Lab 1:

quantum circuits, multiqubit statevector manipulation and evaluation, the Quantum Information Qiskit package and visualization tools

Week 4 : 19/10/2020

TA: Hisham Ashraf Amer

Email: s-hisham.amer@zewailcity.edu.eg

Overview of Lab 1:

1. Build our first quantum circuits using Qiskit.

2. Introduce the Quantum information Module.

3. Building Multiqubit Operators to manipulate statevectors using the module.

4. Build a Bell state and visualize its measurement and elaborate on maximally entangled states.

5. Introduce the GHZ state, and build a quantum circuit for a 4-Qubit GHZ state and run it on IBM's quantum Computer Simulator.

6. Assess the Density matrices of GHZ states on Qiskit from an actual run on an IBM quantum computer, so we will see the effect of noise and decoherence on the density matrix of states resulting from real circuits.

7. Very quickly discuss the concept behind one very promising application of quantum computers, Variational Quantum Eigen-solvers, which we will build in a later lab session.

NB:

Remember to keep the Notebook with the code open while u study since not all the code is included here, I only post a few snippets of the code here

The nature of quantum mechanics and Quantum Hardware limit the type of procedures we can run. We will be mostly discussing the most popular kind, the quantum circuit model.

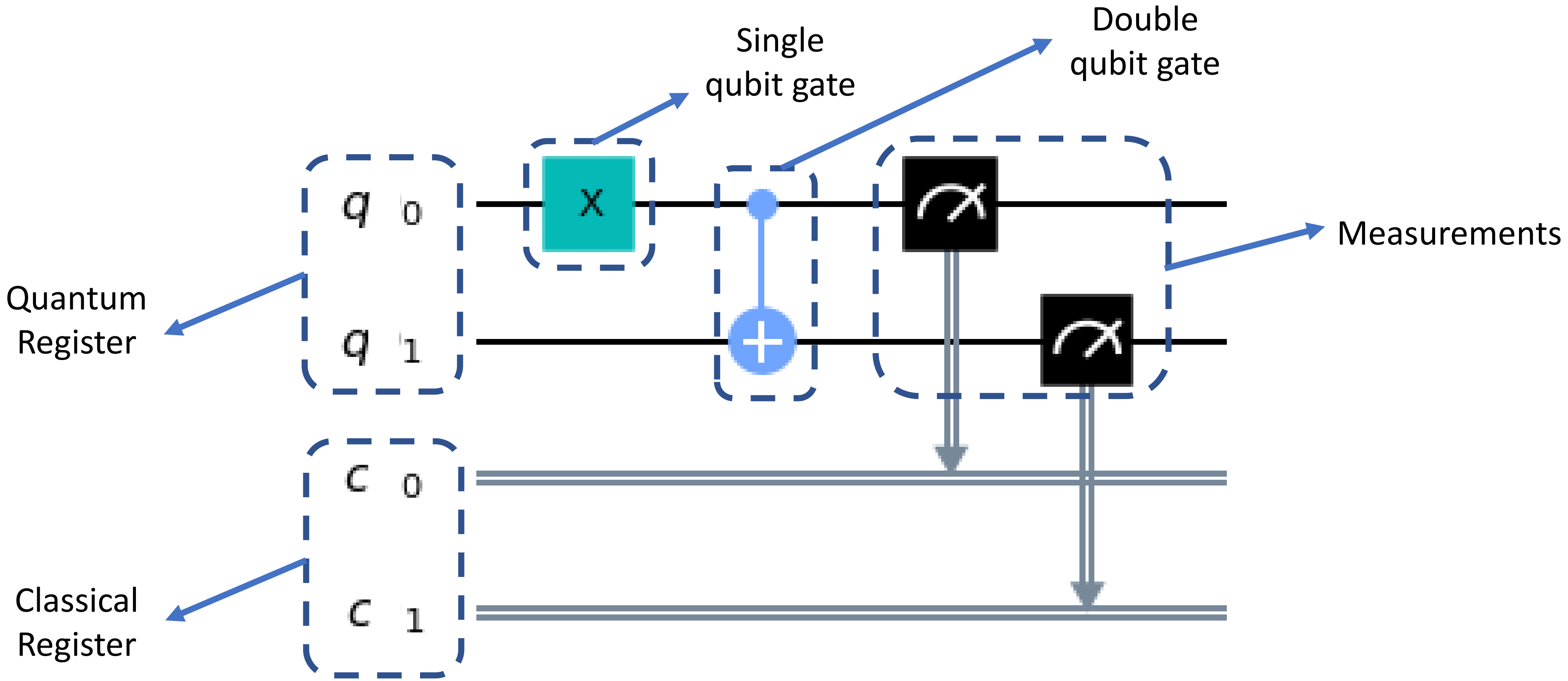
That’s why we have an entire field dedicated to finding ways in which we can translate quantum circuit results into meaningful output

Quantum Algorithms

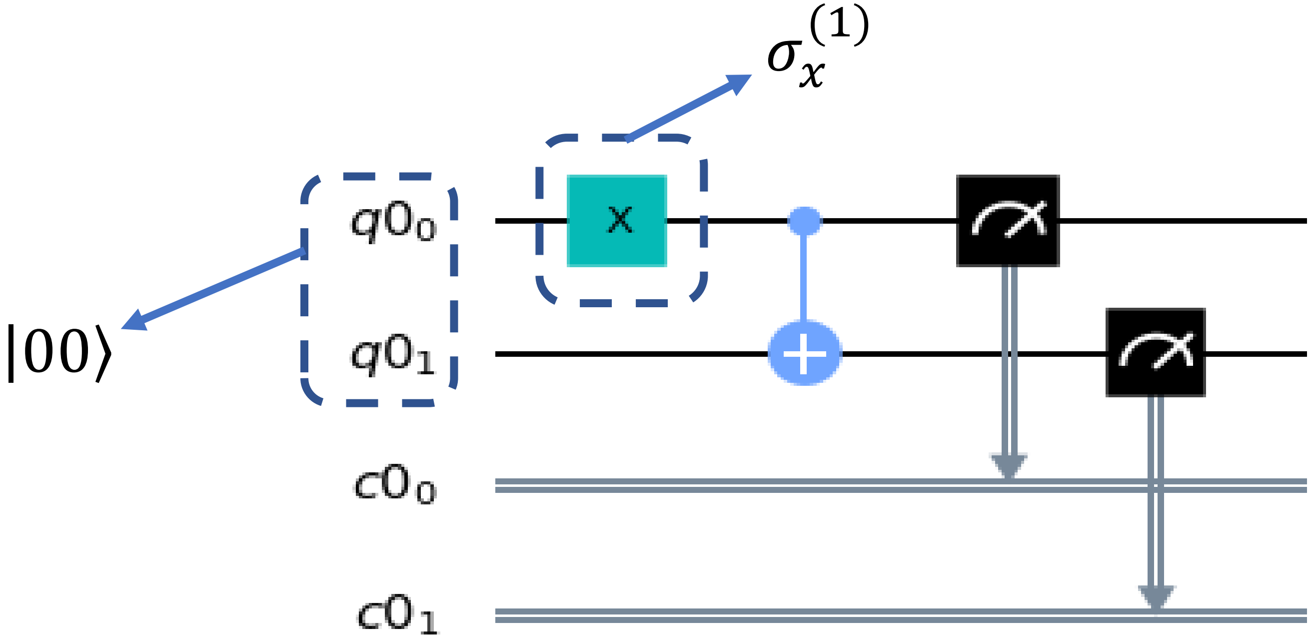
Every function we want to achieve needs its own algorithm and circuit design.

Quantum computers, based on the circuit model, are NOT universal yet.

Quantum circuit constituents



Write down the mathematical equivalent of this circuit



$$\mathrm{CNOT}\left(\sigma_x^{(1)} \otimes \mathbb{I}_{2\times 2}^{(2)}\right)|00\rangle$$

2 minute task :
derive the CNOT gate's matrix form

The CNOT gate flips the TARGET qubit iff the CONTROL qubit is $|1\rangle$

2 minute task: HINT:

$$\text{CNOT}|00\rangle = |00\rangle$$
$$\text{CNOT}|01\rangle = |01\rangle$$
$$\text{CNOT}|10\rangle = |11\rangle$$
$$\text{CNOT}|11\rangle = |10\rangle$$

Input basis

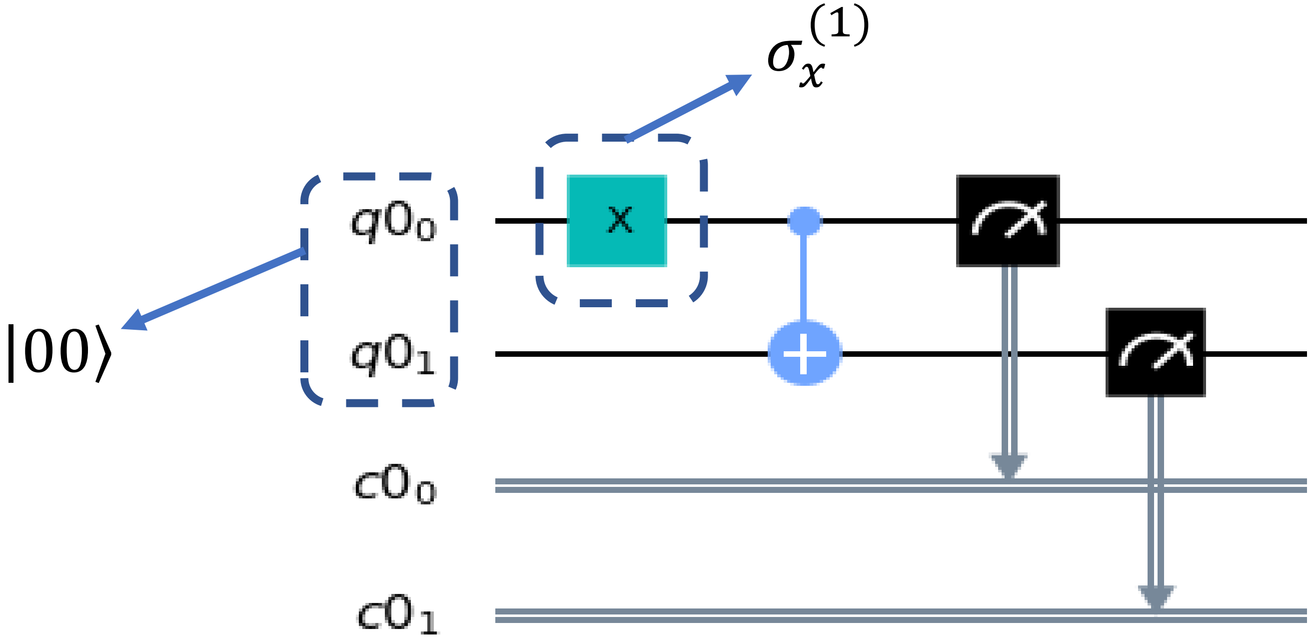
$$
\begin{array}{c|cccc}
\text{output basis} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\
\hline
\begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} &
\multicolumn{4}{c}{\begin{bmatrix}
\boxed{\langle 00|CNOT|00\rangle} & \langle 00|CNOT|01\rangle & \langle 00|CNOT|10\rangle & \langle 00|CNOT|11\rangle \\
\langle 01|CNOT|00\rangle & \boxed{\langle 01|CNOT|01\rangle} & \langle 01|CNOT|10\rangle & \langle 01|CNOT|11\rangle \\
\langle 10|CNOT|00\rangle & \langle 10|CNOT|01\rangle & \langle 10|CNOT|10\rangle & \boxed{\langle 10|CNOT|11\rangle} \\
\langle 11|CNOT|00\rangle & \langle 11|CNOT|01\rangle & \boxed{\langle 11|CNOT|10\rangle} & \langle 11|CNOT|11\rangle
\end{bmatrix}}
\end{array}
=
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}
$$

WHAT IF :

$$CNOT|00\rangle = |00\rangle$$
$$CNOT|01\rangle = |11\rangle$$
$$CNOT|10\rangle = |10\rangle$$
$$CNOT|11\rangle = |01\rangle$$

CNOT in this basis would be $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

Write down the mathematical equivalent of this circuit



$$\text{CNOT}\left(\sigma_x^{(1)} \otimes \mathbb{I}_{2\times 2}^{(2)}\right)|00\rangle$$

Write down the mathematical equivalent of this circuit

$$\mathrm{CNOT}\left(\sigma_x^{(1)} \otimes \mathbb{I}_{2\times 2}^{(2)}\right)|00\rangle = \mathrm{CNOT}\,|10\rangle = |11\rangle$$

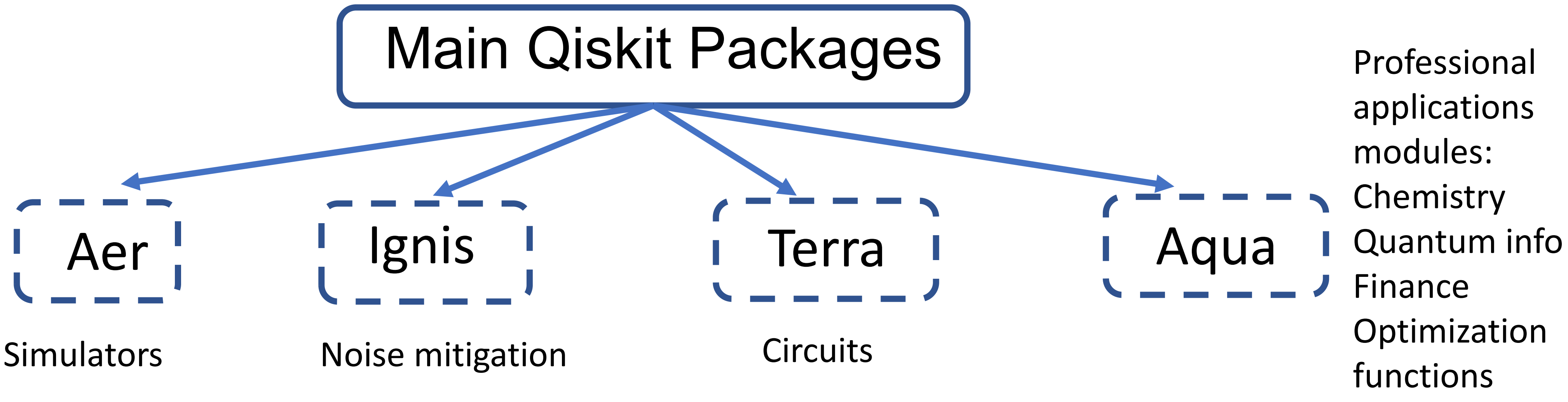
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & 1\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ 1\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & 0\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}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

We will be using

The quantum circuit model

Through

Qiskit
(built on Python) to run IBM Q quantum processors



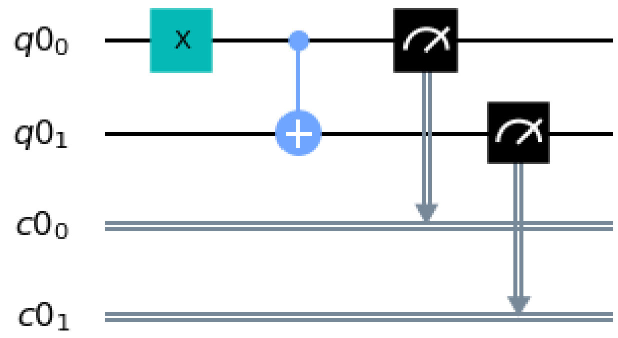
$$\text{CNOT}\left(\sigma_x^{(1)} \otimes \mathbb{I}_{2\times 2}^{(2)}\right)|00\rangle = |11\rangle \longrightarrow$$

```
print(statevector)
```

```
[0.+0.j 0.+0.j 0.+0.j 1.+0.j]
```

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit

'''initializing the circuit with 2 qubits in the quantum register and 2 in
the classical register and their quantum circuit'''

qr = QuantumRegister(2, 'q')
cr = ClassicalRegister(2, 'c')
circ = QuantumCircuit(qr, cr)

## add gates to the quantum circuit

circ.x(qr[0])
circ.cx(qr[0], qr[1])
circ.measure(qr[0], cr[0])
circ.measure(qr[1], cr[1])
```

```
## printing out the circuit in the matplotlib format

circ.draw(output='mpl',cregbundle=False)
```



5 min task :

Find a suitable circuit to generate the **Singlet state**:

$$\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

UP TO A PHASE

In terms of Qiskit it would be :

$$\frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$$

Qasm Simulator and Visualization tools (Bloch sphere and Histograms)

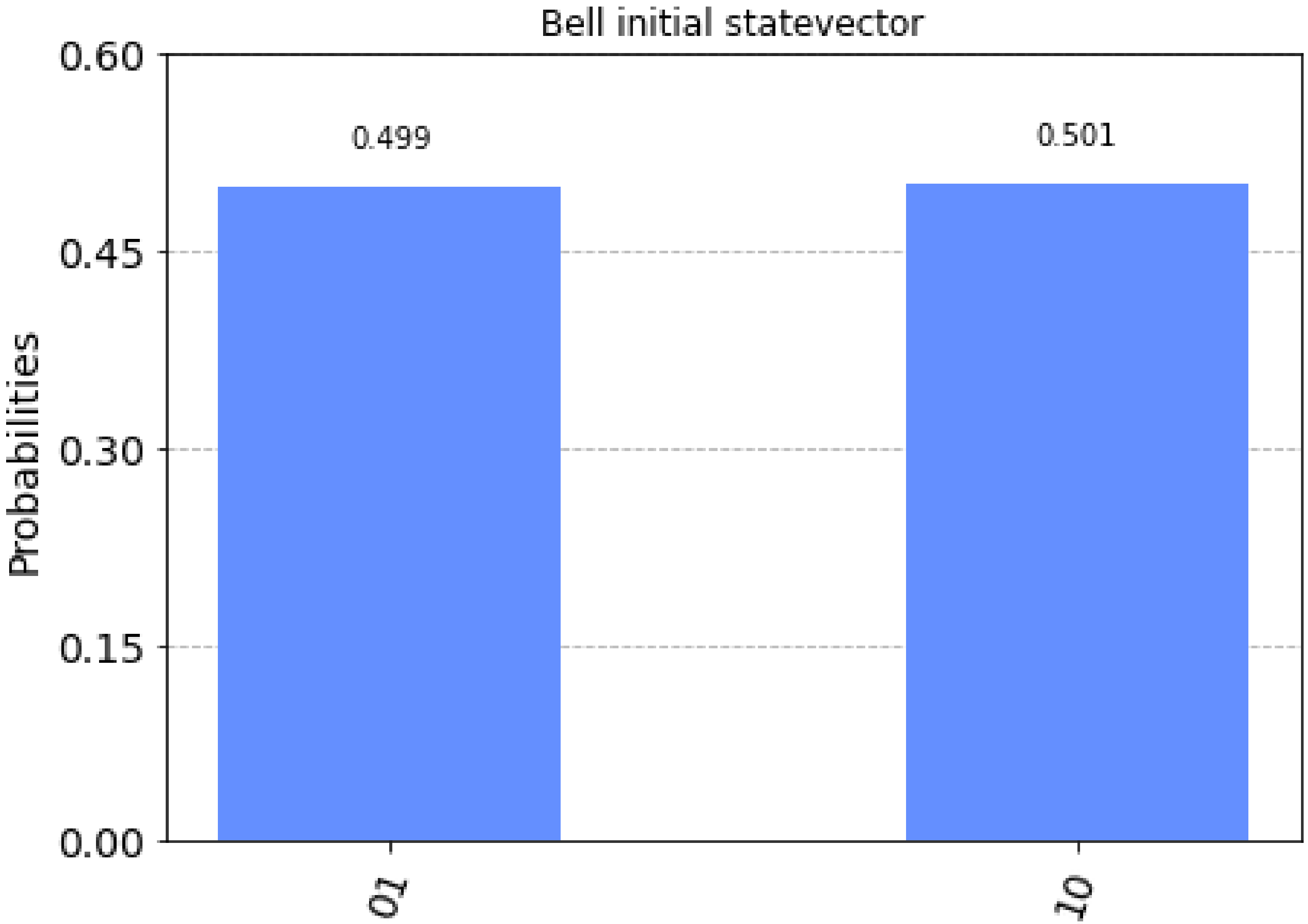
```
## specifying the simulator, we will use the Qasm simulator form the Aer Qiskit package

sim = Aer.get_backend('qasm_simulator')

#############################################
```

```
## run the circuit 'circ' on the Qasm simulator 'sim' storing the results in the "rslt" object
rslt = execute(circ, backend = sim).result()
```

```
#for informative visualization of results we can use qiskit visualization tools but first need
# to import the required tool:

from qiskit.tools.visualization import plot_histogram
```

```
## plotting a histogram of the measurements made by the circuit

plot_histogram(rslt.get_counts(circ),title="Bell initial statevector")
```

Qasm Simulator and Visualization tools (Bloch sphere and Histograms)



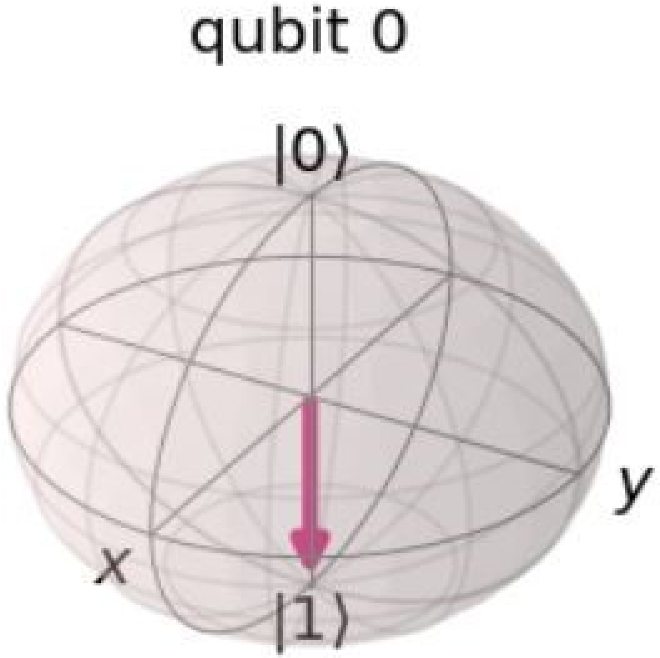
Bloch sphere of a given state

```
## import the plot bloch vector function from Qiksit

from qiskit.tools.visualization import plot_bloch_multivector

# setup a circuit to generate a single qubit state

circ = QuantumCircuit(1,1)
circ.x(0)
S_simulator = Aer.get_backend('statevector_simulator')
rslt = execute(circ, backend = S_simulator).result()
statevector = rslt.get_statevector()

print(statevector)
```

```
[0.+0.j 1.+0.j]
```

Bloch sphere of a general state

```
## given a statevector we can plot it on a bloch sphere
plot_bloch_multivector(statevector)
```

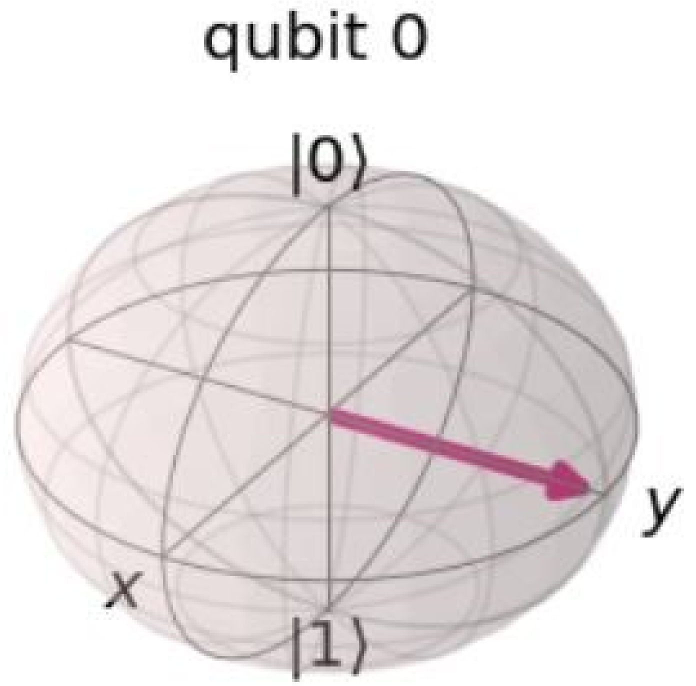


2 Minute Task:

Generate the Bloch sphere of the statevector $|r\rangle$

Visualizing the Bloch sphere of a general state

```
my_statevector = np.array([0.707, 0.0+0.707j])
plot_bloch_multivector(my_statevector)
```



What about General circuits with many qubit gates ?

In 1994, David P. DiVincenzo

“proved that quantum gates operating on just two bits at a time are sufficient to construct a general quantum circuit”

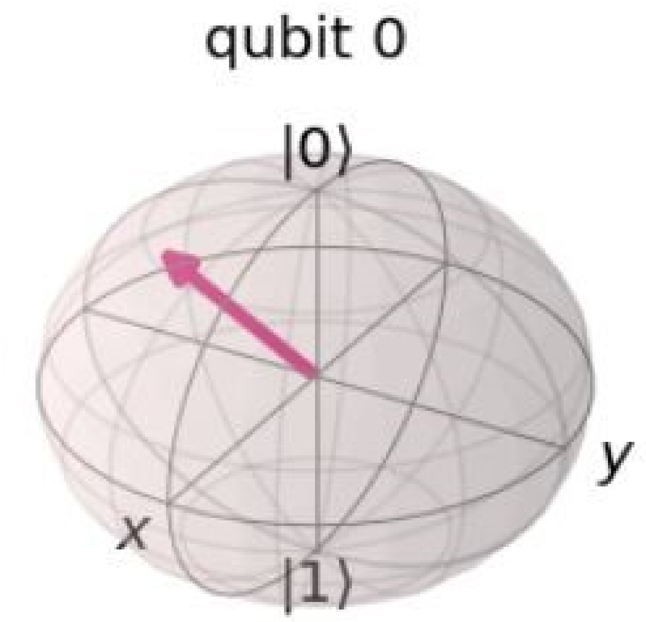
Paper : https://arxiv.org/abs/cond-mat/9407022v1

Quantum information module:

qiskit.quantum_info.<something>....

1. Random statevector

```
from qiskit.quantum_info import random_statevector
our_random_statevector =random_statevector(2, seed=None)
plot_bloch_multivector(our_random_statevector)
```




```
print(our_random_statevector)
```

```
Statevector([-0.42718534-0.64098579j, -0.37066602+0.51889942j],
            dims=(2,))
```

2. Pauli Strings, N-qubit Operators and evolution of statevectors

```
from qiskit.quantum_info.operators import Operator, Pauli
```

Some of the ways you can specify Single or Multiqubit operators:

1. Initiating Pauli Strings ex: XX + ZZ + II

2. Input your own NumPy array for the matrix of the operator

3. Translating a gate into an operator

AGAIN:

In 1994, David P. DiVincenzo

“proved that quantum gates operating on just two bits at a time are sufficient to construct a general quantum circuit”

Paper : https://arxiv.org/abs/cond-mat/9407022v1

3. Density Matrices

Let's find the density matrix of a Singlet:

```
Singlet_statevector_array = np.array([0,-1/np.sqrt(2),1/np.sqrt(2),0])
```

```
Singlet_statevector = Statevector(Singlet_statevector_array)
Singlet_statevector
```

```
Statevector([ 0.        +0.j, -0.70710678+0.j,  0.70710678+0.j,
              0.        +0.j],
            dims=(2, 2))
```

```
My_Singlet_density_matrix = DensityMatrix(Singlet_statevector)
My_Singlet_density_matrix
```

```
DensityMatrix([[ 0. +0.j,  0. -0.j,  0. +0.j,  0. +0.j],
               [ 0. +0.j,  0.5+0.j, -0.5+0.j,  0. +0.j],
               [ 0. +0.j, -0.5-0.j,  0.5+0.j,  0. +0.j],
               [ 0. +0.j,  0. -0.j,  0. +0.j,  0. +0.j]],
              dims=(2, 2))
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & -0.5 & 0 \\ 0 & -0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Let’s Find the Purity of the state:
We can use the statevector or the density matrix

$$\rho^{Singlet} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & -0.5 & 0 \\ 0 & -0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$Purity = tr(\rho^2)$$

For Pure states Purity = 1

Let’s Find the Purity of the state:
We can use the statevector or the density matrix

```
## finding purity given the statevector
Singlet_statevector.purity()
```

```
0.9999999999999996
```

```
## finding purity given the density matrix:
```

```
My_Singlet_density_matrix.purity()
```

```
(0.9999999999999996+0j)
```

Let's Find the reduced density matrix of the Singlet by tracing out one of its subsystems:

2.3.3: Finding reduced density matrices by tracing out subsystems:

```
from qiskit.aqua.utils import get_subsystem_density_matrix
```

```
Singlet_reduced_density_matrix_array = get_subsystem_density_matrix(Singlet_statevector_array,[1])
Singlet_reduced_density_matrix_array
```

```
array([[0.5+0.j, 0. +0.j],
       [0. +0.j, 0.5+0.j]])
```

```
Singlet_reduced_density_matrix = DensityMatrix(Singlet_reduced_density_matrix_array)
Singlet_reduced_density_matrix
```

```
DensityMatrix([[0.5+0.j, 0. +0.j],
               [0. +0.j, 0.5+0.j]],
              dims=(2,))
```

```
Singlet_reduced_density_matrix.purity()
```

```
(0.49999999999999998+0j)
```

This shows that the reduced density matrix of The singlet is MAXIMALLY MIXED with a Purity = I/2

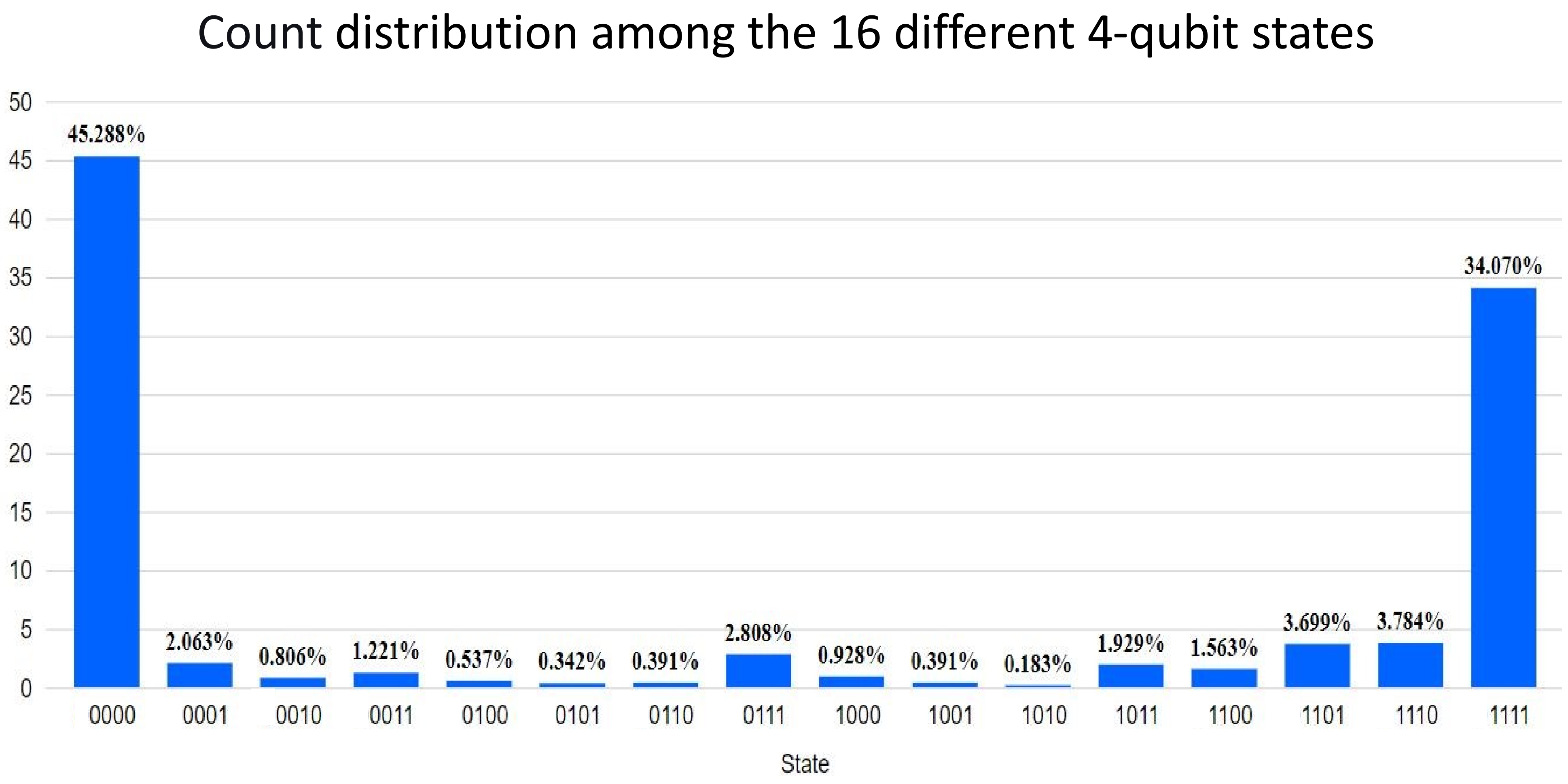
We illustrated how the reduced density matrix of a maximally entangled state, is a maximally mixed state

$$\text{Reduced } \rho^{Singlet} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

Purity of reduced density matrix = 1/2

4-Qubit GHZ State:

$$\frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$$


Count distribution among the 16 different 4-qubit states
50
45
40
35
30
25
20
15
10
5
0
45.288%
2.063%
0.806%
1.221%
0.537%
0.342%
0.391%
2.808%
0.928%
0.391%
0.183%
1.929%
1.563%
3.699%
3.784%
34.070%
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
State

Simulated density matrix

$$\hat{\rho}^{(GHZ)} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

$$\underbrace{\rho^{(/A)} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}}_{\text{Reduced density matrix of partition } A}$$

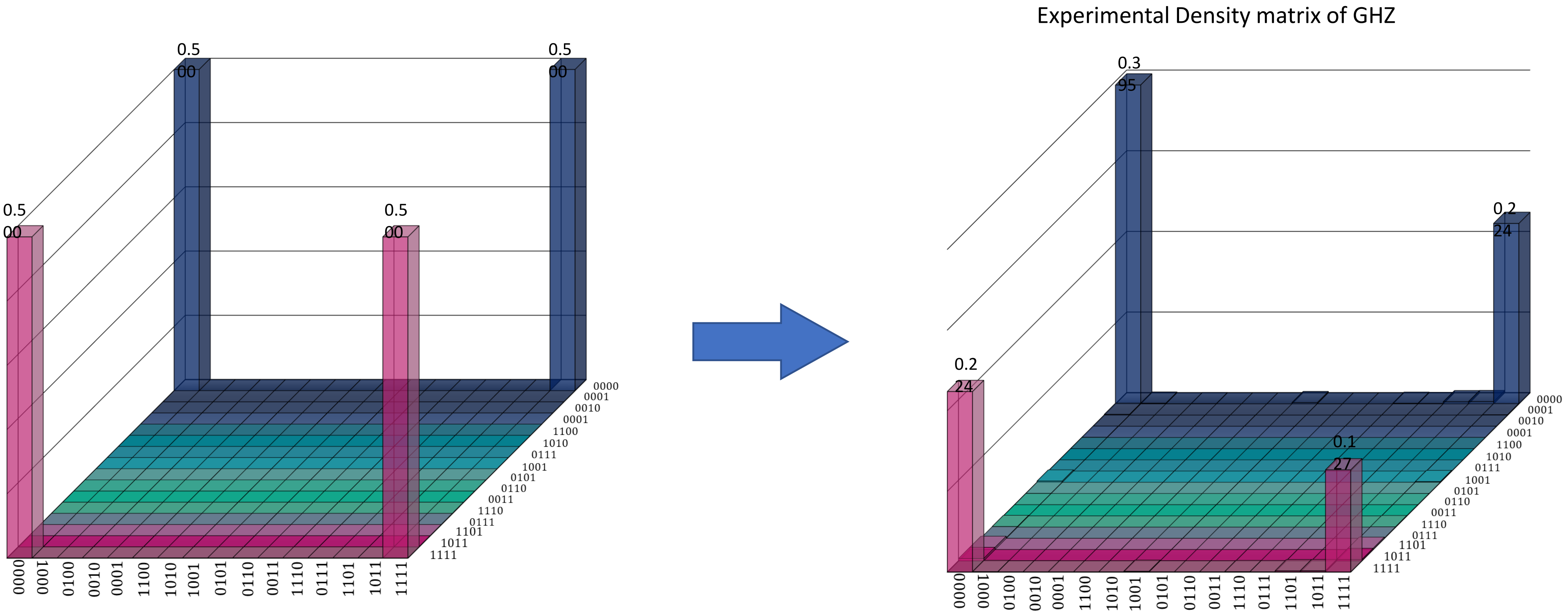
$$\underbrace{\rho^{(/B)} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}}_{\text{Reduced density matrix of partition } B}$$

Quantum computer generated density matrix

$$\hat{\rho}^{GHZ}_{IBM} = \begin{bmatrix} 0.39 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0.002 & 0 & 0 & 0 & 0.001 & 0 & 0.003 & 0.003 & 0.224 \\ 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.002 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.003 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 \\ 0.003 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002 \\ 0.224 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0.002 & 0.127 \end{bmatrix}$$

$$\underbrace{\rho^{(/A)} = \begin{bmatrix} 0.39 & 0.001 \\ 0.001 & 0.127 \end{bmatrix}}_{\text{Reduced density matrix of partition } A}$$

$$\underbrace{\rho^{(/B)} = \begin{bmatrix} 0.39 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0.002 \\ 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002 \\ 0.002 & 0 & 0 & 0 & 0 & 0.001 & 0.002 & 0.127 \end{bmatrix}}_{\text{Reduced density matrix of partition } B}$$


Experimental Density matrix of GHZ
0.5
00
0.5
00
0.5
00
0.5
00
0.3
95
0.2
24
0.2
24
0.1
27
0000
0001
0010
0001
1100
1010
0111
1001
0101
0110
0011
1110
0111
1101
1011
1111
0000
1000
0100
0010
0001
1100
1010
1001
0101
0110
0011
1110
0111
1101
1011
1111