Introduction to Quantum Information and Quantum Machine Learning

Laboratory - class 5

Dr Gustaw Szawioła, docent PUT
D. Sc. Eng. Przemysław Głowacki

1. QFT Quantum Fourier Transform

The Fourier transform

Classical

The discrete Fourier transform acts on a vector $(x_0,\dots,x_{N-1})$ and maps it to the vector $(y_0,\dots,y_{N-1})$ according to the formula

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk} \quad \text{where} \quad \omega_N^{jk} = e^{2\pi i \frac{jk}{N}}$$

Joseph Fourier
*21.03.1768
†16.05.1830

Quantum Fourier Transform **QFT**

The quantum Fourier transform acts on a quantum state $|X\rangle = \sum_{j=0}^{N-1} x_j \, |j\rangle$ and maps it to the quantum state $|Y\rangle = \sum_{k=0}^{N-1} y_k \, |k\rangle$ according to the formula

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk}$$

where

$$\omega_N^{jk} = e^{2\pi i \frac{jk}{N}}$$

Note that only the amplitudes of the state were affected by this transformation.

This can also be expressed as the map:

$$|j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jk} \, |k\rangle$$

or the unitary matrix

$$U_{QFT} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{j=0}^{N-1} \omega_N^{jk} \, |k\rangle\langle j|$$

QFT continued..

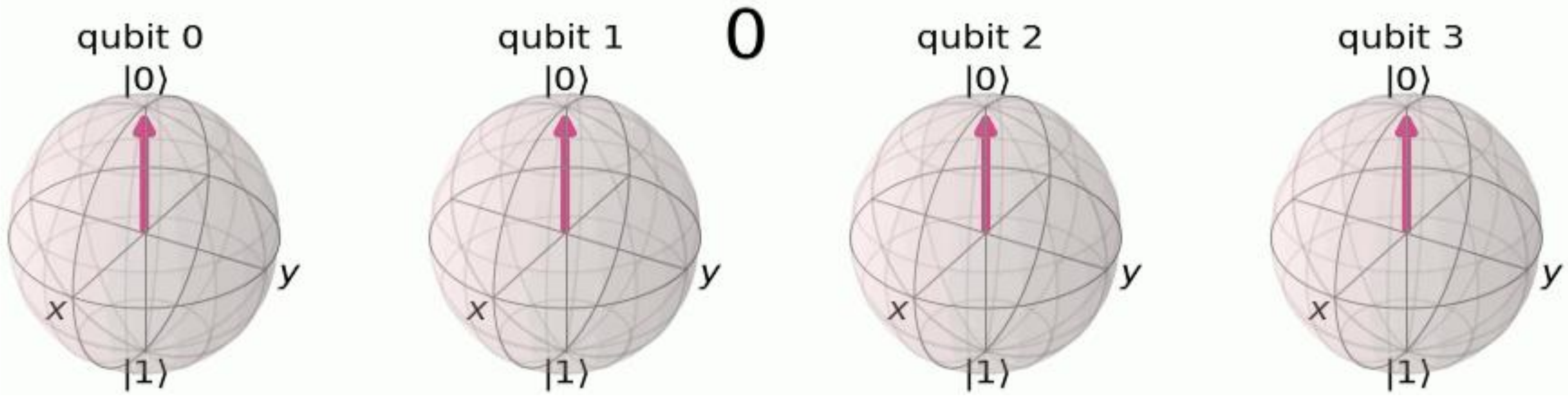
The Quantum Fourier Transform (QFT) transforms between two bases, the computational (Z) basis, and the Fourier basis. The H-gate is the single-qubit QFT, and it transforms between the Z-basis states $|0\rangle$ and $|1\rangle$ to the X-basis states $|+\rangle$ and $|-\rangle$. In the same way, all multi-qubit states in the computational basis have corresponding states in the Fourier basis. The QFT is simply the function that transforms between these bases.

$$|\text{State in Computional Basis}\rangle \xrightarrow{\text{QFT}} |\text{State in Fourier Basis}\rangle$$

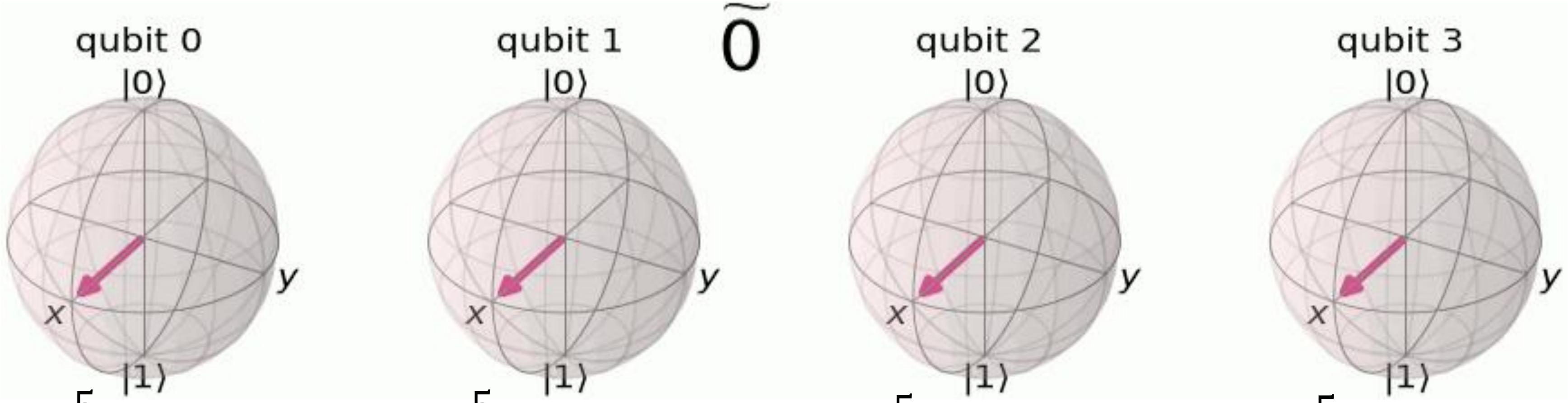
$$\text{QFT}|x\rangle = |\tilde{\text{x}}\rangle$$

QFT continued..

Store decimal numbers in binary using the states $|0\rangle$ and $|1\rangle$:



In the Fourier basis, we store numbers using different rotations around the Z-axis



$$\frac{5}{16} \times 2\pi \qquad \frac{5}{8} \times 2\pi \qquad \frac{5}{4} \times 2\pi \qquad \frac{5}{2} \times 2\pi$$

$$\frac{\text{number } \times 2\pi}{2^k}$$

number =5
k =4 (bits)

1-qubit QFT

Consider how the QFT operator as defined above acts on a single qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. In this case, $x_0 = \alpha,\ x_1 = \beta$ and *N = 2*. Then:

$$y_0 = \frac{1}{\sqrt{2}}\left(\alpha \exp\left(2\pi i \frac{0\times 0}{2}\right) + \beta \exp\left(2\pi i \frac{1\times 0}{2}\right)\right) = \frac{1}{\sqrt{2}}(\alpha+\beta)$$

and

$$y_1 = \frac{1}{\sqrt{2}}\left(\alpha \exp\left(2\pi i \frac{0\times 1}{2}\right) + \beta \exp\left(2\pi i \frac{1\times 1}{2}\right)\right) = \frac{1}{\sqrt{2}}(\alpha-\beta)$$

such that the final result is the state

$$U_{QFT}|\psi\rangle = \frac{1}{\sqrt{2}}(\alpha+\beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|1\rangle$$

This operation is exactly the result of applying the Hadamard operator (*H*) on the qubit:

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

If we apply the ***H*** operator to the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ we obtain a new state:

$$\frac{1}{\sqrt{2}}(\alpha+\beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|1\rangle \equiv \tilde{\alpha}|0\rangle + \tilde{\beta}|1\rangle$$

N-qubit QFT

For $N=2^n$ the QFT_N acting on the state $|x\rangle = |x_1 ... x_n\rangle$,where x_1 is the most significant bit. Then:

$$QFT_N|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle$$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i xy/2^n} |y\rangle \text{ since} : \omega_N^{xy} = e^{2\pi i \frac{xy}{N}} \text{ and } N = 2^n$$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \left(\sum_{k=1}^{n} y_k/2^k\right) x} |y_1 \ldots y_n\rangle : \text{rewriting in fractional binary notation } y = y_1 \ldots y_n, y/2^n = \sum_{k=1}^{n} y_k/2^k$$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{2\pi i x y_k/2^k} |y_1 \ldots y_n\rangle : \text{after expanding the exponential of a sum to a product of exponentials}$$

$$= \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} \left(|0\rangle + e^{2\pi i x/2^k} |1\rangle\right) : \text{after rearranging the sum and products, and expanding } \sum_{y=0}^{N-1} = \sum_{y_1=0}^{1} \sum_{y_2=0}^{1} \ldots \sum_{y_n=0}^{1}$$

$$= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i}{2} x} |1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^2} x} |1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^{n-1}} x} |1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^n} x} |1\rangle\right)$$

Circuit that implements the QFT

The circuit that implements QFT makes use of two gates:

a single-qubit Hadamard gate ***H***

$$H|x_k\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + \exp\left(\frac{2\pi i}{2}x_k\right)|1\rangle\right)$$

The second is a two-qubit controlled rotation $CROT_k$ given in block-diagonal form as

$$CROT_k = \begin{bmatrix} I & 0 \\ 0 & UROT_k \end{bmatrix}$$

where

$$UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi i}{2^k}\right) \end{bmatrix}$$

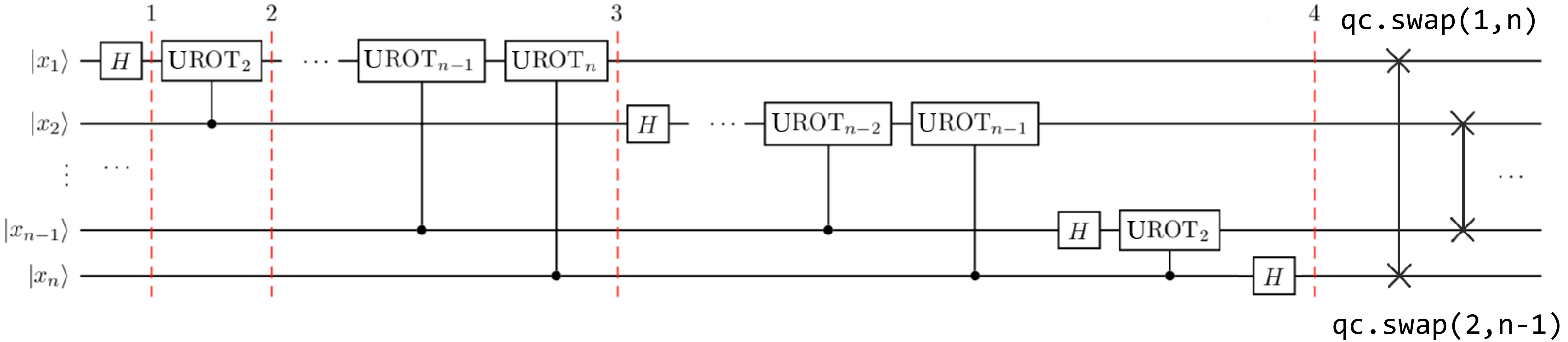
The action of $CROT_k$ on a two-qubit state $|x_i\, x_j\rangle$ where the first qubit is the control and the second is the target is given by

$$CROT_k|0x_j\rangle = |0x_j\rangle$$

and

$$CROT_k|1x_j\rangle = \exp\left(\frac{2\pi i}{2^k}x_j\right)|1x_j\rangle$$

The circuit that implements an n-qubit QFT is shown below



Qiskit Implementation

In **Qiskit**, the implementation of the ***CROT* gate** used in the discussion above is a controlled **phase rotation gate**.

$$CP(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{bmatrix}$$

Hence, the mapping from the ***CROT$_k$*** gate into the ***CP*** gate is found from the equation $\theta = 2\pi/2^k = \pi/2^{k-1}$

QFT Example on 3 Qubits

```
from numpy import pi
# importing Qiskit
from qiskit import QuantumCircuit, transpile
from qiskit_aer import Aer
from qiskit.visualization import plot_histogram, plot_bloch_multivector
```

Encode a number in the computational basis.
We can see the number 5 in binary is 101:

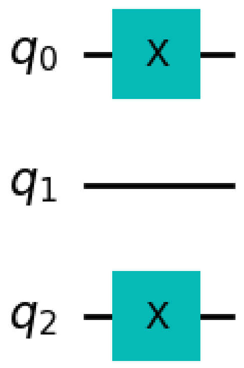
```
bin(5)
```

```
Out['0b101']
```

```
# Create the circuit
qc = QuantumCircuit(3)

# Encode the state 5
qc.x(0)
qc.x(2)
qc.draw()
```

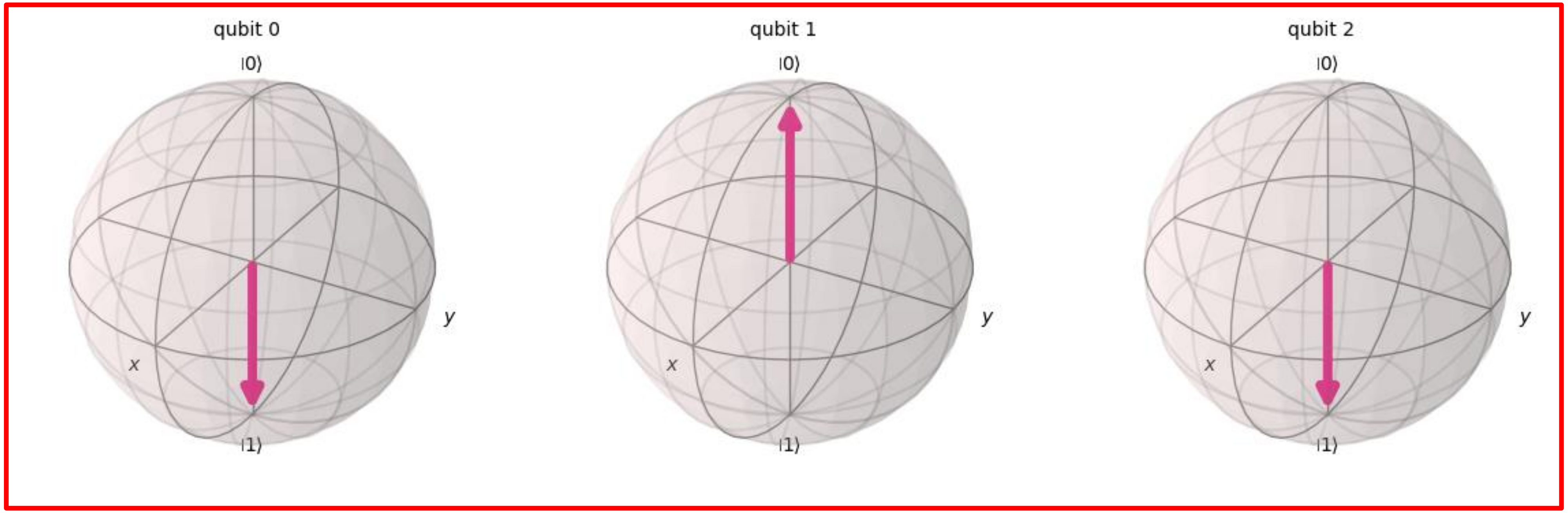
```
Out[]
```



Let's check the qubit's states using the aer simulator

```
sim = Aer.get_backend("aer_simulator")
qc_init = qc.copy()
qc_init.save_statevector()
statevector = sim.run(qc_init).result().get_statevector()
plot_bloch_multivector(statevector)
```

- Out[]



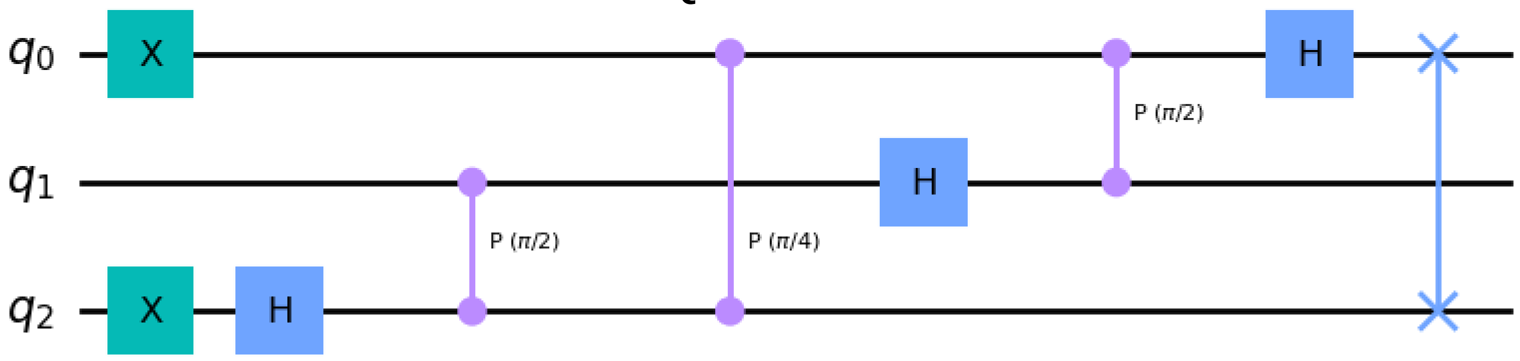
Build the QFT circuit and view the final state of our 3 qubits ($|101\rangle$):

Qiskit order $|q_2q_1q_0\rangle$

```
# Create the circuit
qc = QuantumCircuit(3)
qc.x(0)
qc.x(2)
qc.h(2)
qc.cp(pi/2, 1, 2) # CROT from qubit 1 to qubit 2
qc.cp(pi/4, 0, 2) # CROT from qubit 0 to qubit 2
qc.h(1)
qc.cp(pi/2, 0, 1) # CROT from qubit 0 to qubit 1
qc.h(0)
qc.swap(0,2)
qc.draw('mpl')
```

- Out[]

QFT

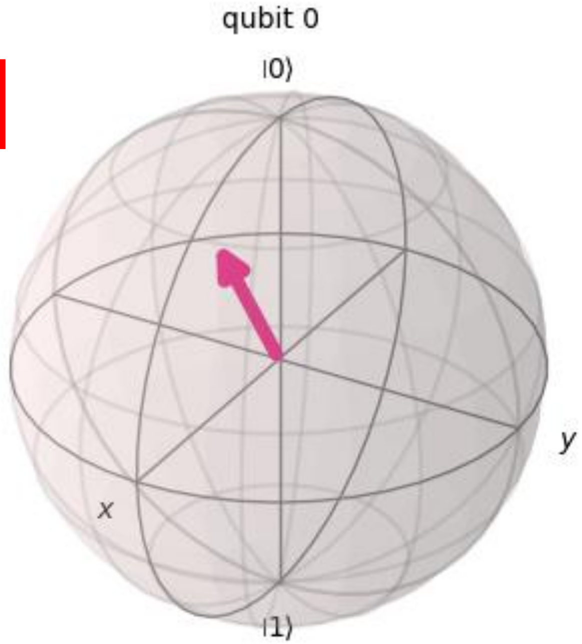


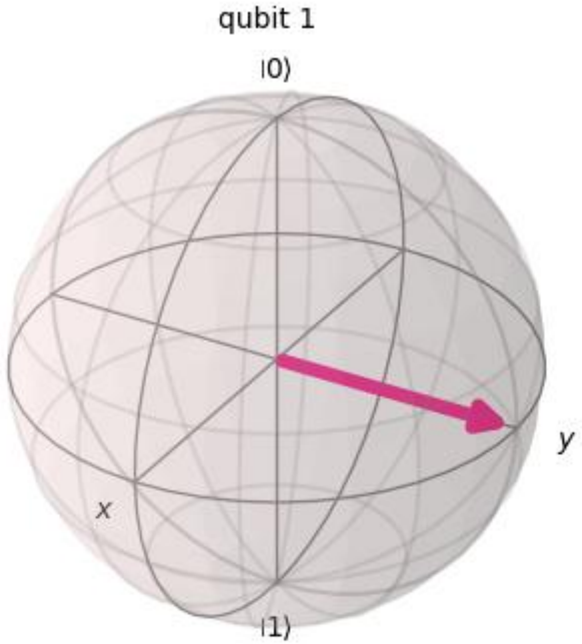
Build the QFT circuit and view the final state of our 3 qubits (|101⟩): 5

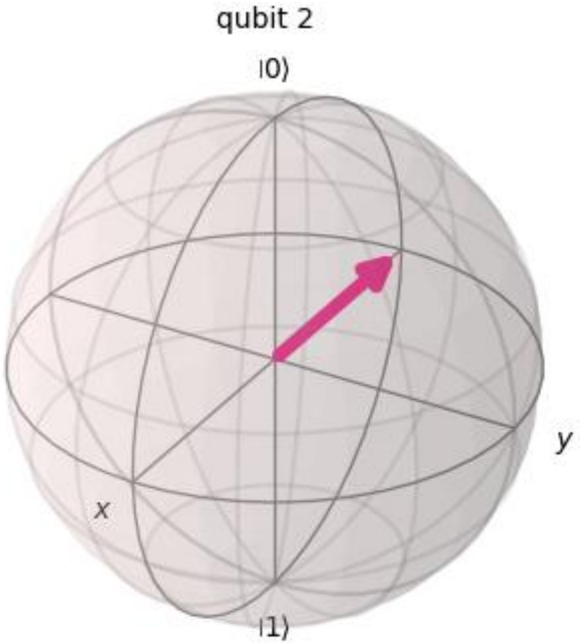
```
qc.save_statevector()
statevector = sim.run(qc).result().get_statevector()
plot_bloch_multivector(statevector)
```

Qiskit order $|q_2q_1q_0\rangle$

- Out[]







QFT rotation

$$\frac{5\cdot 2\pi}{2^3}=\frac{5\pi}{4}$$

$$\frac{5\cdot 2\pi}{2^2}=\frac{5\pi}{2}$$

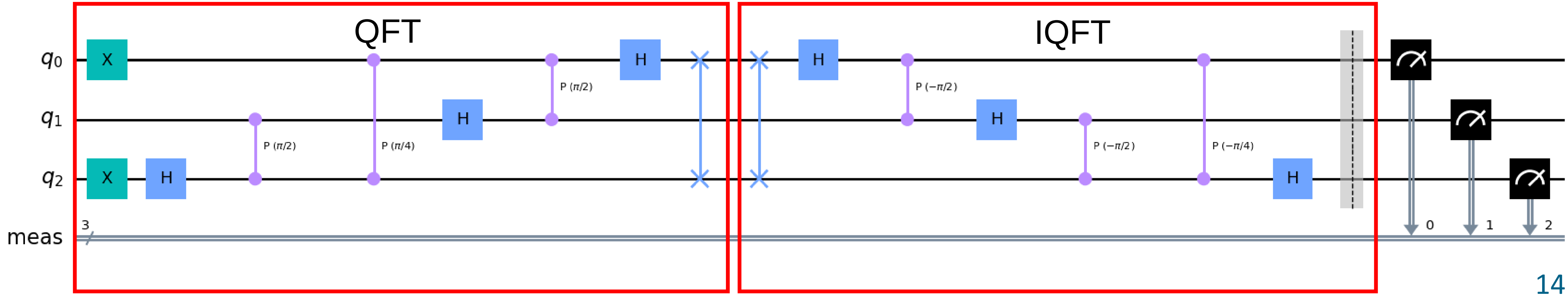
$$\frac{5\cdot 2\pi}{2^1}=5\pi$$

Build the Inverse QFT (IQFT) circuit :

Qiskit order $|q_2q_1q_0\rangle$

```
# inverse QFT - IQFT
qc.swap(0,2)
qc.h(0)
qc.cp(-pi/2, 0, 1) # CROT from qubit 0 to qubit 1
qc.h(1)
qc.cp(-pi/2, 1, 2) # CROT from qubit 1 to qubit 2
qc.cp(-pi/4, 0, 2) # CROT from qubit 2 to qubit 0
qc.h(2)
qc.measure_all()
qc.draw('mpl’)
```

- Out[]



Build the Inverse QFT (IQFT) circuit and check, if $\text{IQFT}|\tilde{5}\rangle = |101\rangle$?

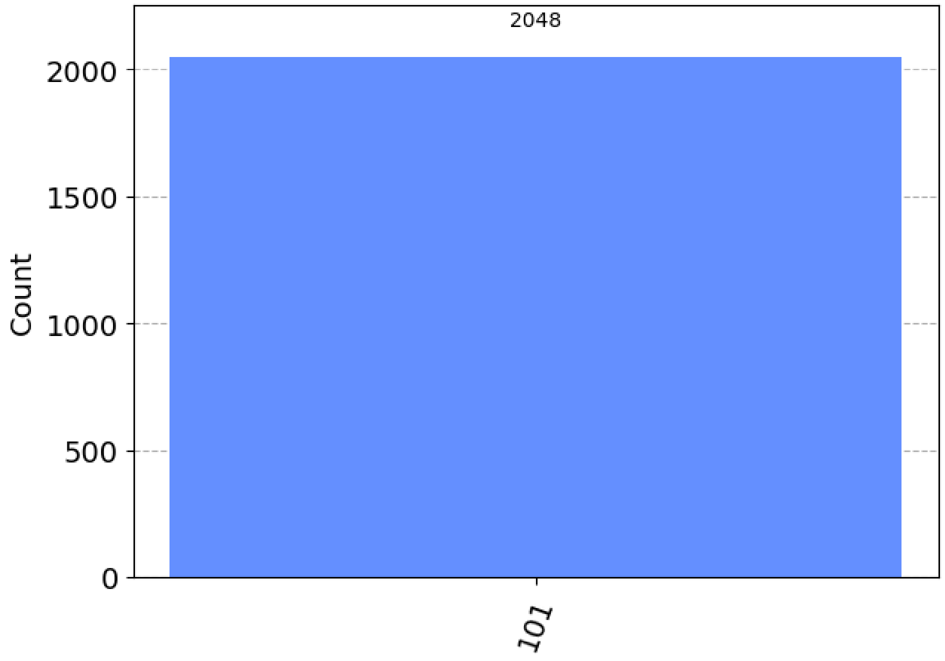
```
backend = Aer.get_backend('qasm_simulator')
shots = 2048
transpiled_qc = transpile(qc, backend,
optimization_level=3)
job = backend.run(transpiled_qc, shots=shots)
```

$$\text{IQFT}|\tilde{5}\rangle = |101\rangle$$

$$\text{QFT}^{\dagger}|\tilde{5}\rangle = |101\rangle$$

```
counts = job.result().get_counts()
plot_histogram(counts)
```

- Out[]



Tasks:

1. Try to find the state $|\tilde{a}\rangle$ such that $\mathrm{QFT}^{\dagger}|\tilde{a}\rangle = |100\rangle$ (show state $|\tilde{a}\rangle$ on Bloch sphers (statevector))
2. Try to find the state $|\tilde{b}\rangle$ such that $\mathrm{QFT}^{\dagger}|\tilde{b}\rangle = |011\rangle$ (show state $|\tilde{b}\rangle$ on Bloch sphers (statevector))
3. For the number of qubits n=3, present on the Bloch sphere (statevector) the Fourier transform of states from $|000\rangle$ to $|111\rangle$
4. (for grade 5.0) Try to write the QFT function with recursion for n=1 to n=8 qubits.
   Use *Qiskit's unitary simulator („*aer_symulator” ,'qasm_simulator’) to verify your results.

The End