پردازش اطلاعات کوانتومی

نام و نام خانوادگی: مهدی وجهی

۸۱۰۱۰۱۵۵۸

پروژه ۲

# ۱ حالت درهم تنیده بل

حالت‌های بل چهار حالت کوانتومی خاص برای دو کیوبیت هستن که بهشون درهم‌تنیده‌ی ماکزیمم می‌گن. ویژگی اصلی‌شون اینه که اگرچه وضعیت کل سیستم دقیقاً مشخصه، اما وضعیت هر کیوبیت به تنهایی قابل تعیین نیست.
از نظر ریاضی این چهار حالت یک پایه برای فضا می‌سازن. دو تا حالت اول که بهشون حالت‌های فی می‌گن، همبستگی مستقیم دارن (یعنی هر دو کیوبیت یا صفرن یا یک):

$$\left|\Phi^{\pm}\right\rangle = \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}$$

دو تا حالت بعدی حالت‌های سای هستن که همبستگی معکوس دارن :

$$\left|\Psi^{\pm}\right\rangle = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}$$

برای ساختن این حالت‌ها، مثلاً برای رسیدن به $|\Phi^{+}\rangle$، کار رو از حالت پایه $|00\rangle$ شروع می‌کنیم. اول یک گیت هادامارد روی کیوبیت اول اعمال می‌کنیم تا برهم‌نهی ایجاد بشه و بلافاصله بعدش یک گیت سی‌نات (با کنترل کیوبیت اول و هدف کیوبیت دوم) می‌زنیم تا درهم‌تنیدگی شکل بگیره:

$$|00\rangle \xrightarrow{H\otimes I} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{CNOT} \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \left|\Phi^{+}\right\rangle$$

کد آن به شکل زیر است:

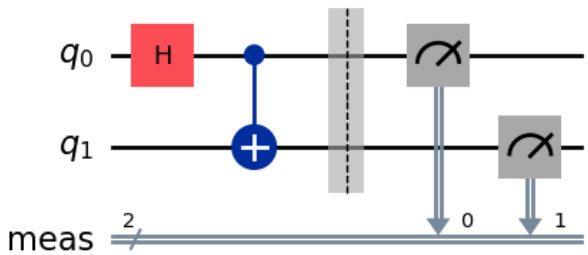
```
qc = QuantumCircuit(2)
qc.h([0])
qc.cx([0],[1])
qc.measure_all()
```

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]]
```

یک مدار دو کیوبیتی می‌سازیم که پیش‌فرض توی حالت پایه $|00\rangle$ قرار داره. قدم اول اینه که با دستور $\mathrm{qc.h}(0)$ یه گیت هادامارد روی کیوبیت اول اعمال کنیم. این کار باعث می‌شه کیوبیت اول از حالت قطعی دربیاد و بره توی حالت برهم‌نهی متوازن.

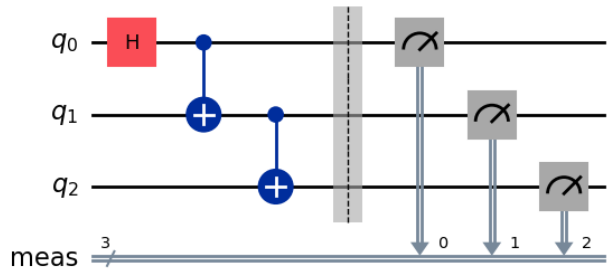
بعد از اون، با دستور $\mathrm{qc.cx}(0,\ 1)$ گیت سی‌نات را استفاده. توی این گیت، کیوبیت اول نقش کنترل‌کننده رو داره و کیوبیت دوم هدف هست. این کار باعث می‌شه وضعیت کیوبیت دوم به اولی وابسته بشه و اون درهم‌تنیدگی‌ای که دنبالش بودیم شکل بگیره.

در نهایت با استفاده از کلاس اپراتور، ماتریس یکانی کل مدار رو می‌کشیم بیرون که نشون می‌ده بردار حالت چطوری تغییر کرده. آخر سر هم با دستور measure_all، یه اندازه‌گیری روی هر دو تا کیوبیت انجام می‌دیم که اگه مدار درست کار کرده باشه، خروجی‌مون باید فقط حالت‌های $|00\rangle$ و $|11\rangle$ با احتمال ۵۰-۵۰ باشه.

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 $QFT$

تبدیل فوریه کوانتومی دقیقاً همون کاری رو می‌کنه که تبدیل فوریه گسسته (□□□) توی پردازش سیگنال کلاسیک انجام می‌ده، با این تفاوت که اینجا روی دامنه‌ی احتمالاتی حالت‌های کوانتومی اعمال می‌شه. ایده‌ی اصلی اینه که بردار حالت رو از پایه محاسباتی به پایه فوریه ببریم. این تبدیل بخش اصلی خیلی از الگوریتم‌های مهم مثل الگوریتم شور و تخمین فاز رو تشکیل می دهد. اگه بخوایم معادله ریاضی رو بنویسیم، برای یک حالت پایه $|j\rangle$ در یک سیستم $N$ بعدی (که $N = 2^n$)، تبدیل به صورت زیر تعریف می‌شه:

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{jk}{N}} |k\rangle$$

ساختار مدارش خیلی جالبه و برخلاف نوع کلاسیکش که پیچیدگی محاسباتی زیادی داره، اینجا با تعداد گیت‌های کمی اجرا می‌شه. مدار کیو-اف-تی از دو نوع گیت اصلی تشکیل شده: یکی گیت هادامارد که برهم‌نهی رو می‌سازه و دیگری گیت‌های فاز کنترل‌شده. روند کار این‌طوریه که روی هر کیوبیت اول یه هادامارد می‌زنیم و بعدش با بقیه‌ی کیوبیت‌ها گیت فاز کنترل‌شده اعمال می‌کنیم. نکته‌ی مهم اینجاست که خروجی این مدار ترتیب بیت‌هاش برعکس می‌شه، پس آخر سر باید با چند تا گیت اسواپ ترتیب کیوبیت‌ها رو درست کنیم تا خروجی قابل خوندن بشه.

پیچیدگی زمانی این الگوریتم یکی از بزرگترین دستاوردهای محاسبات کوانتومی محسوب می‌شه. در حالی که نیاز به $O(n2^n)$ عملیات داره، تبدیل فوریه کوانتومی فقط با $O(n^2)$ گیت انجام می‌شه که یک شتاب‌دهی نمایی رو نشون می‌ده.

مطابق توضیحات کد را می نویسیم.

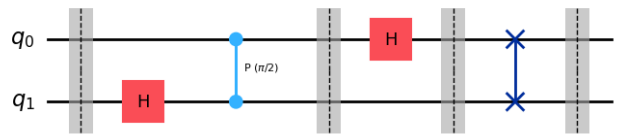
```
def QFT(qc):
  qc.barrier()
  for i in range(qc.num_qubits - 1, -1,-1):
    qc.h(i)
    for j in range(i-1, -1, -1):
      qc.cp(np.pi/2**(i-j), j, i)
    qc.barrier()
  for i in range(qc.num_qubits//2):
    qc.swap(i, qc.num_qubits - i - 1)
  qc.barrier()
```

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 $QFT$

در تابعی که برای پیاده‌سازی کیو-اف-تی نوشتیم، منطق مدار به دو بخش تقسیم می‌شه. بخش اول شامل دو حلقه‌ی تو در تو هست؛ حلقه‌ی بیرونی روی کیوبیت‌ها حرکت می‌کنه و با اعمال گیت هادامارد، برهم‌نهی رو می‌سازه. بلافاصله بعد از اون، حلقه‌ی داخلی وارد عمل می‌شه و $Controlled-Phase$ رو با زوایای $\frac{\pi}{2^k}$ اعمال می‌کنه که $k$ فاصله‌ی بین دو کیوبیت هست. بخش دوم تابع هم مربوط به اصلاح ترتیب بیت‌هاست؛ چون خروجی ریاضی تبدیل فوریه ترتیبش عکس ورودی‌هاست، ما با یک حلقه و استفاده از گیت اسواپ، کیوبیت اول رو با آخر و به همین ترتیب بقیه رو جابجا می‌کنیم.

در مورد خروجی عددی، چون در کد ما گیت‌های ایکس کامنت شده بودن، ورودی مدار حالت پایه $|00\rangle$ بود. ماتریس ۴ در ۴ که چاپ شده، در واقع همون ماتریس عملگر تبدیل فوریه برای دو کیوبیت هست. اگر دقت کنیم تمام درایه‌ها اندازشون ۰.۵ هست (که همون $\frac{1}{\sqrt{4}}$ می‌شه).

ستون اول این ماتریس نشون می‌ده که حالت $|00\rangle$ به چه چیزی تبدیل شده. می‌بینیم که تمام عناصر ستون اول ۰.۵ هستن، یعنی حالت $|00\rangle$ به یک برهم‌نهی کاملاً متوازن تبدیل شده:

$$QFT\,|00\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = |{+}{+}\rangle$$

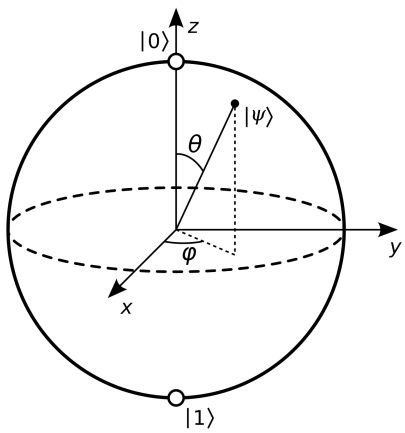
# ۳ کره بلاچ

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

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

این نمایش هندسی خیلی به درک شهودی ما کمک می‌کنه و در واقع یک نگاشت از فضای پیچیده‌ی ریاضی به یک فضای سه بعدی.اگرچه یک کیوبیت با دو عدد مختلط توصیف می‌شه (۴ درجه آزادی)، اما چون مجموع احتمالات باید یک بشه و فاز کلی یا همون گلوبال فاز هم توی آزمایش‌های فیزیکی بی‌اثیره، ما می‌تونیم تمام حالت‌های ممکن رو فقط با همین دو تا زاویه $\theta$ و $\phi$ خلاصه کنیم.

توی این مدل، زاویه‌ی $\theta$ نشون‌دهنده‌ی عرض جغرافیاییه؛ اگر صفر باشه روی قطب شمالیم (حالت $|0\rangle$) و اگر $\pi$ باشه می‌رسیم به قطب جنوب (حالت $|1\rangle$). هر چقدر این زاویه تغییر کنه یعنی داریم سهم احتمالات صفر و یک رو عوض می‌کنیم (مثلاً روی خط استوا احتمال‌ها ۵۰-۵۰ می‌شه). زاویه‌ی $\phi$ هم طول جغرافیایی رو نشون می‌ده و بیانگر فاز نسبی بین دو حالت پایه است که باعث چرخش دور محور $Z$ می‌شه. در نهایت، هر گیت تک‌کیوبیتی که روی سیستم اعمال کنیم، دقیقاً مثل اینه که داریم این بردار رو روی سطح کره می‌چرخونیم.
بعد به صورت زیر آن را روی کره بلاچ می توانیم نمایش دهیم:




شکل ۴: کره بلاچ

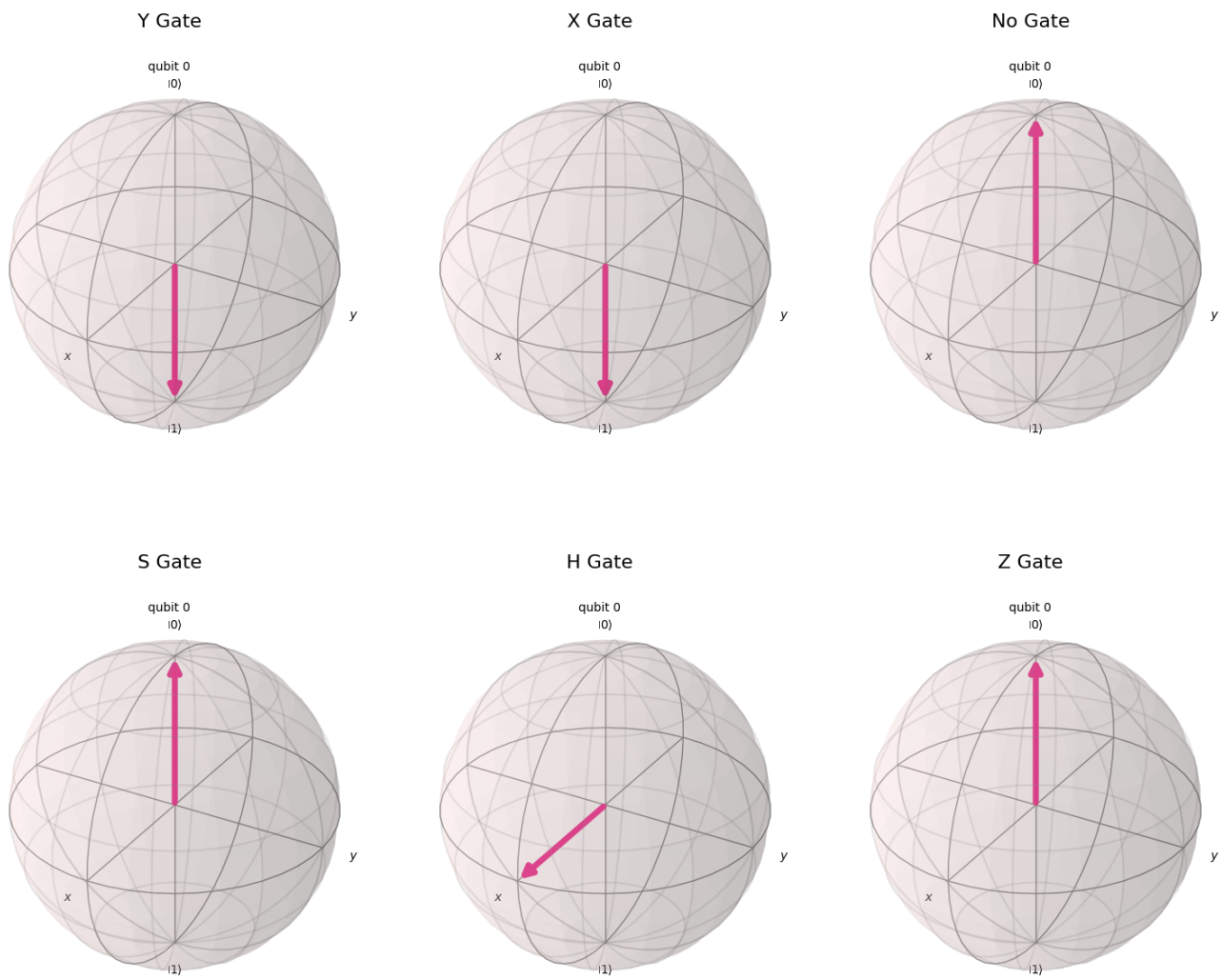
سپس گیت های مختلف را اعمال می کنیم و خروجی را نمایش می دهیم. کد آن به شکل زیر است:

```
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)
  g(qc)
  display(qc.draw(output="mpl"))

  qc.save_statevector()
  simulator = AerSimulator()
  job = simulator.run(qc)
  result = job.result()
```

```
statevector = result.get_statevector(qc)
display(plot_bloch_multivector(statevector, title=t))
```




شکل ۵: کره های بلاچ

اول از همه حالت اولیه یا بدون گیت رو داریم. کیوبیت‌ها پیش‌فرض با $|0\rangle$ مقداردهی می‌شن، پس طبیعیه که بردار ما دقیقاً روی قطب شمال (محور $+Z$) ایستاده باشه.

وقتی گیت $X$ (یا همون نقیض) رو اعمال می‌کنیم، مثل این می‌مونه که توپ رو ۱۸۰ درجه حول محور $X$ چرخونده باشیم؛ پس بردار از قطب شمال مستقیم می‌ره به قطب جنوب (حالت $|1\rangle$). گیت $Y$ هم نتیجه‌اش مشابهه و بردار رو به قطب جنوب می‌بره ($i\,|1\rangle$)، فقط محور چرخش فرق داره ولی چون روی کره بلاچ فاز جهانی دیده نمی‌شه، خروجی

بصری‌ش دقیقاً مثل $X$ هست. اما گیت $Z$ فرق می‌کنه؛ این گیت یک چرخش حول محور عمودیه. چون بردار ما خودش الان روی محور عمودی (قطب شمال) قرار داره، هرچقدر هم دور خودش بچرخه، جاش عوض نمی‌شه و همون بالا می‌مونه.

حالا می‌رسیم به گیت هادامارد که ابزار اصلی تولید برهم‌نهی هست. این گیت بردار رو از قطب شمال می‌کشه پایین و می‌ذاره روی خط استوا در جهت محور $+X$ که بهش حالت $|+\rangle$ می‌گیم.

در نهایت گیت $S$ رو داریم که در واقع ریشه‌ی دوم گیت $Z$ هست (چرخش ۹۰ درجه فازی). دقیقاً مثل گیت $Z$، چون داره حول محوری می‌چرخه که بردارمون روی اون قرار داره، هیچ تغییر بصری روی حالت $|0\rangle$ ایجاد نمی‌کنه.