پردازش اطلاعات کوانتومی

نام و نام خانوادگی: مهدی وجهی

۸۱۰۱۰۱۵۵۸

پروژه ۲

۱ حالت درهم تنیده بل

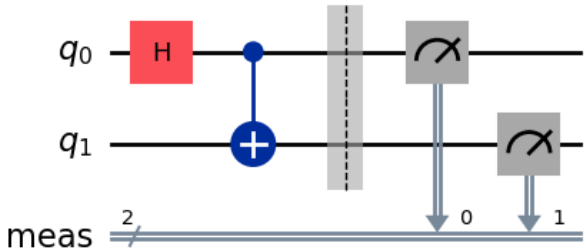
کد آن به شکل زیر است:

```
qc = QuantumCircuit(2)
qc.h([0])
qc.cx([0],[1])
qc.measure_all()
```

سپس با دستور زیر مدار را نمایش می دهیم:

```
unitary_operator = Operator(qc)
final_matrix = unitary_operator.data
np.set_printoptions(precision=1, suppress=True, linewidth=np.inf)
print(final_matrix)
display(qc.draw(output="mpl"))
```

که خروجی آن هست:



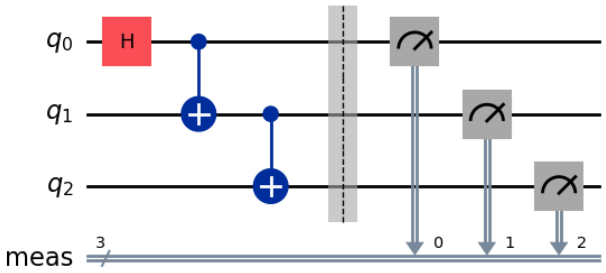
شکل ۱: مدار سوال ۱

```
[[ 0.7+0.j  0.7+0.j  0. +0.j  0. +0.j]
 [ 0. +0.j  0. +0.j  0.7+0.j -0.7+0.j]
 [ 0. +0.j  0. +0.j  0.7+0.j  0.7+0.j]
 [ 0.7+0.j -0.7+0.j  0. +0.j  0. +0.j]]
```

تقریبا ما در تمام مدار های کوانتومی از این حالت در هم تنیدگی استفاده می کنیم. این موضوع در الگوریتم هایی مانند دویچ، دویچ جوزا، وزیرانی و... همگی از آن بهره می برند علاوه بر آن در کدگذاری متراکم وتوزیع کلید کوانتومی ضروری است.

مدار حالت ۳ کیوبیتی را پیاده می کنیم:

```
qc = QuantumCircuit(3)
qc.h([0])
qc.cx([0,1],[1,2])
```



شکل ۲: مدار سوال ۱ بخش ۳

۲ QFT

مدار را با کد زیر تولید می کنیم. به دلیل سادگی از شرح آن صرف نظر می کنیم اما به صورت کلی روی هر کیوبیت حلقه می زند و هادامارد و گیت کنترلی را می گذارد و در آخر هم سواپ می کند.

```
def QFT(qc):
  qc.barrier()
  for i in range(qc.num_qubits - 1, -1,-1):
    qc.h(i)
```

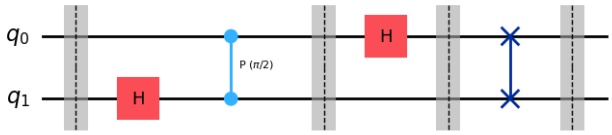
```
    for j in range(i-1, -1, -1):
      qc.cp(np.pi/2**(i-j), j, i)
    qc.barrier()
  for i in range(qc.num_qubits//2):
    qc.swap(i, qc.num_qubits - i - 1)
  qc.barrier()
```

حال یک مدار دو کیوبیتی ورودی می دهیم و خروجی ها را نمایش می دهیم:

```
n = 2
qc = QuantumCircuit(n)
# qc.x([i for i in range(n) if random.random() < 0.5])
# qc.x([1])
QFT(qc)
display(qc.draw(output="mpl"))
unitary_operator = Operator(qc)
final_matrix = unitary_operator.data
np.set_printoptions(precision=1, suppress=True, linewidth=np.inf)
print(final_matrix)
qc.save_statevector()
simulator = AerSimulator()
job = simulator.run(qc)
result = job.result()
statevector = result.get_statevector(qc)
display(plot_bloch_multivector(statevector))
```

خروجی آن به شکل زیر است:

```
[[ 0.5+0.j   0.5+0.j   0.5+0.j   0.5+0.j ]
 [ 0.5+0.j   0. +0.5j -0.5+0.j  -0. -0.5j]
 [ 0.5+0.j  -0.5+0.j   0.5+0.j  -0.5+0.j ]
 [ 0.5+0.j  -0. -0.5j -0.5+0.j   0. +0.5j]]
```



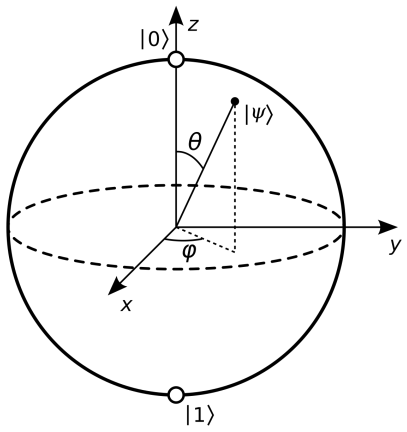
شکل ۳: مدار QFT

۳ کره بلاچ

ما یک حالت کوانتومی را می توان به صورت زیر نمایش دهیم:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

بعد به صورت زیر آن را روی کره بلاچ می توانیم نمایش دهیم:



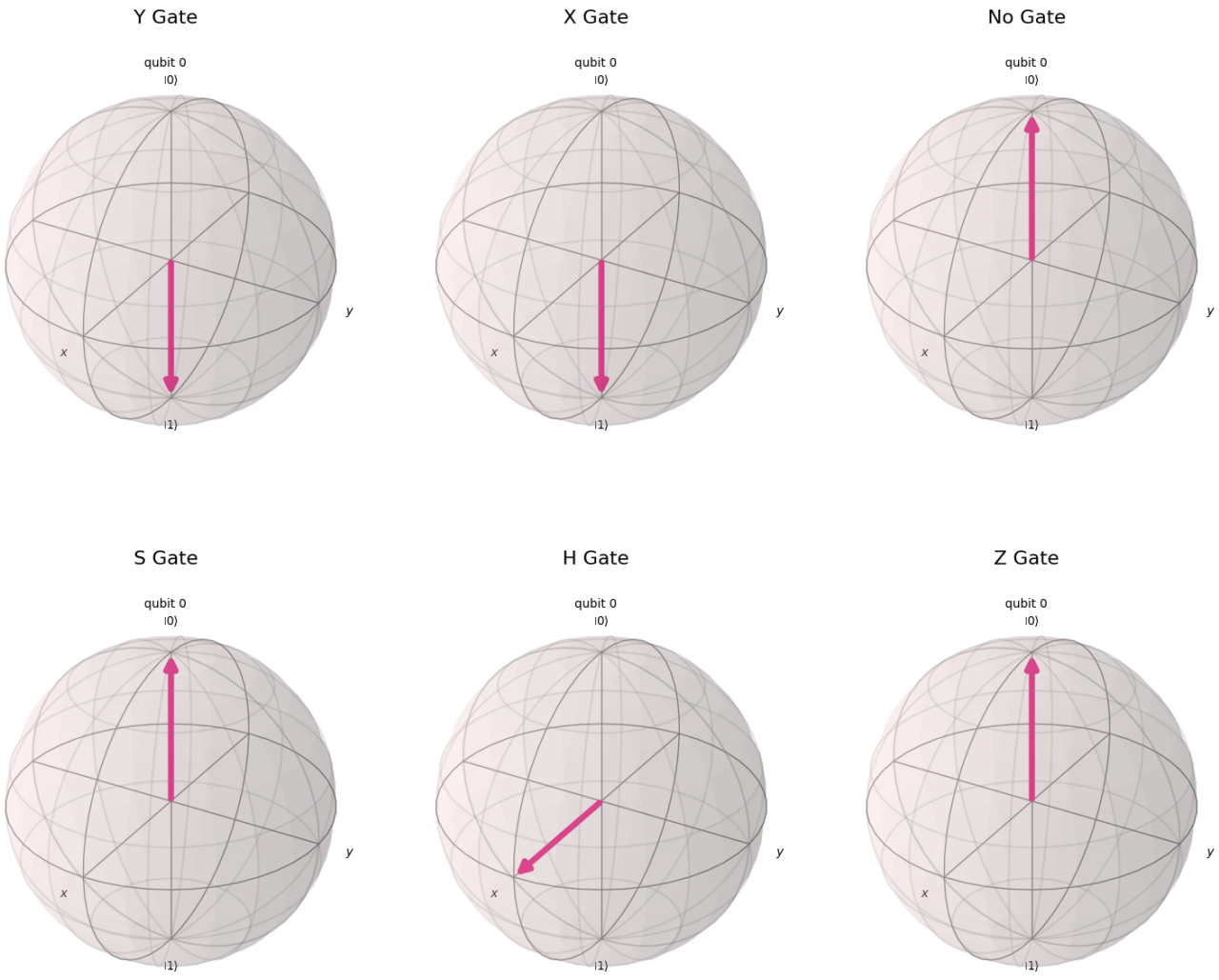
شکل ۴: کره بلاچ

سپس گیت های مختلف را اعمال می کنیم و خروجی را نمایش می دهیم. کد آن به شکل زیر است:

```
gate = [
    ['No Gate', lambda qc: None],
    ['X Gate', lambda qc: qc.x(0)],
    ['Y Gate', lambda qc: qc.y(0)],
    ['Z Gate', lambda qc: qc.z(0)],
    ['H Gate', lambda qc: qc.h(0)],
    ['S Gate', lambda qc: qc.s(0)],
]

for t, g in gate:
  print(t)
  qc = QuantumCircuit(1)
```

```
13 g(qc)
14 display(qc.draw(output="mpl"))
15
16 qc.save_statevector()
17 simulator = AerSimulator()
18 job = simulator.run(qc)
19 result = job.result()
20 statevector = result.get_statevector(qc)
21 display(plot_bloch_multivector(statevector, title=t))
```



شکل ۵: کره های بلاچ