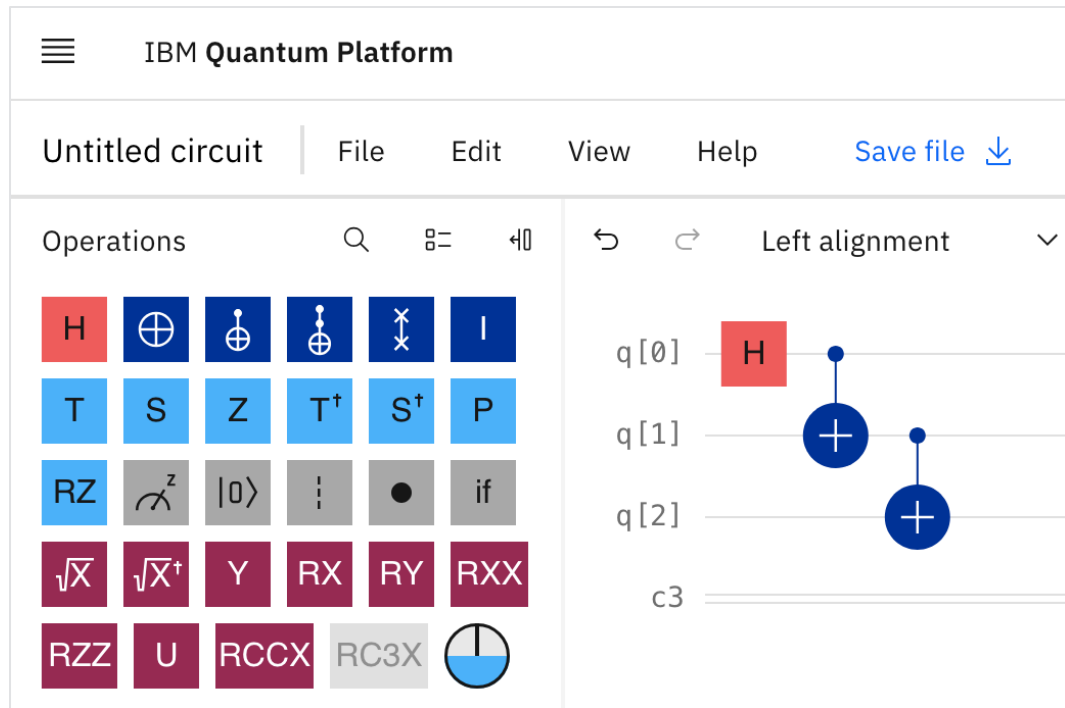


Kawasaki Quantum Summer Camp 2025

量子ゲート基礎 IBM Quantum Composer

Jul 30, 2025

沼田祈史
Kifumi Numata
IBM Quantum



いつも使っている
コンピューターのビット

0 または 1

どちらか

コイン



おもて

コイン



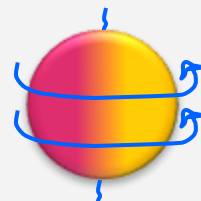
うら

量子コンピューターの
量子ビット

0 と 1

両方

くるくる回っているコイン (イメージ)



測定すると表か裏にバシッと決まる

いつも使っている
コンピューターのビット

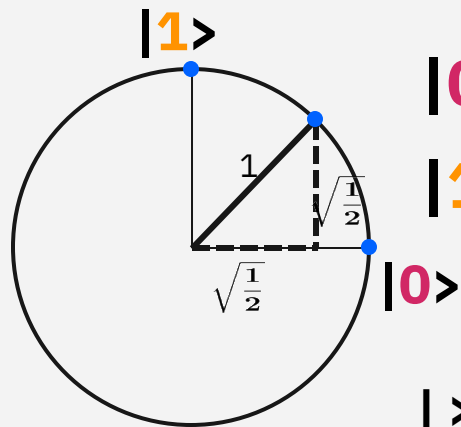
0 または **1**

どちらか

量子コンピューターの
量子ビット

$$\alpha \times |0\rangle + \beta \times |1\rangle$$

0 と 1 の「重ね合わせ」

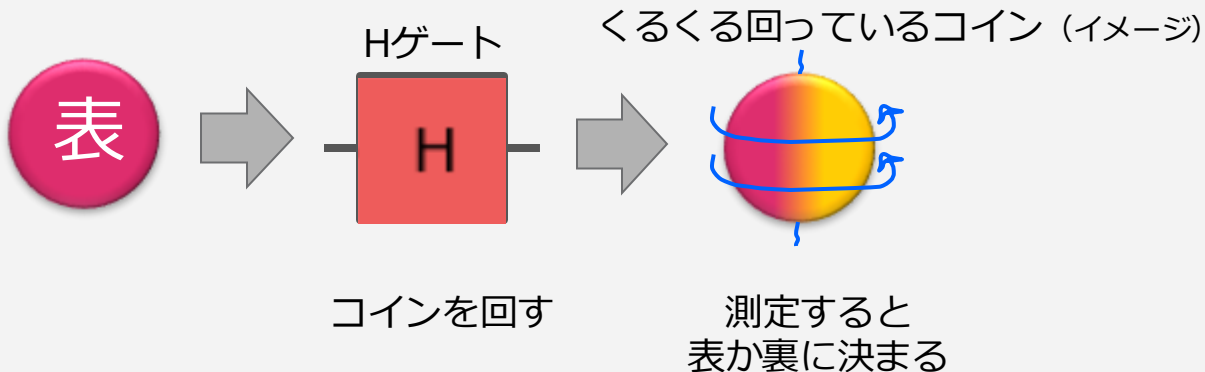
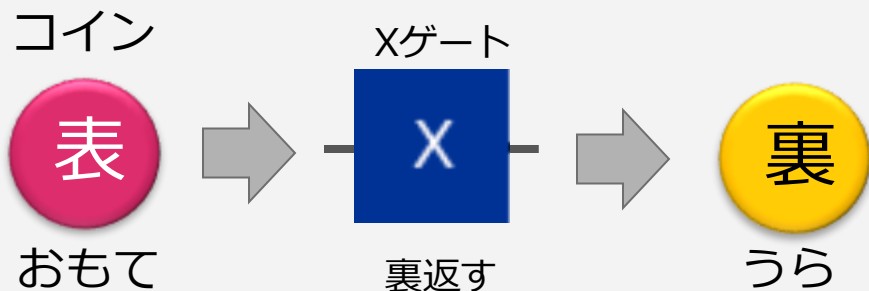


$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

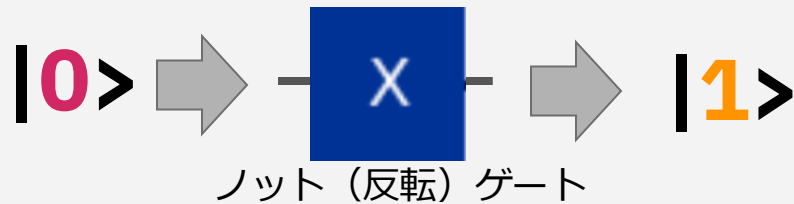
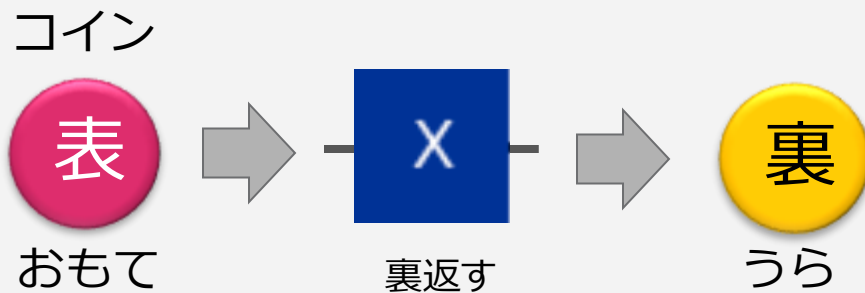
$$|1\rangle = 0 \times |0\rangle + 1 \times |1\rangle$$

$| \rangle$: 量子ビットを表す記号⁶

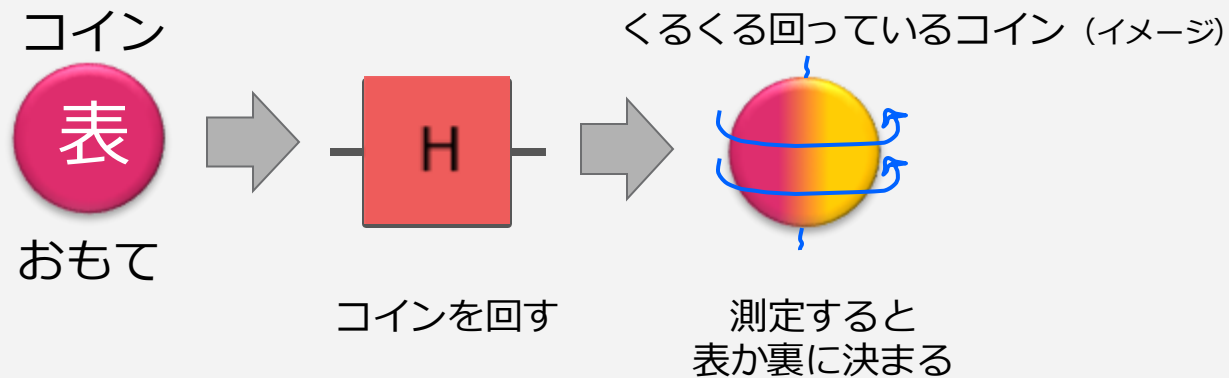
量子コンピューターの計算方法



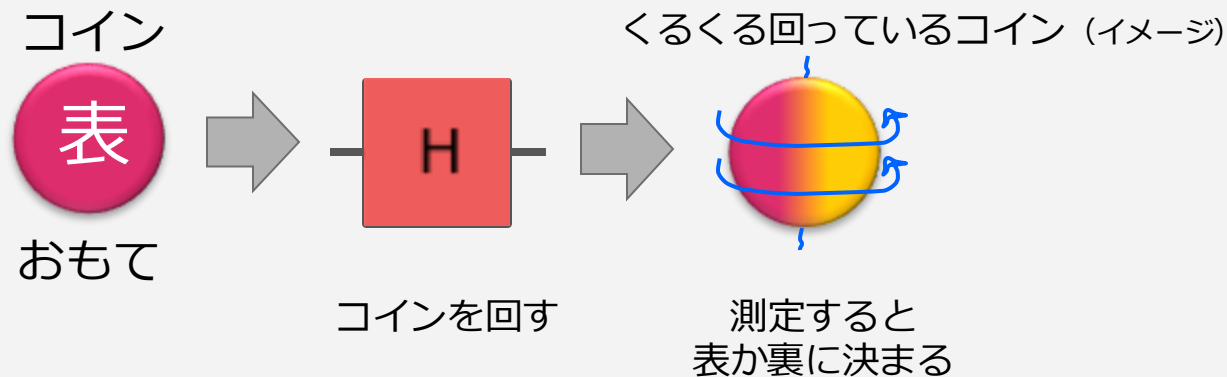
Xゲート



Hゲート



Hゲート

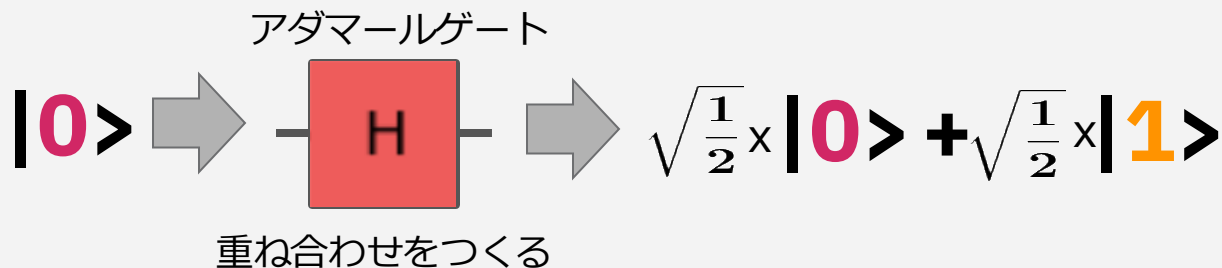
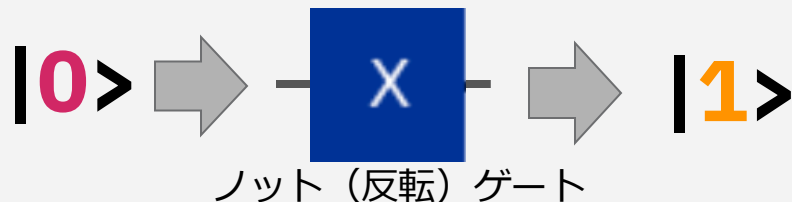


$$|0\rangle \rightarrow \text{H} \rightarrow \sqrt{\frac{1}{2}} \times |0\rangle + \sqrt{\frac{1}{2}} \times |1\rangle$$

重ね合わせをつくる

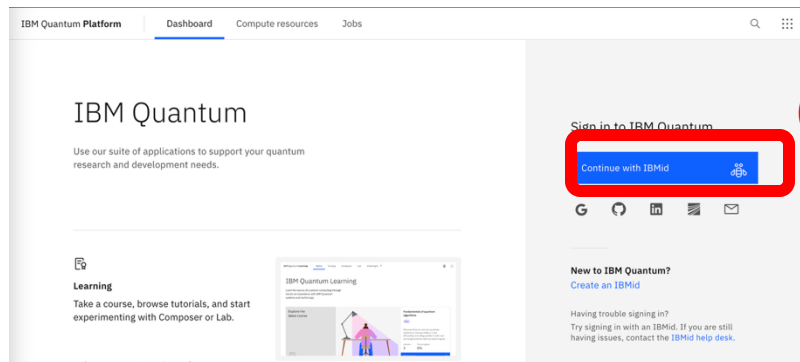
0 と 1 の重ね合わせ

量子コンピューターの計算方法

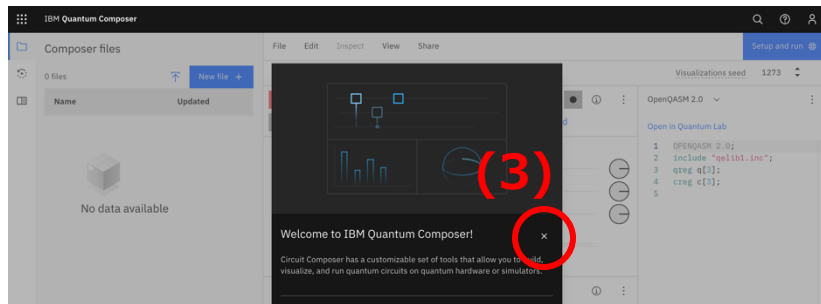


ハンズオン: IBM Quantum Composer

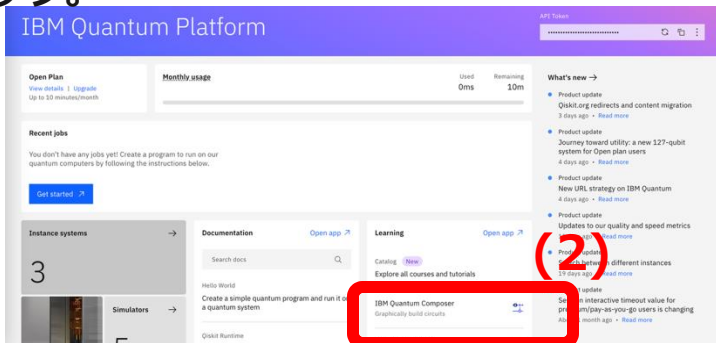
(1) IBM Quantum にログインします。URL:
<https://quantum.ibm.com/>



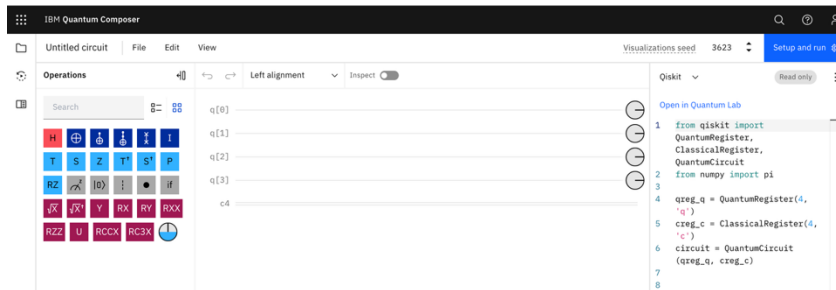
(3) ポップアップウィンドウは「x」をクリックして、閉じます。



(2) 中央下の方の「IBM Quantum Composer」をクリック。



(4) この画面になったら準備完了です。



ハンズオン: IBM Quantum Composer

<https://quantum.cloud.ibm.com/composer>

短縮URL: ibm.biz/cmpsr25

The screenshot displays the IBM Quantum Composer web application interface. At the top, the header includes the IBM Quantum Platform logo and navigation links for search, globe, and sign-in. Below the header is a menu bar with 'Untitled circuit', 'File', 'Edit', 'View', and 'Help'. The main workspace is divided into three sections:

- Operations:** A grid of quantum gates and symbols including H, \oplus , \otimes , \otimes , \otimes , I, T, S, Z, T^\dagger , S^\dagger , P, RZ, $\frac{\pi}{2}$, $|0\rangle$, $|1\rangle$, \bullet , if, \sqrt{X} , \sqrt{X}^\dagger , Y, RX, RY, RXX, RZZ, U, RCCX, RC3X, and a phase meter.
- Circuit Workspace:** A central area for building the quantum circuit. It shows five qubits labeled $q[0]$, $q[1]$, $q[2]$, $q[3]$, and $c4$. The circuit is currently empty. Controls for 'Left alignment' and 'Inspect' are visible.
- Code Editor:** A panel on the right showing OpenQASM 2.0 code. The code includes a version declaration, an include statement for 'qelib1.inc', and register declarations for $q[4]$ and $c[4]$.

```
1 OPENQASM 2.0;  
2 include "qelib1.inc";  
3  
4 qreg q[4];  
5 creg c[4];  
6  
7
```

ハンズオンの資料

URL: ibm.biz/kwskgit

「day2」フォルダー



quantum-tokyo / kawasaki-quantum-camp

Code Issues Pull requests Actions Projects Wiki Security Insights

kawasaki-quantum-camp Public

main 1 Branch 0 Tags

Go to file t + Code

kifumi Merge pull request #16 from quantum-tokyo/kifumi 215a32d · 43 minutes ago 107 Commits

2022	update readme	2 years ago
2023	create 2023 folder	last year
2024	updated readme	44 minutes ago
day1	move folders	54 minutes ago
day2	move folders	54 minutes ago
day3	move folders	54 minutes ago
.gitignore	upload day1 materials	3 years ago
LICENSE	Create LICENSE	6 months ago
README.md	updated readme	44 minutes ago
vector_matrix.pdf	Add files via upload	last year

README Apache-2.0 license

Kawasaki Quantum Summer Camp 2025

1量子ビット回路

Operations

Search

q[0]

q[1]

q[3]

c4

Left alignment

Inspect

Qiskit

Open in Quantum Lab

```
1 from qiskit import
   QuantumCircuit
2 from numpy import
3
4 qreg_q = QuantumRe
5 creg_c = Classical
6 circuit = QuantumC
```

マウスでq[1]をクリックするとゴミ箱マークが出てくるので、クリックして消します。

q[0]だけにして、1量子ビット回路の準備をします。

1量子ビット回路

The screenshot displays the IBM Quantum Composer interface. On the left, the 'Operations' panel contains a grid of quantum gates. The CNOT gate, represented by a blue square with a white circle and a plus sign, is circled in red. An orange arrow points from this gate to the circuit diagram. The circuit diagram shows two horizontal lines: the top line is labeled 'q[0]' and the bottom line is labeled 'c4'. An orange arrow points from the CNOT gate to the 'c4' line. On the right, the 'Qiskitのコード' (Qiskit code) panel shows the following code:

```
1 from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(1, 'q')
5 creg_c = ClassicalRegister(4, 'c')
6 circuit = QuantumCircuit(qreg_q, creg_c)
7
8
```

Orange arrows indicate the flow from the gate selection to the circuit and from the code panel to the circuit.

量子ゲートをマウスでドラッグ&ドロップして、量子回路を作ります。

右側には、Qiskitのコードが自動生成されます。

1. Xゲート(NOTゲート)

図の回路を作ってみてください。

下に表示される棒グラフの変化を確認しましょう。

1-1) $q[0]$ 

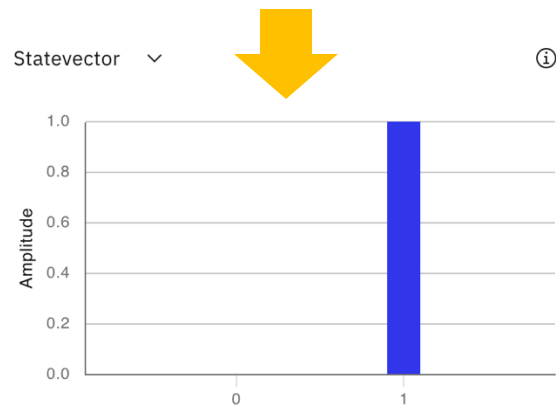
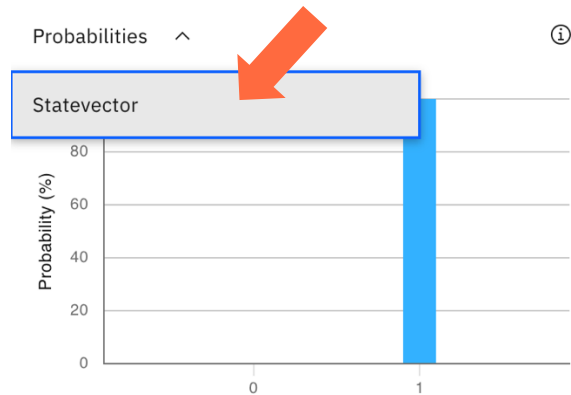


1-2) $q[0]$  

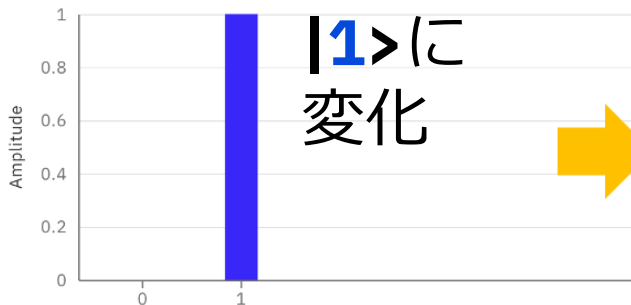
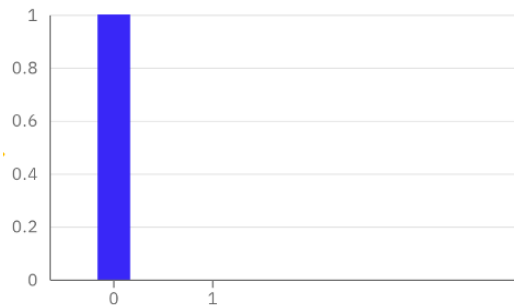


1-3) $q[0]$   

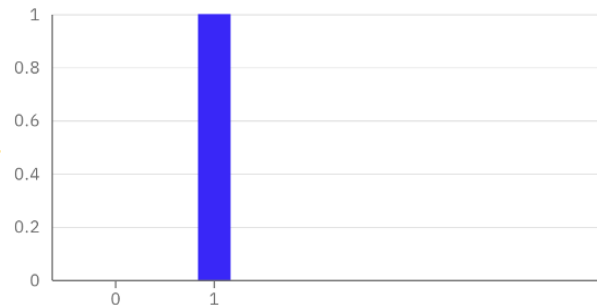
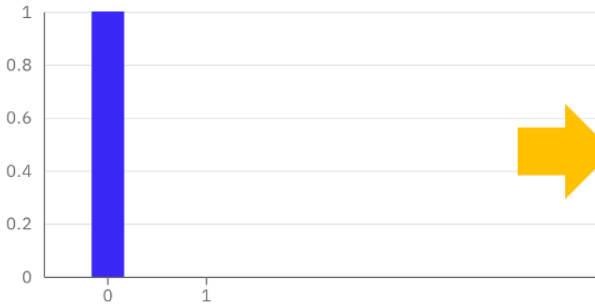
左下のグラフは青棒の「Statevector」表示にしてください。



初期状態は $|0\rangle$



$|1\rangle$ に
変化



棒グラフ (Statevector 表示) は
量子ビットの状態

$$\alpha \times |0\rangle + \beta \times |1\rangle$$

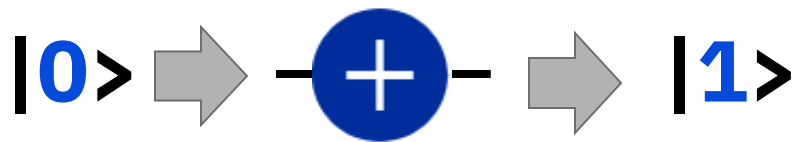
の α, β (確率振幅) です。

$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

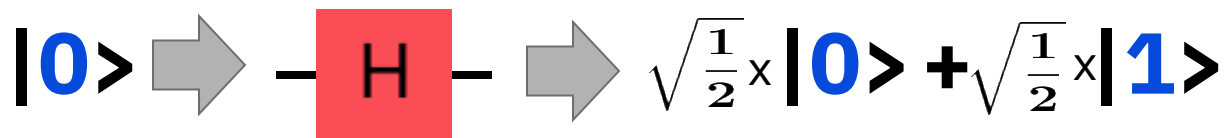
$$|1\rangle = 0 \times |0\rangle + 1 \times |1\rangle$$



量子コンピューターの計算方法

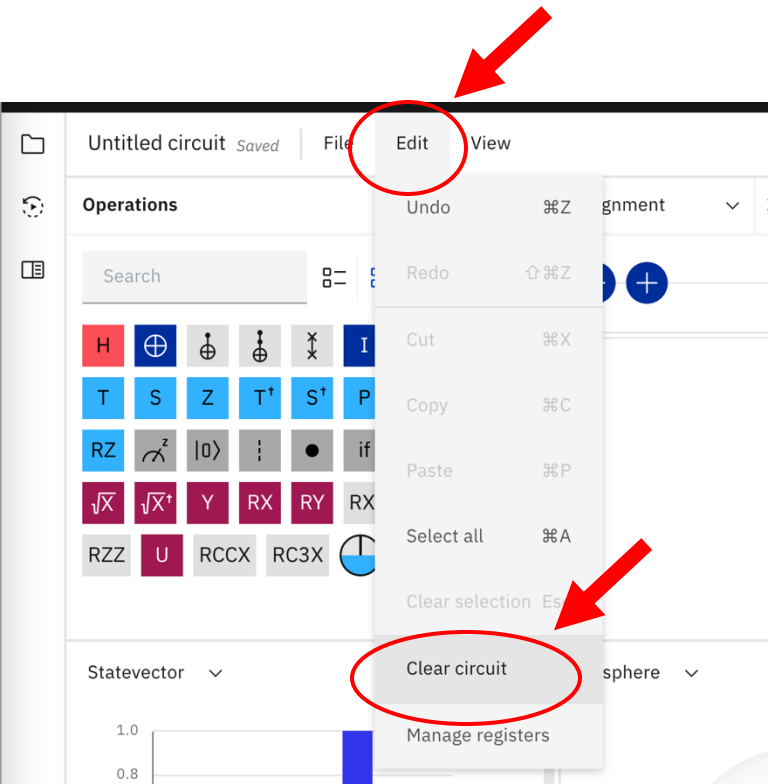


ノット（反転）ゲート

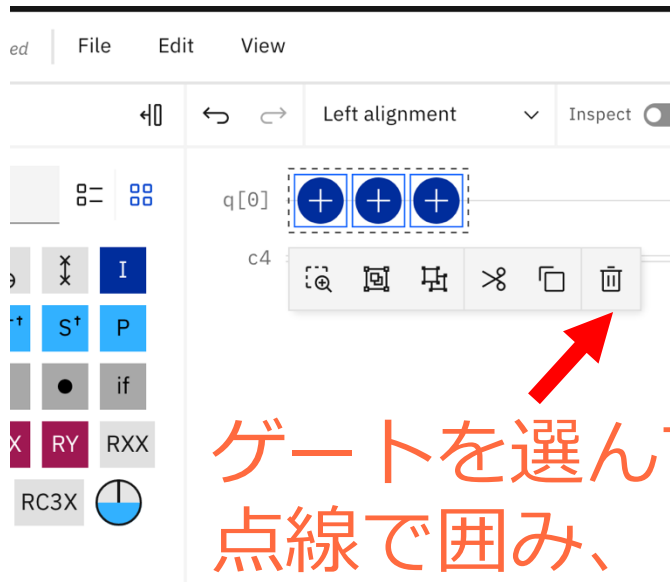


重ね合わせをつくる

置いたゲートを取り除く



または

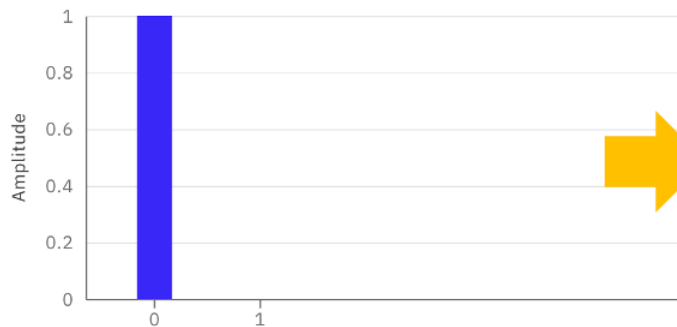
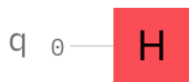


ゲートを選んで
点線で囲み、
ゴミ箱マークを
クリック

2. Hゲート

図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

2-1)



$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

重ね合わせ



例えば1000回同じ状態を作って、測定すると約500回は0が観測され、約500回は1が観測される状態。

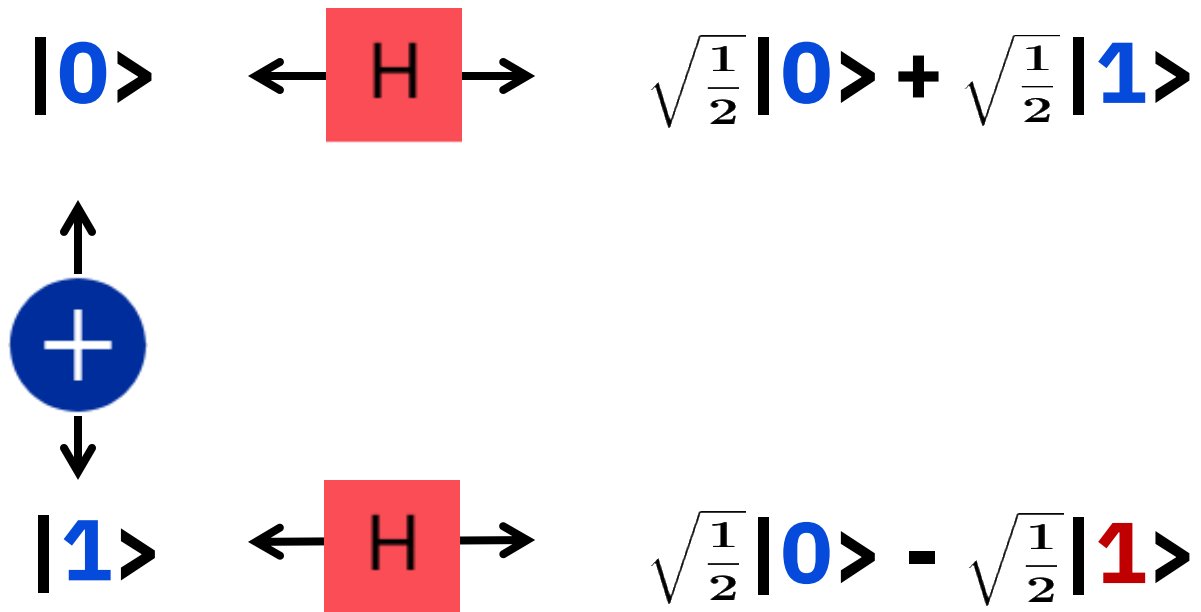
$$\begin{aligned} & 0.707 \times |0\rangle + 0.707 \times |1\rangle \\ &= \frac{1}{\sqrt{2}} \times |0\rangle + \frac{1}{\sqrt{2}} \times |1\rangle \end{aligned}$$

2. Hゲート

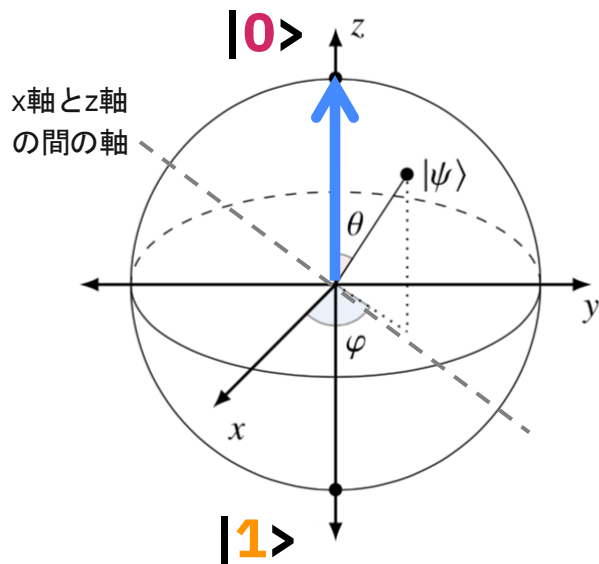
図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。



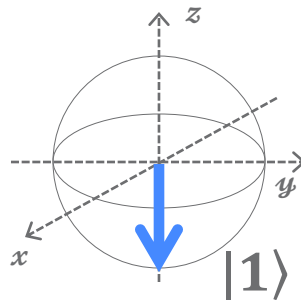
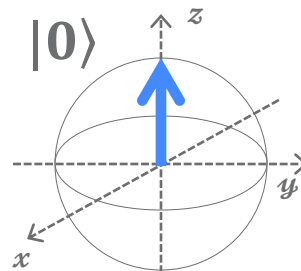
量子コンピューターの計算方法



ブロッホ球

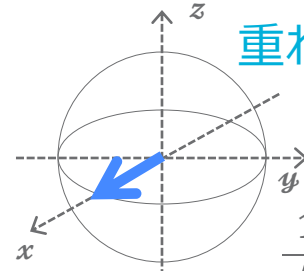


北極 : 0の確率が100%

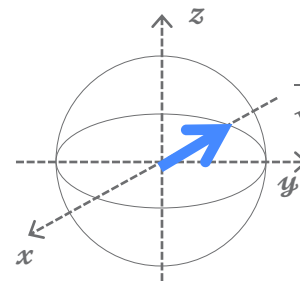


南極 : 1の確率が100%

赤道 : 0と1が50%ずつの
重ね合わせ状態

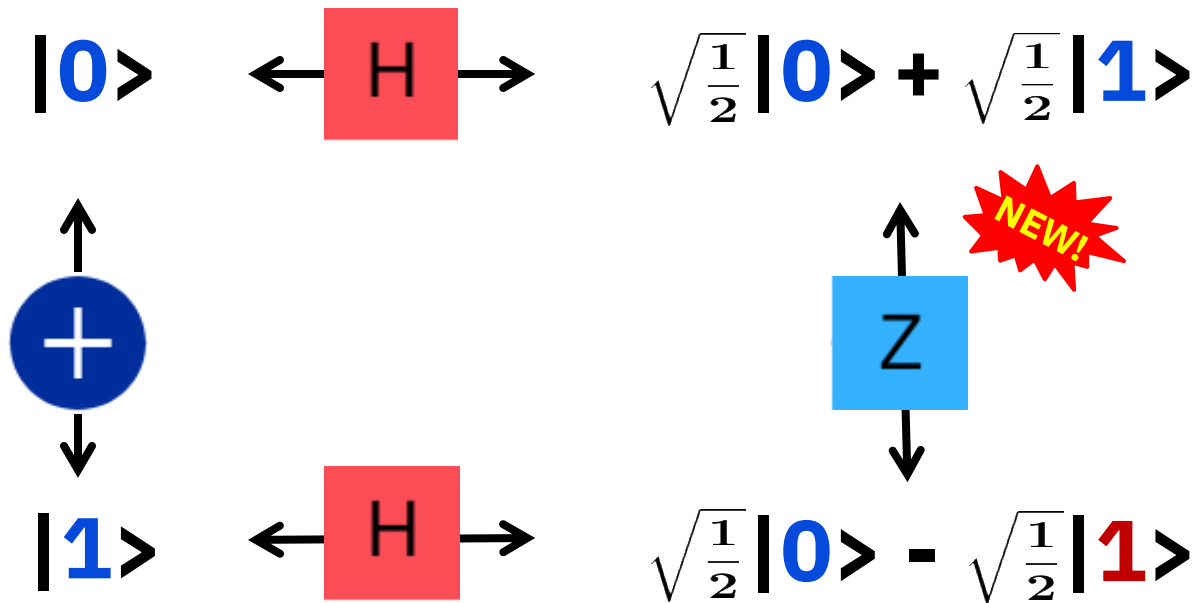


$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$



$$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

量子コンピューターの計算方法



3. Zゲート

図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

3-1)



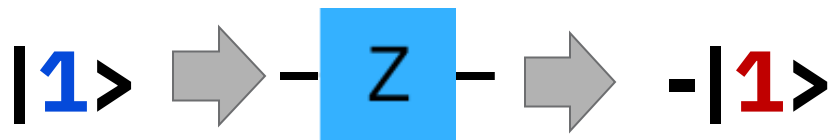
3-2)



3-3)

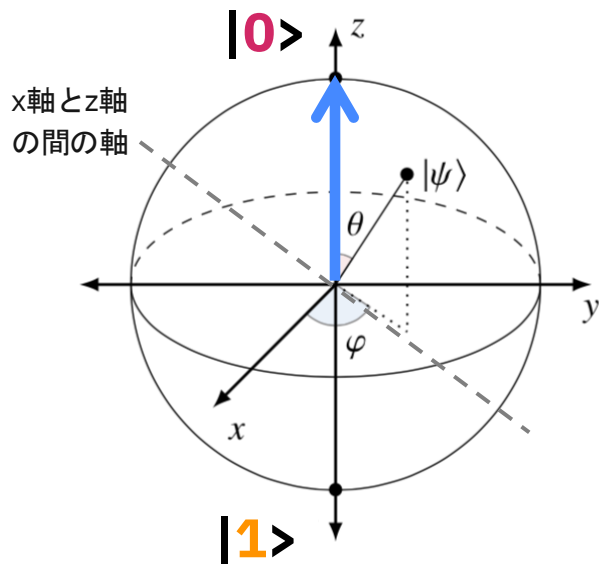


量子コンピューターの計算方法

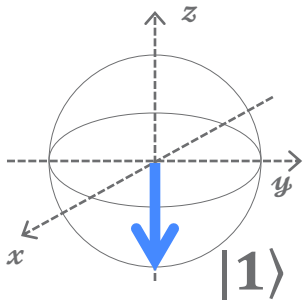
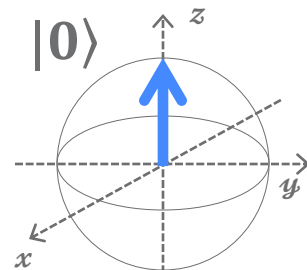


1 の符号を反転する

ブロッホ球

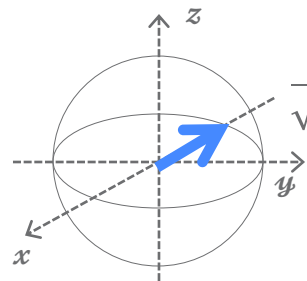
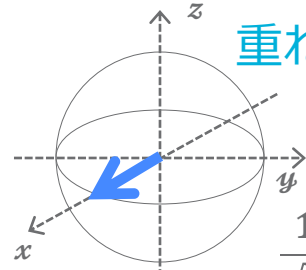


北極 : 0の確率が100%



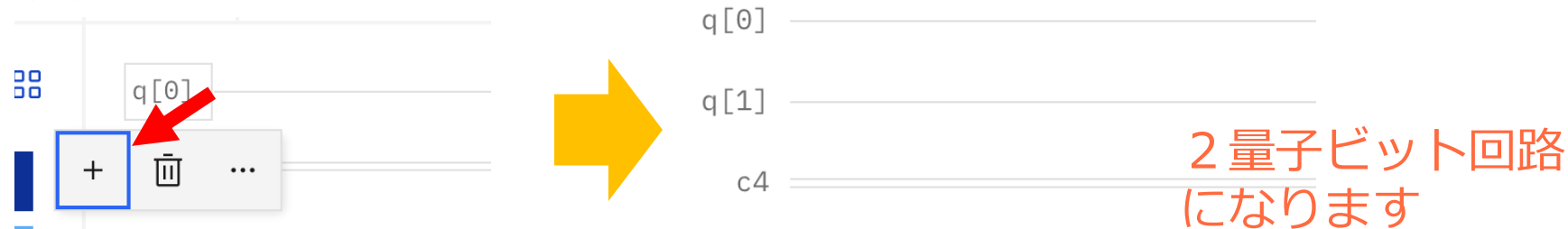
南極 : 1の確率が100%

赤道 : 0と1が50%ずつの
重ね合わせ状態



4. 量子重ね合わせ

q[0]をクリックして、さらに「+」マークをクリックして、2量子ビットの回路を準備します。



図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

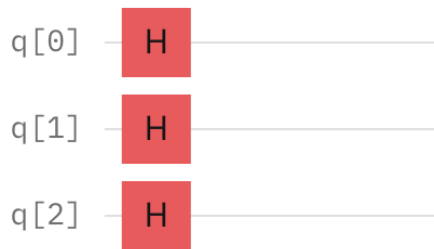
4-1)



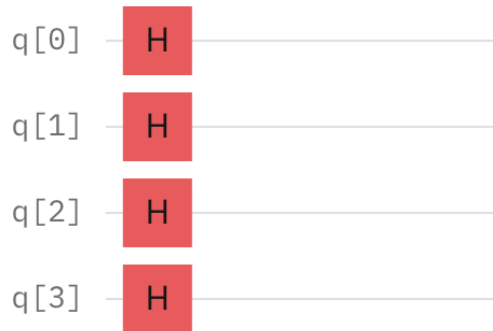
さらにq[1]をクリックして、さらに「+」マークをクリックして、3量子ビット、4量子ビット、5量子ビットの時の重ね合わせ状態を確認します。



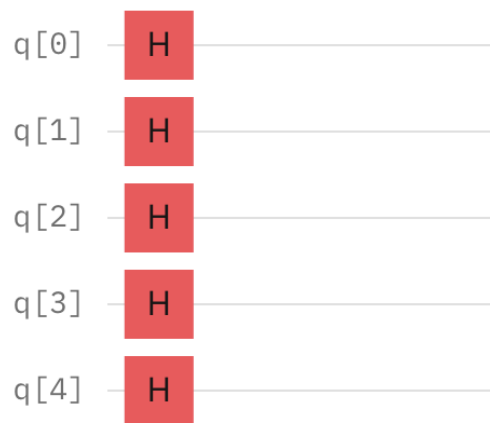
4-2)



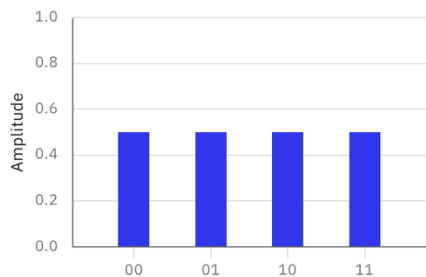
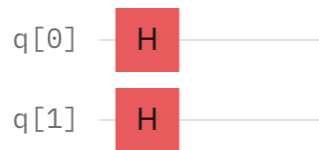
4-3)



4-4)

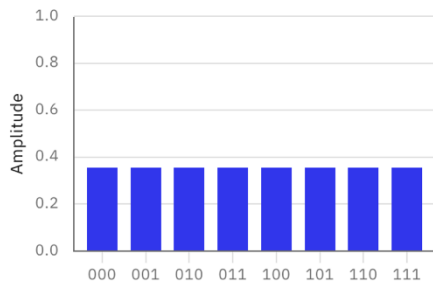
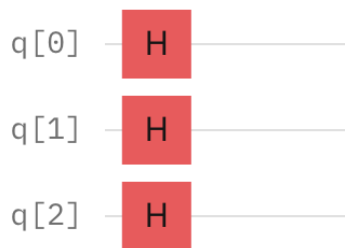


4-1)



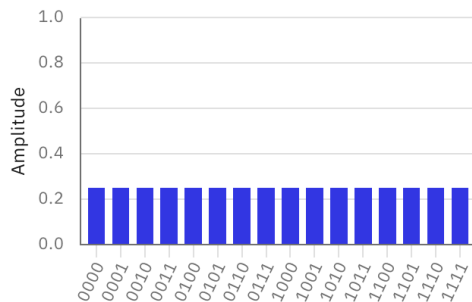
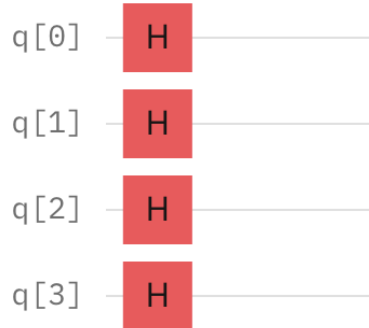
4個

4-2)



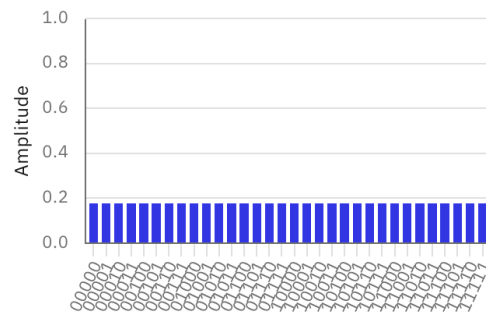
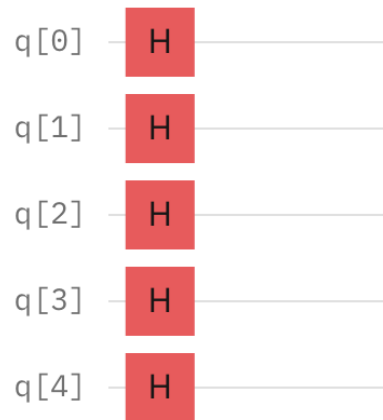
8個

4-3)



16個

4-4)



32個

量子ビット数(n)が増えるにつれて、量子状態が倍々に(2^n 個に)増えていくことがわかります。

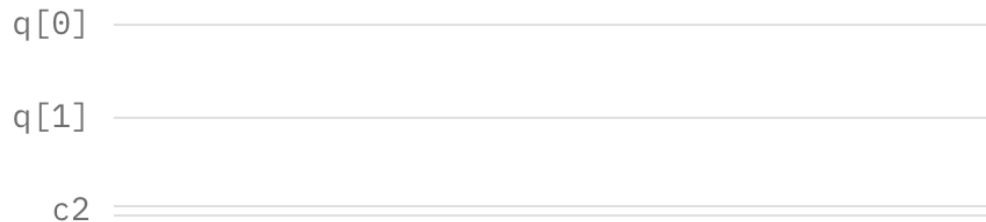
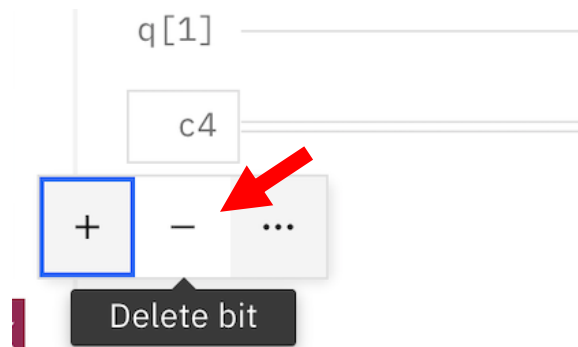
2量子ビット・2古典ビットの状態を作る

q[0]をクリックして、さらに「ゴミ箱」マークをクリック、を繰り返して、2量子ビットの回路を準備します。



2量子ビット回路
にします

次に、c4をクリックして、「-」マークをクリックするを2回繰り返して、2古典ビットにします。



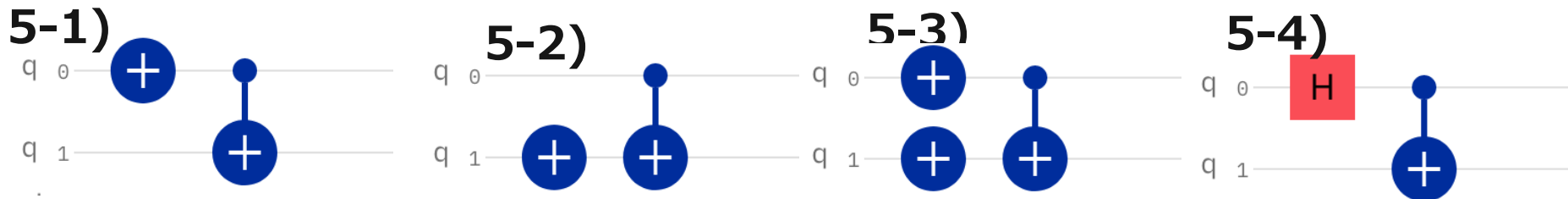
2古典ビットになります

5. CNOTゲート(制御Xゲート)

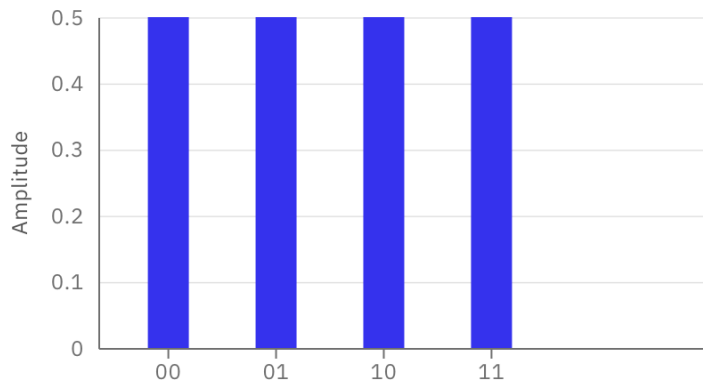
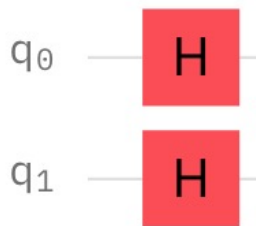
制御ビットが $|1\rangle$ のときのみ、目標ビットを反転（NOT）するゲートです。



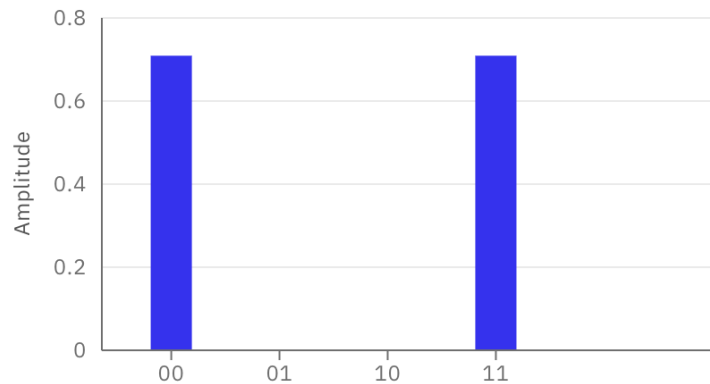
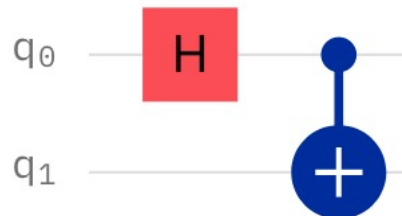
入力		出力	
目標ビット	制御ビット	目標ビット	制御ビット
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	1



量子重ね合わせ



量子もつれ (エンタングルメント)

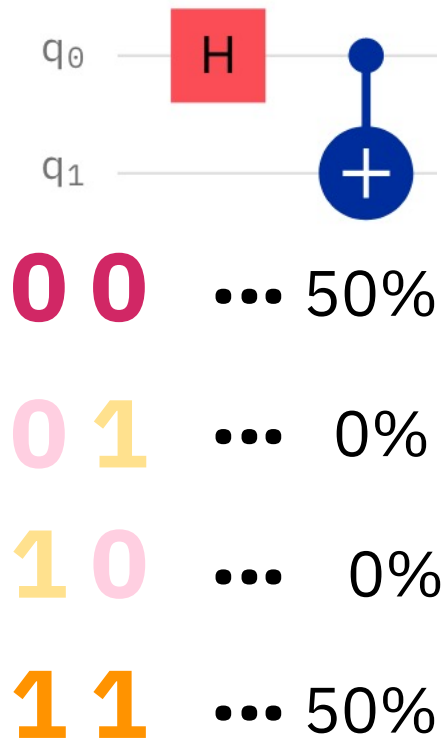


CNOTゲートは、
エンタングルメントを作ります。

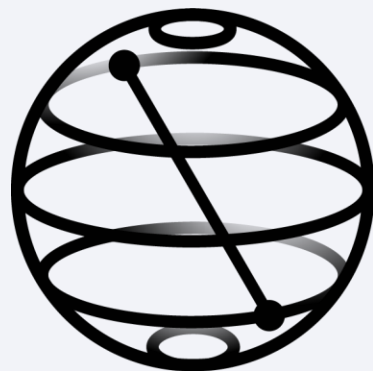
量子重ね合わせ



量子もつれ (エンタングルメント)



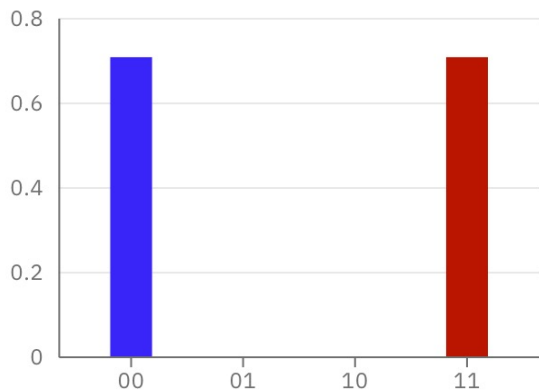
1量子ビット目が0だと分かったら
2量子ビット目も0



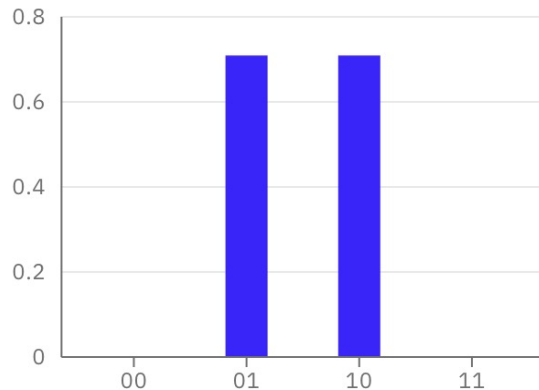
演習問題

2量子ビットのエンタングル状態を作ってみましょう。
答えは一つではないので、どんな作り方でもOKです。

(1) $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$



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



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

