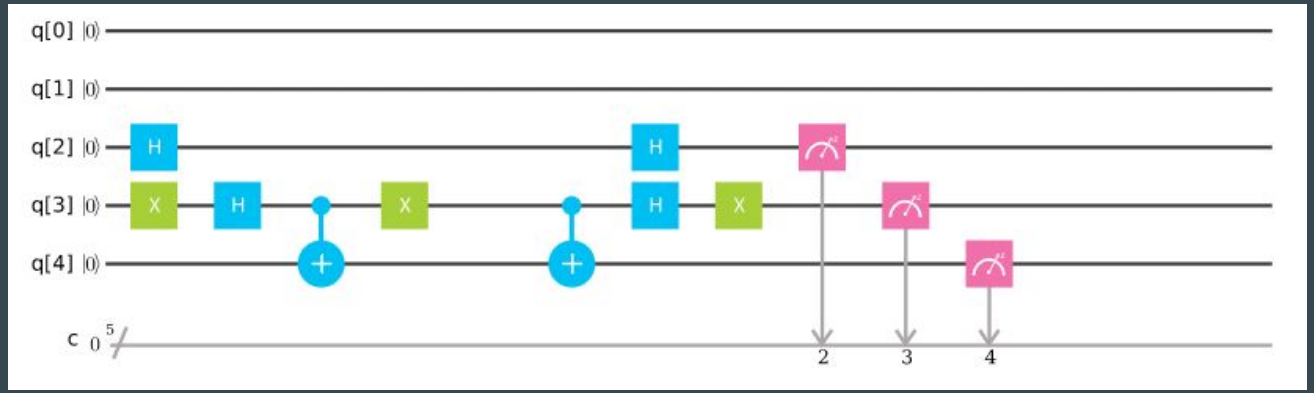
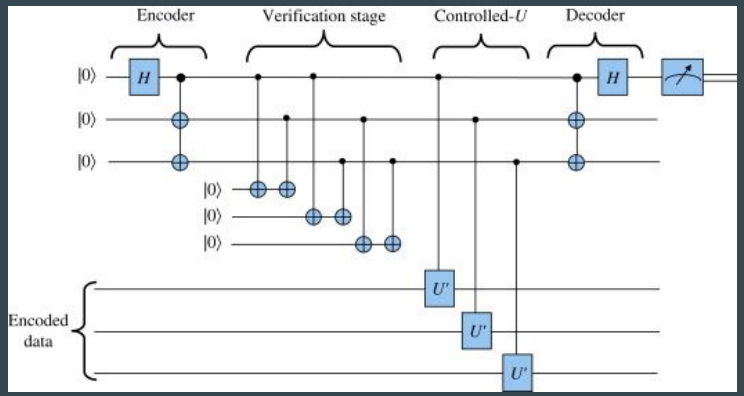
Quantum Computing with Python & IBM Quantum Experience

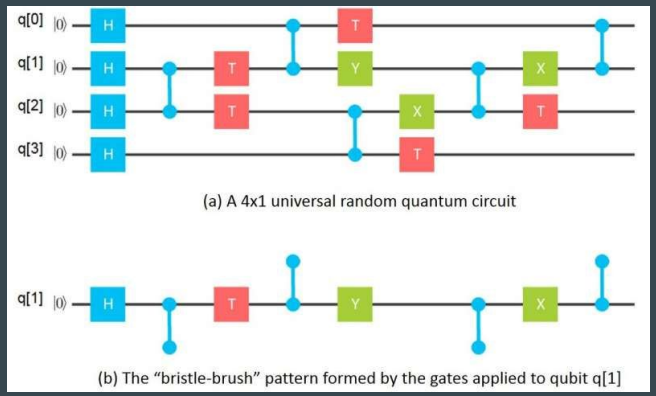
•••

Week 3 - 2019 AI Inspire


q[0] |0⟩
q[1] |0⟩
q[2] |0⟩
q[3] |0⟩
q[4] |0⟩
c 0 5
H
X
H
X
H
H
X
2
3
4

Background


Encoder
Verification stage
Controlled-U
Decoder
|0⟩
|0⟩
|0⟩
|0⟩
|0⟩
|0⟩
H
H
U
U
U
Encoded data


q[0] |0⟩
q[1] |0⟩
q[2] |0⟩
q[3] |0⟩
H
T
Y
X
(a) A 4x1 universal random quantum circuit
q[1] |0⟩
(b) The “bristle-brush” pattern formed by the gates applied to qubit q[1]

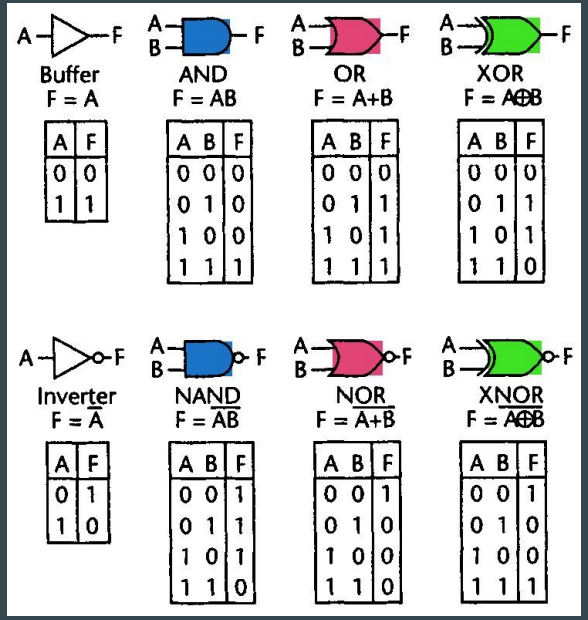
Cirq

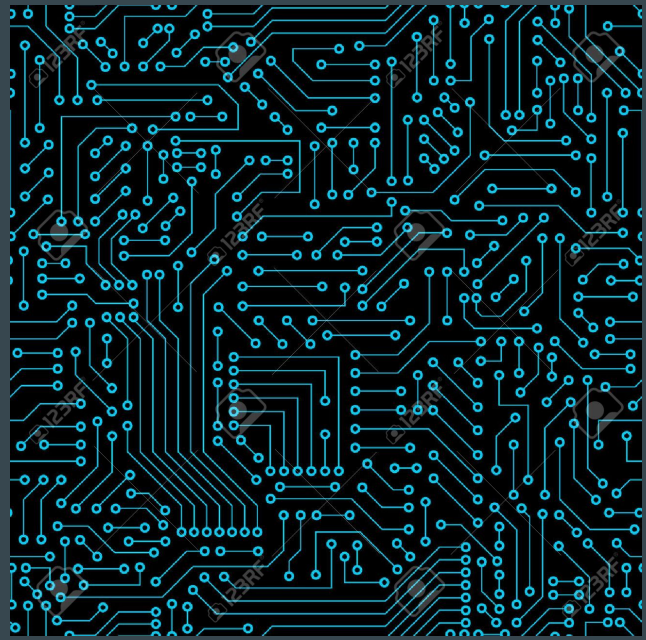
- Python Software Library
- Created by Google
- For NISQ Quantum Algos
  - NISQ Computer = Noisy Intermediate Scale Quantum Computer
  - NISQ → computers with smaller # qubits (50-100) & require error correction
- Gives control over quantum circuits & gates
  - Use these quantum circuit programs and run with quantum computer/simulator

Cirq

Qiskit

- Open source library framework for Python
- Running quantum programs + algos
- Run on either IBM's quantum computers (backend) or simulators
- Can create quantum circuits through quantum gates, hardware


Buffer
F = A
AND
F = AB
OR
F = A+B
XOR
F = A⊕B
Inverter
F = Ā
NAND
F = $\overline{AB}$
NOR
F = $\overline{A+B}$
XNOR
F = $\overline{A \oplus B}$




Classical Computer
Logic Gates

https://www.youtube.com/watch?v=gI-qXk7XojA

Classical Gates

- Logic Gate
  - Physical structure with binary input & output
    - Output determined from boolean function
- Operate on classical bits
- Irreversible
  - Many lose info

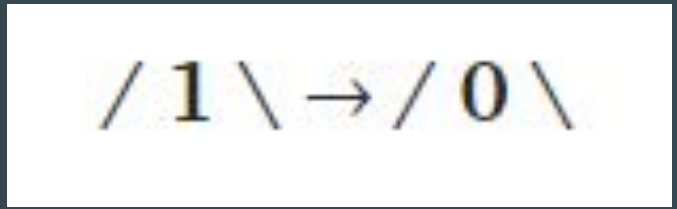
Quantum Gates

- Logic Gate
- Operate on qubits so they can perform operations
  - Superposition and entanglement
- Reversible
  - Never lose information
  - “Undo”
  - Both input and output → same # qubits
- Transform quantum states

Vectors & Matrices

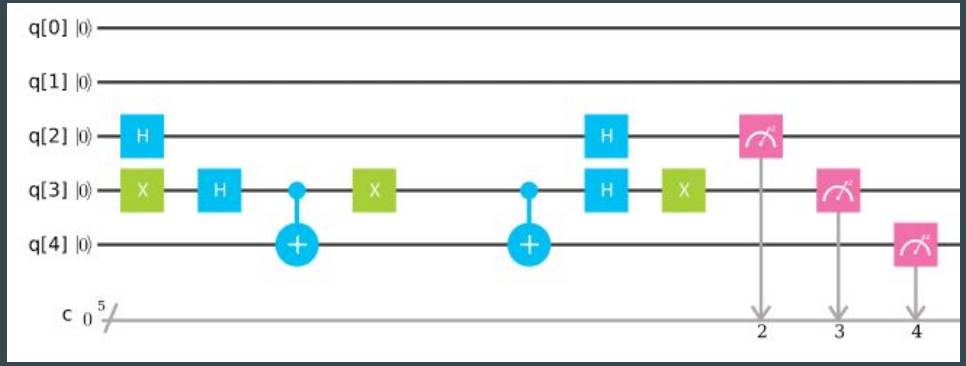
- Vectors denote state of qubit
- Binary state
  - 0 and 1 vectors
  - 0 = up spin
  - 1 = down spin

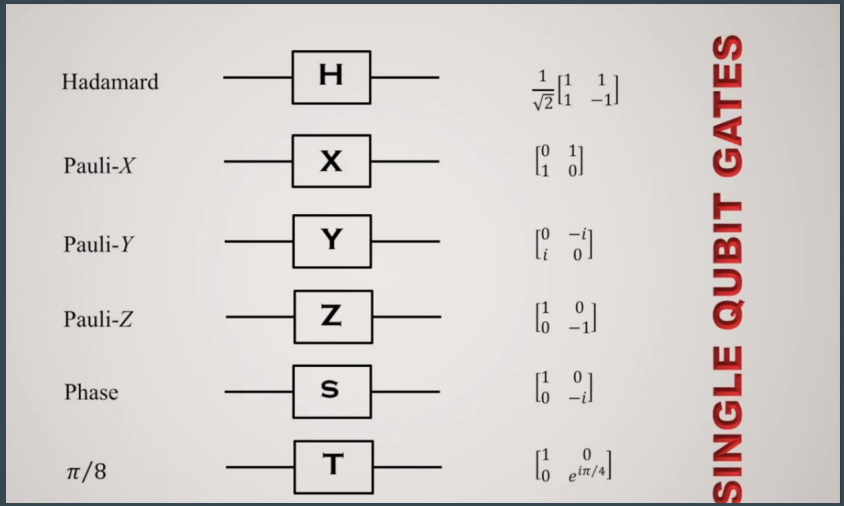
- Matrices transform vectors into new states
  - Matrix mult
- Matrices represent gates which change the state of these vectors
  - Ex : up state to down state or down state to up state



Pauli Gate

- Wolfgang Pauli
- Operate on Pauli spin matrices
  - Computer changes to spin of a particle
  - Quantum gates → supposed to transform the state (in this case, the state = spin of electron)
- 3 axes on Bloch sphere
  - X, Y, Z → one Pauli gate/matrix
  - Each axis → 1 matrix rotates the vector
- Pauli gate → acts on 1 qubit





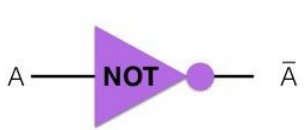
Pauli X Gate

- Rotates vector along x axis
- One of simplest gates
- Operates with negation
- Analogous to classical NOT gate
  - Simply flips the states of qubits once it passes through gate with matrix (switching amplitudes also)
  - Ex - transform spin up to spin down

/ 0 1 \

\ 1 0 /

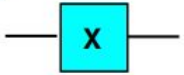
$$\alpha_0|0\rangle + \alpha_1|1\rangle \rightarrow \alpha_1|0\rangle + \alpha_0|1\rangle$$



A	$\bar{A}$
0	1
1	0

PAULI X GATE

$\|A\rangle$	$\|\bar{A}\rangle$
0	1
1	0

> Input state: $c_0|0\rangle + c_1|1\rangle$
> Output state: $c_1|0\rangle + c_0|1\rangle$
> Graphic symbol: 

applying the quantum NOT-gate to our qubit (in this case the spin-up state of an electron), looks like this:

X|0> = / 0 1 \ /1\ = /0\ = |1>

\ 1 0 / \0/ \1/

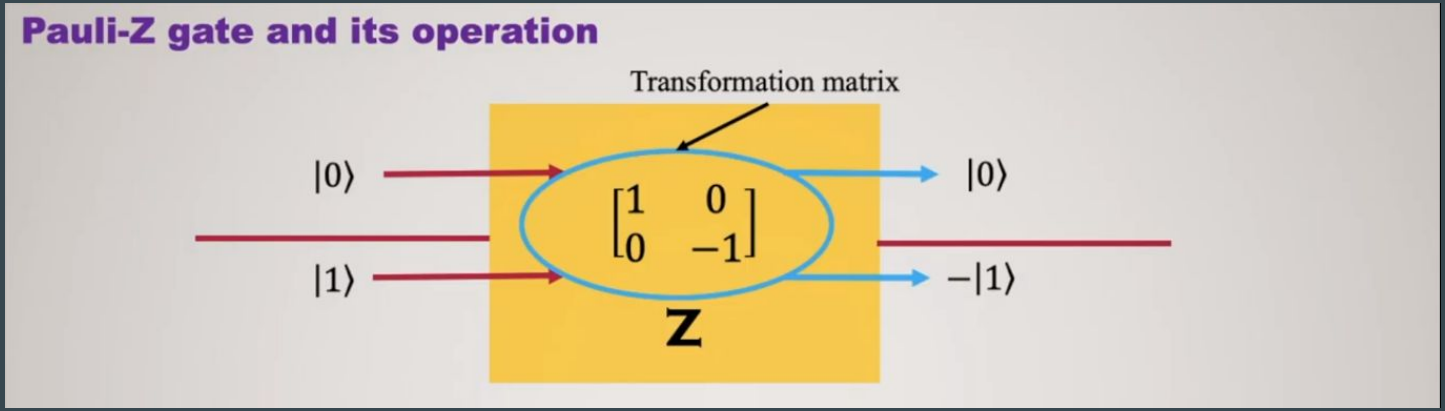
Applied to a spin-*down* vector, the complete notation looks like this:

X|1> = / 0 1 \ /0\ = /1\ = |0>

\ 1 0 / \1/ \0/

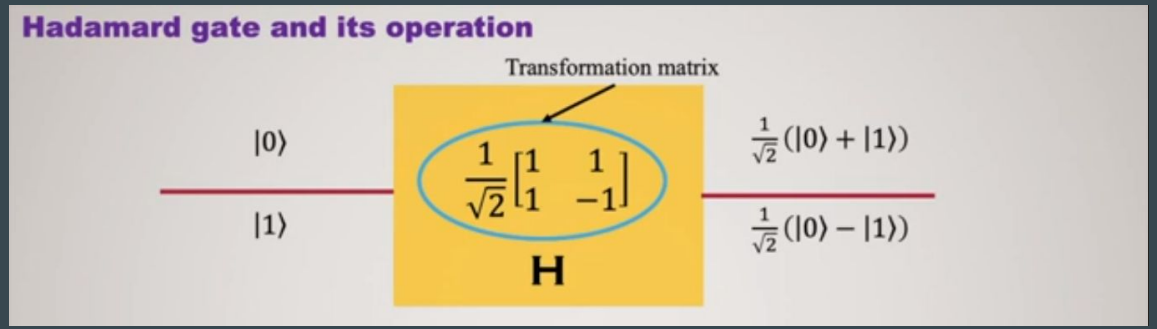
Pauli Z Gate

- Change spin of qubit electron
- Similar to Pauli X Gate
- Rotates vector along Z axis
- Also switches amplitude
- Multiples by -1 along with switching



Hadamard Gate

- Very famous
- Turns original gate input into superposition of two gates (prob ½ for both states in output)
- From spin up to spin down (qubit existing in exactly 1 state) to being in a combination of both by passing through gate
- One of the first gates in circuit
  - Beginning - qubit exists in solely 1 state
  - Hadamard gate makes it exist in both states @ same time → unleash quantum potential early in circuit
- Applying H gate twice = performing nothing

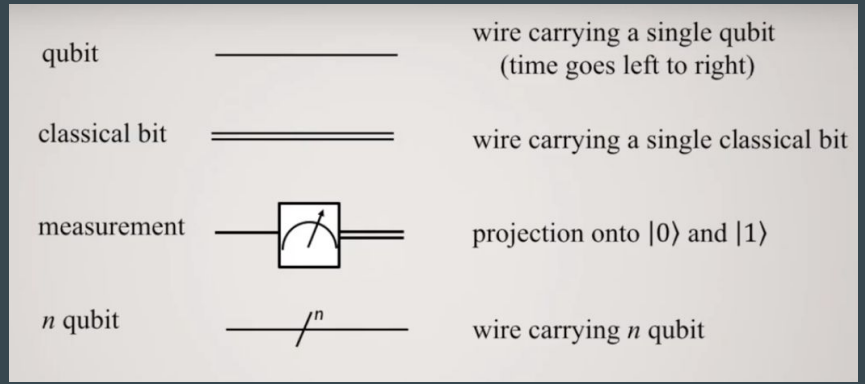


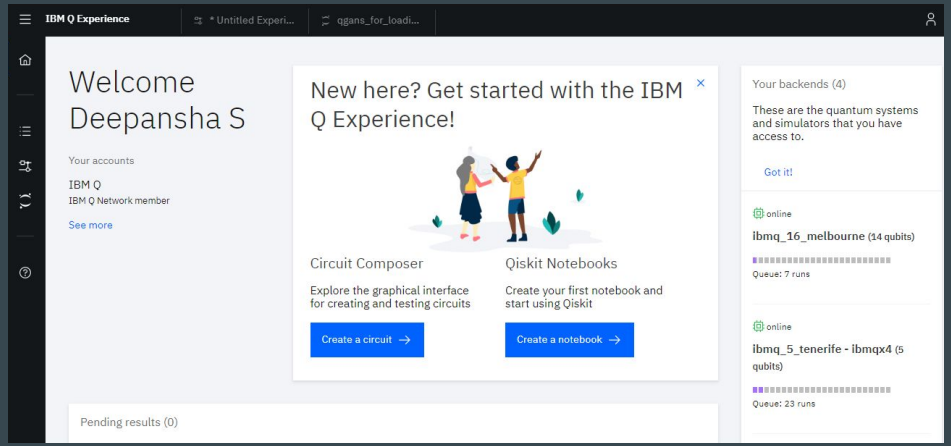
Building Quantum Circuits

https://www.youtube.com/watch?v=pYD6bvKLI_c

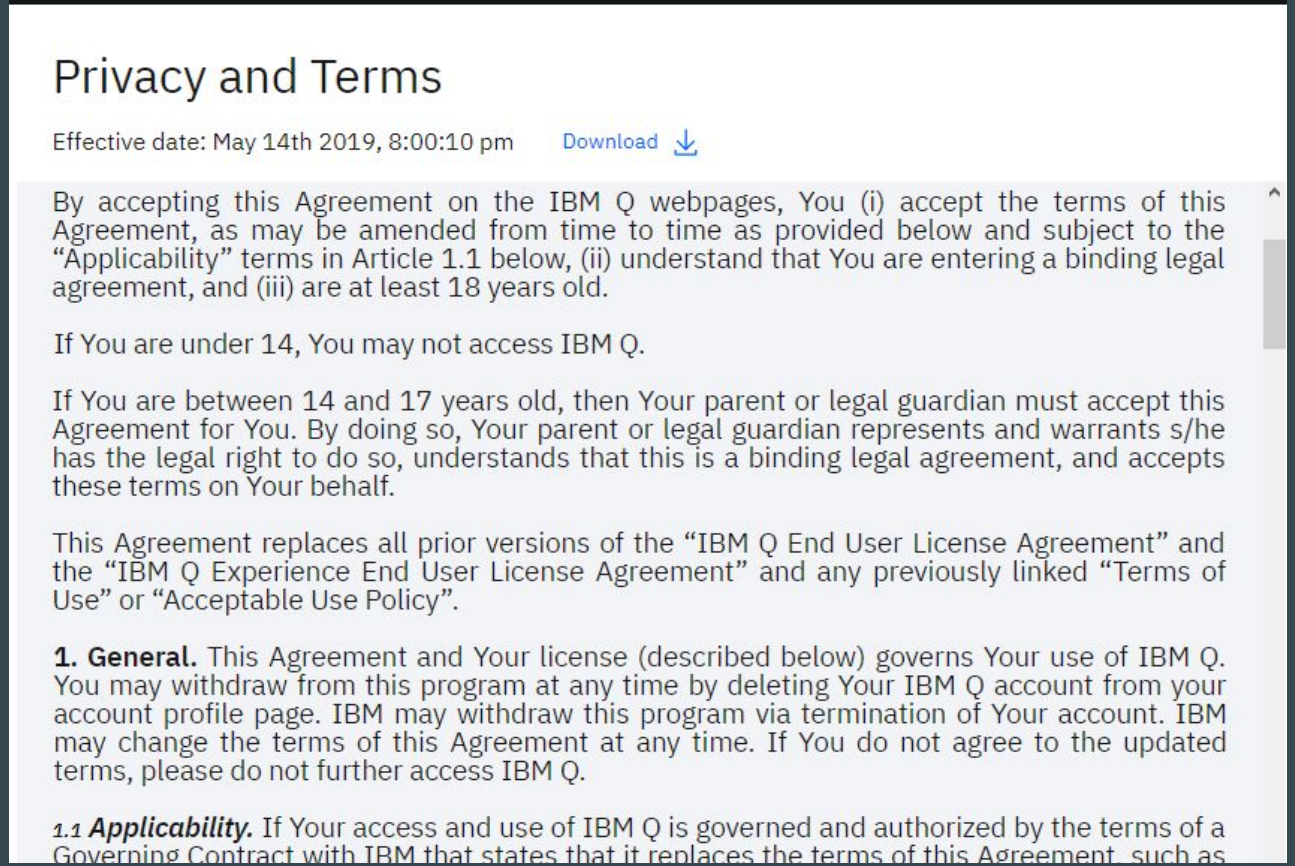
What is a Quantum Circuit?

- Quantum Circuit - sequence of quantum gates
- IBM Quantum Experience platform
  - Run = real quantum computer, Simulate = just simulation





IBM Q Experience



Privacy and Terms

Effective date: May 14th 2019, 8:00:10 pm Download

By accepting this Agreement on the IBM Q webpages, You (i) accept the terms of this Agreement, as may be amended from time to time as provided below and subject to the "Applicability" terms in Article 1.1 below, (ii) understand that You are entering a binding legal agreement, and (iii) are at least 18 years old.

If You are under 14, You may not access IBM Q.

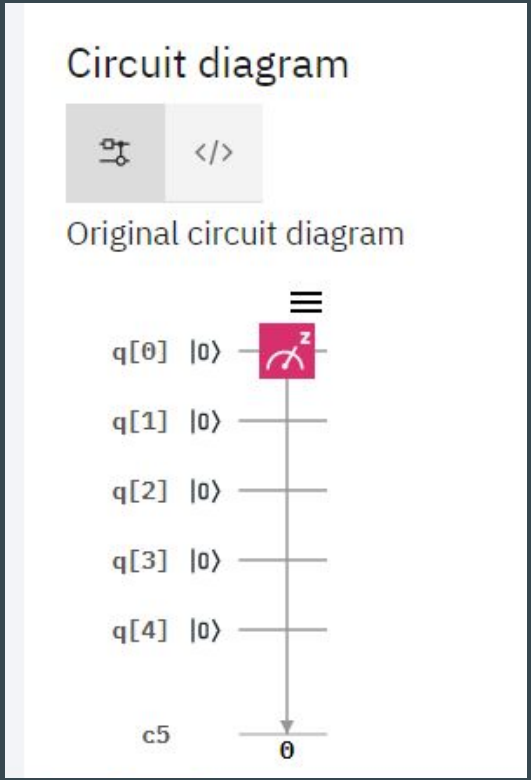
If You are between 14 and 17 years old, then Your parent or legal guardian must accept this Agreement for You. By doing so, Your parent or legal guardian represents and warrants s/he has the legal right to do so, understands that this is a binding legal agreement, and accepts these terms on Your behalf.

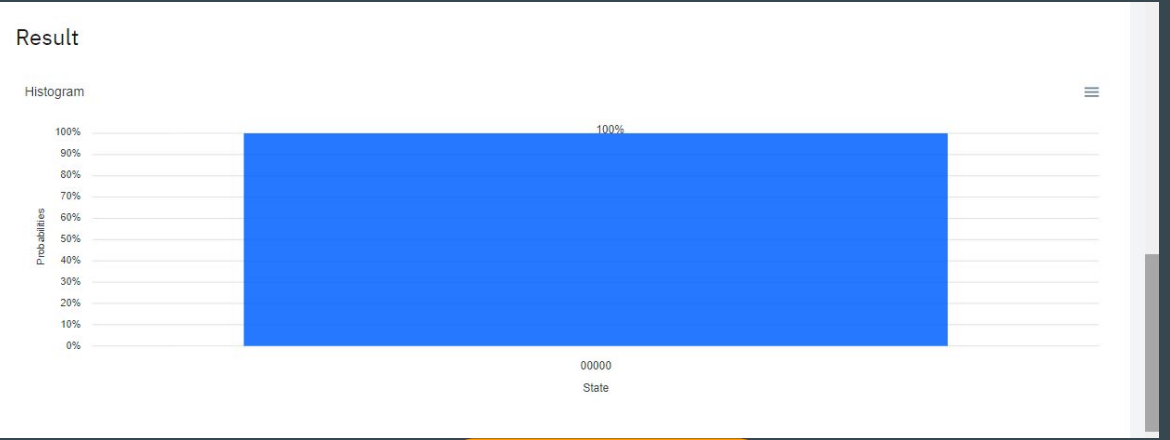
This Agreement replaces all prior versions of the "IBM Q End User License Agreement" and the "IBM Q Experience End User License Agreement" and any previously linked "Terms of Use" or "Acceptable Use Policy".

1. General. This Agreement and Your license (described below) governs Your use of IBM Q. You may withdraw from this program at any time by deleting Your IBM Q account from your account profile page. IBM may withdraw this program via termination of Your account. IBM may change the terms of this Agreement at any time. If You do not agree to the updated terms, please do not further access IBM Q.

1.1 ***Applicability.*** If Your access and use of IBM Q is governed and authorized by the terms of a Governing Contract with IBM that states that it replaces the terms of this Agreement, such as

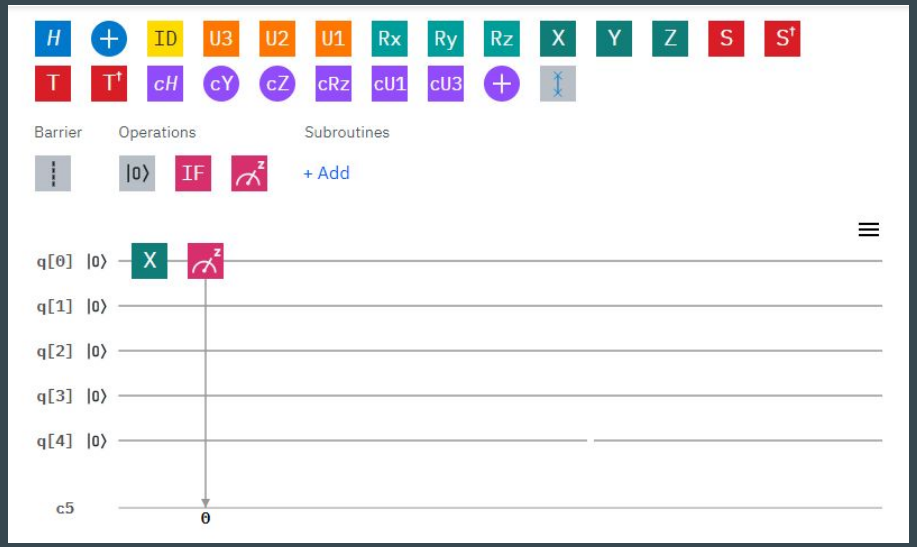
IBM Q Experience Demo 1

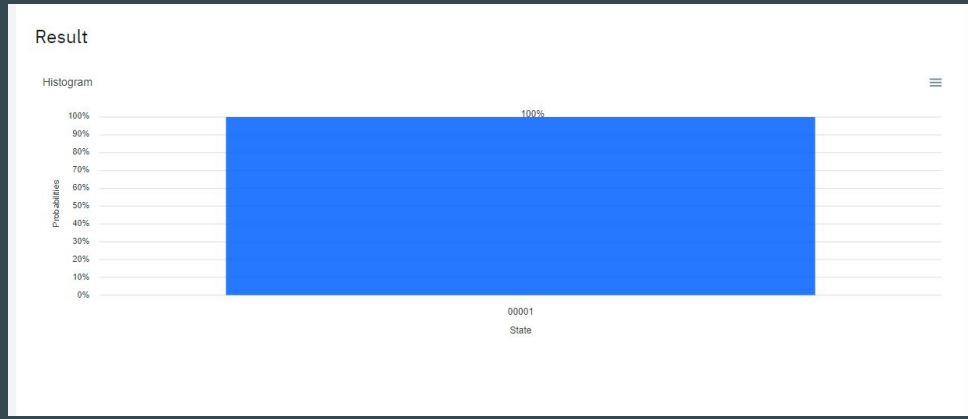




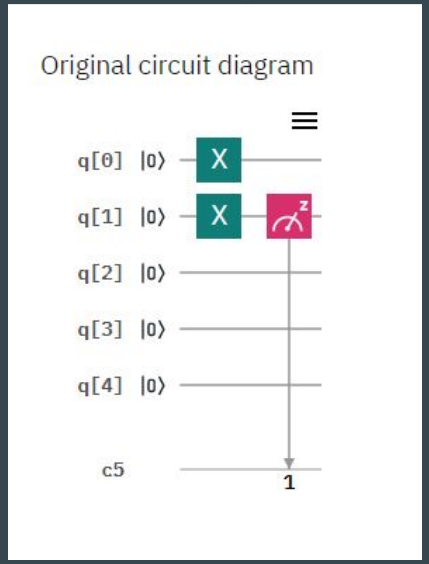
5 qubits (00000)
measurement 1

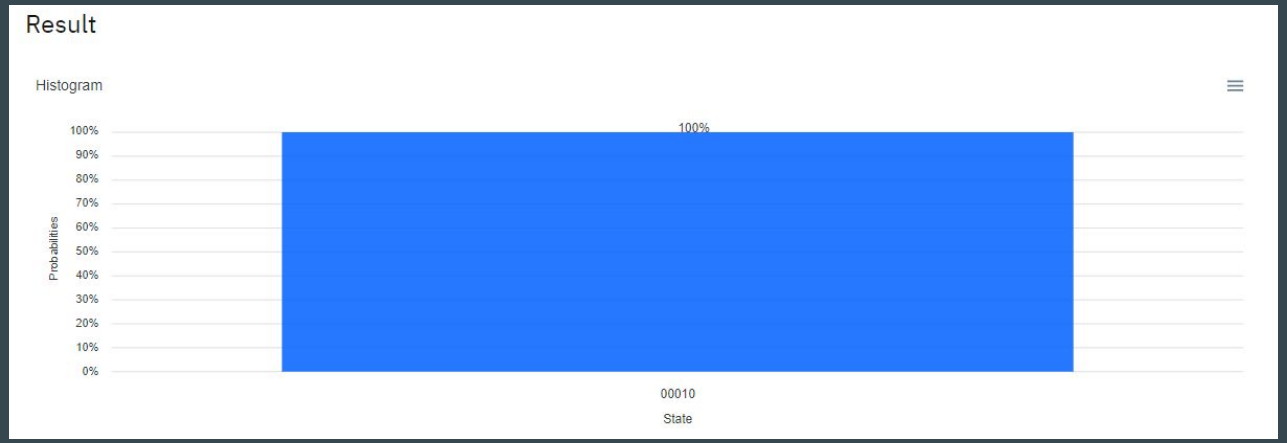
IBM Q Experience Pauli Gate Demo



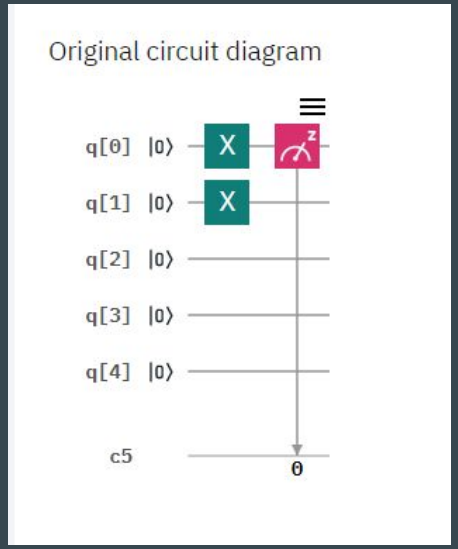


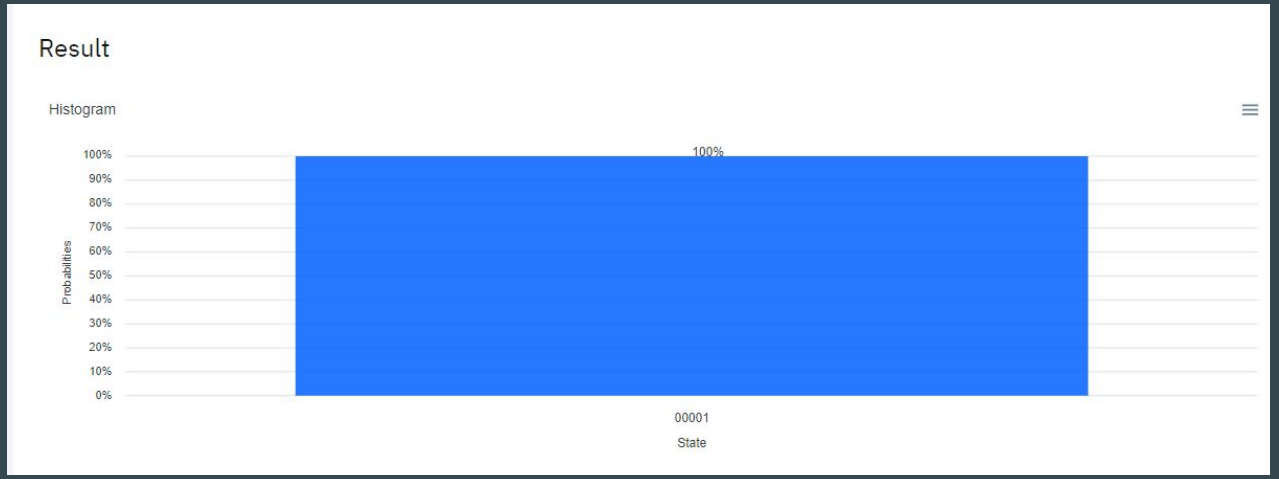
IBM Q Experience Pauli Gate Demo



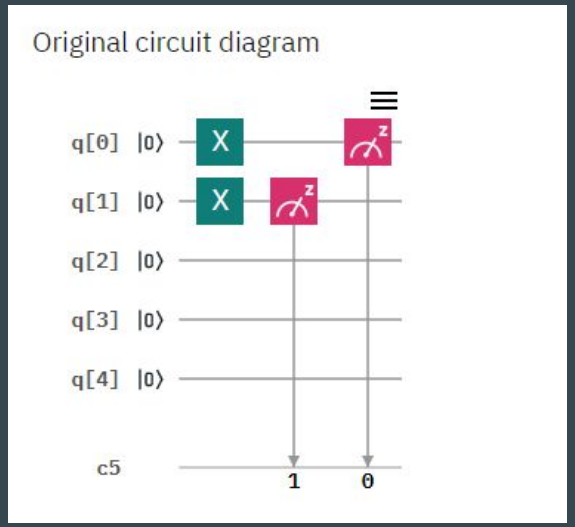


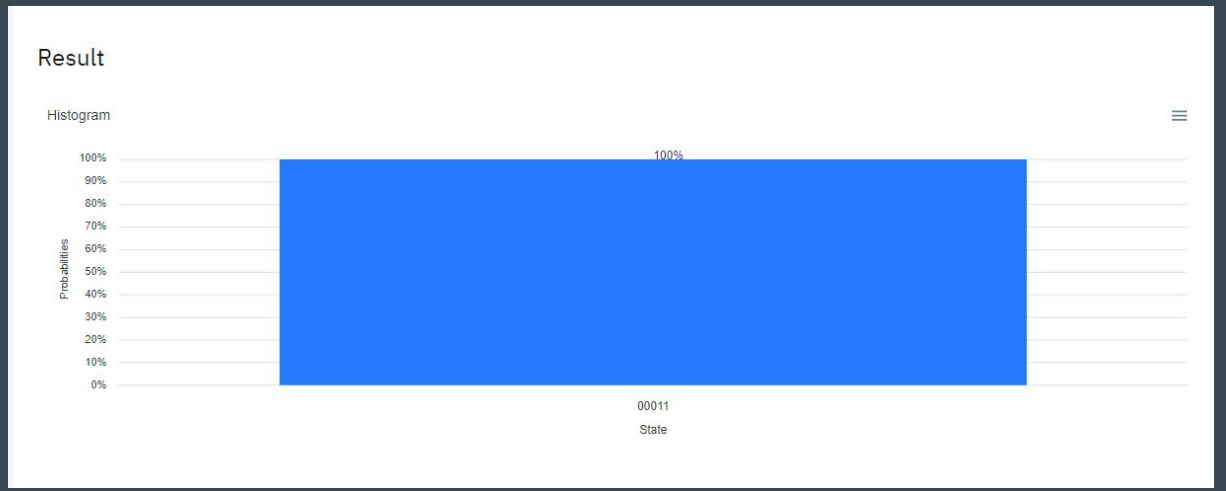
IBM Q Experience Pauli Gate Demo



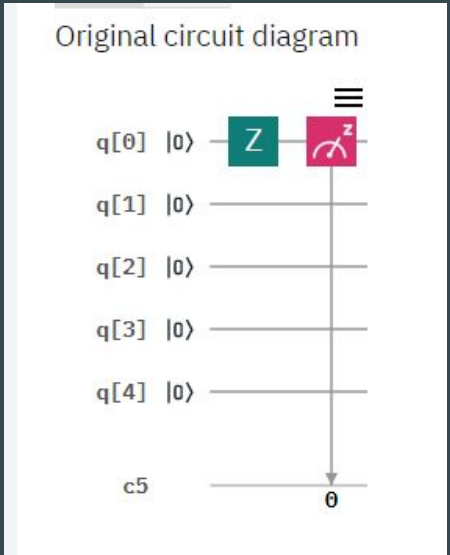


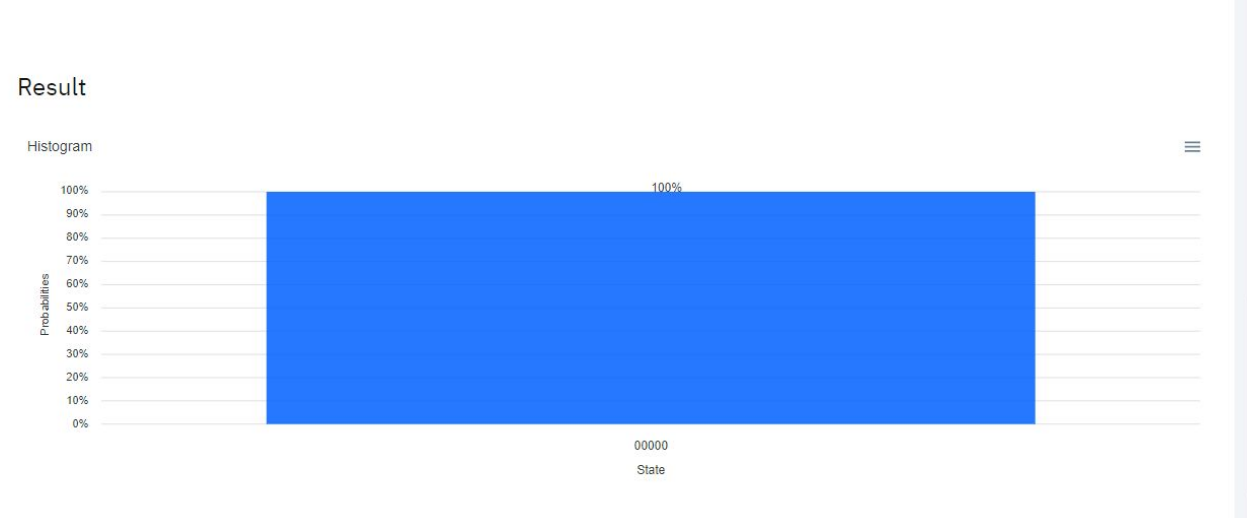
IBM Q Experience Pauli Gate Demo



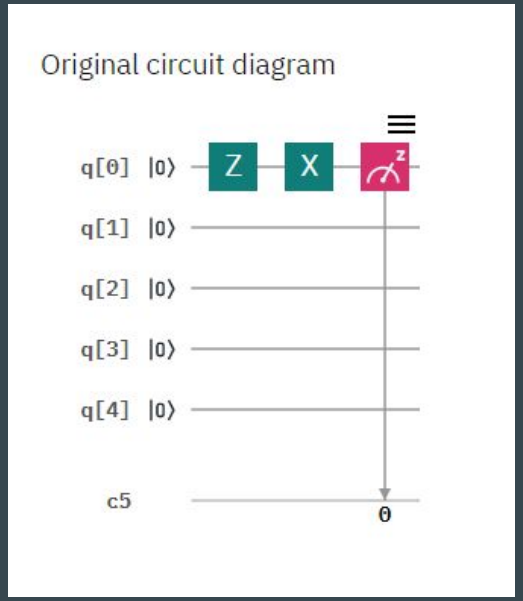


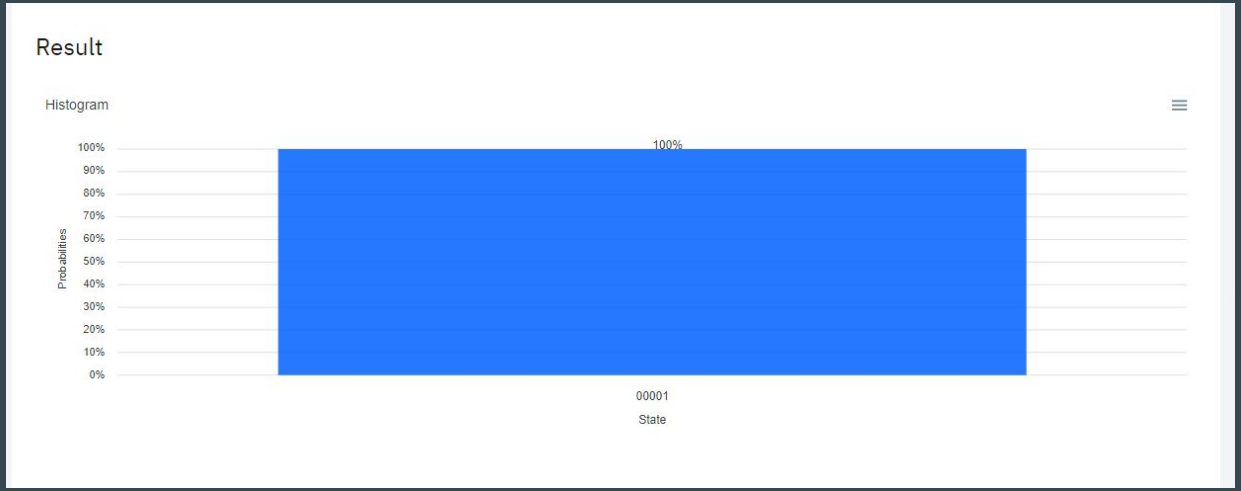
IBM Q Experience Pauli Gate Demo



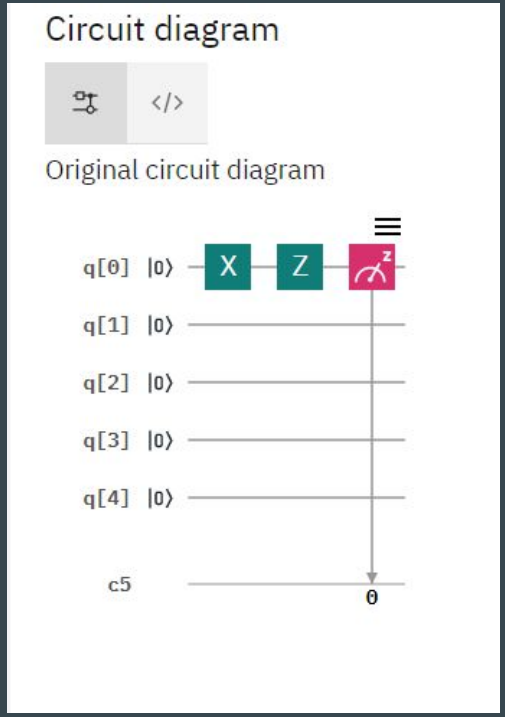


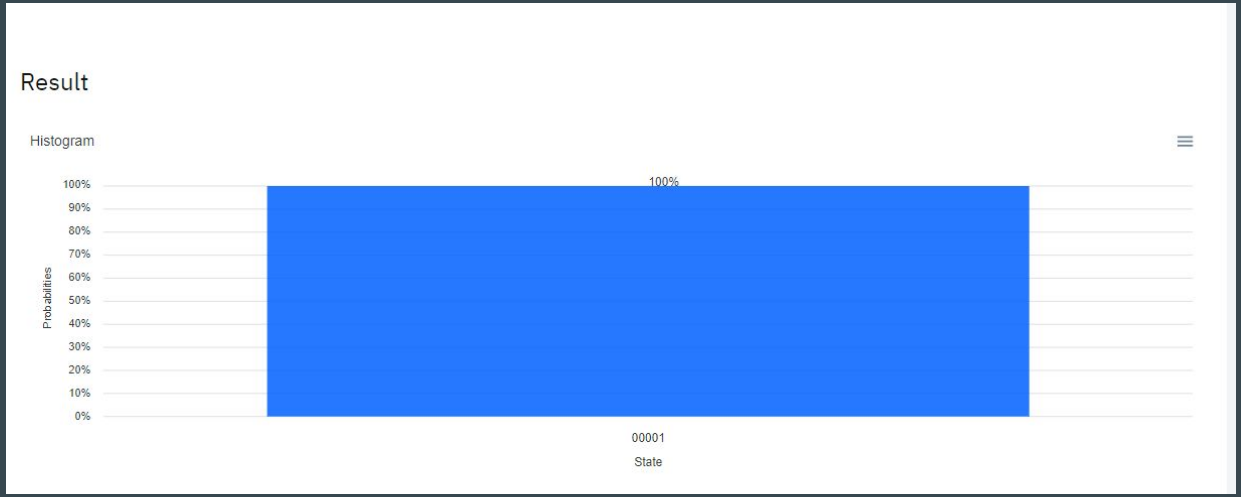
IBM Q Experience Pauli Gate Demo





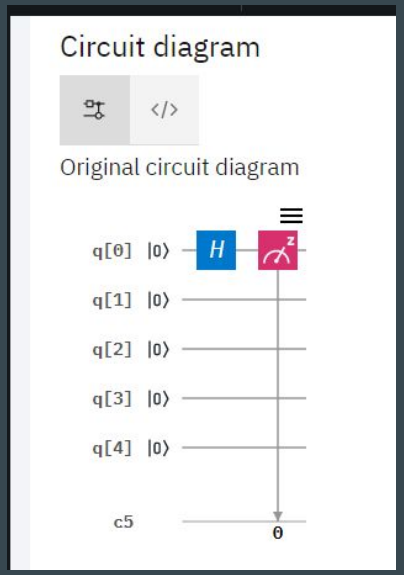
IBM Q Experience Pauli Gate Demo

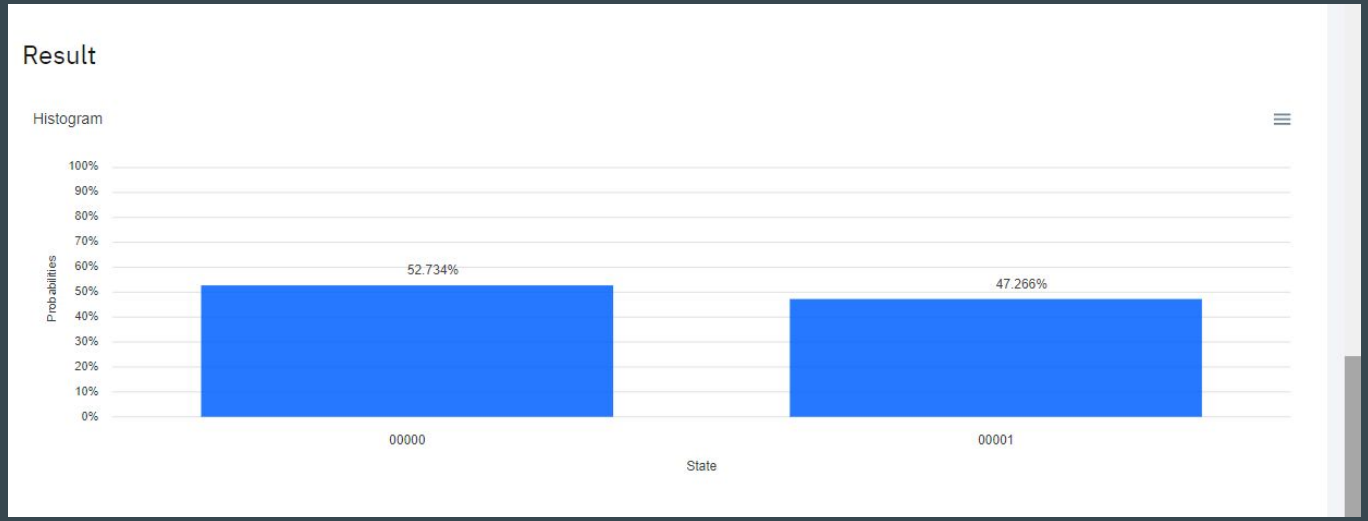




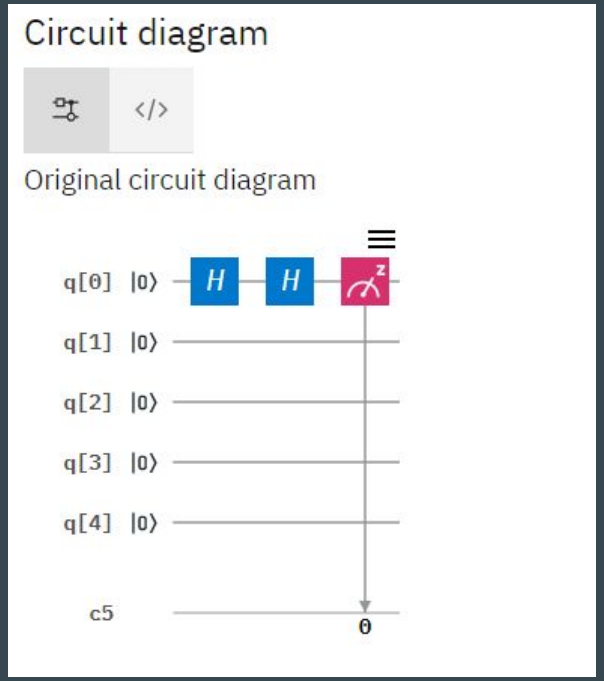
where is there no -1?

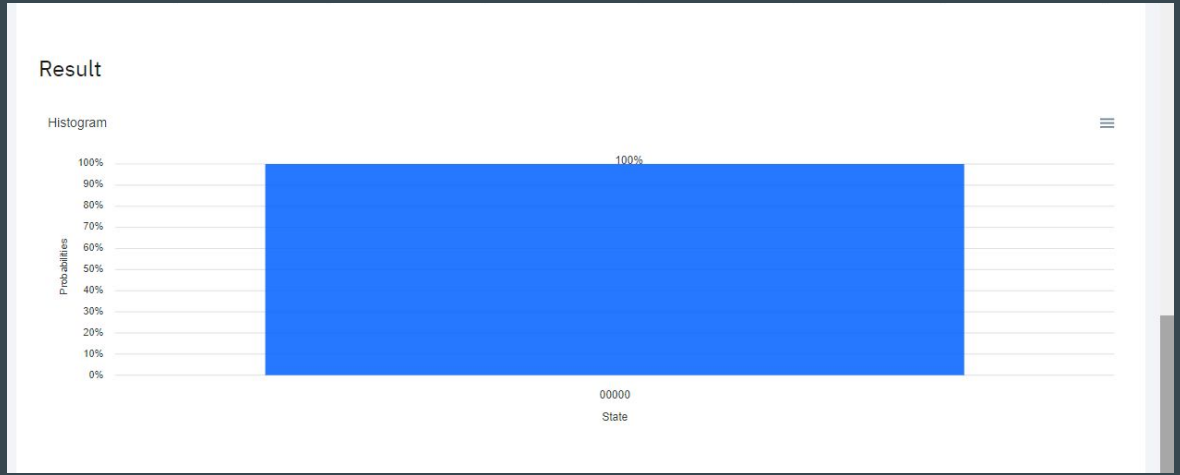
IBM Q Experience Hadamard Gates Demo



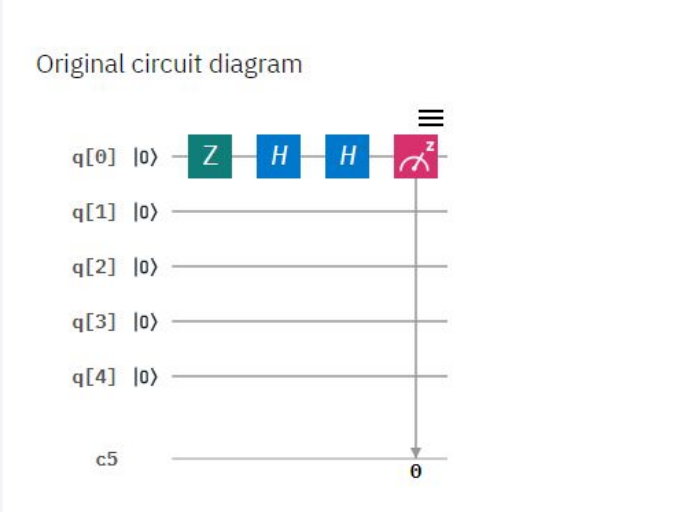


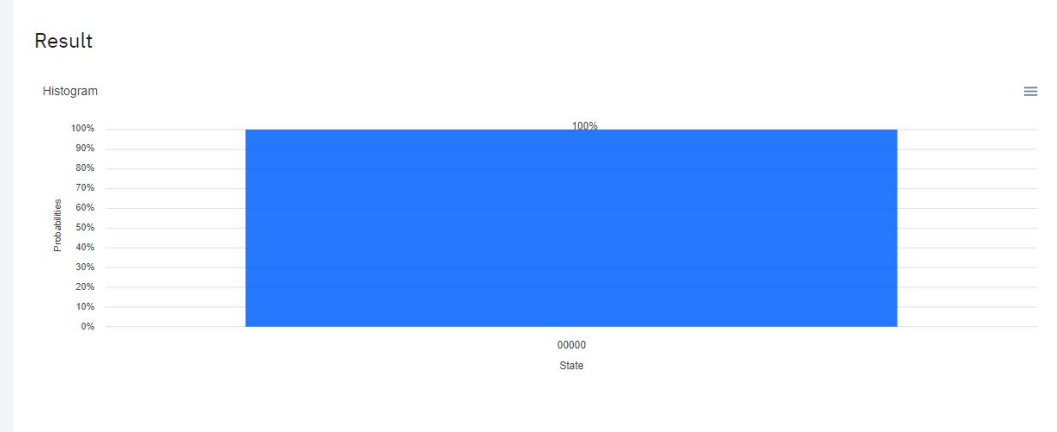
IBM Q Experience Hadamard Gates Demo



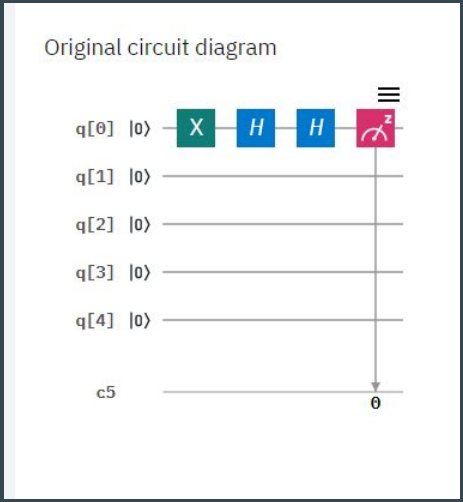


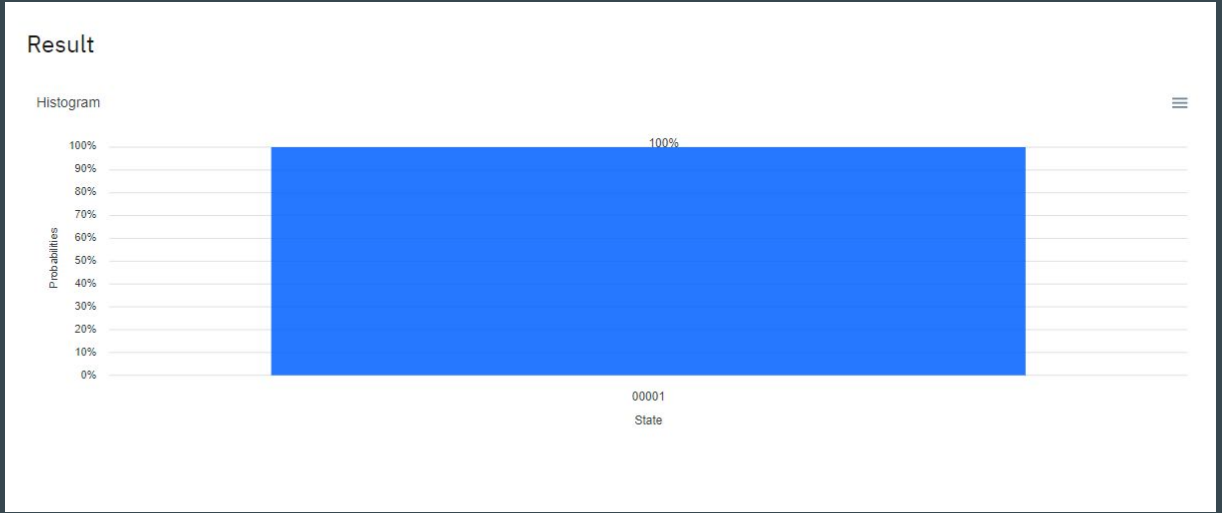
IBM Q Experience Hadamard Gates Demo



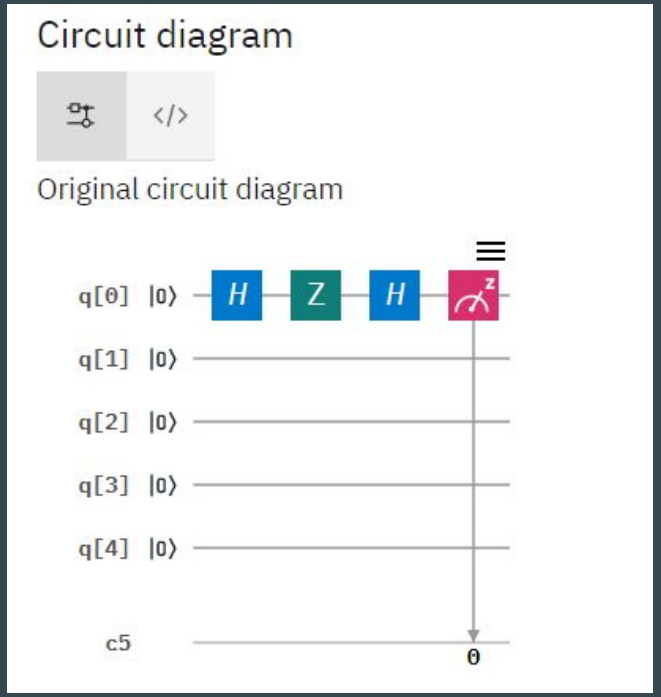


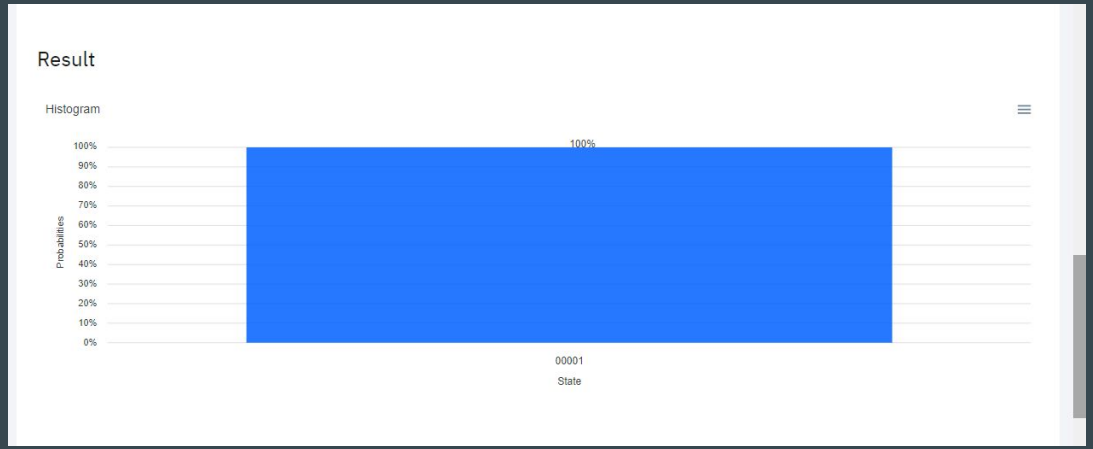
IBM Q Experience Hadamard Gates Demo



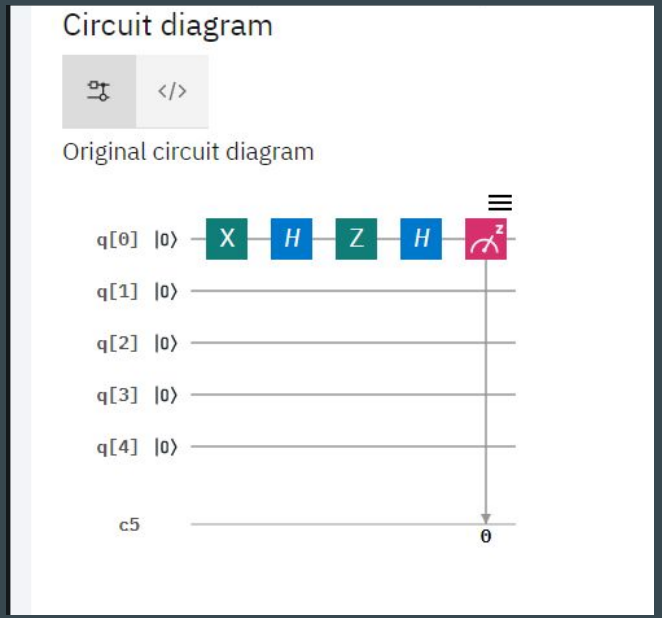


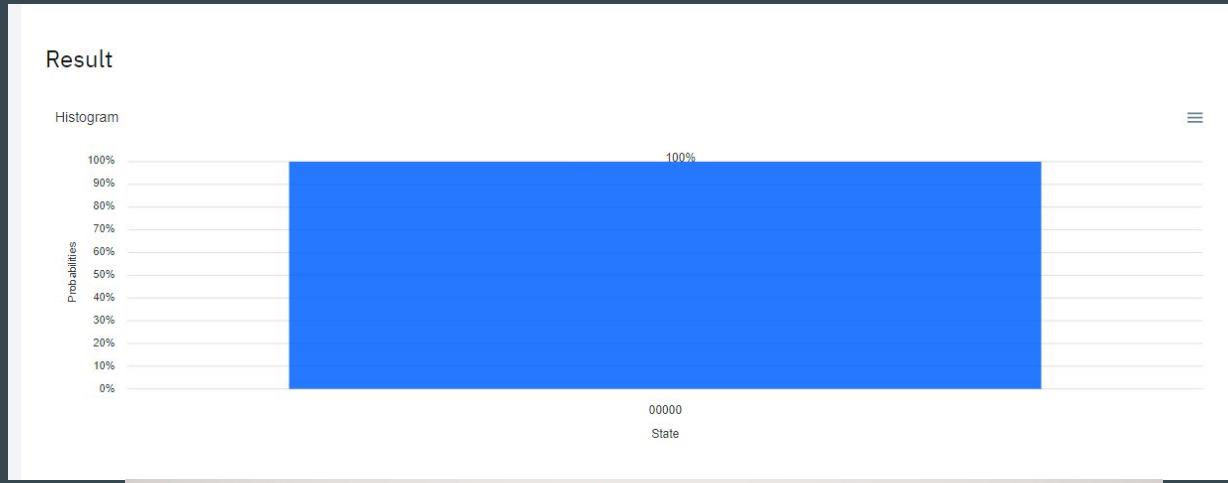
IBM Q Experience Pauli Gate + Hadamard Gates Demo

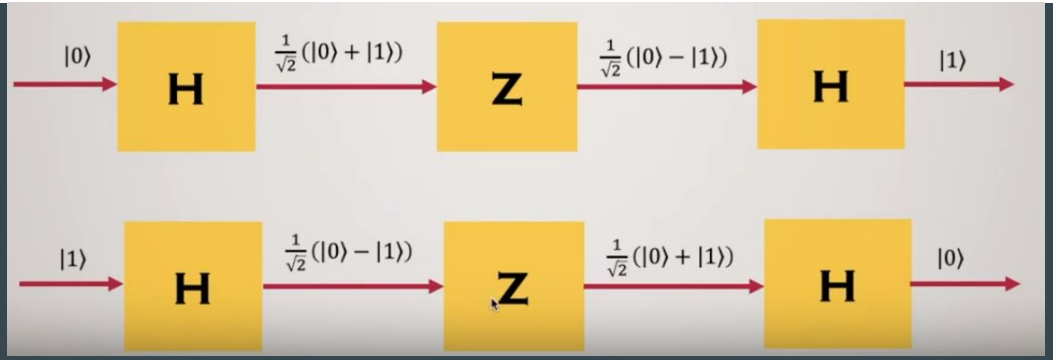




IBM Q Experience Pauli Gate + Hadamard Gates Demo







Python Code

Python Code Explained

QuantCircuit.py