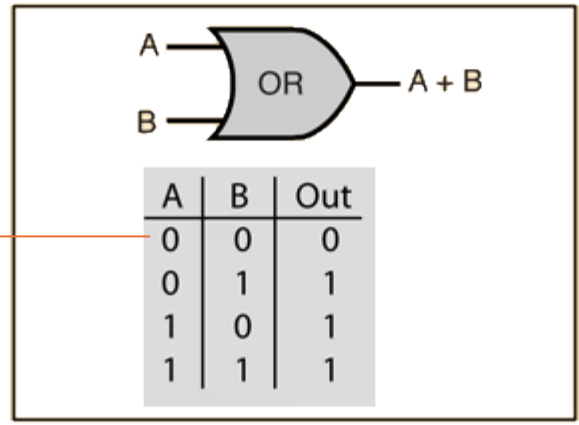
OR gate Implementation with Qiskit



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

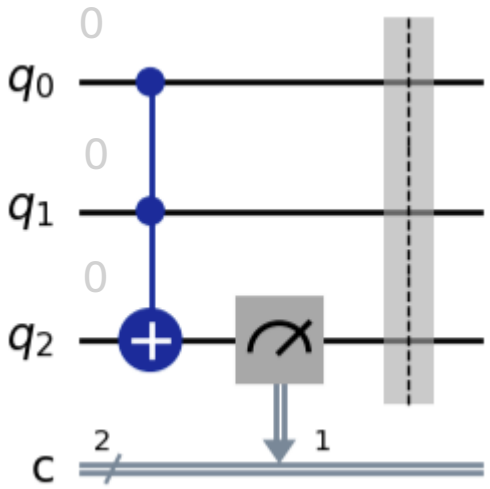
Code

```
from qiskit import *
from qiskit.visualization import plot_histogram
from qiskit_aer import Aer
from math import pi
from qiskit.visualization import  plot_bloch_multivector
from qiskit.visualization import array_to_latex
%matplotlib inline
```

Case 1
A= 0, B= 0, Output = 0

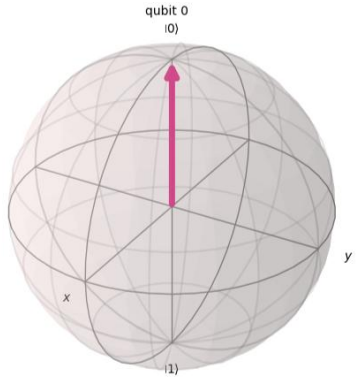
```
circuit= QuantumCircuit(3,2)
circuit.ccx(0,1,2)
circuit.measure(2,1)
circuit.barrier()
circuit.draw('mpl')
```

Output Circuit Code

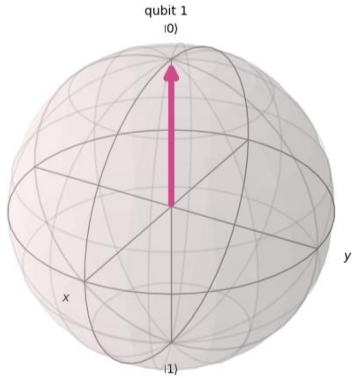



```
]: simulator = Aer.get_backend('statevector_simulator')
   result = simulator.run(circuit).result()
   plot_bloch_multivector(result.get_statevector())
```

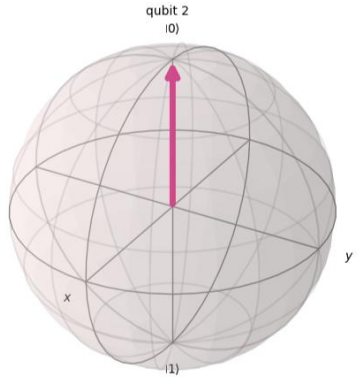
Bloch Sphere



$q_0 = 0$



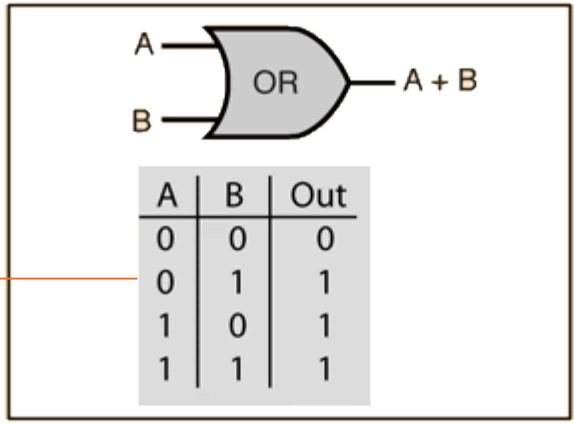
$q_1 = 0$



$q_2 = 0$

umair

OR gate Implementation with Qiskit



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

Case 2
A= 0, B= 1, Output = 1

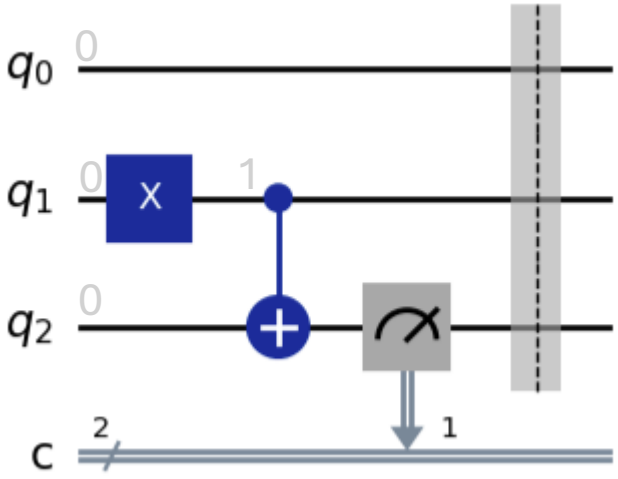
Code

```
from qiskit import *
from qiskit.visualization import plot_histogram
from qiskit_aer import Aer
from math import pi
from qiskit.visualization import  plot_bloch_multivector
from qiskit.visualization import array_to_latex
%matplotlib inline
```

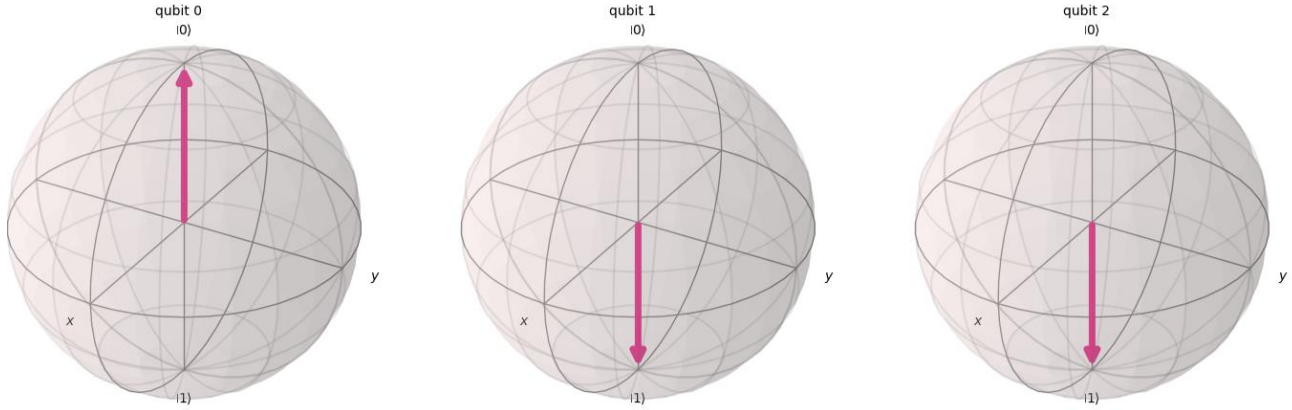
```
circuit= QuantumCircuit(3,2)
#circuit.x(0)
circuit.x(1)
circuit.cx(1,2)
circuit.measure(2,1)
circuit.barrier()
circuit.draw('mpl')
```

```
]: simulator = Aer.get_backend('statevector_simulator')
result = simulator.run(circuit).result()
plot_bloch_multivector(result.get_statevector())
```

Output Circuit Code



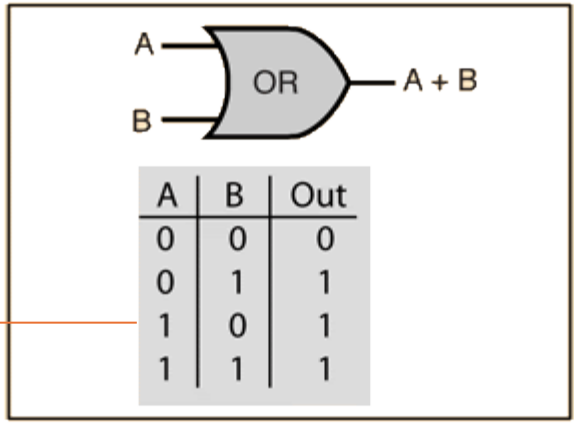
Bloch Sphere



$q_0 = 0$ $q_1 = 1$ $q_2 = 1$

umair

OR gate Implementation with Qiskit



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

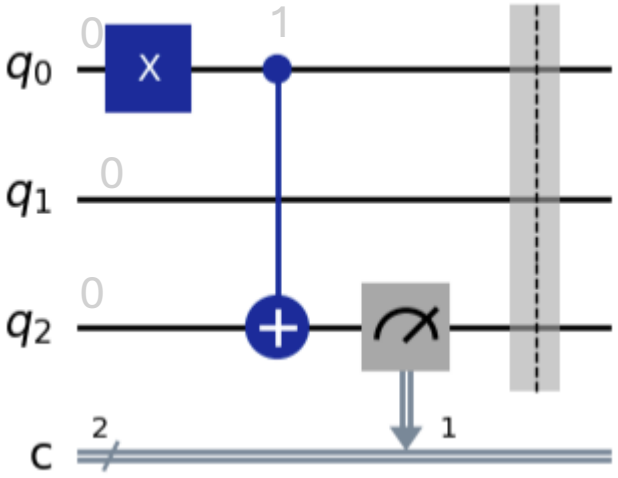
Code

```
from qiskit import *
from qiskit.visualization import plot_histogram
from qiskit_aer import Aer
from math import pi
from qiskit.visualization import  plot_bloch_multivector
from qiskit.visualization import array_to_latex
%matplotlib inline
```

Case 3
A= 1, B= 0, Output = 1

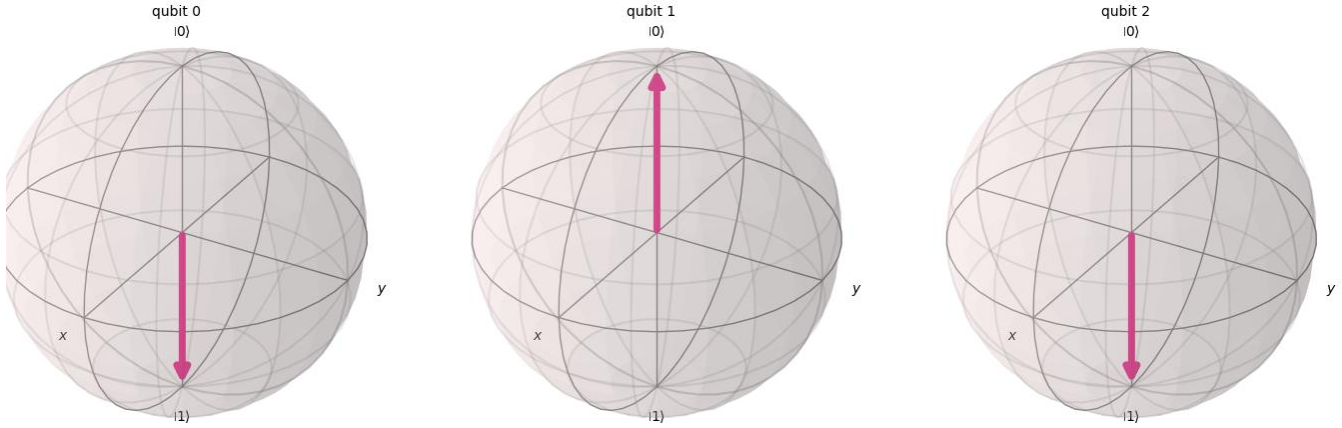
```
circuit= QuantumCircuit(3,2)
#circuit.x(0)
circuit.x(1)
circuit.cx(1,2)
circuit.measure(2,1)
circuit.barrier()
circuit.draw('mpl')
```

Output Circuit Code




```
]: simulator = Aer.get_backend('statevector_simulator')
result = simulator.run(circuit).result()
plot_bloch_multivector(result.get_statevector())
```

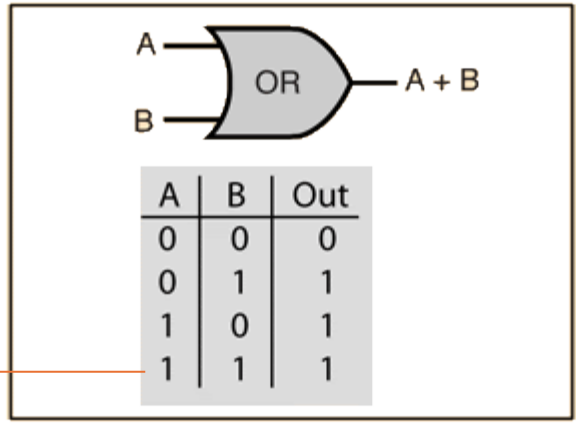
Bloch Sphere



$q_0 = 1$ $q_1 = 0$ $q_2 = 1$

umair

OR gate Implementation with Qiskit



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

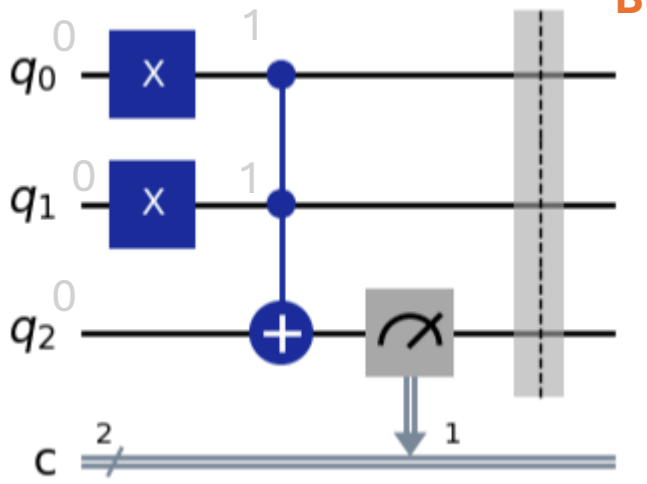
Case 4
A= 1, B= 1, Output = 1

Code

```
from qiskit import *
from qiskit.visualization import plot_histogram
from qiskit_aer import Aer
from math import pi
from qiskit.visualization import  plot_bloch_multivector
from qiskit.visualization import array_to_latex
%matplotlib inline
```

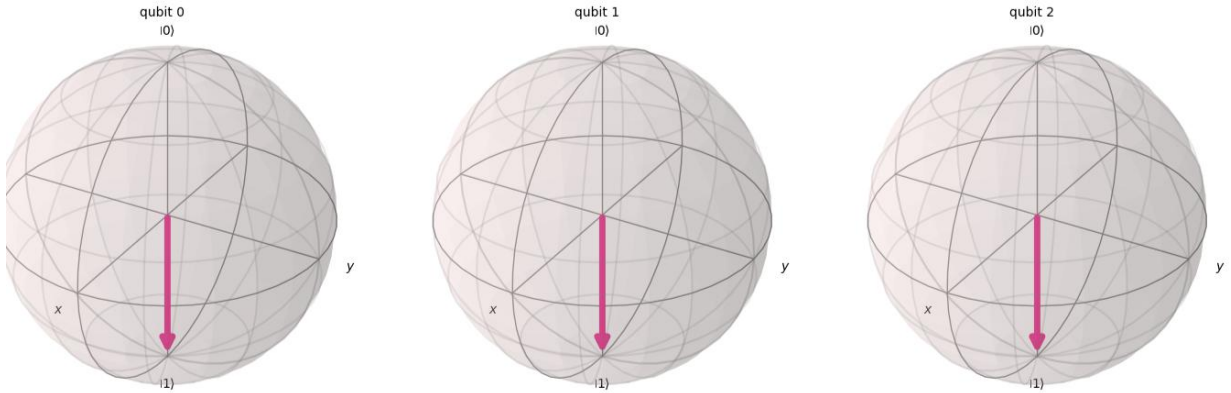
```
circuit= QuantumCircuit(3,2)
circuit.x(0)
circuit.x(1)
#circuit.cx(0,2)
circuit.ccx(0,1,2)
circuit.measure(2,1)
circuit.barrier()
circuit.draw('mpl')
```

Output Circuit Code




```
]: simulator = Aer.get_backend('statevector_simulator')
   result = simulator.run(circuit).result()
   plot_bloch_multivector(result.get_statevector())
```

Bloch Sphere



$q_0 = 1$ $q_1 = 1$ $q_2 = 1$

umair