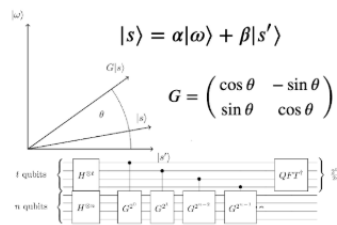
Preface

Learn Quantum Computation using Qiskit

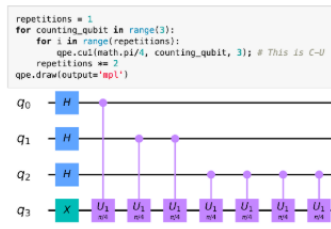


Greetings from the Qiskit Community team! We initiated this open-source textbook in collaboration with IBM Research as a university quantum algorithms/computation course supplement based on Qiskit. The Qiskit textbook helps you:

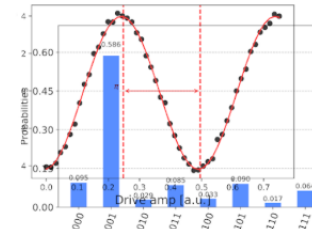
Learn Quantum Computing Theory



Learn to Code Quantum Programs



Experiment with Real Devices



Get Started!

FI/2019/2020

Graficzne środowisko programistyczne komputera kwantowego – QUANTUM COMPOSER

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

The screenshot displays the IBM Quantum Composer web interface in a browser window. The address bar shows the URL `quantum-computing.ibm.com/composer/new-experiment`. The interface includes a top navigation bar with "Save", "Clear", and "Help" buttons, and a "Run" button with a warning for "Unsaved changes".

The main workspace is divided into several panels:

- Left Panel:** Contains a "Statevector" dropdown menu, a "Statevector" label, and a bar chart. The chart has a y-axis from 0.0 to 1.0 and a single blue bar at the 1.0 position. Below the chart, there is a text box with the following content:
The height of the bar is the complex modulus.
The color of the bar is based in the complex argument or phase.
 $[1+0j]$
The qubit 0 is the one that is furthest to the right on the state.
- Center Panel:** Features a "Gates" section with a grid of quantum gate icons (H, ID, U3, U2, U1, Rx, Ry, Rz, X, Y, Z, S, S', T, T', cH, cY, cZ, cRz, cU1, cU3, and a plus sign). Below this is an "Operations" section with icons for $|0\rangle$, IF, and a plus sign, and a "Subroutines" section with a "+ Add" button.
- Right Panel:** Shows a "Barrier" icon and a list of qubits: `q[0]`, `q[1]`, `q[2]`, `q[3]`, and `q[4]`, each followed by $|0\rangle$. Below the qubit list is a label `c5`.

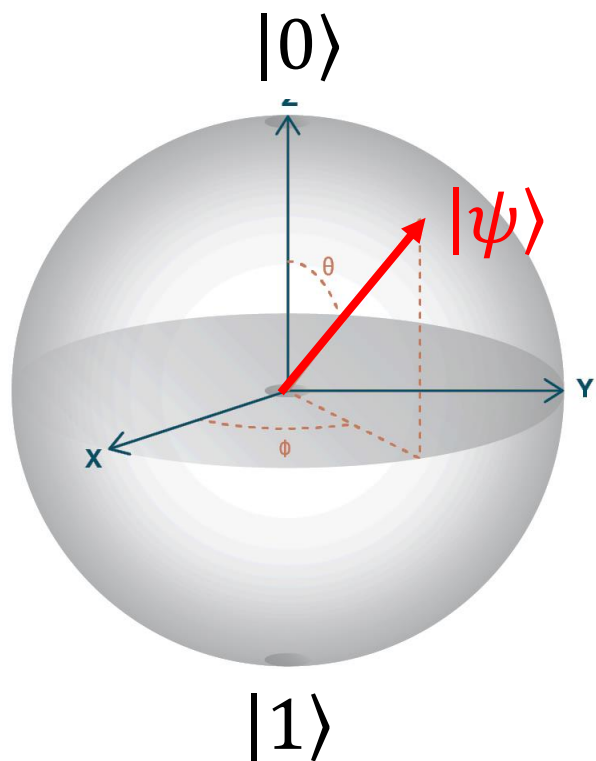
The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 08:04 on 27.09.2019.

Sugestia

- Utwórz osobiste konto na platformie IBM
- Znajdź przedstawione na slajdach przewodniki

1. QUBIT

Stan kubit – superpozycja stanów bazowych



$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle$$

Algebraiczna notacja

KET

Baza przestrzeni stanów

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$$

$$^\dagger : |\psi\rangle \rightarrow \underbrace{(|\psi\rangle)^\dagger}_{\langle\psi|}$$

BRA

Baza dualnej (sprzężonej)
przestrzeni stanów

$$\langle 0| = (1, 0) \quad \langle 1| = (0, 1)$$

$$\begin{aligned} \langle\psi| &= c_0^* \langle 0| + c_1^* \langle 1| \\ &= (c_0^*, c_1^*) \end{aligned}$$

$$a = |z| \cos \varphi \quad b = |z| \sin \varphi$$

Iloczyn skalarny (wewnętrzny)

$$|\psi\rangle = a_0 |0\rangle + b_1 |1\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$$

$$\langle\varphi| = b_0^* \langle 0| + b_1^* \langle 1| = (b_0^* \ , \ b_1^*)$$

$$z = a + ib$$

$$z^* = a - ib$$

$$z = |z| e^{i\varphi} = |z| (\cos \varphi + i \sin \varphi)$$

$$\langle\varphi|\psi\rangle = (b_0^* \ , \ b_1^*) \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = b_0^* a_0 + b_1^* a_1$$

Iloczyn skalarny

$$z^* z = |z|^2$$

$$\langle 0|0\rangle = 1 \quad \langle 1|1\rangle = 1$$

$$\langle 0|1\rangle = 0$$

$$\begin{aligned} |\langle 0|\psi\rangle|^2 &= |c_0|^2 = p \\ |\langle 1|\psi\rangle|^2 &= |c_1|^2 = 1 - p \end{aligned}$$

$$\begin{aligned} \langle 0|\psi\rangle &= c_0 \\ \langle 1|\psi\rangle &= c_1 \end{aligned}$$

Normalizacja wektora stanu

$$c_0 = e^{i\varphi_0}|c_0| \quad c_1 = e^{i\varphi_1}|c_1| \quad \text{--- } |\psi\rangle = c_1|0\rangle + c_2|1\rangle$$

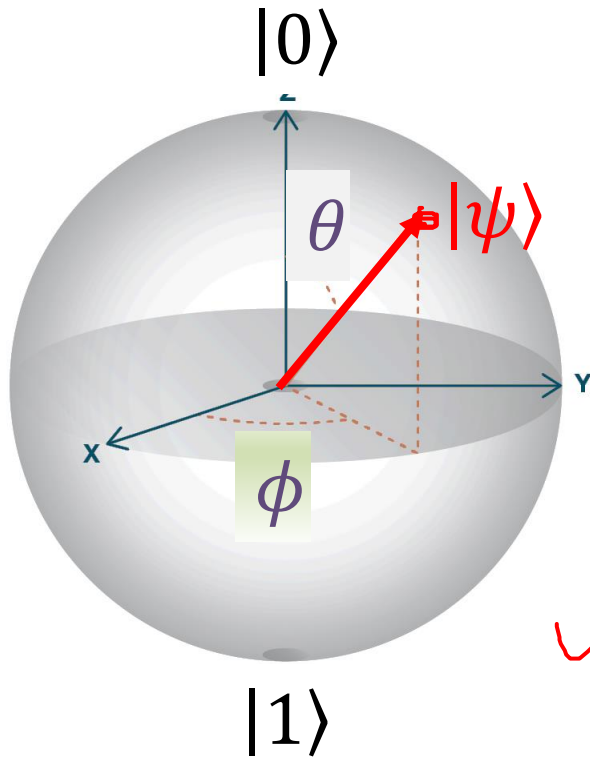
$$|\psi\rangle = e^{i\varphi_0} \left(|c_0| |0\rangle + |c_1| e^{i(\varphi_1 - \varphi_0)} |1\rangle \right)$$
$$|c_0|^2 + |c_1|^2 = 1$$

$$|\psi\rangle = e^{i\varphi_0} \left(\cos \alpha |0\rangle + e^{i(\varphi_0 - \varphi_1)} \sin \alpha |1\rangle \right)$$

$$\cos^2 \alpha + \sin^2 \alpha = 1$$

Stan kubitů a sfera Blocha

$$\frac{\langle \psi | \psi \rangle^2 = \langle \psi | \psi \rangle}{- \uparrow}$$



$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\frac{\langle \psi | \psi \rangle = \cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = 1}{\uparrow \quad \uparrow}$$

$$\underbrace{|\psi\rangle}_{\checkmark \nearrow} = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle$$

$$p = p_{|0\rangle} = \cos^2\left(\frac{\theta}{2}\right)$$

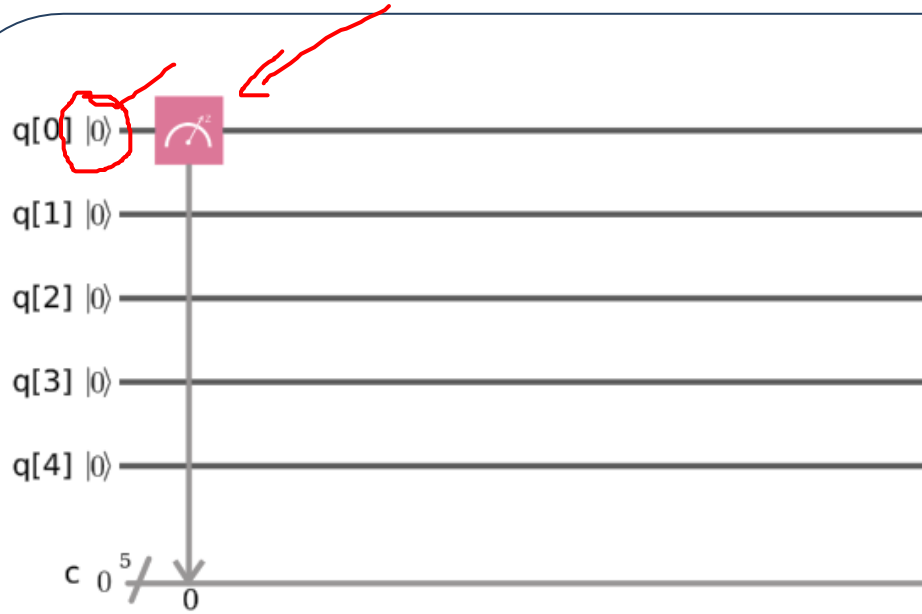
$$p_{|1\rangle} = \sin^2\left(\frac{\theta}{2}\right) = 1 - p$$

Pomiar rzutowy typu Z – odczyt stanu kubit

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

$$p_0 = |c_0|^2 = |\langle 0 | \psi \rangle|^2$$

$$p_1 = |c_1|^2 = |\langle 1 | \psi \rangle|^2$$



Kod graficzny

```
1 include "qelib1.inc";
2
3 qreg q[5];
4 creg c[5];
5
6 measure q[0] -> c[0];
```

Kod w asemblerze kwantowym (QASM)

$$p_0 = c_0^2 = \left(\frac{1}{\sqrt{2}}\right)^2$$

$$p_1 = c_1^2 = \left(\frac{1}{\sqrt{2}}\right)^2$$

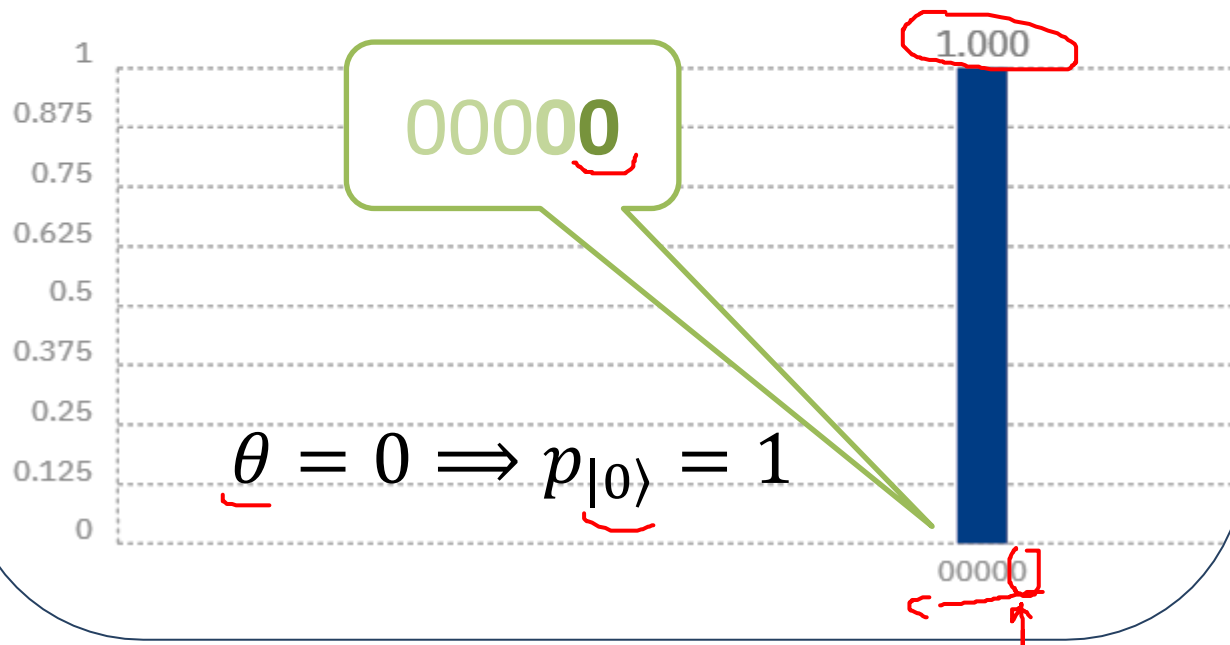
Klasyczna symulacja pomiaru

 e1

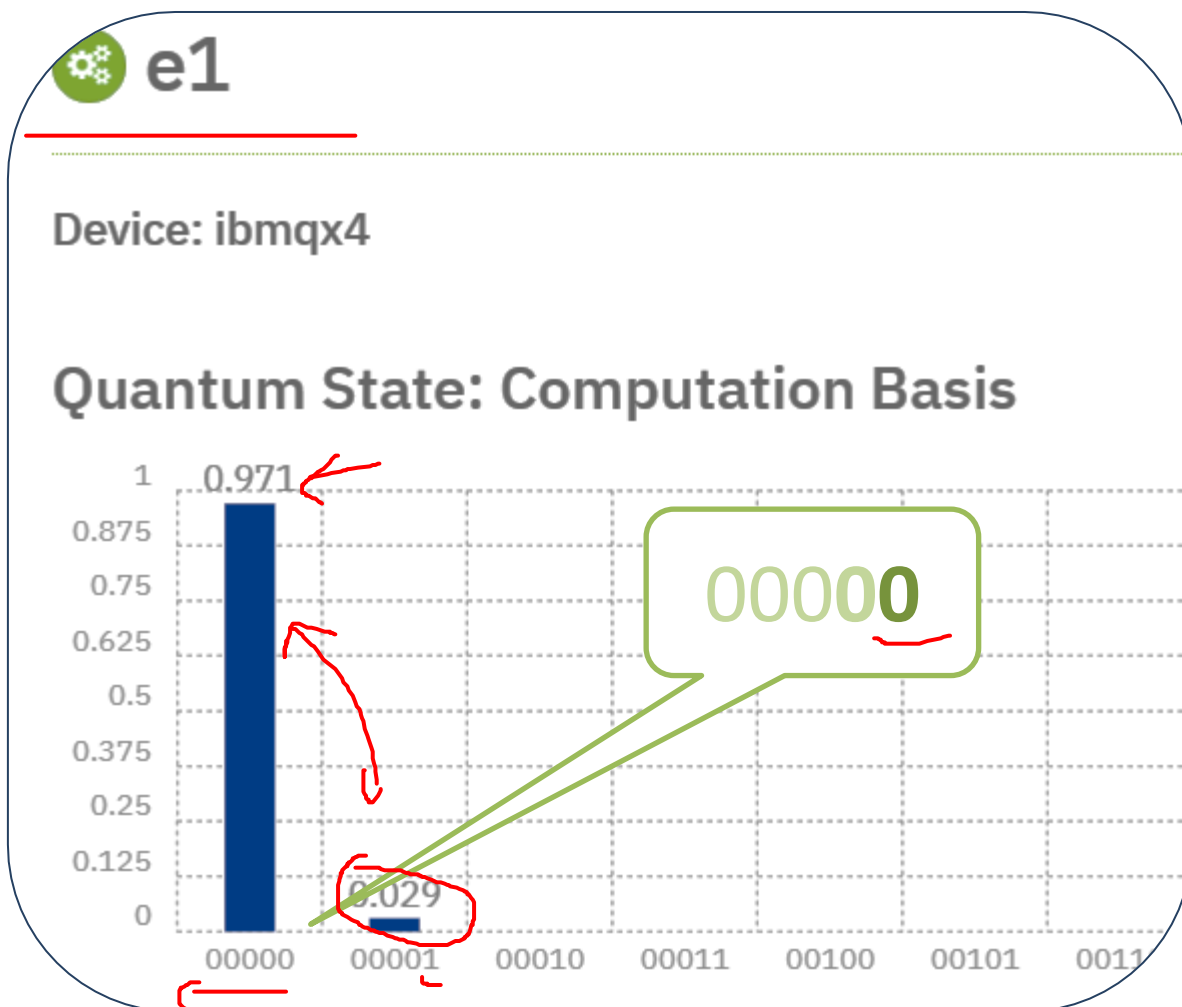
Device: Simulator

$$p_{|0\rangle} = \cos^2\left(\frac{\theta}{2}\right)$$

Quantum State: Computation Basis



Wynik ściśle kwantowych obliczeń na kwantowym komputerze



Python i biblioteka QISKIT

```
[1]: %matplotlib inline
# Importowanie standardowej biblioteki Qiskit i konfiguracja konta
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, execute, \
    Aer, IBMQ
from qiskit.compiler import transpile, assemble
from qiskit.tools.jupyter import *
from qiskit.visualization import *
# Określenie konta IBM Q
provider = IBMQ.load_account() ✓
```

Wczytywanie biblioteki QISKIT

→ qiskit.org ✓

git

Tworzenie algorytmu kwantowego- -obwodu kwantowego

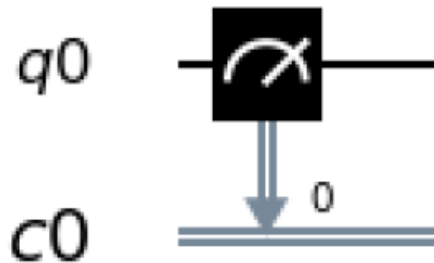
```
[2]: n0 = 1 # Liczba kubitów i bitów  
q0 = QuantumRegister(n0) # Rejestr kwantowy  
c0 = ClassicalRegister(n0) # Rejestr klasyczny  
  
circuit0 = QuantumCircuit(q0, c0) # Algorytm kwantowy - kwantowy obwód  
circuit0.measure(q0[0], c0[0]) # Sprawdzenie stanów kubitów - kwantowy pomiar
```

```
[2]: <qiskit.circuit.instructionset.InstructionSet at 0x7f6380873850>
```

```
[3]: circuit0.draw(output='mpl') # Rysowanie obwodu
```

```
[3]:
```

Pomiar stanu
1 kubit



Wykonanie algorytmu kwantowego - obliczenia

```
[4]: # Wykonaj obliczeń kwantowych  
backend = Aer.get_backend('qasm_simulator')  
job_sim0 = execute(circuit0, backend)  
sim_result0 = job_sim0.result()
```

```
# Liczbowa prezentacja wyników  
print(sim_result0.get_counts(circuit0))
```

```
{'0': 1024}
```

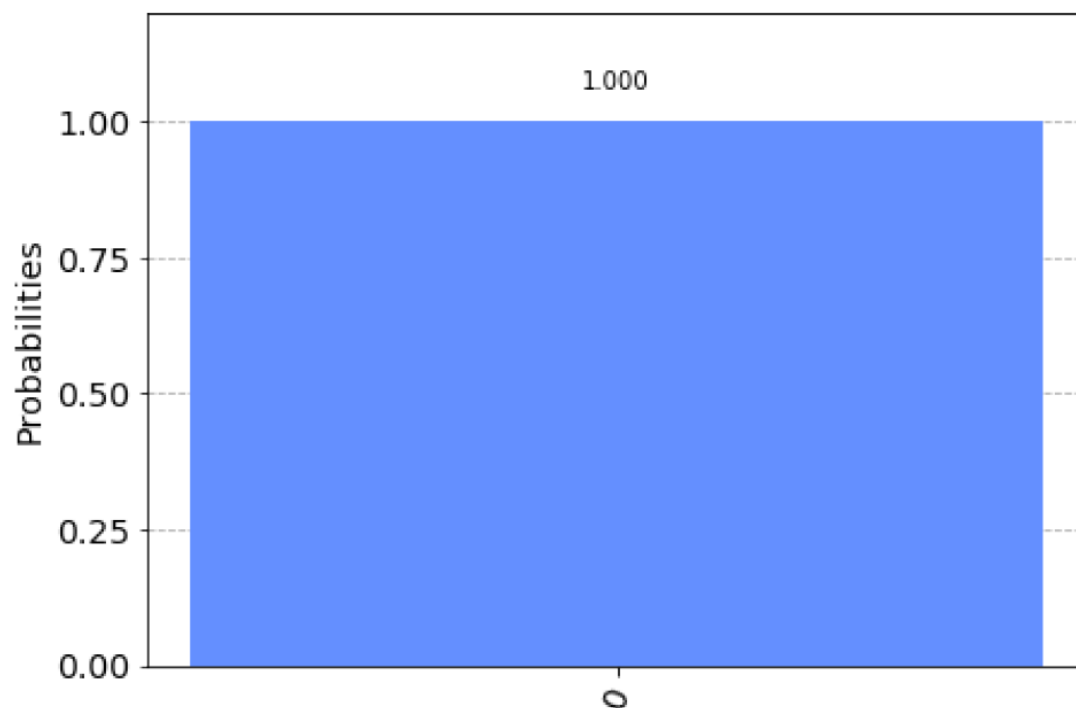


wynik liczbowy

Graficzna prezentacja wyników

```
[5]: # Graficzna prezentacja wyników  
plot_histogram(sim_result0.get_counts(circuit0))
```

[5]:

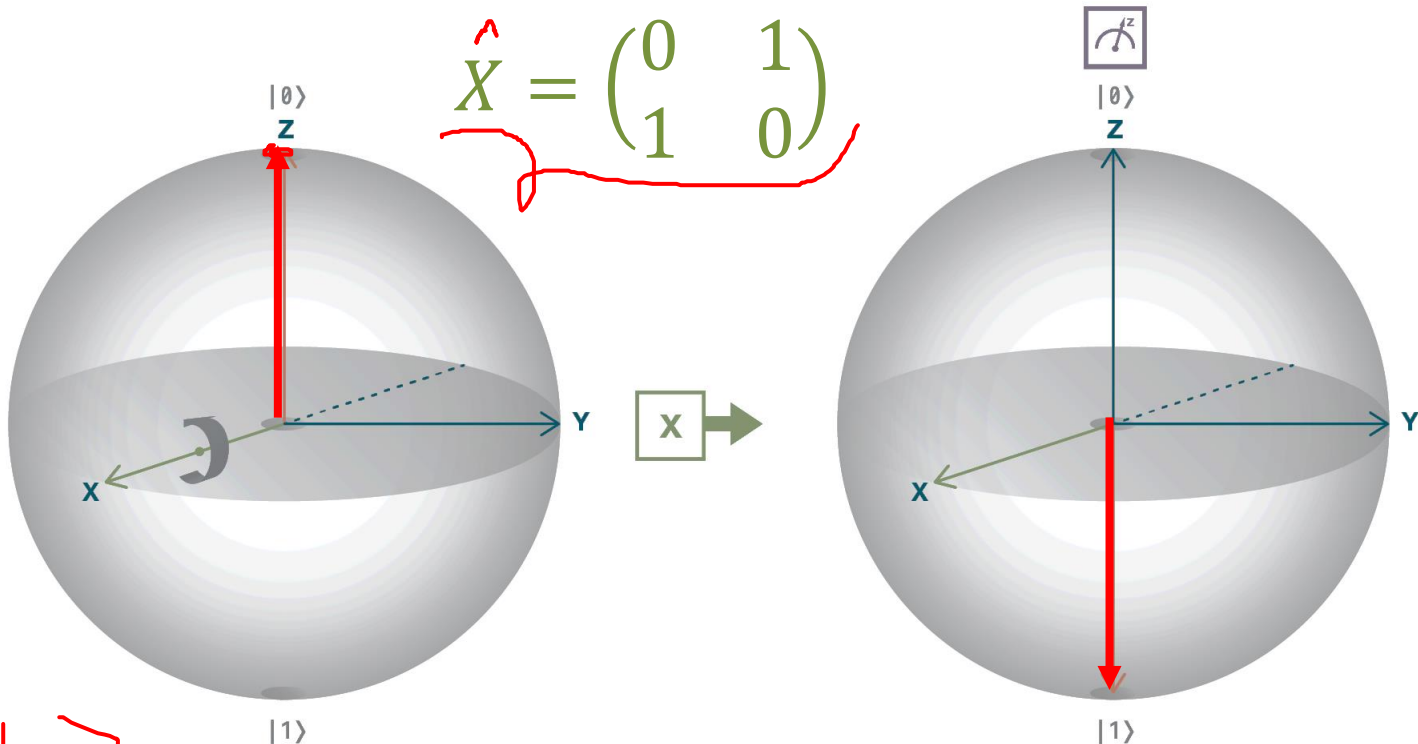


2. OPERACJE (BRAMKI) JEDNOKUBITOWE

Bramka kwantowa X = kwantowa negacja

$$\hat{G}_x, \hat{G}_1$$

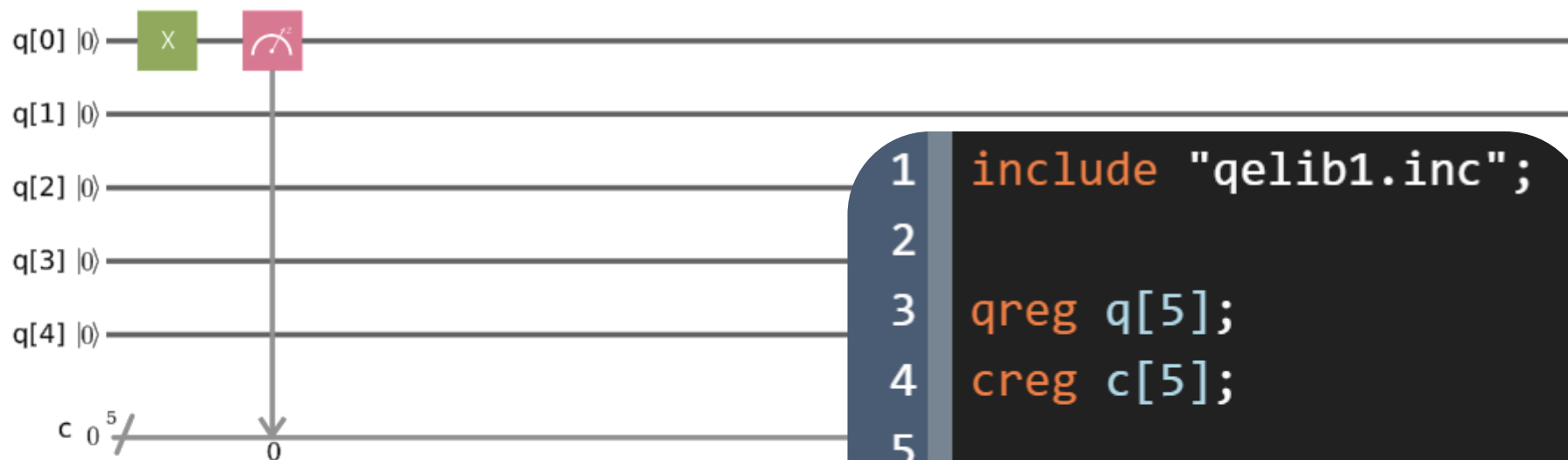
$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$



$$\hat{X}|1\rangle = |0\rangle$$

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

Działanie i odczyt wyniku działania bramki X

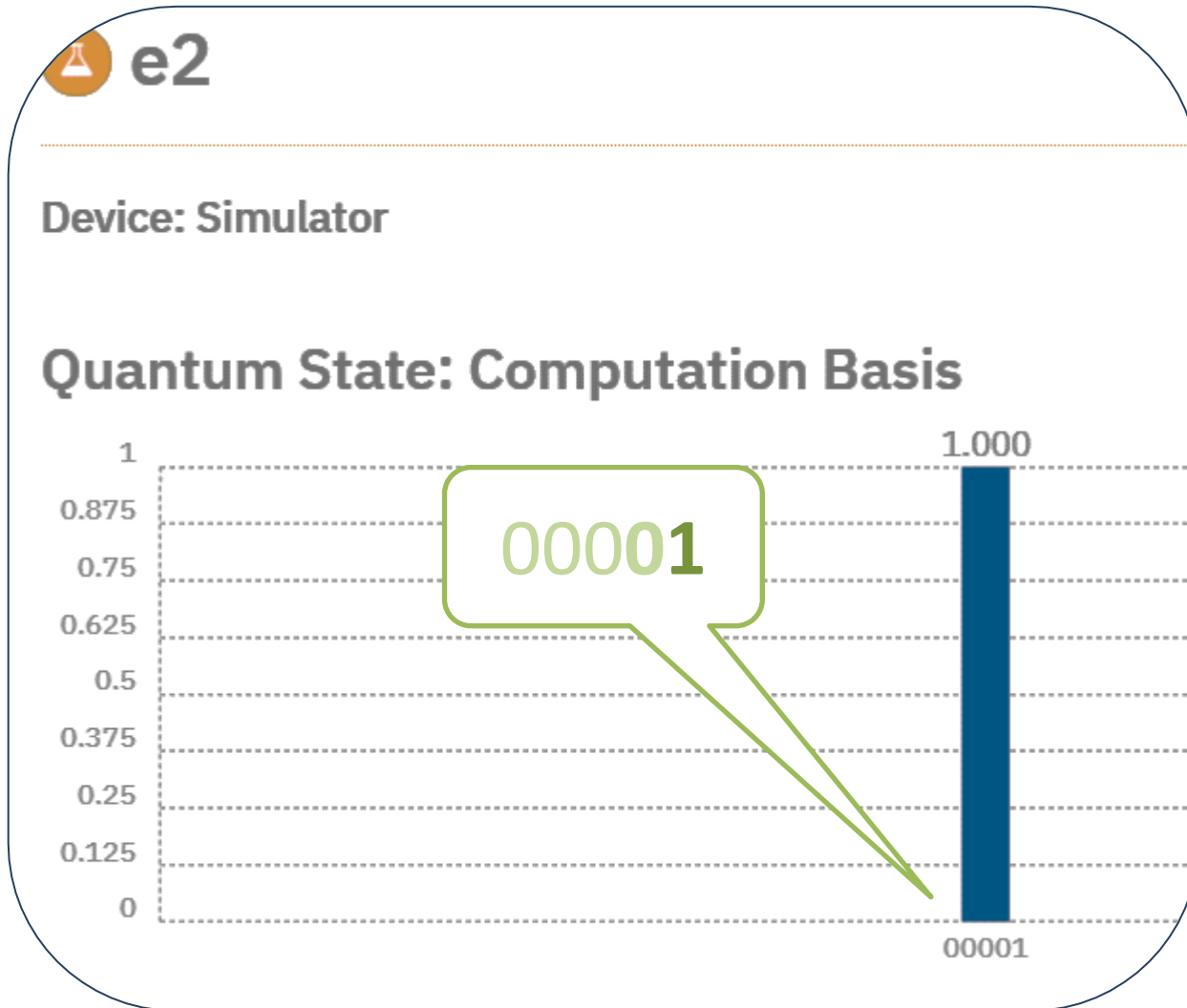


Kod graficzny

```
1 include "qelib1.inc";  
2  
3 qreg q[5];  
4 creg c[5];  
5  
6 x q[0];  
7 measure q[0] -> c[0];
```

Kod w asemblerze kwantowym
(QASM)

Wynik symulacji działania bramki X

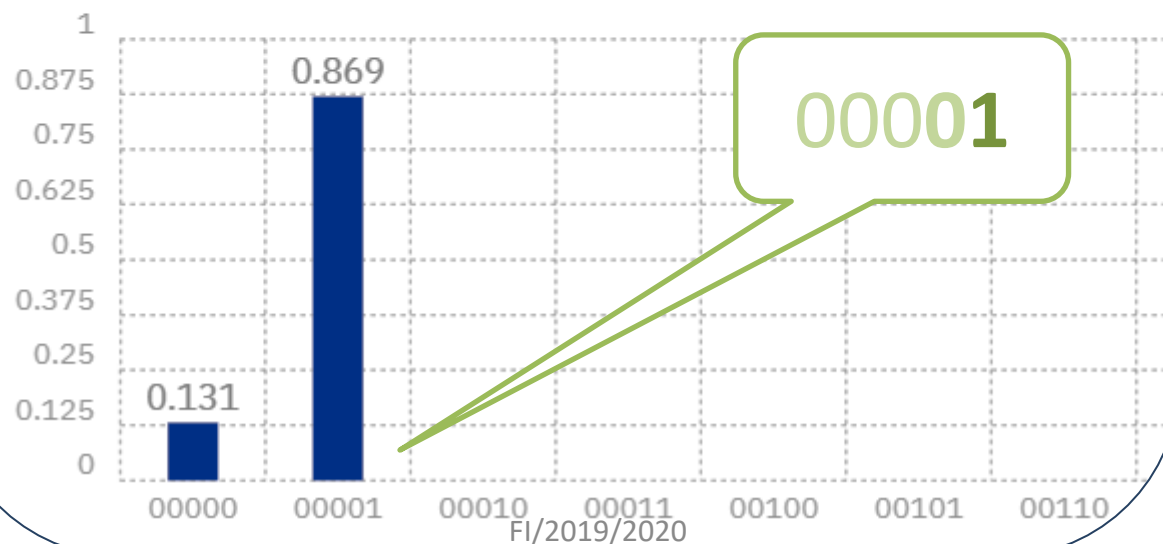


Wynik ściśle kwantowych obliczeń na kwantowym komputerze



Device: ibmqx4

Quantum State: Computation Basis



Tworzenie algorytmu kwantowego-obwodu kwantowego

```
[6]: # Zbuduj obwód kwantowy

n = 1 # liczba kubitów
q1 = QuantumRegister(n) # rejestr kwantowy
c1 = ClassicalRegister(n) # rejestr klasyczny

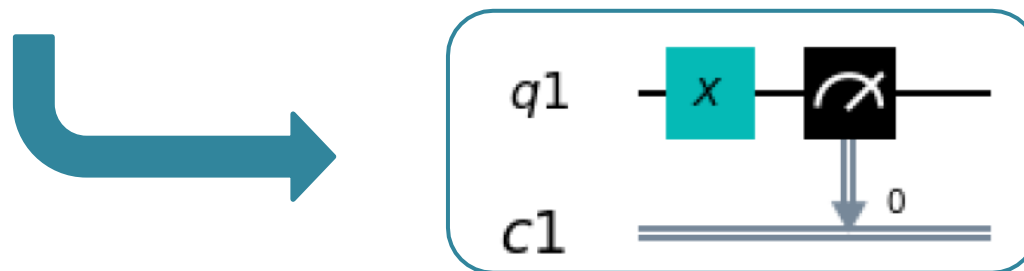
circuit1 = QuantumCircuit(q1, c1)

circuit1.x(q1[0]) # Operacja X realizowana na kubicie 1[0]
circuit1.measure(q1[0], c1[0])
```

```
[6]: <qiskit.circuit.instructionset.InstructionSet at 0x7f63807786d0>
```

```
[7]: circuit1.draw(output='mpl') # rysowanie obwodu
```

```
[7]:
```



Wykonywanie obliczeń kwantowych

```
[8]: # Wykonaj i zaprezentuj obliczenia  
backend = Aer.get_backend('qasm_simulator')  
job_sim1 = execute(circuit1, backend)  
sim_result1 = job_sim1.result()  
  
print(sim_result1.get_counts(circuit1))
```

```
{'1': 1024}
```

```
[9]: plot_histogram(sim_result1.get_counts(circuit1))
```

Wykonywanie obliczeń kwantowych

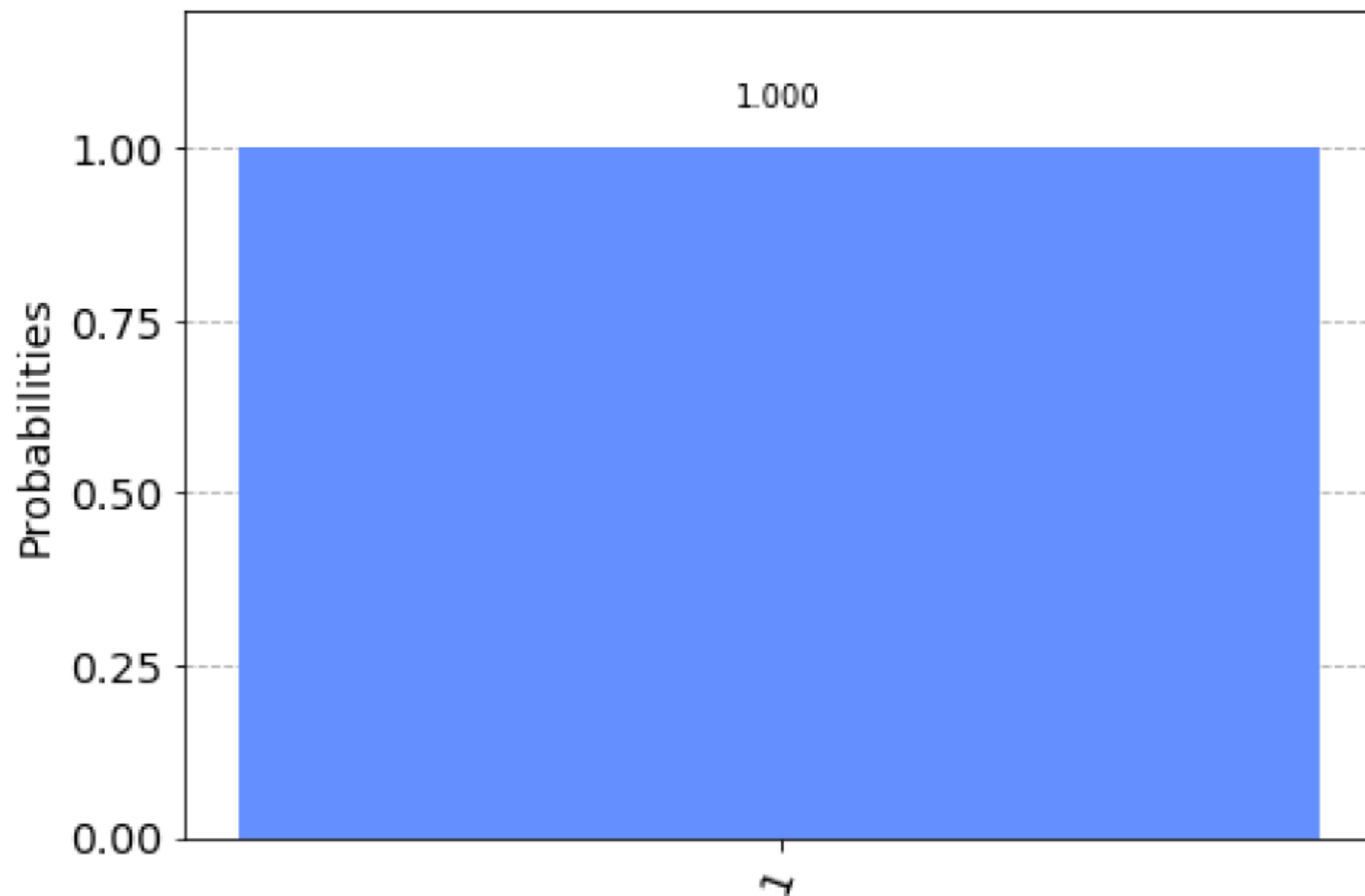
```
[8]: # Wykonaj i zaprezentuj obliczenia  
backend = Aer.get_backend('qasm_simulator')  
job_sim1 = execute(circuit1, backend)  
sim_result1 = job_sim1.result()  
  
print(sim_result1.get_counts(circuit1))
```

```
{'1': 1024}
```

```
[9]: plot_histogram(sim_result1.get_counts(circuit1))
```

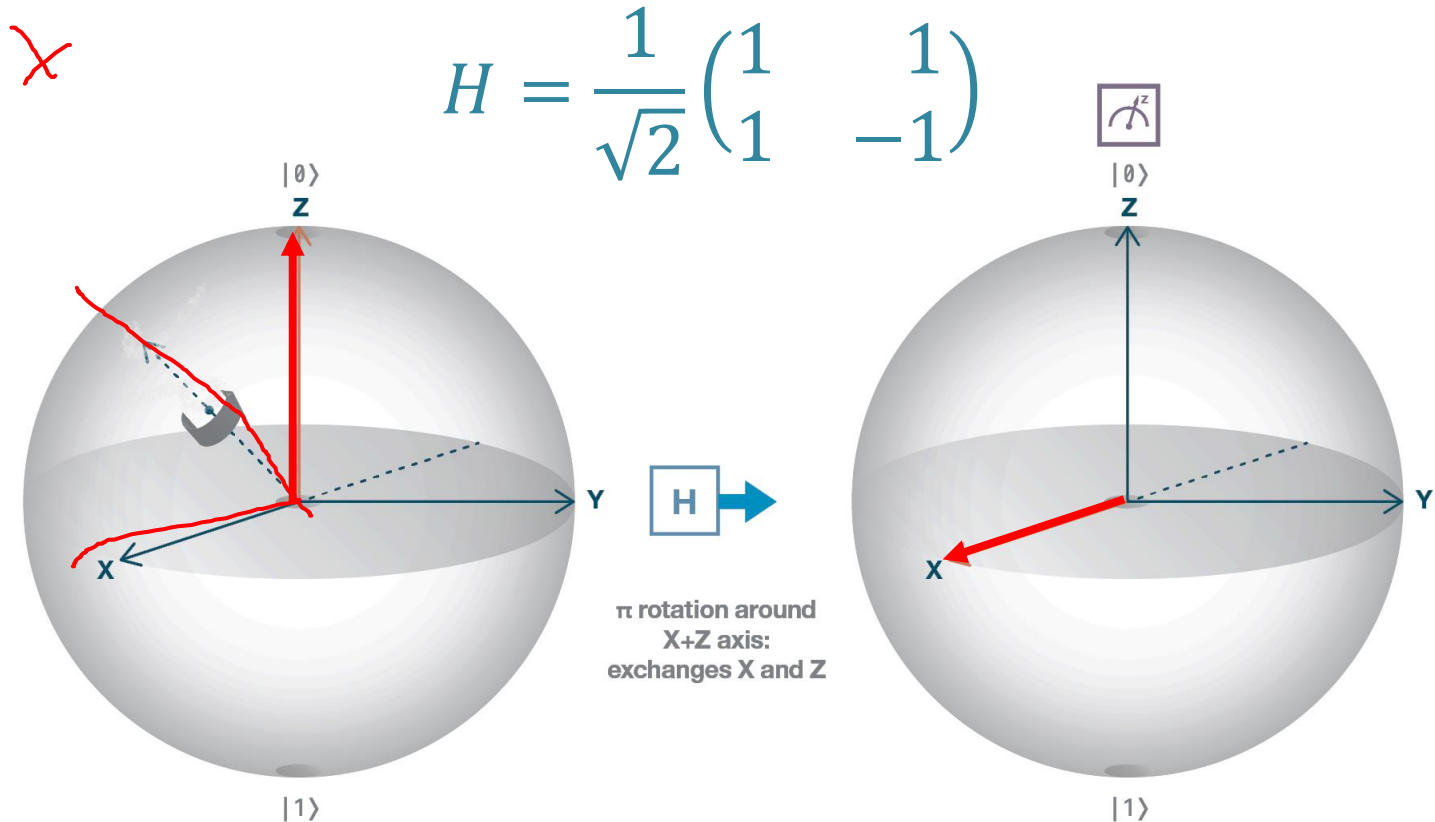
Graficzne przedstawienie wyniku

[9]:



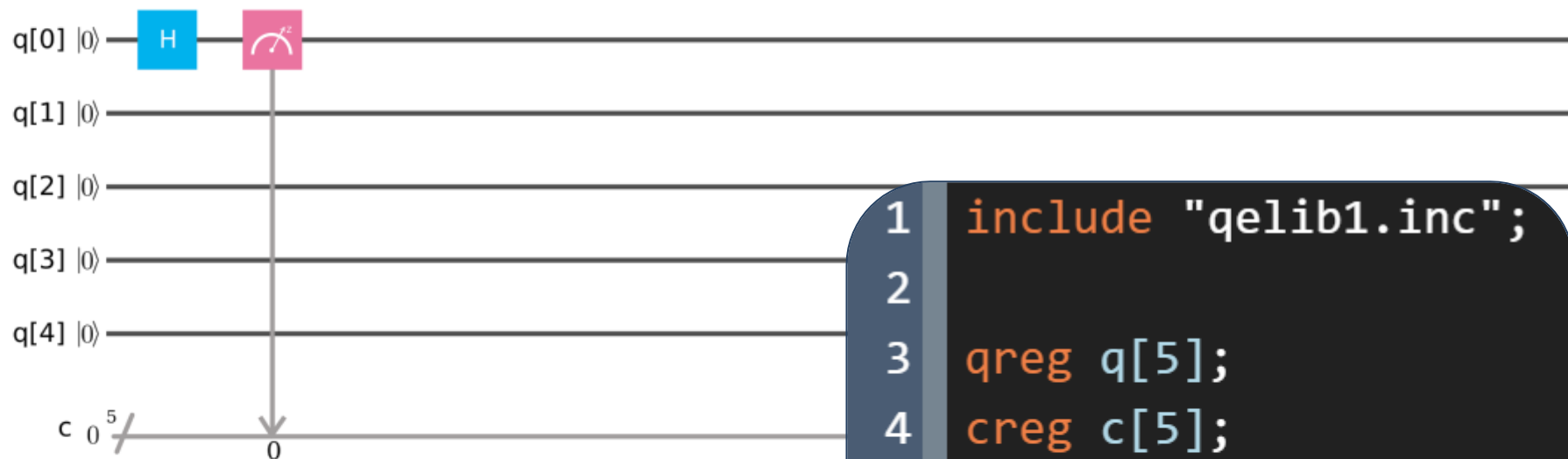
3. SUPERPOZYCJA STANÓW

Bramka kwantowa Hadamarda



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

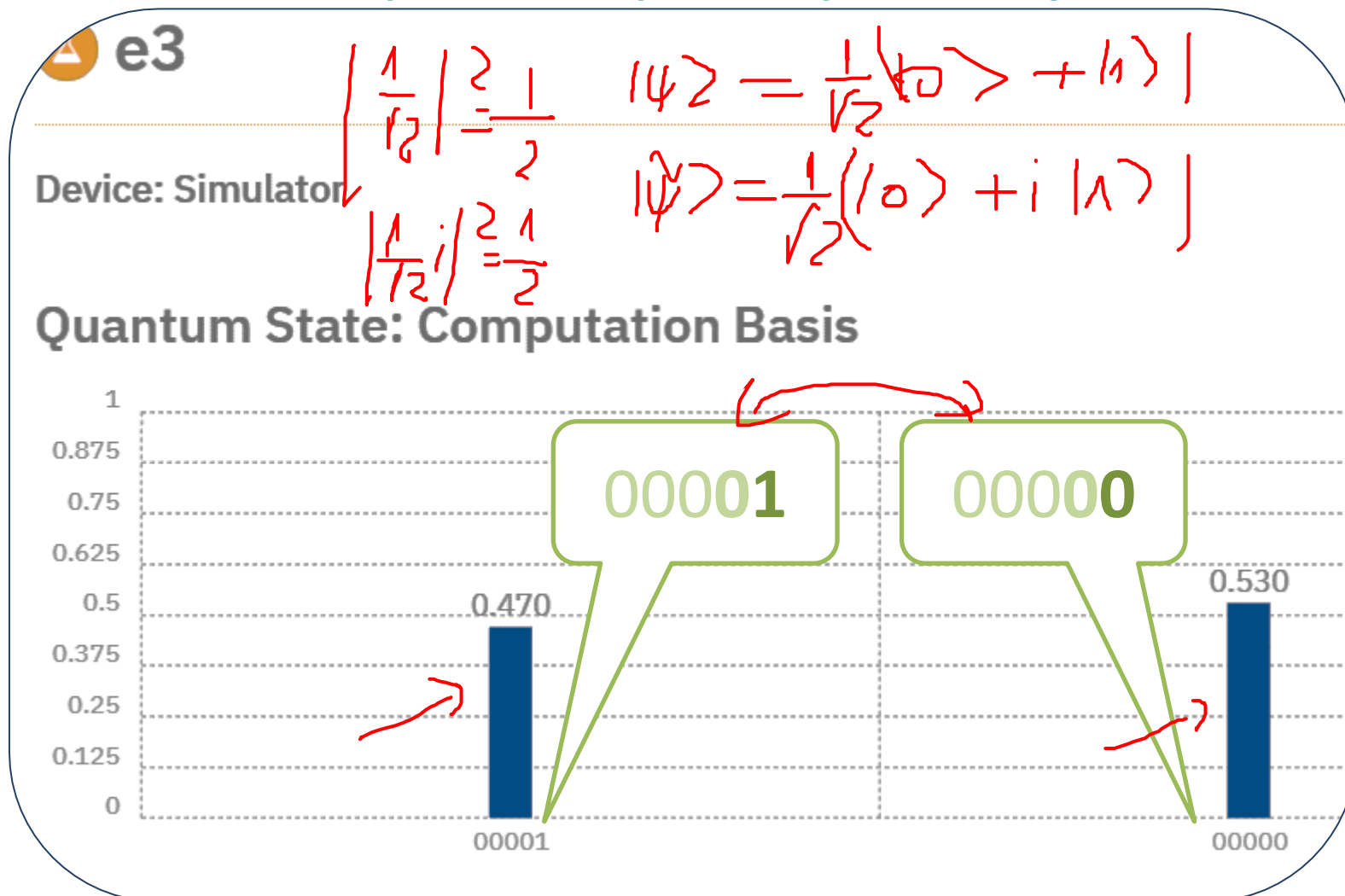
Działanie i odczyt wyniku działania bramki H (Hadamarda)



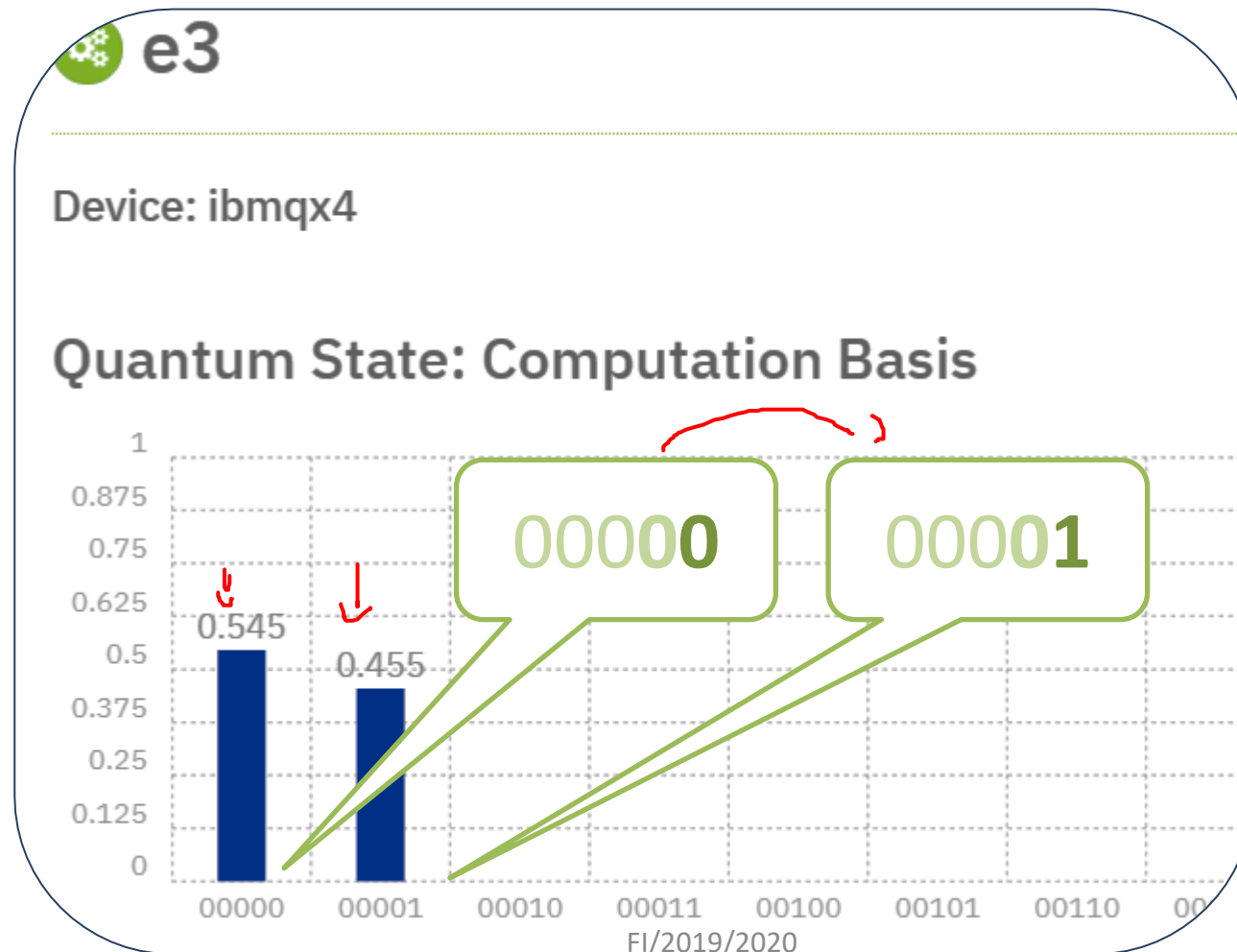
```
1  include "qelib1.inc";  
2  
3  qreg q[5];  
4  creg c[5];  
5  
6  h q[0];  
7  measure q[0] -> c[0];
```

Kod w asemblerze kwantowym
(QASM)

Bramka H pozwala zdefiniować generatorem liczb losowych – wynik symulacji



Bramka H – wynik obliczeń z zastosowaniem fizycznej bramki Hadamarda



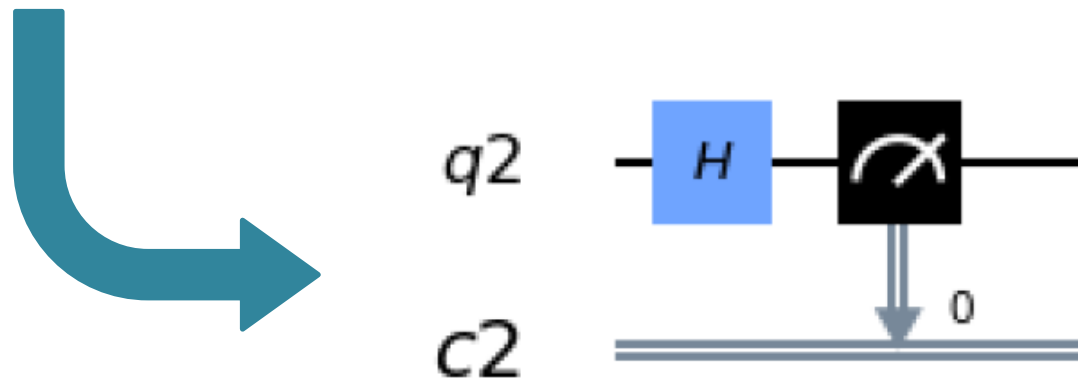
Tworzenie obwodu z bramką Hadamarda

```
[10]: q2 = QuantumRegister(n) # rejestr kwantowy  
      c2 = ClassicalRegister(n) # rejestr klasyczny  
  
      circuit2 = QuantumCircuit(q2, c2)  
  
✓ circuit2.h(q2[0]) # Operacja H realizowana na kubicie q2[0]  
  circuit2.measure(q2[0], c2[0])
```

```
[10]: <qiskit.circuit.instructionset.InstructionSet at 0x7f637fe08ad0>
```

```
[11]: circuit2.draw(output='mpl') # rysowanie obwodu
```

```
[11]:
```



Wykonywanie obliczeń

```
[12]: # Wykonaj i zaprezentuj obliczenia
      job_sim2 = execute(circuit2, backend)
      sim_result2 = job_sim2.result()

      print(sim_result2.get_counts(circuit2))
```

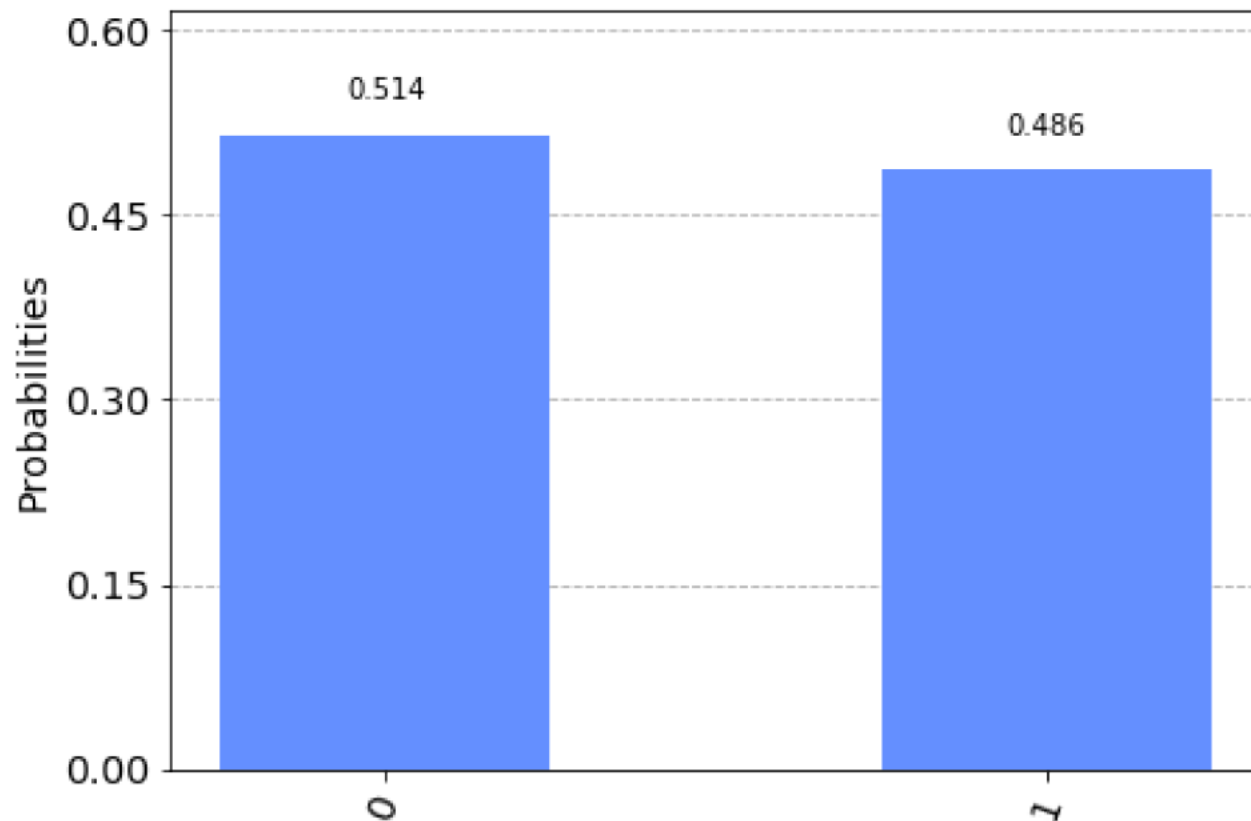
{'0': 526, '1': 498}

10 2 L

Wykonywanie obliczeń i prezentacja wyników

```
[13]: plot_histogram(sim_result2.get_counts(circuit2))
```

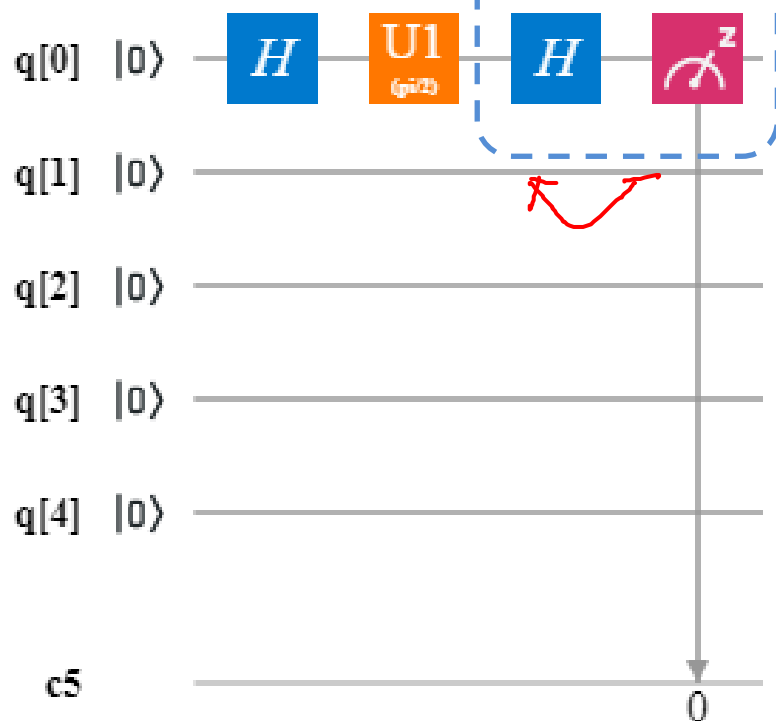
[13]:



4. Tomografia stanu jednego kubitu – realizacje na komputerze kwantowym

$\langle + | - \rangle = 0$ Pomiar w bazie X ✓

$$\langle + | + \rangle = \langle - | - \rangle = 1$$



$$P_0 = |\langle 0 | \psi \rangle|^2$$

$$P_1 = |\langle 1 | \psi \rangle|^2$$

$$P_{\pm} = |\langle \pm | \psi \rangle|^2$$

$$\sigma_1 = \underline{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

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

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

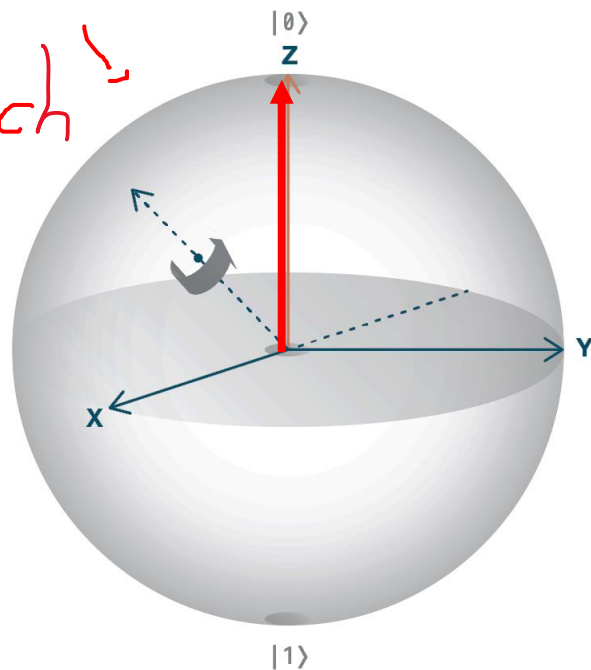
$$\begin{cases} |0\rangle \\ |1\rangle \end{cases} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\begin{cases} |0\rangle \rightarrow \lambda = 1 \\ |1\rangle \rightarrow \lambda = -1 \end{cases}$$

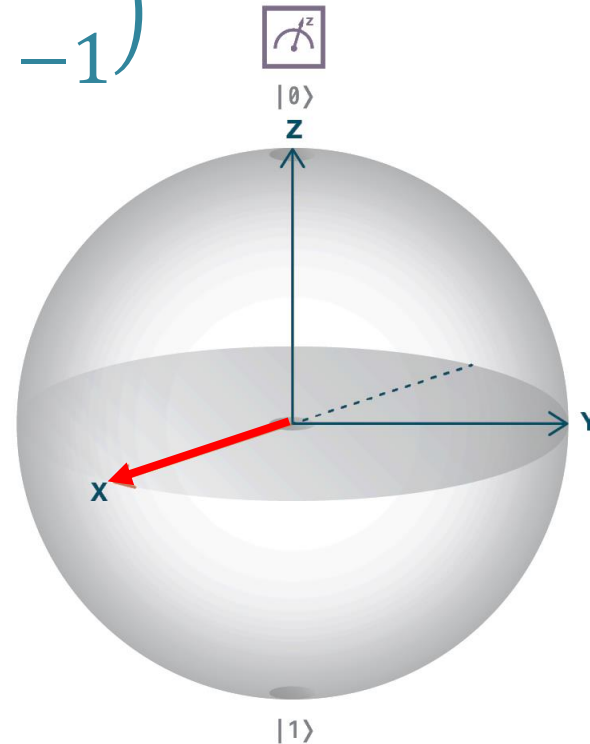
Bramka kwantowa Hadamarda

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

Bloch



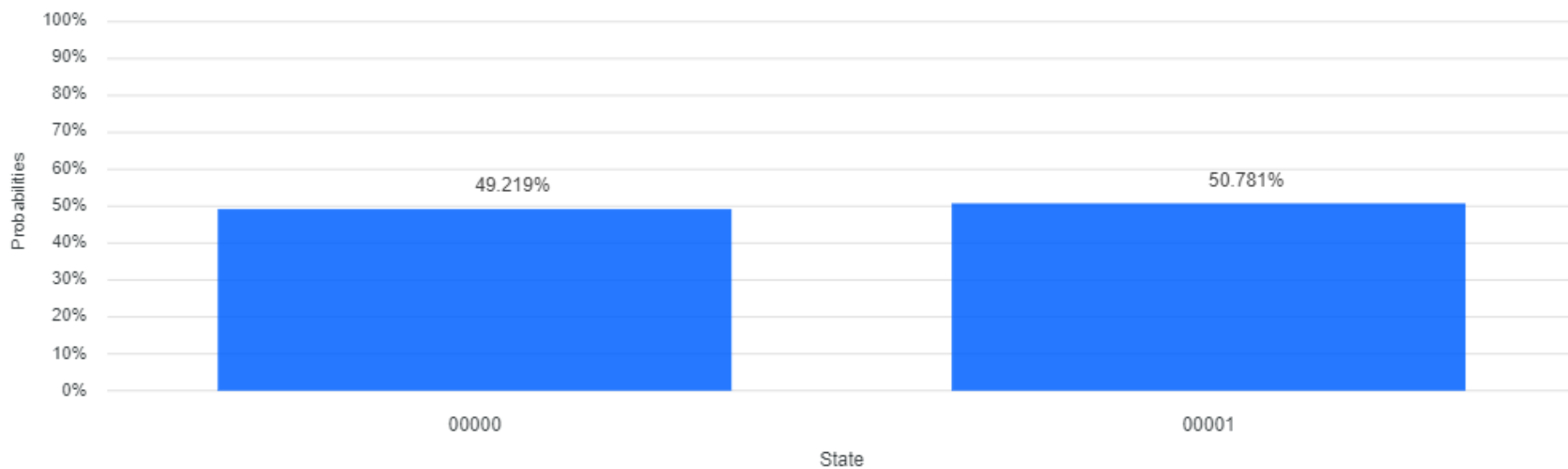
π rotation around
X+Z axis:
exchanges X and Z



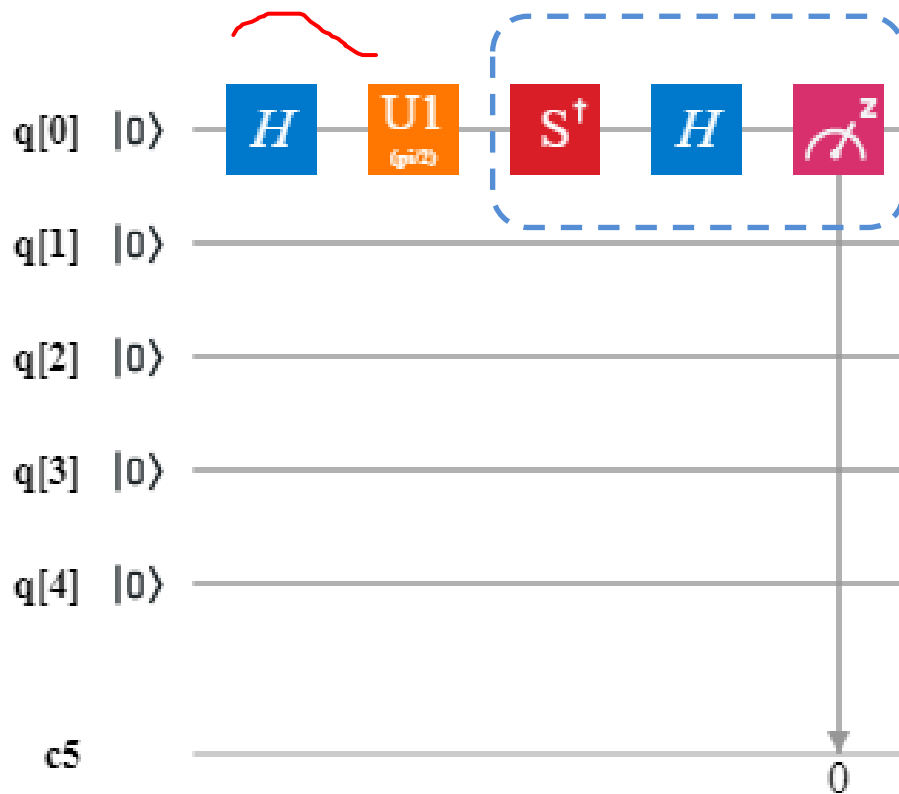
$$H |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

Wynik pomiaru w bazie X

Histogram



Pomiar w bazie Y



$$P_{\pm} = |\langle e | \psi \rangle|^2$$

$$\sigma_2 = \cancel{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

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

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

$$\hat{Y}|R\rangle = +|L\rangle$$

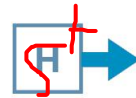
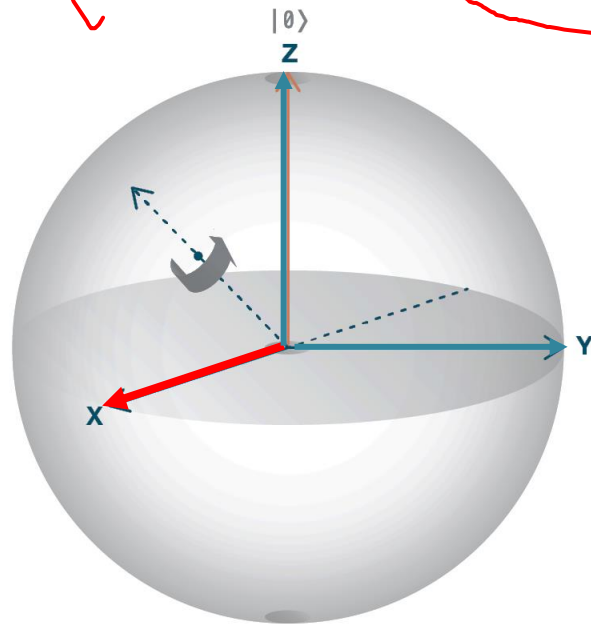
$$|L\rangle = -|R\rangle$$

Bramka kwantowa Hadamarda

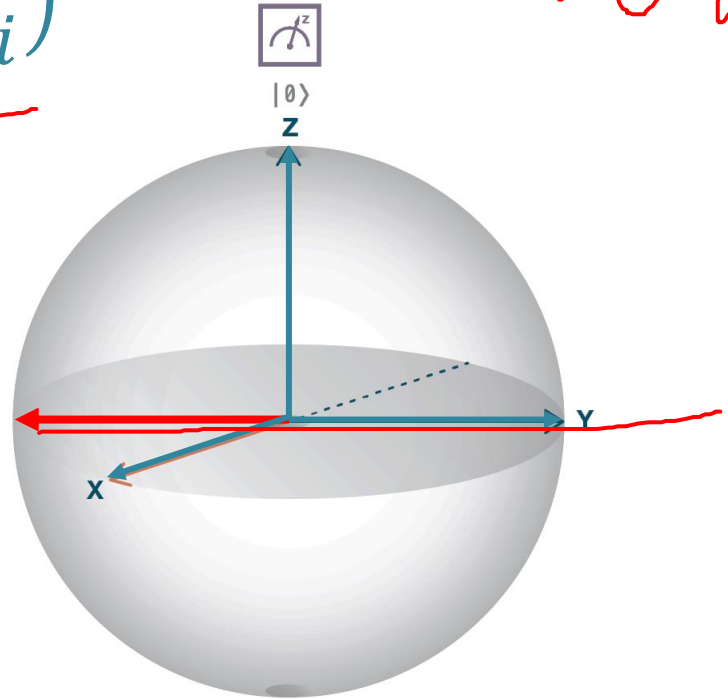
$$S^\dagger = (S^\dagger)^\dagger$$

$$S^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

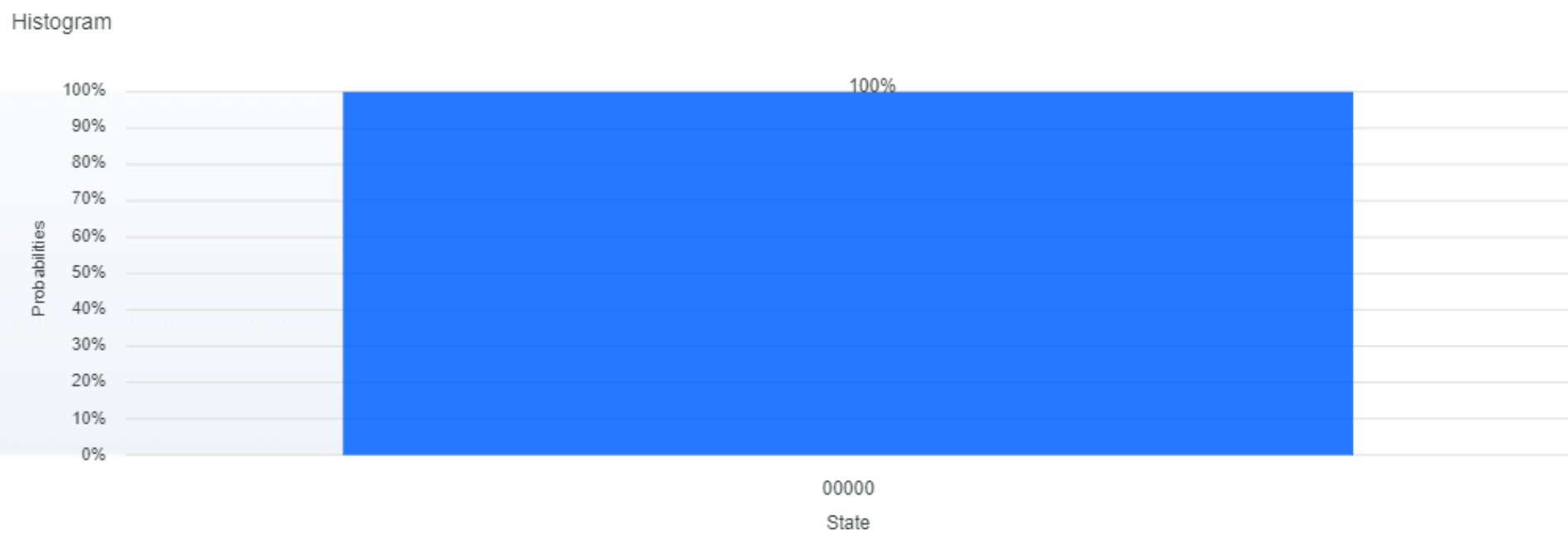


π rotation around
X+Z axis:
exchanges X and Z

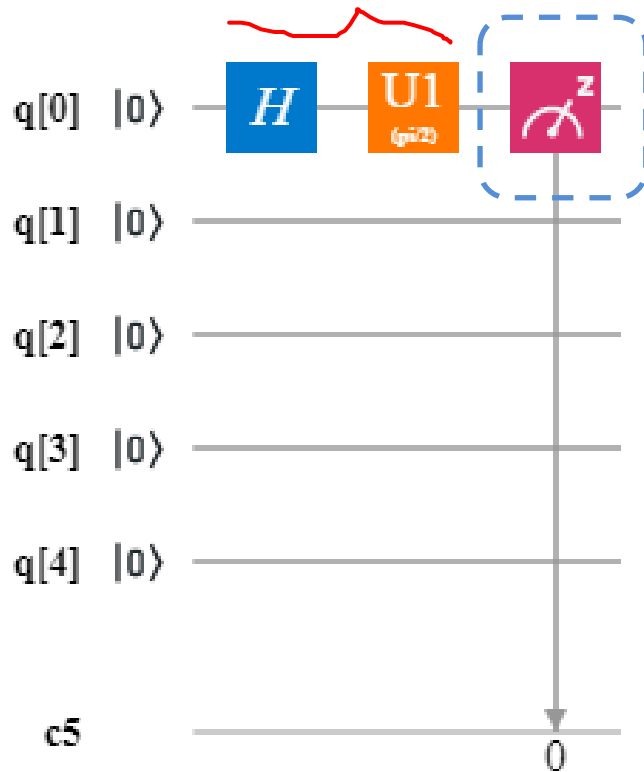


$$S^\dagger |0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - i|1\rangle)$$

Wynik pomiaru w bazie Y



Pomiar w bazie Z

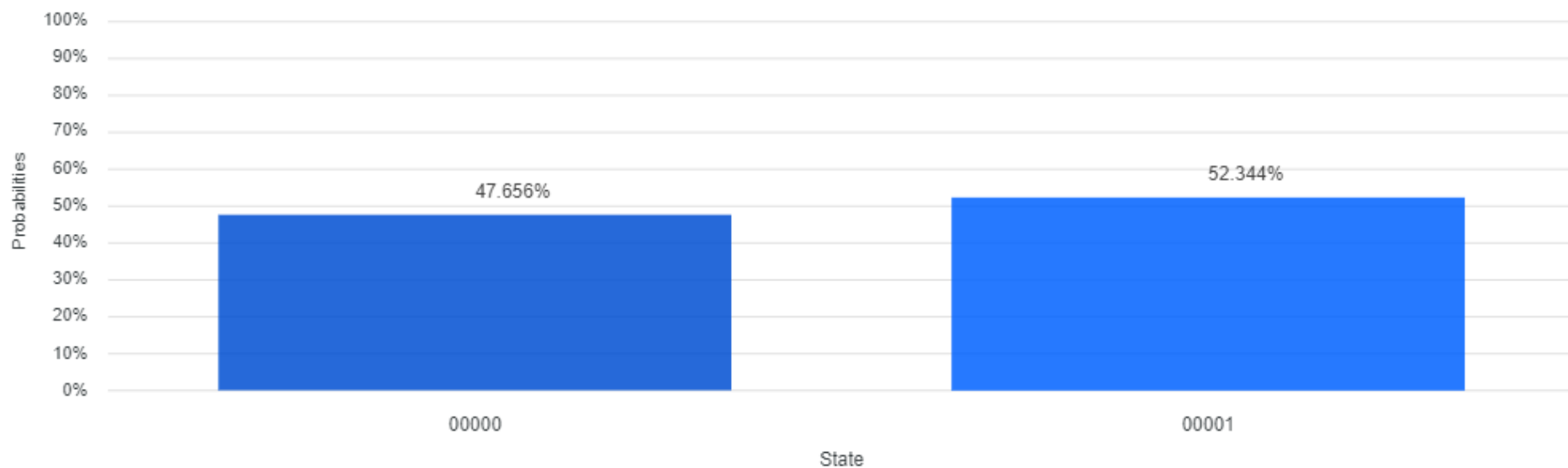


$$\sigma_3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

→ $\underline{|0\rangle}$
 $\underline{|1\rangle}$

Wynik pomiaru w bazie Z

Histogram



POMIAR WEKTORA STANU- -POMIAR PO ZESPOLE

$$p_{\psi \rightarrow \pm} = |\langle 0/1 | \psi \rangle|^2 = |c_{\pm}|^2$$

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$$

$$p = \cos^2 \frac{\theta}{2} \quad 1-p = \sin^2 \frac{\theta}{2}$$

σ_3

dla bardzo dużej liczby cząstek pomiar

$$p_{\psi \rightarrow 0/1} \approx \frac{n_{0/1}}{n}$$

$$\cos \varphi = \frac{q - 1/2}{\sqrt{p(1-p)}}$$

q – prawdopodobieństwo znalezienia kubit „0” w pomiarze X

dla bardzo dużej liczby cząstek pomiar

σ_1

ZADANIE?

Inne jednokubitowe bramki kwantowe

$$U_1(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix},$$

$$U_2(\lambda, \phi) = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{e^{i\lambda}}{\sqrt{2}} \\ \frac{e^{i\phi}}{\sqrt{2}} & \frac{e^{i(\lambda+\phi)}}{\sqrt{2}} \end{bmatrix}$$

$$U_3(\theta, \phi, \lambda) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} \sin\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i(\lambda+\phi)} \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Pobieranie notatnika *Jupyter* w formacie pdf

The screenshot shows the IBM Quantum Experience Jupyter notebook interface. The browser address bar displays `quantum-computing.ibm.com/jupyter/user/Untitled1.ipynb`. The notebook's 'File' menu is open, showing options like 'Save and Checkpoint', 'Download as', and 'PDF via LaTeX (.pdf)'. A red arrow points to the 'PDF via LaTeX (.pdf)' option. The notebook content includes a code cell with the following text:

```
%matplotlib inline

# Import Qiskit libraries and configuring account
from qiskit import QuantumCircuit, execute, Aer, IBMQ
qiskit_ = IBMQ_provider()
# Compile, transpile and assemble a circuit
qc = QuantumCircuit(1)
qc.h(0)
qc.measure(0,0)
# Compile and execute the circuit
job = execute(qc, backend=Aer.get_backend('aer_simulator'))
result = job.result()
print(result.get_counts(qc))
```

The bottom of the interface shows the IBM Quantum footer with links to Privacy Policy, End User Agreement, IBM Terms of Use, and IBM Privacy Statement, along with the version number v1.14.0. The Windows taskbar at the bottom shows the search bar and several application icons.