Qiskit

Quantum Computing from Theory to Practice

Chapter 1

Topics

- Quantum Platform and Circuit Composer
    - ✓ Bell
    - ✓ GHZ
    - ✓ Toffoli
- Quantum Lab with Qiskit
    - ✓ Hello World
    - ✓ Simulators and Real Devices
    - ✓ Mystery Circuit
    - ✓ Quantum Teleportation
    - ✓ Bloch Sphere

IBM Quantum Platform and Circuit Composer

- Start here: https://quantum.ibm.com/
- Compute Resources
- API Token
- Learning > Explore Courses and Tutorials
- IBM Quantum Composer
  - ✓ Play with Quantum Gates
  - ✓ Bell State (4 variants)
  - ✓ GHZ State
  - ✓ Toffoli Gate
  - ✓ Show results on real device

Topics

- Quantum Platform and Circuit Composer
  - ✓ Bell
  - ✓ GHZ
  - ✓ Toffoli
- Quantum Lab with Qiskit
  - ✓ Hello World
  - ✓ Simulators and Real Devices
  - ✓ Mystery Circuit
  - ✓ Quantum Teleportation
  - ✓ Bloch Sphere

Hello World

https://www.qbraid.com/

Exercise

- Start from qBraid
- File > New Notebook
- Together with the instructor, create a Jupyter Notebook from Scratch: **“Hello World”**

Mystery Circuit

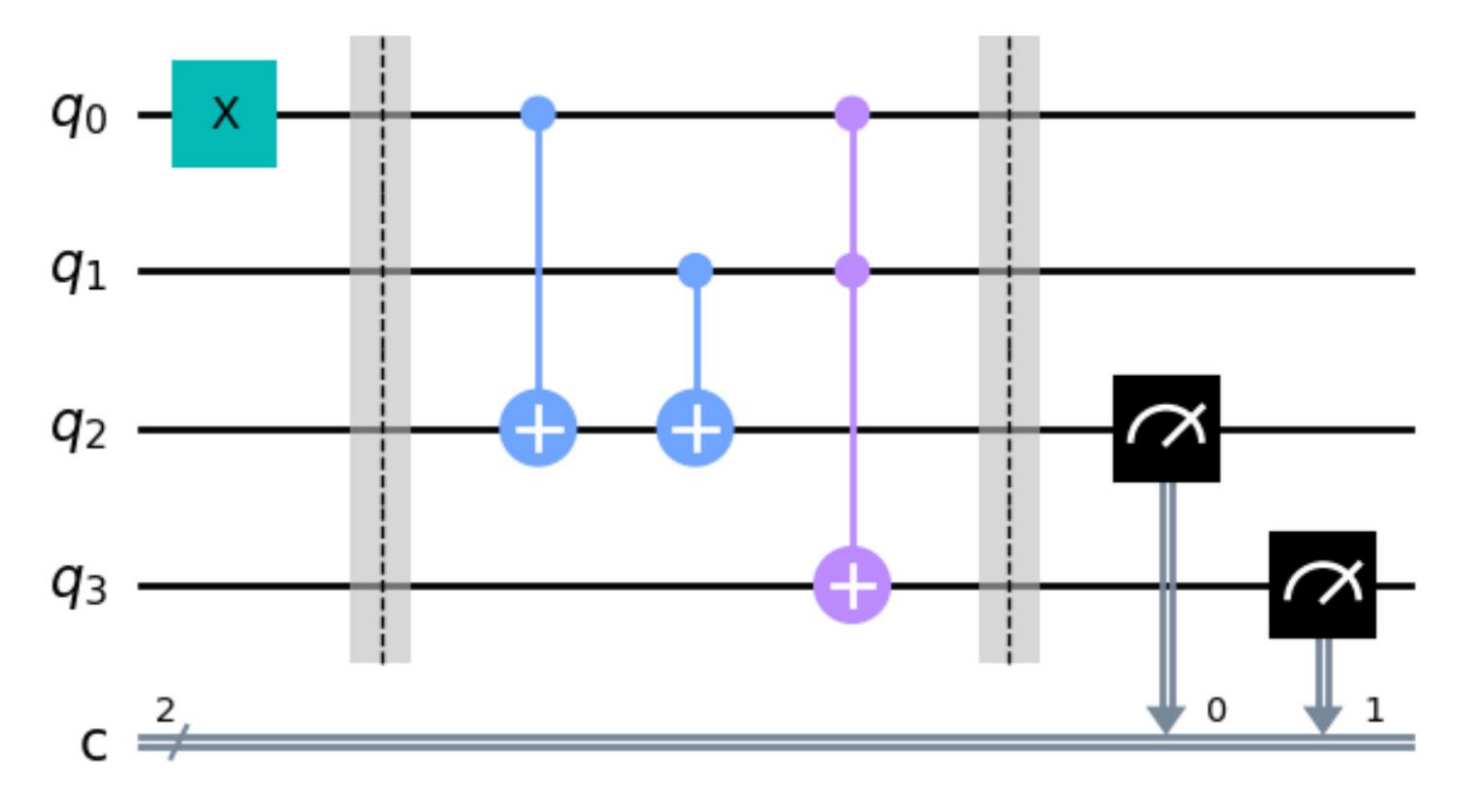
qBraid

https://www.qbraid.com/

Develop this Quantum Circuit starting from a copy of the result of Exercise 1.
Try out with different Initialization Gates I and X on q0 and q1.
How would you name this Circuit?

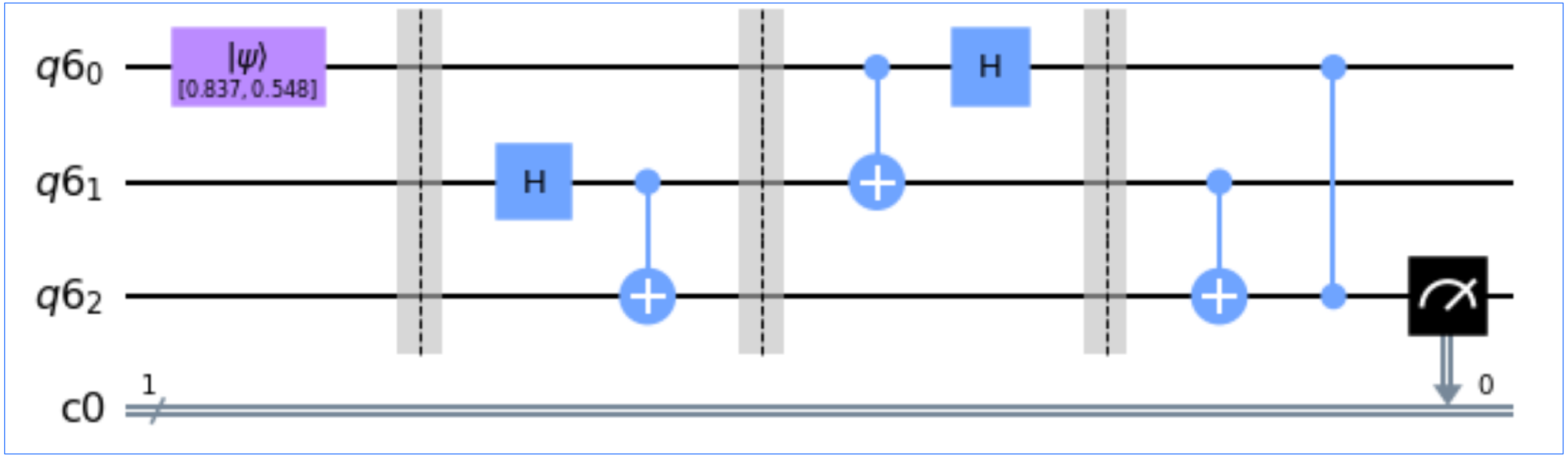
In other words:

- $I \otimes I\, |00\rangle$
- $\boldsymbol{I \otimes X\, |00\rangle}$
- $X \otimes I\, |00\rangle$
- $X \otimes X\, |00\rangle$

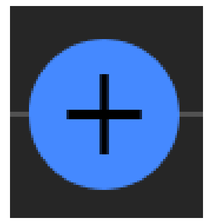


Quantum Teleportation

- ***No-Cloning Theorem**: it is impossible to create an independent and identical copy of an arbitrary unknown quantum state*
- How to transfer a Quantum State from Qubit 0 (Alice) to Qubit 2 (Bob), using Qubit 1 (Auxiliary or Spock)?
- This is the Quantum Circuit that allows to do so and that we are going to program in Qiskit (by completing a Jupyter Notebook)



Pauli Matrices (2 out of 4)

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

BIT FLIP

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

SIGN FLIP

Quantum Teleportation Protocol

Bell State ("EPR pair")

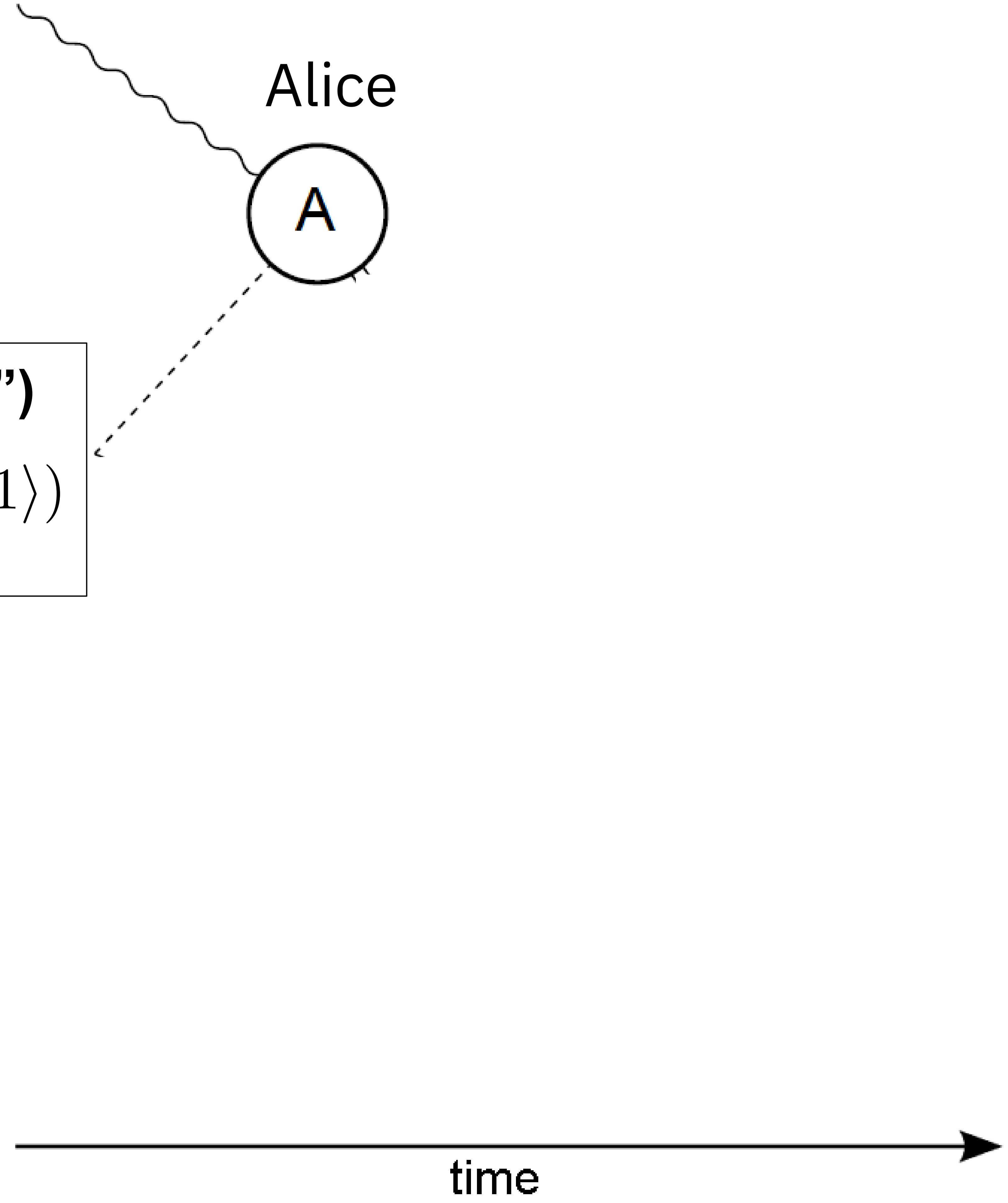
$$\left|\Phi^{+}\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

time

Source: https://en.wikipedia.org/wiki/Quantum_teleportation

Quantum Teleportation Protocol

Photon with quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (state to be *teleported*)

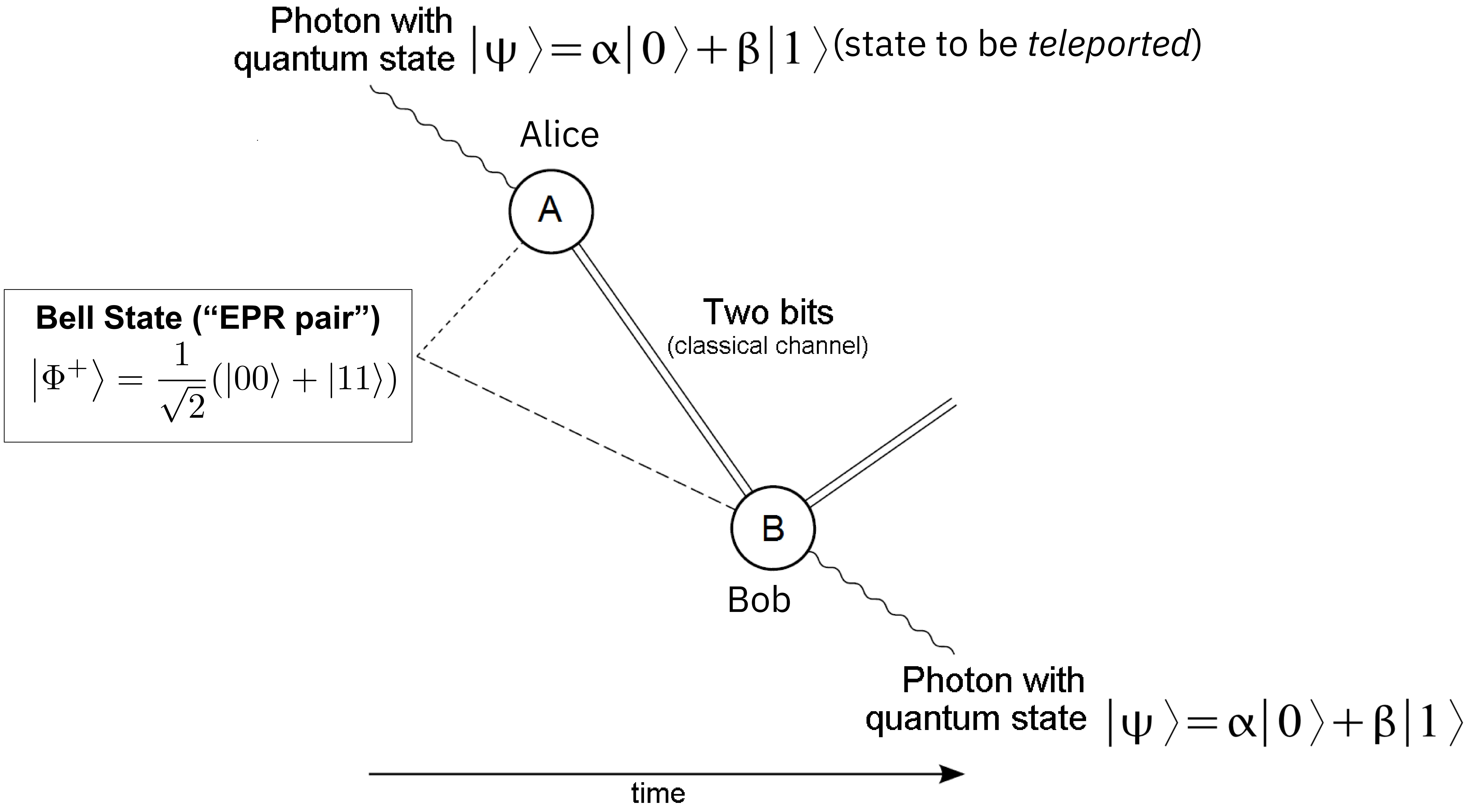


Bell State ("EPR pair")

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

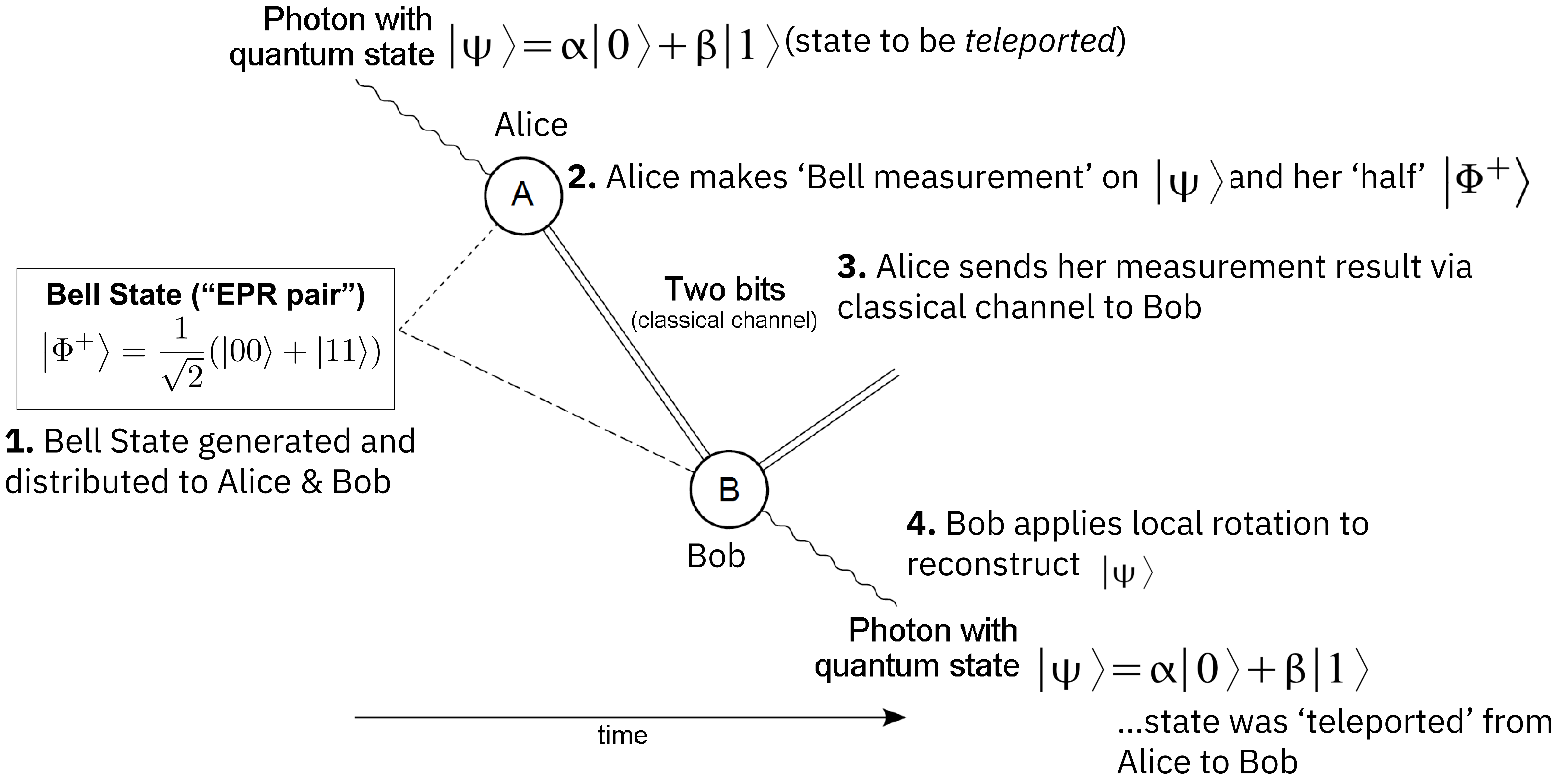
Source: https://en.wikipedia.org/wiki/Quantum_teleportation

Quantum Teleportation Protocol



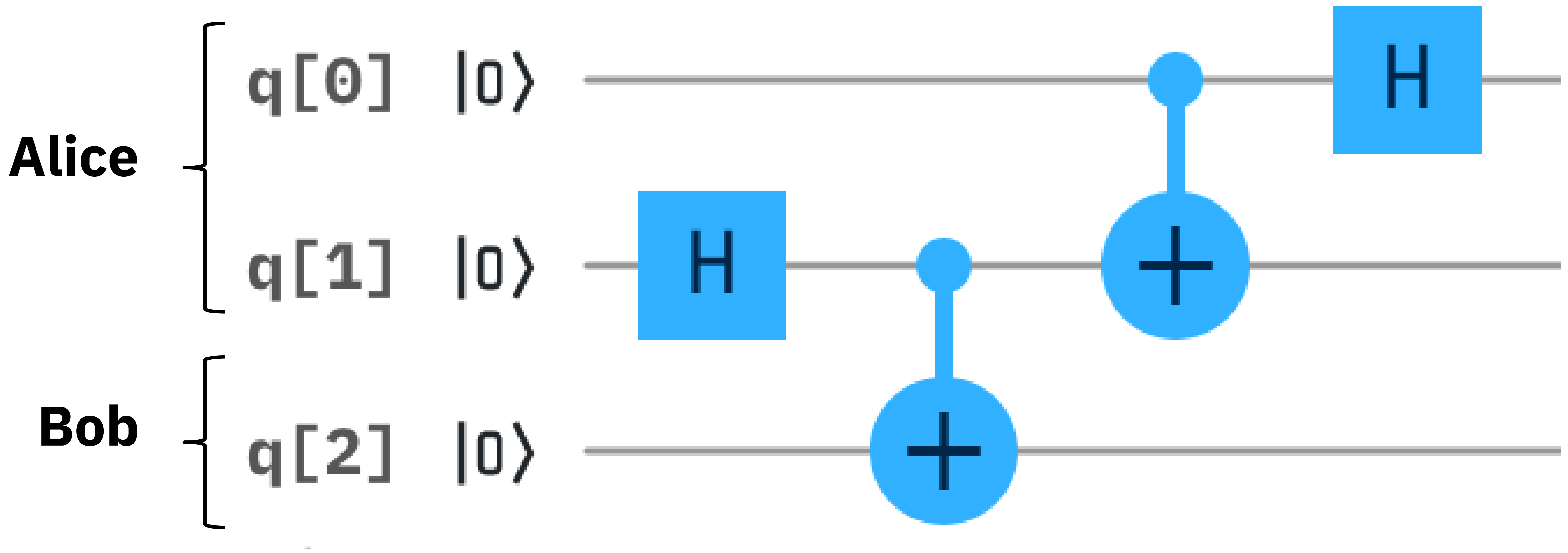
Source: https://en.wikipedia.org/wiki/Quantum_teleportation

Quantum Teleportation Protocol

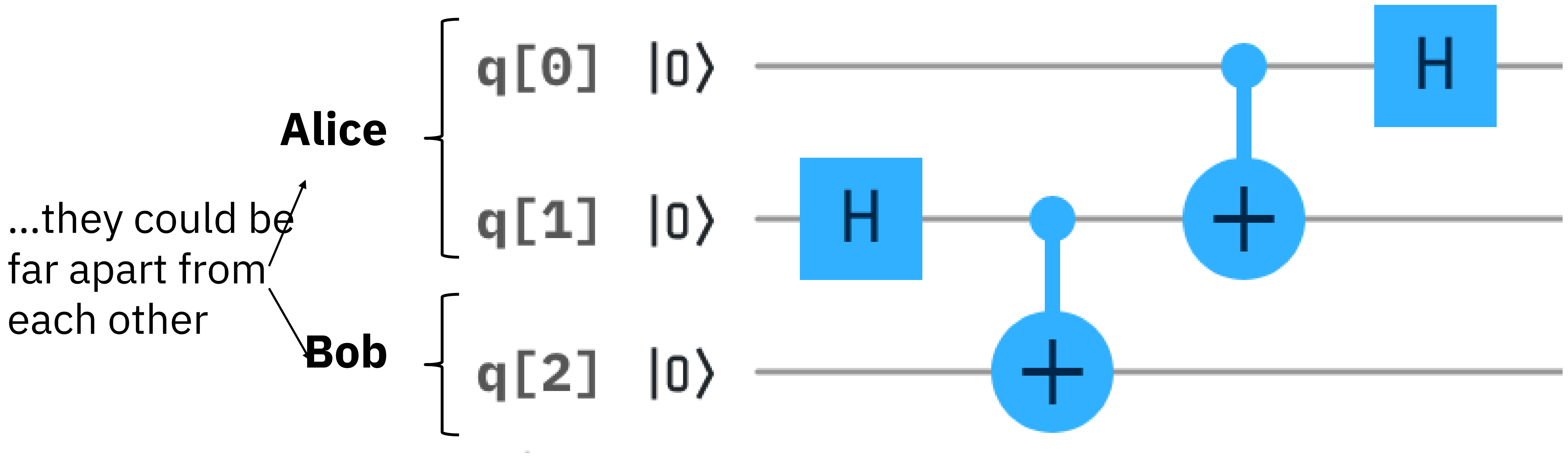


Source: https://en.wikipedia.org/wiki/Quantum_teleportation

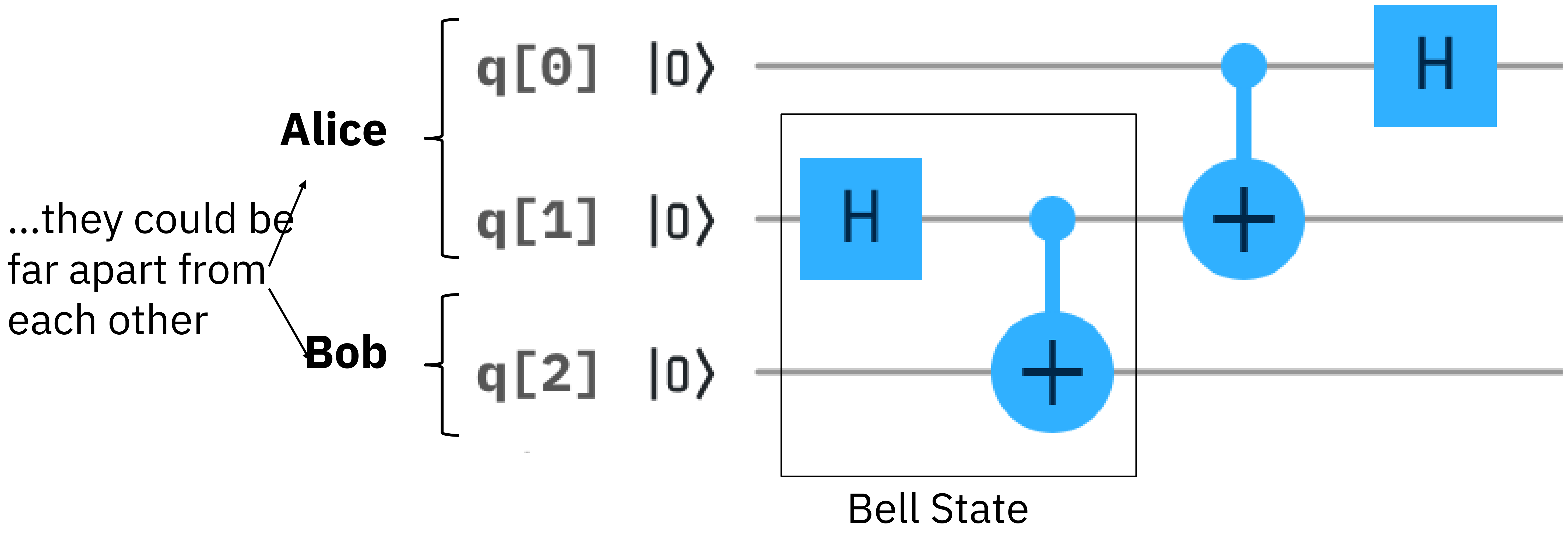
Preparing Quantum Teleportation Protocol: 3 Qubit Circuit with Bell measurement



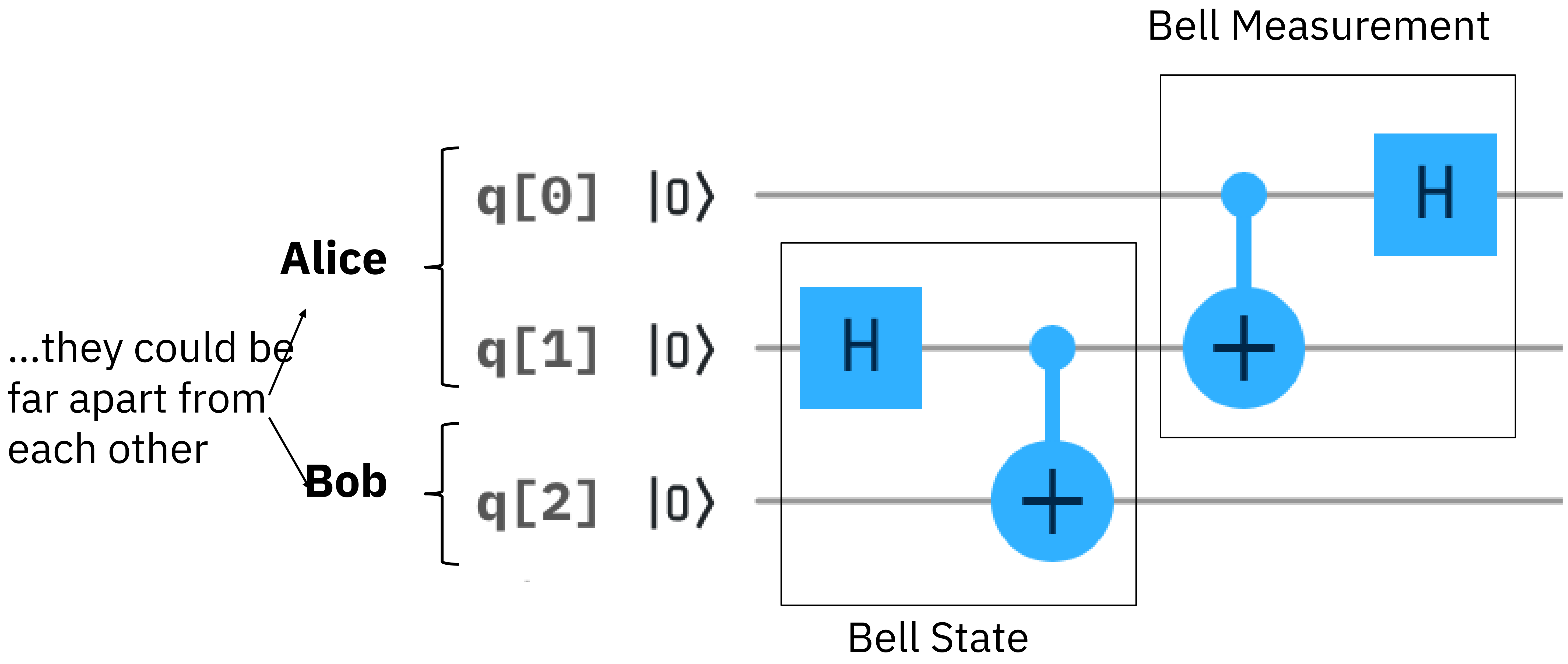
Preparing Quantum Teleportation Protocol:
3 qubit circuit with Bell measurement



Preparing Quantum Teleportation Protocol: 3 qubit circuit with Bell measurement

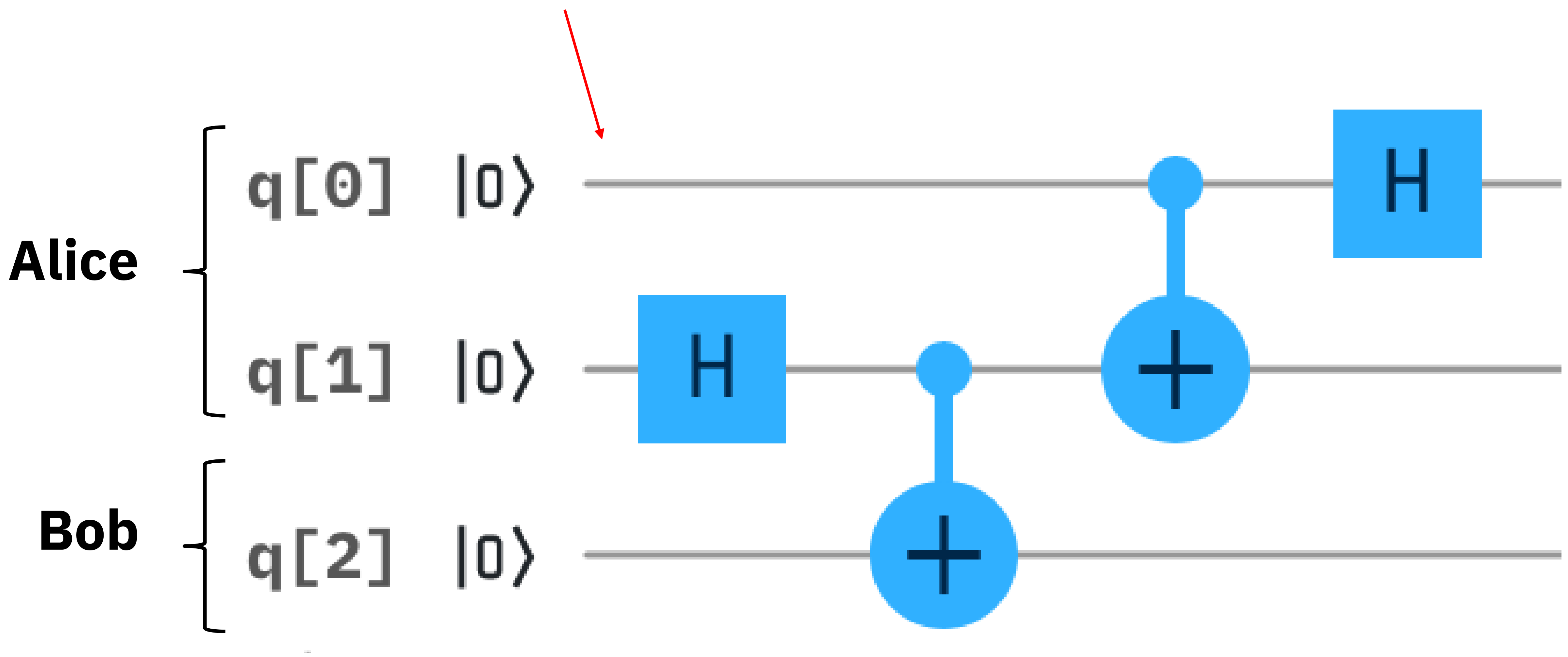


Preparing Quantum Teleportation Protocol: 3 qubit circuit with Bell measurement



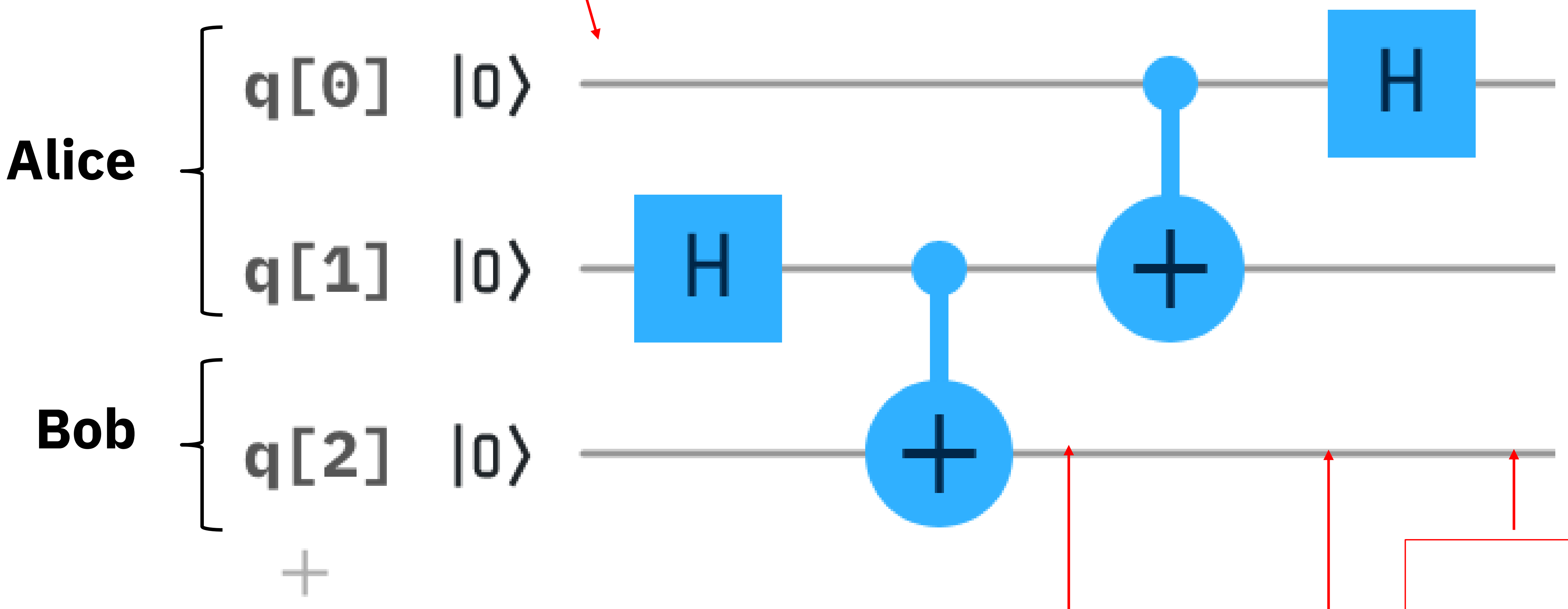
Preparing Quantum Teleportation Protocol: 3 qubit circuit with Bell measurement

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (Alice's State to be teleported)



Preparing Quantum Teleportation Protocol: 3 Qubit Circuit with Bell measurement

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$ (Alice's State to be teleported)



$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle)$$

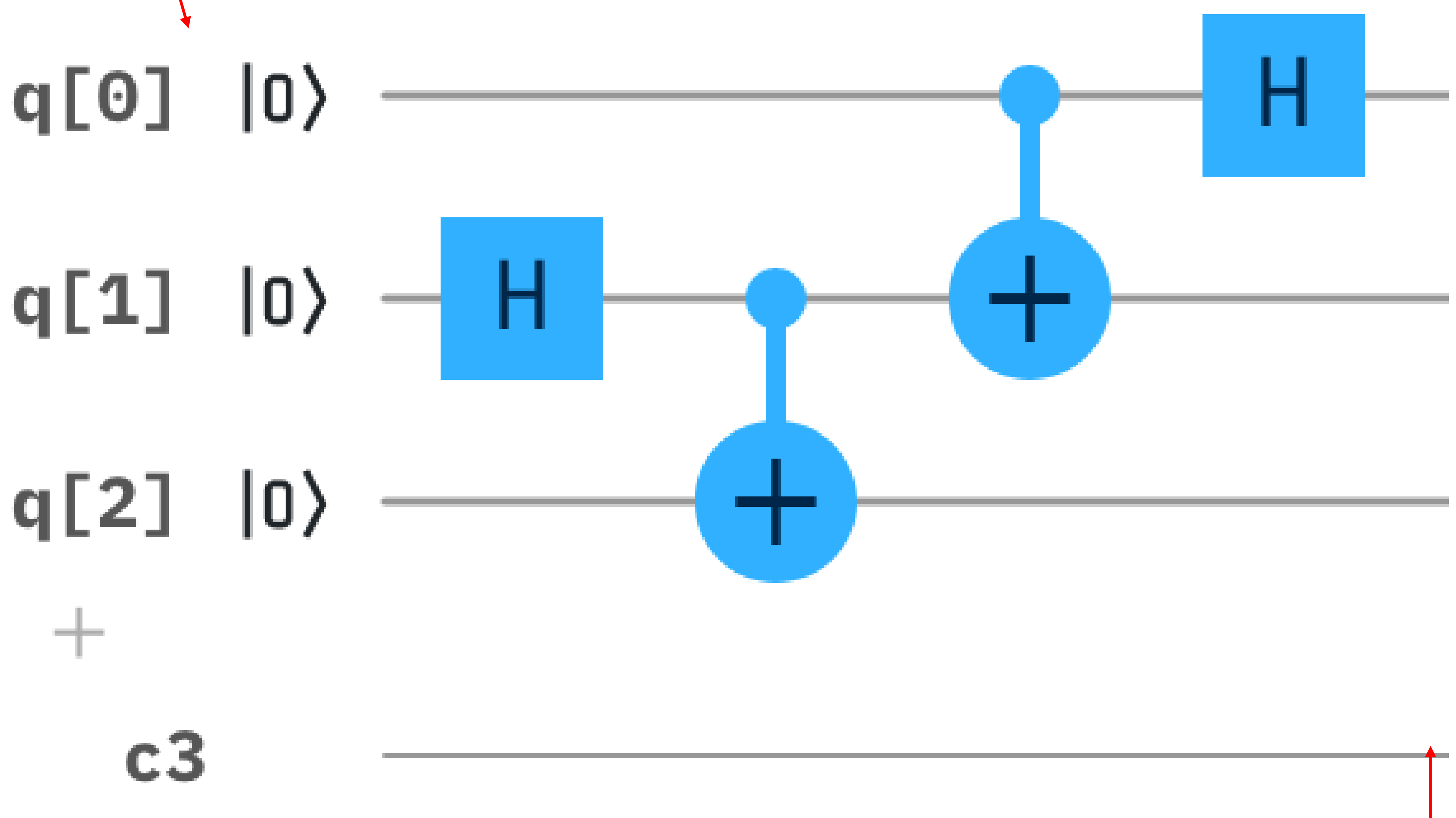
$$= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|110\rangle + \beta|001\rangle + \beta|111\rangle)$$

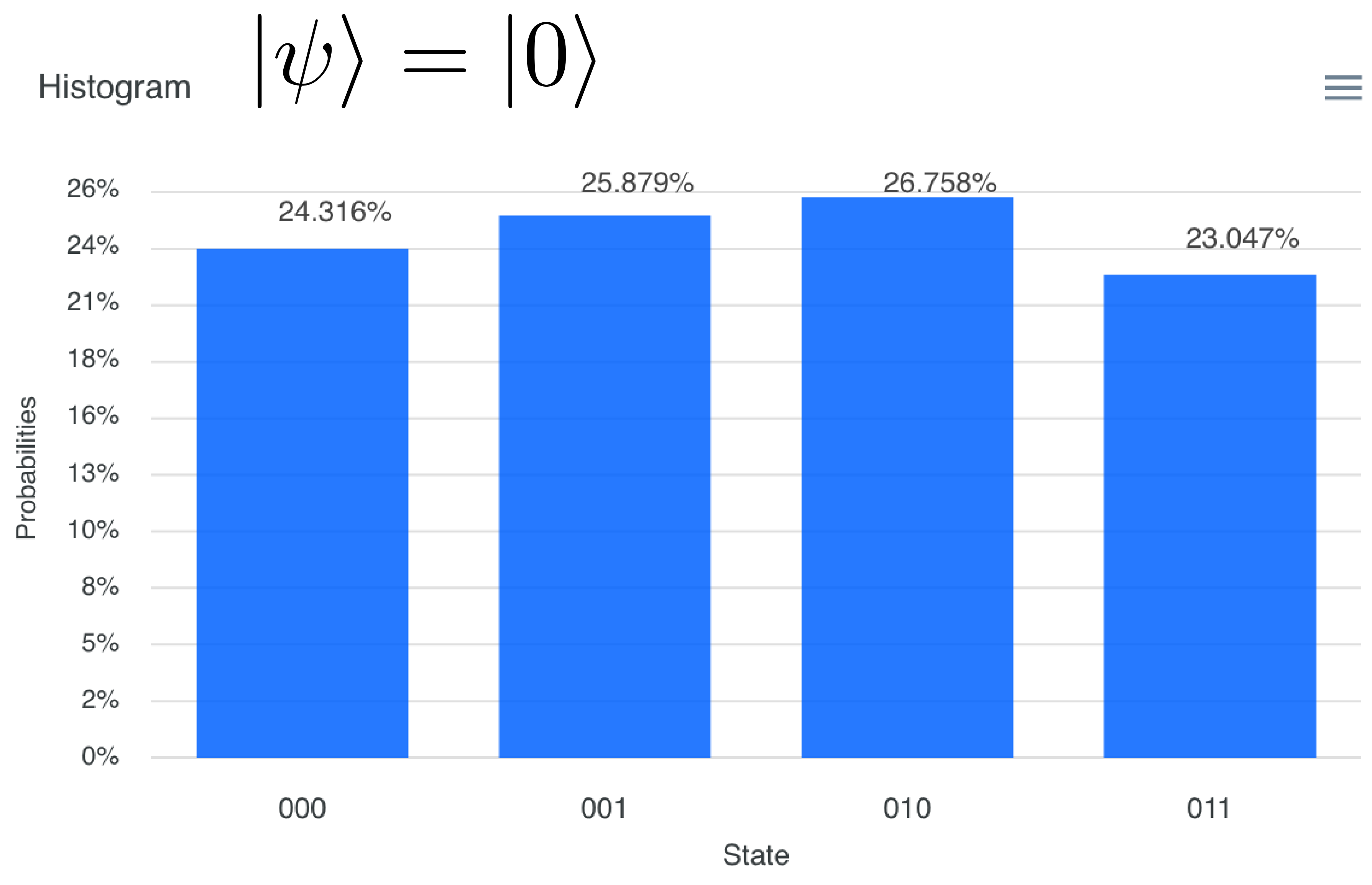
$$= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|110\rangle + \beta|011\rangle + \beta|101\rangle)$$

$$= \frac{1}{2}\begin{pmatrix} \alpha|000\rangle + \alpha|001\rangle + \alpha|110\rangle + \alpha|111\rangle \\ \beta|010\rangle - \beta|011\rangle + \beta|100\rangle - \beta|101\rangle \end{pmatrix}$$

3 Qubit Circuit with Bell measurement

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$ (Alice's State to be teleported)





$$= \frac{1}{2}\left(\boldsymbol{\alpha}\,|\mathbf{000}> + \alpha\,|001> + \alpha\,|110> + \alpha\,|111> + \beta|010> - \beta|011> + \boldsymbol{\beta}\,|\mathbf{100}> - \beta\,|101> \right)$$

$$= \frac{1}{2}\ (\alpha\,|0> + \beta\,|1>)|00> + (\alpha\,|0> - \beta\,|1>)|01> + (\alpha\,|1> + \beta\,|0>)|10> + (\alpha\,|1> - \beta\,|0>)|11>)$$

Bob receives bits from Alice and Applies Appropriate Gate

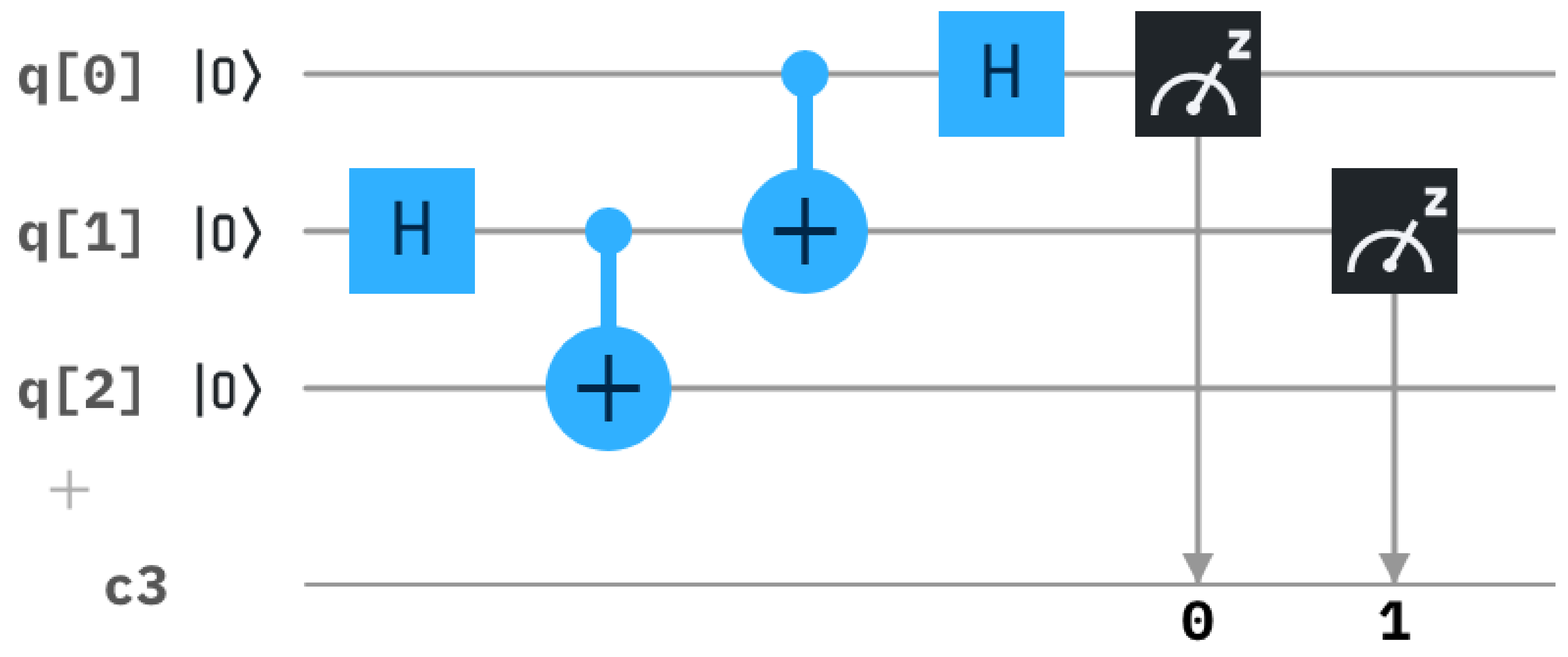
$$= \frac{1}{2}\ (\alpha\,|0> + \beta\,|1>)|00> \ + (\alpha\,|0> - \beta\,|1>)|01> + (\alpha\,|1> + \beta\,|0>)|10> + (\alpha\,|1> - \beta\,|0>)|11>)$$

Bob receives the following Qubits:

$$(\alpha\,|0> + \beta\,|1>)$$
$$(\alpha\,|0> - \beta\,|1>)$$
$$(\alpha\,|1> + \beta\,|0>)$$
$$(\alpha\,|1> - \beta\,|0>)$$

Bits Received	→	Apply Gate
00	→	I
01	→	Z
10	→	X
11	→	ZX

Measurements

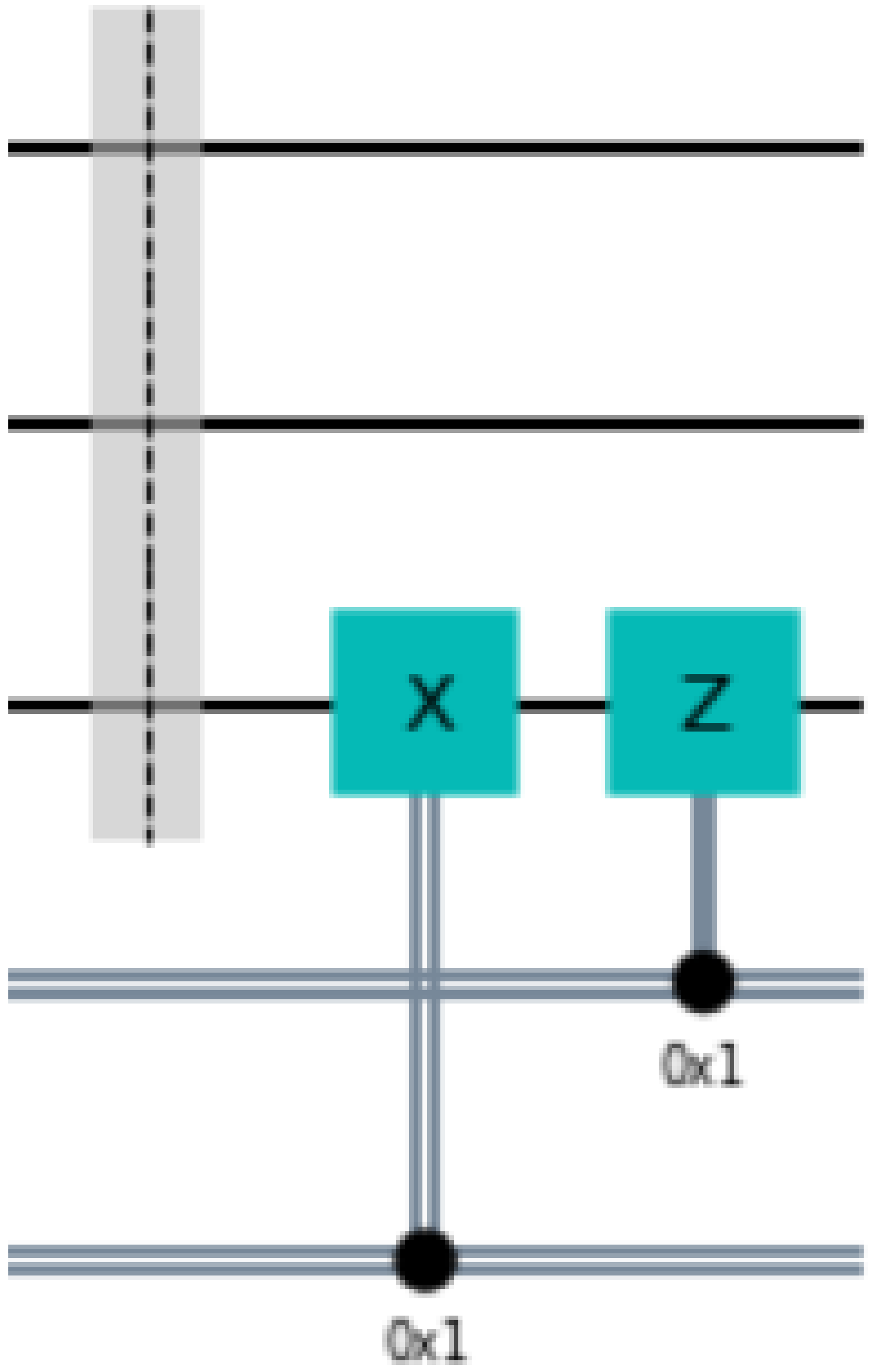


$$\mathrm{I}\ (\alpha\,|0> + \beta\,|1>) = (\alpha\,|0> + \beta\,|1>)$$
$$\mathrm{Z}\ (\alpha\,|0> - \beta\,|1>) = (\alpha\,|0> + \beta\,|1>)$$
$$\mathrm{X}(\alpha\,|1> + \beta\,|0>) = \ (\alpha\,|0> + \beta\,|1>)$$
$$ZX\ (\alpha\,|1> - \beta\,|0>) = (\alpha\,|0> + \beta\,|1>)$$

= Qubit sent by Alice !!!

But how to apply the right gates before measurement?

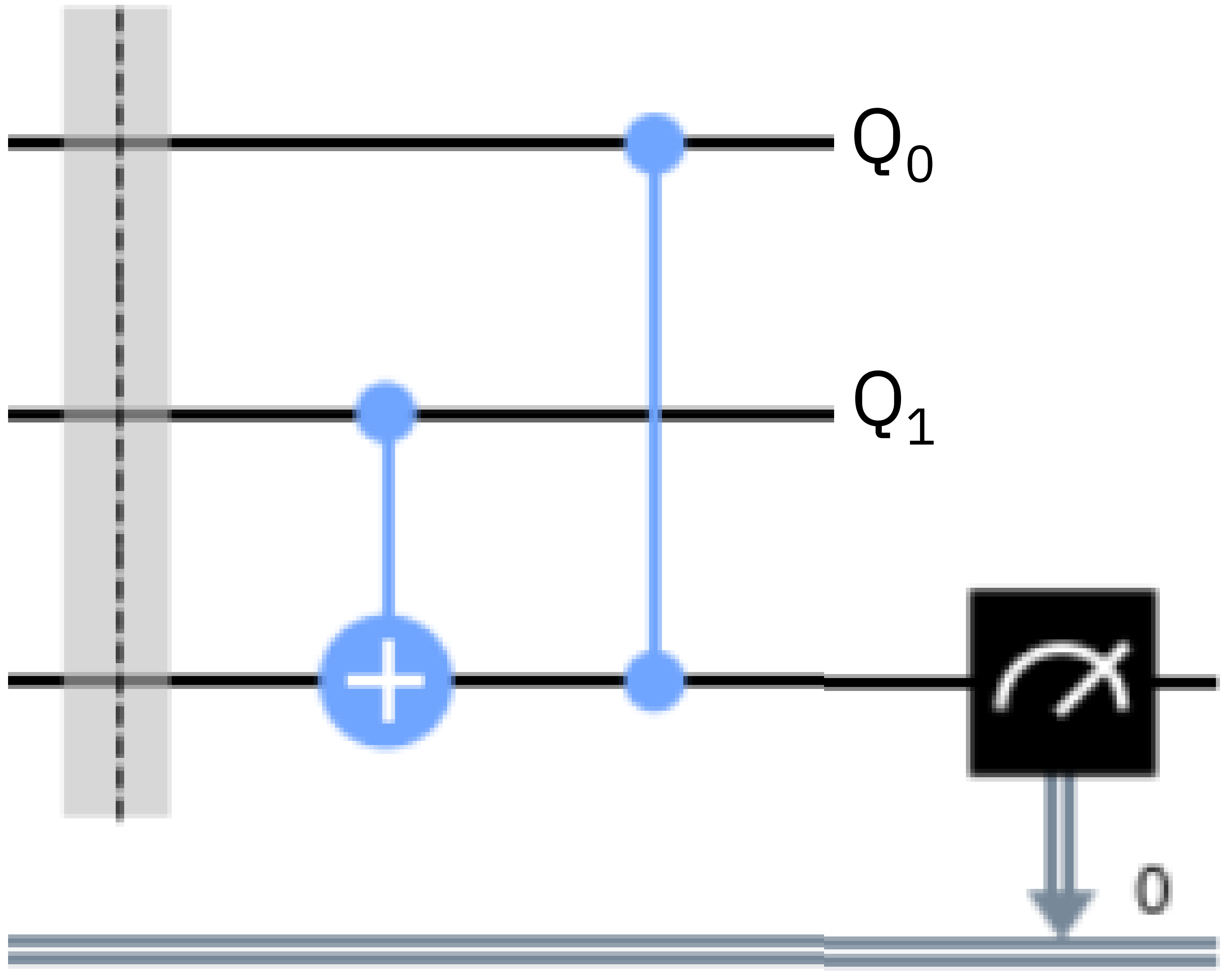
Make use of
- Conditional bit-flip X
- Conditional sign-flip Z



c_0 = Measurement of Q_0

c_1 = Measurement of Q_1

$\sim$



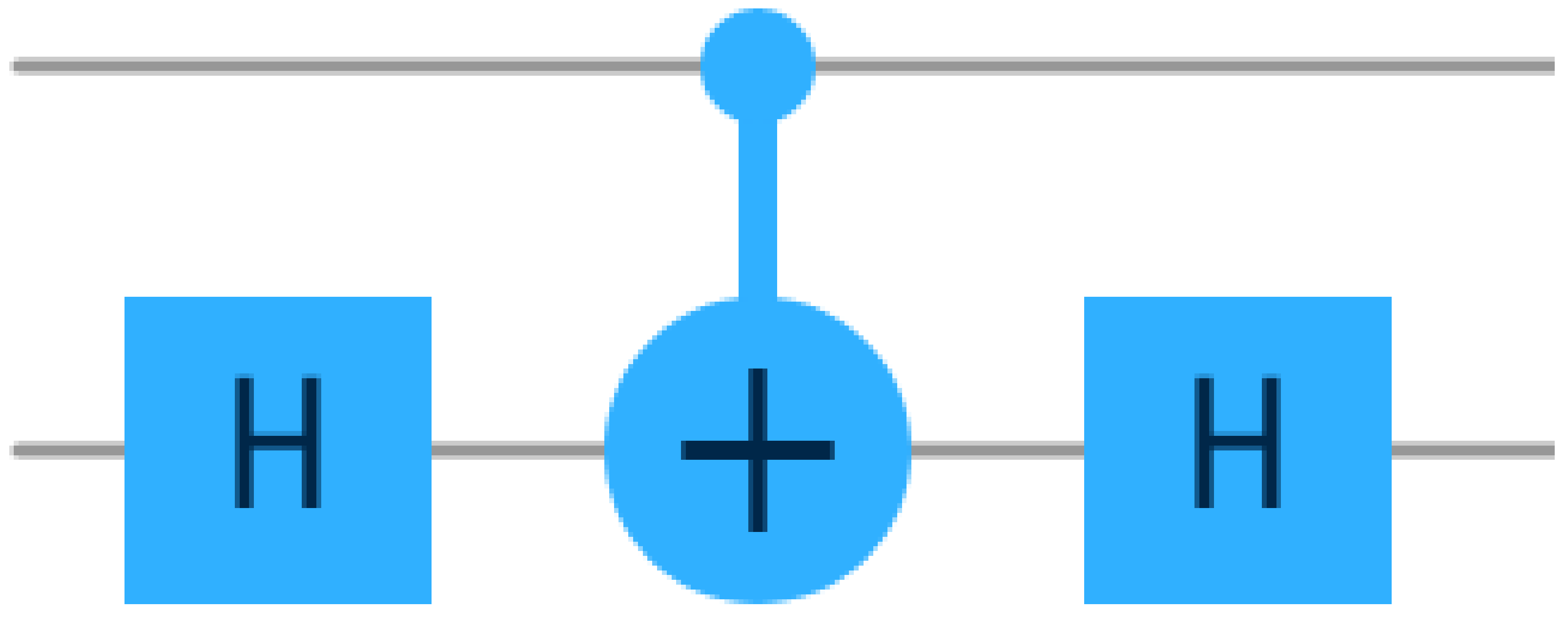
C1

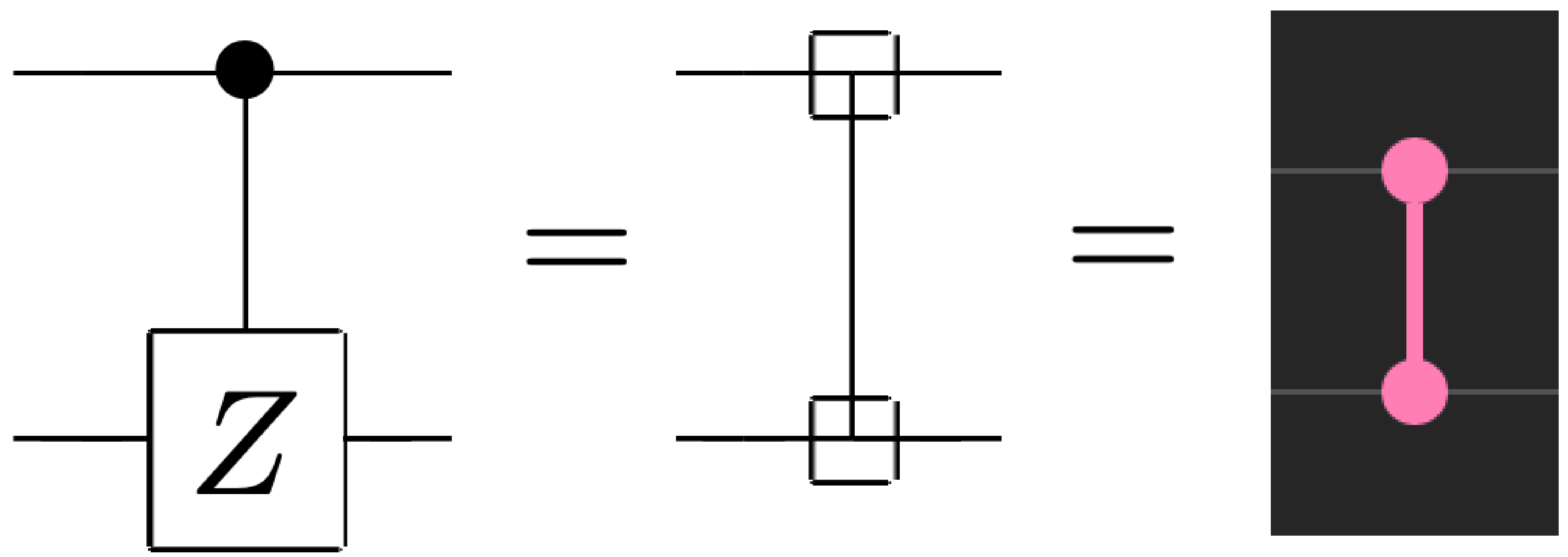
Remember (or check):

$$H\,X\,H = Z$$

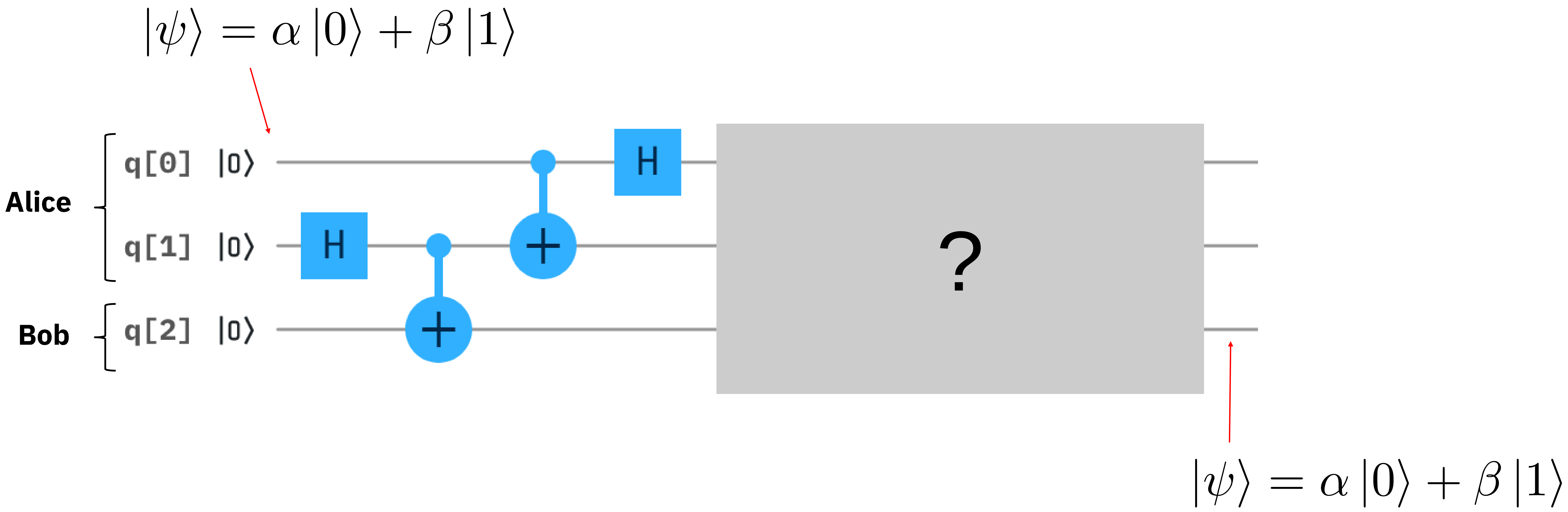
Controlled Z Gate (This is an Optional Step)

Note: $$H\ X\ H = Z$$

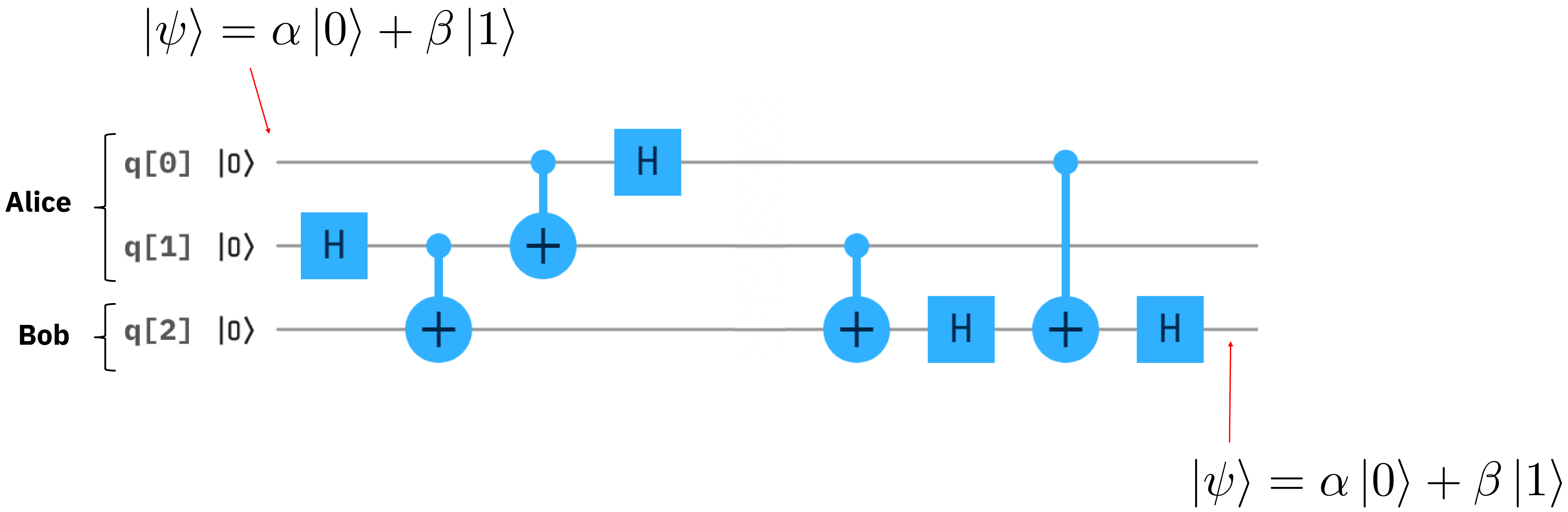




3 Qubit Circuit for Quantum Teleportation

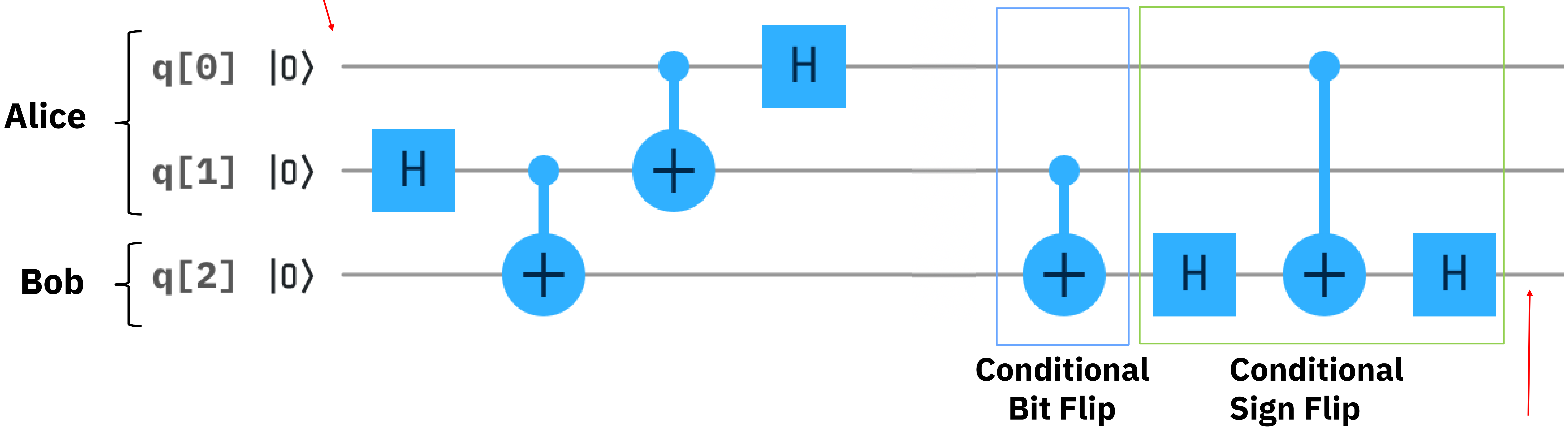


3 Qubit Circuit for Quantum Teleportation



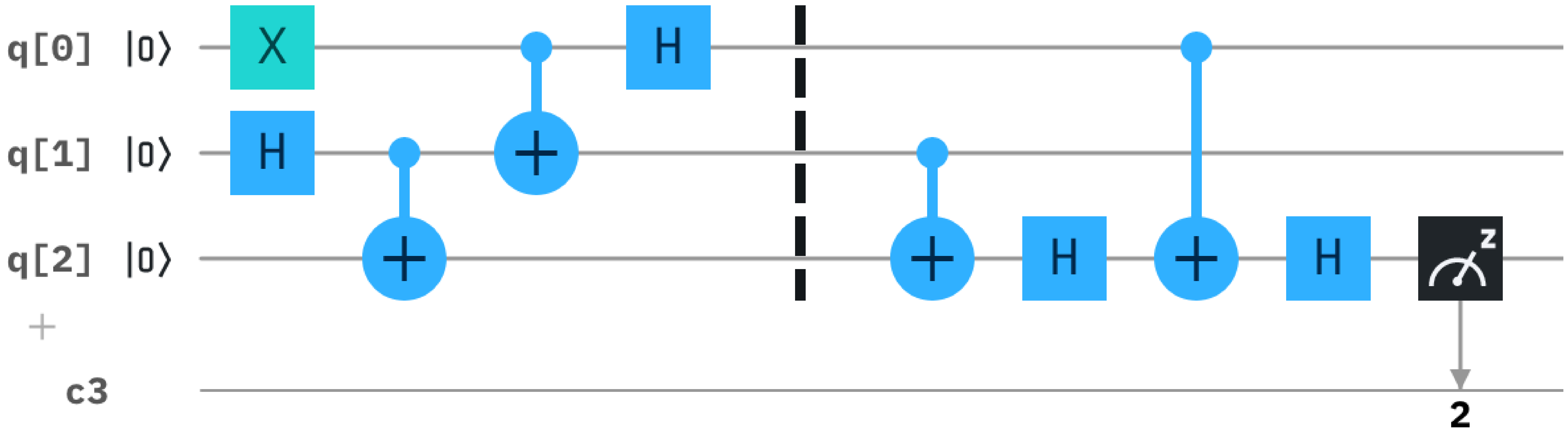
3 Qubit Circuit for Quantum Teleportation

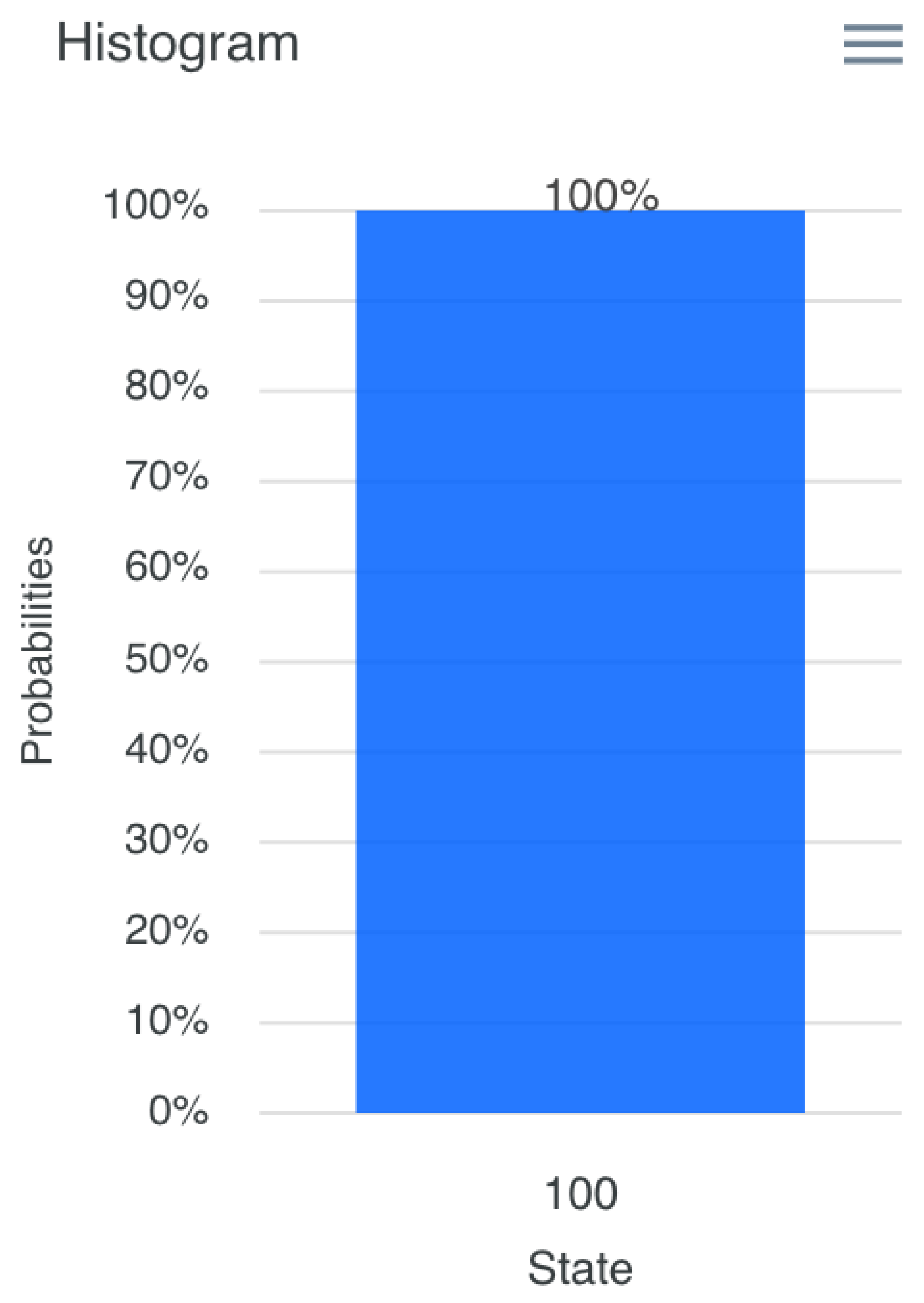
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$



$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

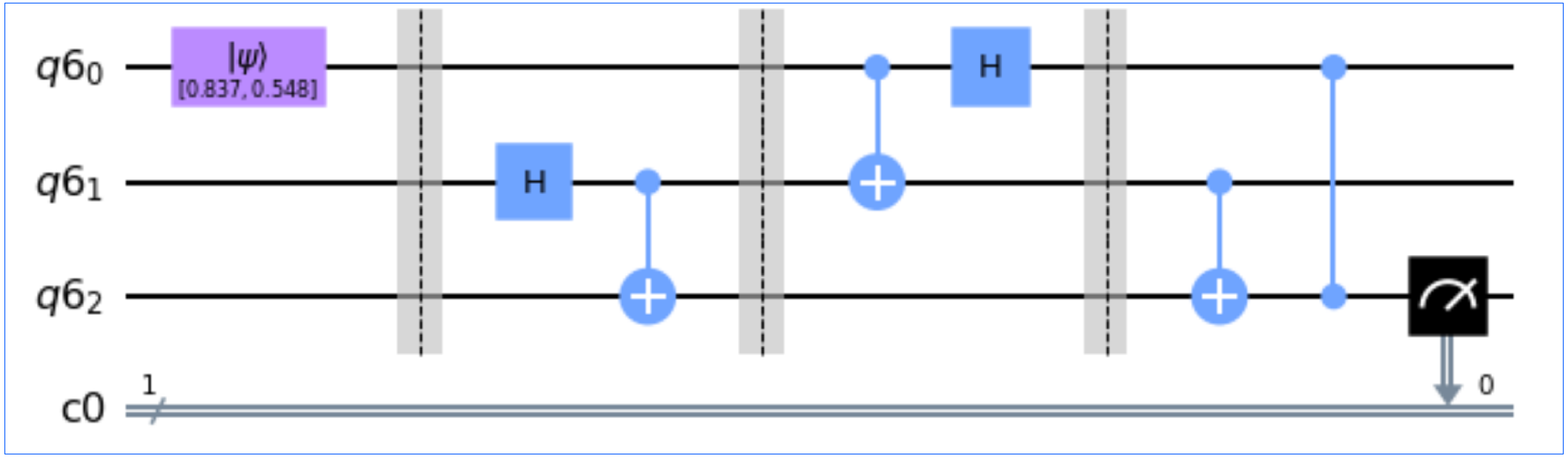
3 Qubit Circuit for Quantum Teleportation: $|1\rangle$





Let us now complete the Jupyter Notebook

- This is the Quantum Circuit that allows to run a Quantum Teleportation
- We are going to program this with Qiskit (by completing a Jupyter Notebook)
- We teleport the Quantum State $\psi = \sqrt{0.7}\,|0\rangle + \sqrt{0.3}\,|1\rangle$



An Overview of some famous Quantum Algorithms

Quantum Algorithms

- Deutsch-Josza
- Bernstein-Vazirani
- Simon
- Shor
    - Quantum Fourier Transform
    - Quantum Phase Estimation
- Grover
- Counting
- Teleportation
- Superdense Coding
- Key Distribution

Quantum Algorithms for Applications

- HHL
- VQE
- QAOA
- Quantum Neural Network
- QSVM
- ...

Bernstein-Vazirani

Guess a Secret Number "s" that is represented as a bit string and that is hidden in a "Black Box"

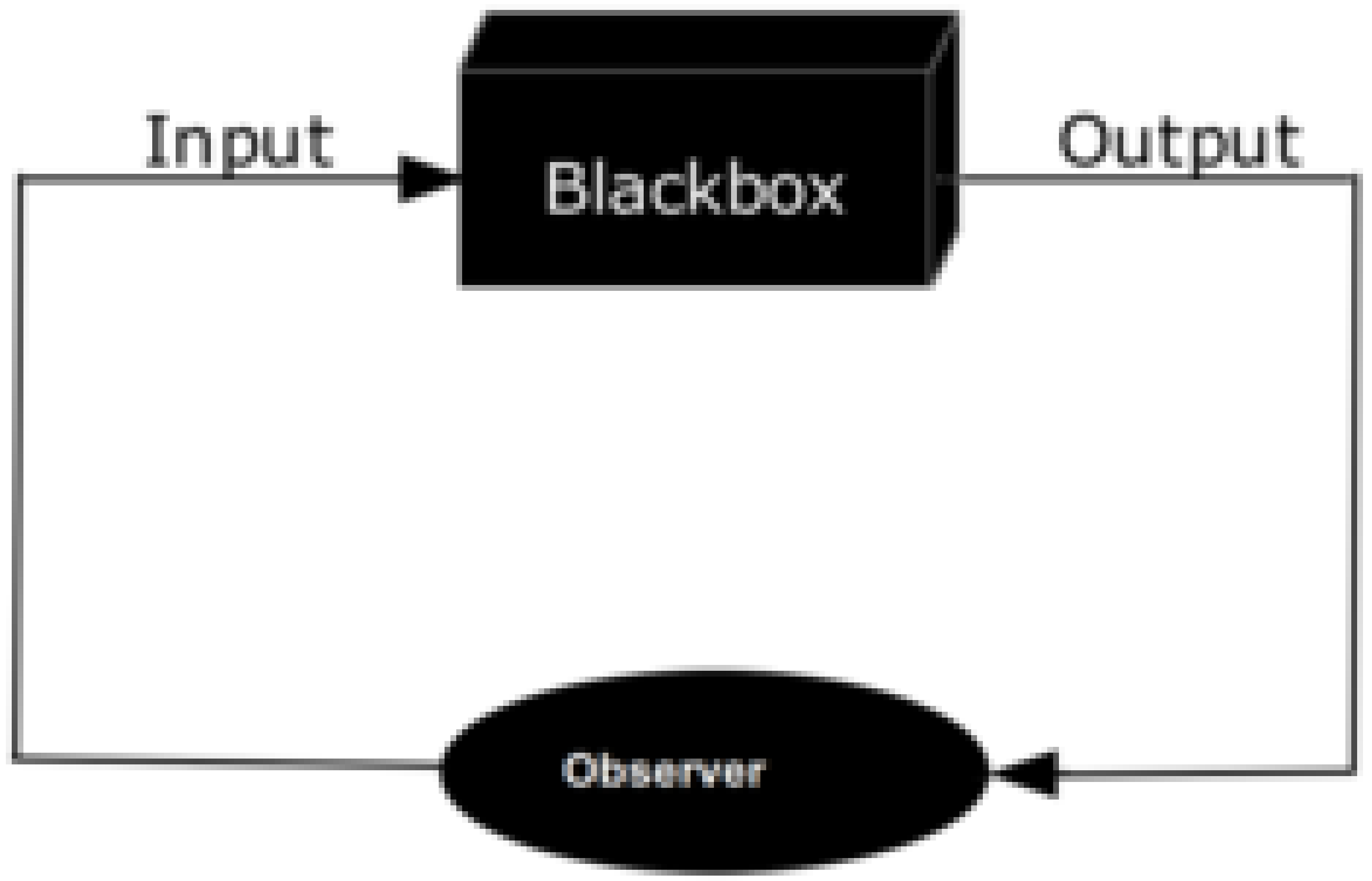
What is the Challenge?

Guess a Secret Number "s" that is represented as a bit string and that is hidden in a "Black Box"

<u>Preliminary</u>

An "Oracle" is an operation that has some properties you do not know, and you are trying the find out what these properties are

- Equivalent term: "<u>Black Box</u>"
- You cannot see inside it
- You do not know what it is doing upfront
- But… you can supply inputs and receive outputs ➔ <u>"Queries"</u>



Formal Explanations

- **Let s be an unknown non-negative integer less than 2^n**
- Let f(x) take any integer x into the modulo-2 sum of the products of corresponding bits of s and x, which we denote by s · x (bitwise modulo-2 inner product):

  $$s \cdot x = s_0 x_0 \oplus s_1 x_1 \oplus s_2 x_2 \cdots$$
- Example:
  - *n = 6*
  - *s = 110101*
  - *x = 001111*
  - *f(x) = s · x = 1.0 + 1.0 + 0.1 + 1.1 + 0.1 + 1.1 = 1 + 1 modulo 2 = 0*

- Suppose that we have a subroutine that evaluates f(x) = s · x
- How many times do we have to call that subroutine f to determine the value of the integer s?
- **With a classical computer** we can learn the n bits of s by applying f just n times:
  - *For n=6, this are 000001, 000010, 000100, 001000, 010000, 100000*

- **This requires n different invocations of the subroutine**

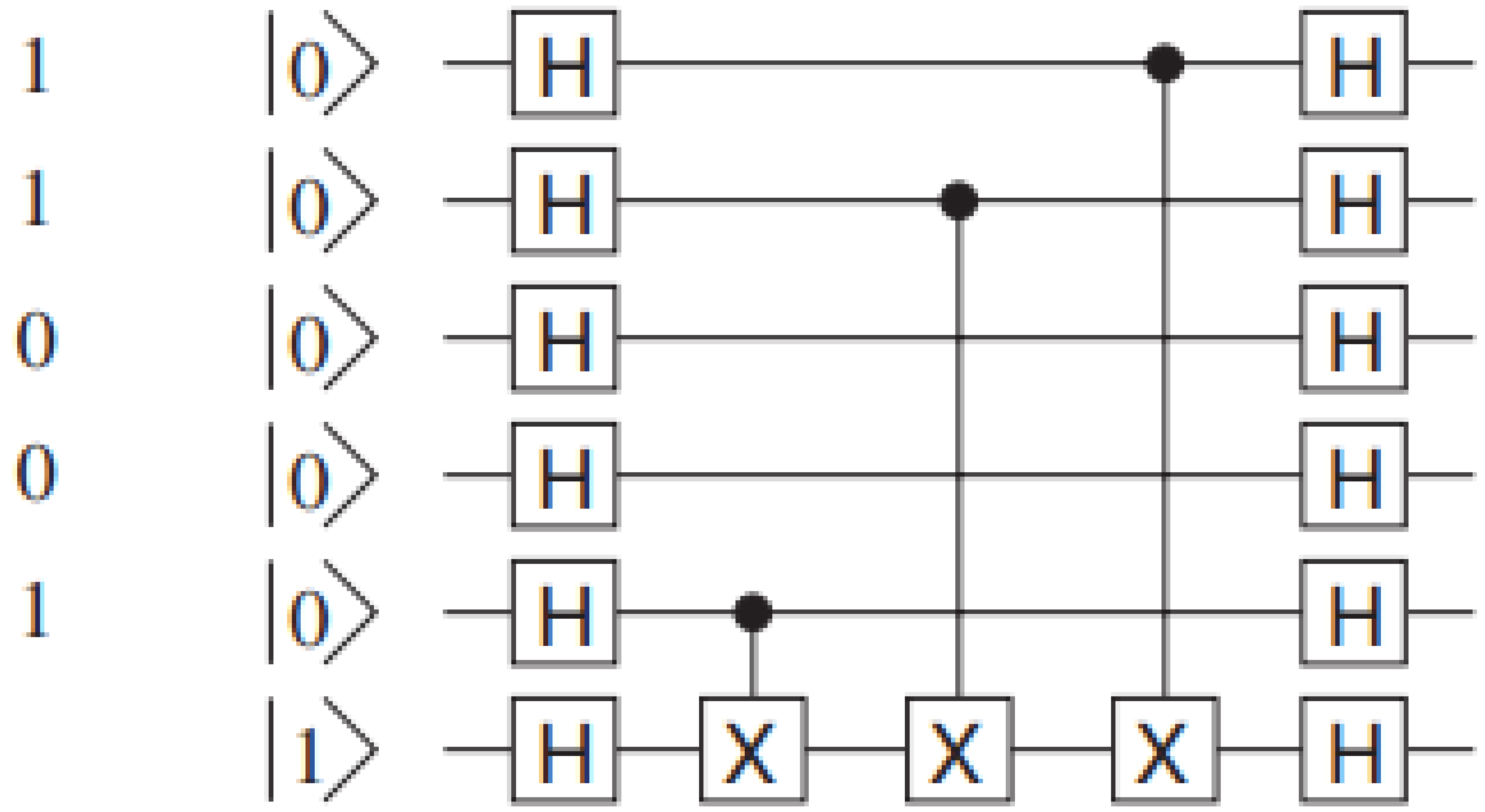
More Explanations

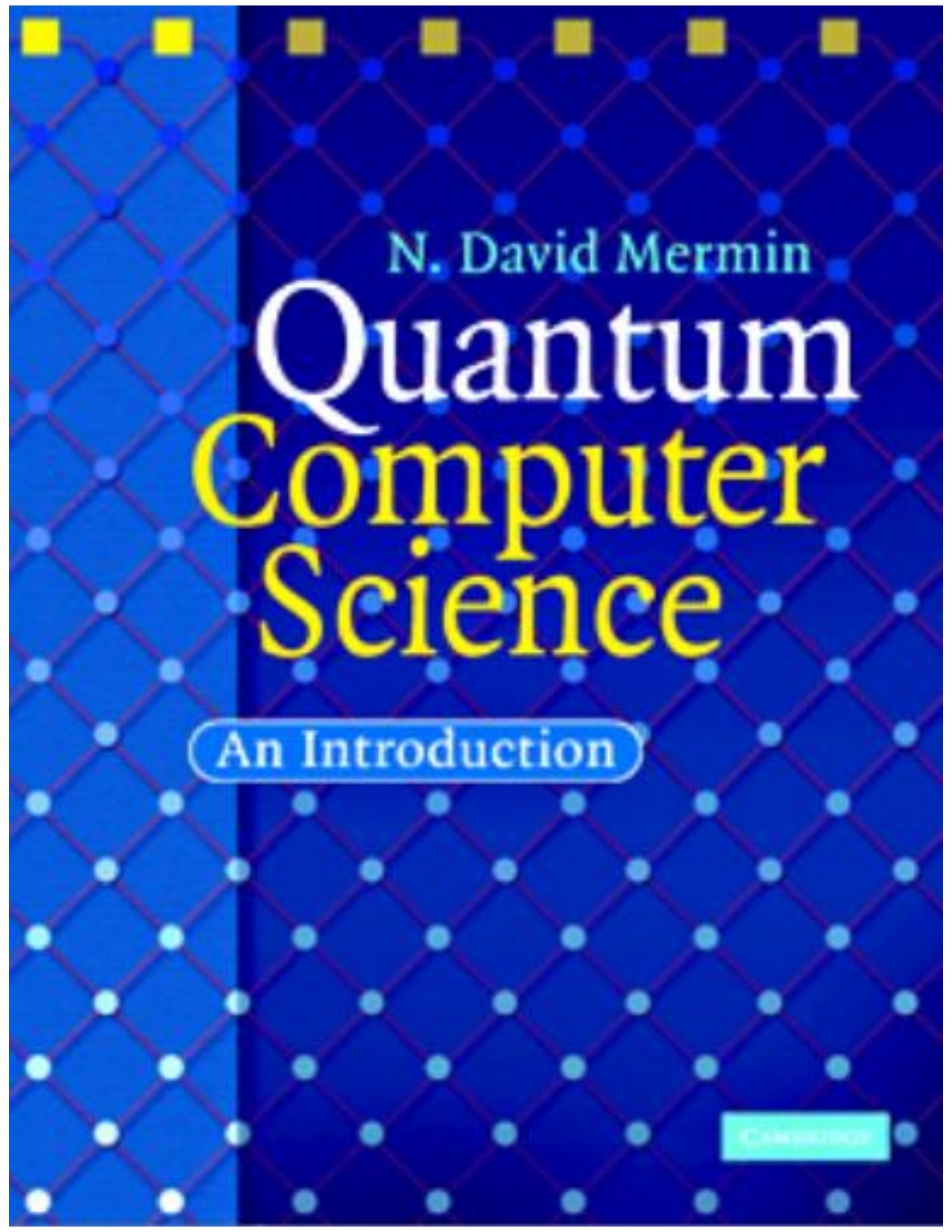
With a Quantum Computer a single invocation is enough to determine a completely, regardless of how big n is!

Formula: $H^{\otimes(n+1)} U_f H^{\otimes(n+1)} |0\rangle_n |1\rangle_1 = |s\rangle_n |1\rangle_1$

The crucial operator is U_f

With cNOT gates the action of U_f is produced





Pages 50-54

Define the Secret Number

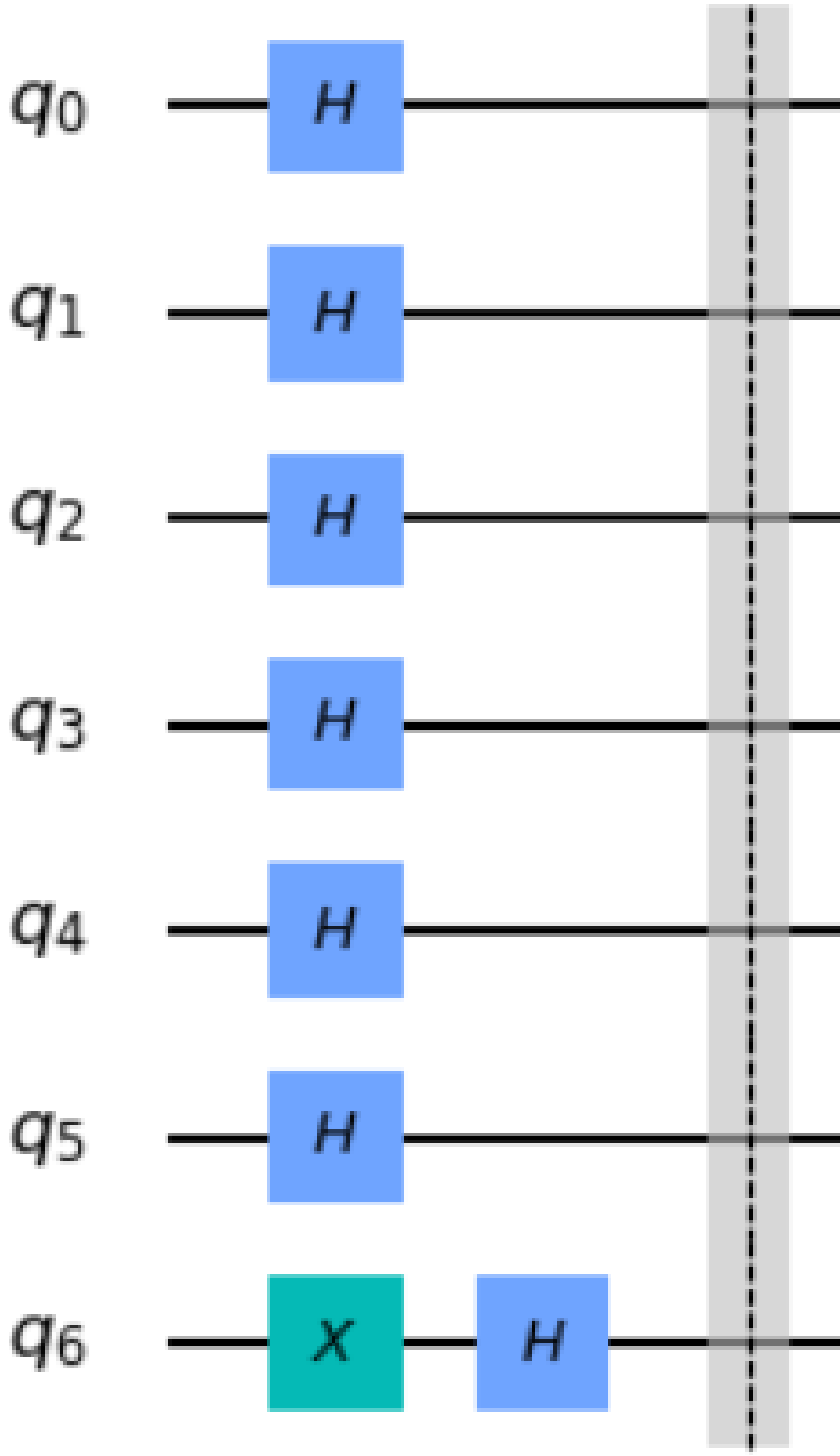
For example, suppose it consists of 6 bits

```
# Define the secret number
secretnumber = '101001'
```

Create and Initialize the Qubits

```
ericcircuit = QuantumCircuit(6+1, 6)

ericcircuit.h([0,1,2,3,4,5])
ericcircuit.x([6])
ericcircuit.h([6])
ericcircuit.barrier()
```



Apply CNOT Gates to implement the Oracle

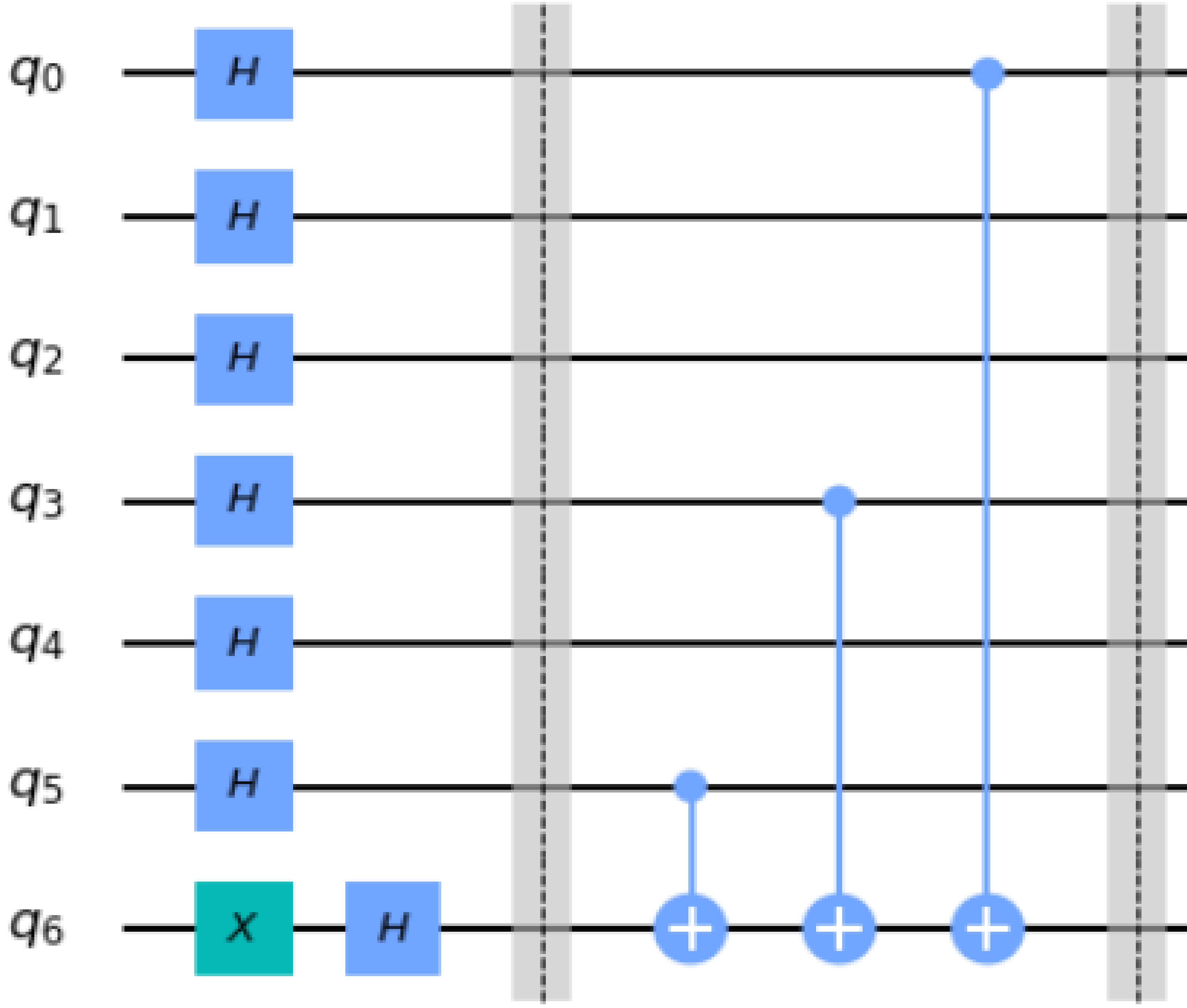
Control Qubit is a Qubit that has a corresponding 1 in the Secret Number

```
ericcircuit.cx(5,6)
ericcircuit.cx(3,6)
ericcircuit.cx(0,6)
ericcircuit.barrier()
```

S = 101001

Reverse sequence !

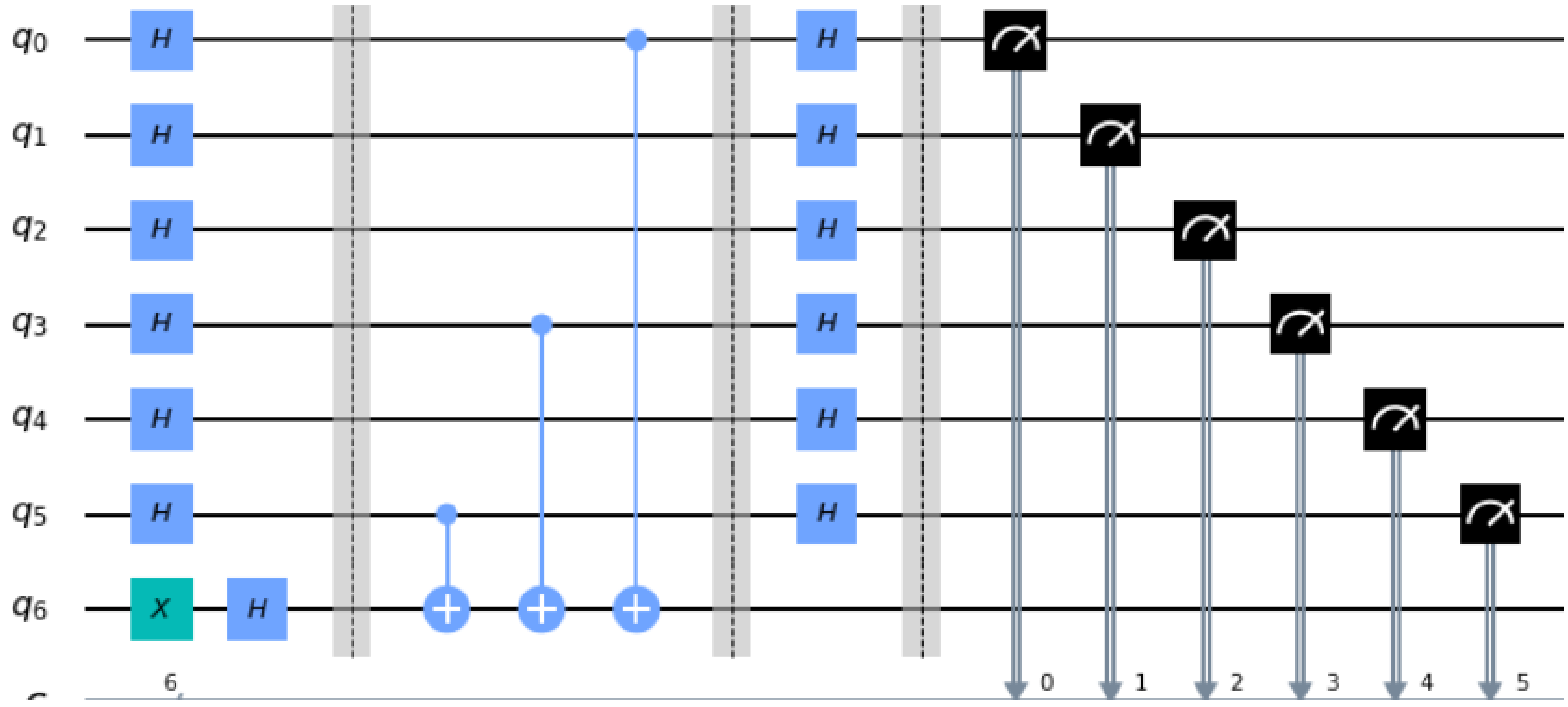
From MSB to LSB



Apply Hadamard Gates and Measure

```
ericcircuit.h([0,1,2,3,4,5])
ericcircuit.barrier()

ericcircuit.measure([0,1,2,3,4,5], [0,1,2,3,4,5])
```



Execute with just 1 shot

```
from qiskit.visualization import plot_histogram
result = AerSimulator().run(ericcircuit).result()
statistics = result.get_counts()
display(plot_histogram(statistics))
print(statistics)
plot_histogram = image = circuit.draw('mpl')
plt.show()
```

- But … what if we have a secret number with a variable length and variable bit values?
- E.g. We want to find secret numbers:
- ‘101010101’, ‘10011’, ‘1111101010010101’, and so on…
- Apply Python programming constructs !

Intuitive Proof

- Suppose the secret number is ‘110101’ = ‘$q_5\ q_4\ q_3\ q_2\ q_1\ q_0$’
- Initialize the 6 Qubits with H Gate: $H\ |0\rangle = \ |+\rangle$
- Initialize the Ancilla Qubit with H.X Gate: $H\ X\ |1\rangle = \ |-\rangle$
- For example, assume q0 and q6, then we initialize as follows:

$$q0 = \frac{1}{\sqrt{2}}(|0\rangle + \ |1\rangle) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$q6 = \frac{1}{\sqrt{2}}(|0\rangle - \ |1\rangle) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\Longrightarrow \qquad |q6\ q0\rangle = |q6\ \rangle \otimes \ |q0\rangle = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

Intuitive Proof (Cont.)

- So, suppose there is a 1 in position 0 of the bit string, then we apply a CNOT between q0 and q6
- Then we apply a unitary matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

- So, we see how q0 was changed: $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
- This is called "phase kickback": we see how q0 was changed from $|+\rangle$ into $|-\rangle$
- Qubits representing a 1 of the secret are turned into $|-\rangle$ state
- Qubits representing a 0 of the secret stay in $|+\rangle$ state
- **But at the end we measure with H Gate:**
  - $H\,|-\rangle = |1\rangle$
  - $H\,|+\rangle = |0\rangle$
- <u>And this is what we need !</u>
  - If the bit in the secret number was a 1, we will measure a 1 → Measure bit = 1
  - If the bit in the secret number was 0, we will measure 0 → Measure bit = 0

Additional Materials

- *Qiskit: https://qiskit.org/textbook/ch-algorithms/bernstein-vazirani.html*
- *Videos*
  - *https://www.youtube.com/watch?v=sqJIpHYl7oo*
  - *https://www.youtube.com/watch?v=m9KtmAi7iW4*
  - *https://www.youtube.com/watch?v=xtD8e91kfxc*
- *And there is so much more ☺*

What is Efficient?

In complexity theory, Algorithms that take at most polynomial time are considered "efficient"

Not Efficient

Name	Running Time $T(n)$	Example
Constant Time	$O(1)$	Check if integer is even or odd
Loglinear Time	$O(\mathrm{n} \log n)$	Sort, Fast Fourier Transform
Logarithmic Time	$O(\log n)$	Binary search in sorted list
Linear Time	$O(n)$	Find largest number in unsorted list
Quadratic Time	$O(n^2)$	Multiply two n-digit numbers
Polynomial Time	$O(n^3)$	Naive matrix multiplication
Exponential Time	$O(2^n)$	Integer Factorization
Factorial Time	$O(n!)$	Brute –Force TSP

$$\langle 10|Q\rangle$$

$$for\ your\ attention\,!$$