

# 日本語訳 『Qiskit Textbook』 勉強会 第2章 (2.1-2.2)



---

Hironobu Takamatsu  
Associate Certified Architect

# 自己紹介



名前：高松洋亘

所属：NO.3 FSS CTL3

仕事：

- Technical Solution Architect (TSA、旧CTL、旧旧CITA)
- 金融のお客様（主に大手信託銀行様）担当のアーキテクト

保有資格

- CSM (Certified ScrumMaster)
- AWS Solution Architect Associate
- AWS Solution Architect Professional
- Red Hat Certified Specialist in Openshift Administration

# はじめに

# Qisikit Textbook 目次

## Learn Quantum Computation using Qiskit

### 0. Prerequisites

### 1. Quantum States and Qubits

#### 1.1 Introduction

#### 1.2 The Atoms of Computation

#### 1.3 Representing Qubit States

#### 1.4 Single Qubit Gates

### 2. Multiple Qubits and Entanglement

#### 2.1 Introduction

#### 2.2 Multiple Qubits and Entangled States

#### 2.3 Phase Kickback

#### 2.4 Proving Universality

#### 2.5 More Circuit Identities

## 3. Quantum Protocols and Quantum Algorithms

### 3.1 Defining Quantum Circuits

### 3.2 Quantum Teleportation

### 3.3 Superdense Coding

### 3.4 Deutsch-Josza Algorithm

### 3.5 Bernstein-Vazirani Algorithm

### 3.6 Simon's Algorithm

### 3.7 Quantum Fourier Transform

### 3.8 Quantum Phase Estimation

### 3.9 Grover's Algorithm

### 3.10 Quantum Counting

### 3.11 Quantum Key Distribution

## 4. Quantum Algorithms for Applications

### 4.1 Applied Quantum Algorithms

#### 4.1.1 Solving Linear Systems of Equations using HHL

#### 4.1.2 Simulating Molecules using VQE

#### 4.1.3 Solving combinatorial optimization problems using QAOA

#### 4.1.4 Solving Satisfiability Problems using Grover's Algorithm

#### 4.1.5 Hybrid quantum-classical Neural Networks with PyTorch and Qiskit

### 4.2 Implementations of Recent Quantum Algorithms

#### 4.2.1 Variational Quantum Linear Solver

## 5. Investigating Quantum Hardware Using Quantum Circuits

### 5.1 Introduction to Quantum Error Correction using Repetition Codes

### 5.2 Measurement Error Mitigation

### 5.3 Randomized Benchmarking

### 5.4 Measuring Quantum Volume

## 6. Investigating Quantum Hardware Using Microwave Pulses

### 6.1 Calibrating Qubits with Qiskit Pulse

### 6.2 Accessing Higher Energy States

## 7. Problem Sets & Exercises

### Set 1. Classical Logic Gates with Quantum Circuits

### Set 2. Basic Synthesis of Single-Qubit Gates

### Set 3. Building the Best AND Gate

## 8. Games & Demos

### Hello Qiskit Game

### Estimating Pi Using Quantum Phase Estimation Algorithm

### Interactivity Index

Qiskit Textbook

<https://qiskit.org/textbook/preface.html>

# 本日の範囲

## Learn Quantum Computation using Qiskit

### 0. Prerequisites

### 1. Quantum States and Qubits

#### 1.1 Introduction

#### 1.2 The Atoms of Computation

#### 1.3 Representing Qubit States

#### 1.4 Single Qubit Gates

### 2. Multiple Qubits and Entanglement

#### 2.1 Introduction

#### 2.2 Multiple Qubits and Entangled States

#### 2.3 Phase Kickback

#### 2.4 Proving Universality

#### 2.5 More Circuit Identities

## 3. Quantum Protocols and Quantum Algorithms

### 3.1 Defining Quantum Circuits

### 3.2 Quantum Teleportation

### 3.3 Superdense Coding

### 3.4 Deutsch-Josza Algorithm

### 3.5 Bernstein-Vazirani Algorithm

### 3.6 Simon's Algorithm

### 3.7 Quantum Fourier Transform

### 3.8 Quantum Phase Estimation

### 3.9 Grover's Algorithm

### 3.10 Quantum Counting

### 3.11 Quantum Key Distribution

## 4. Quantum Algorithms for Applications

### 4.1 Applied Quantum Algorithms

#### 4.1.1 Solving Linear Systems of Equations using HHL

#### 4.1.2 Simulating Molecules using VQE

#### 4.1.3 Solving combinatorial optimization problems using QAOA

#### 4.1.4 Solving Satisfiability Problems using Grover's Algorithm

#### 4.1.5 Hybrid quantum-classical Neural Networks with PyTorch and Qiskit

### 4.2 Implementations of Recent Quantum Algorithms

#### 4.2.1 Variational Quantum Linear Solver

## 5. Investigating Quantum Hardware Using Quantum Circuits

### 5.1 Introduction to Quantum Error Correction using Repetition Codes

### 5.2 Measurement Error Mitigation

### 5.3 Randomized Benchmarking

### 5.4 Measuring Quantum Volume

## 6. Investigating Quantum Hardware Using Microwave Pulses

### 6.1 Calibrating Qubits with Qiskit Pulse

### 6.2 Accessing Higher Energy States

## 7. Problem Sets & Exercises

### Set 1. Classical Logic Gates with Quantum Circuits

### Set 2. Basic Synthesis of Single-Qubit Gates

### Set 3. Building the Best AND Gate

## 8. Games & Demos

### Hello Qiskit Game

### Estimating Pi Using Quantum Phase Estimation Algorithm

### Interactivity Index

Qiskit Textbook

<https://qiskit.org/textbook/preface.html>

# 2.1

## Introduction

### はじめに

# 量子ビット操作と 量子ゲート

- 量子ビットを使用して計算するためには量子ビットの操作が必要
- 量子ビットの操作には量子ゲートを使用する
- 一般的にはハードウェアに実装できる量子ゲートは、1量子ビットまたは2量子ビットの操作が可能なもの
- 量子ビットに対する任意の操作は量子ゲートを組み合わせることで実現可能

ビット数	古典コンピュータ（論理ゲート）	量子コンピュータ（量子ゲート）
1ビット	・ NOTゲート	・ Xゲート ・ Yゲート ・ Zゲート ・ Hゲート（アダマール） ・ Rゲート ・ Iゲート ・ Sゲート ・ Tゲート ・ Uゲート ※「1.4 Single Qubit Gates」参照
2ビット	・ ANDゲート ・ ORゲート	・ CNOTゲート など

## 2.2

# Multiple Qubits and Entangled States

## 複数量子ビットともつれ 状態



# 目次

## Contents

1. Representing Multi-Qubit States
  - 1.1 Exercises
2. Single Qubit Gates on Multi-Qubit Statevectors
  - 2.1 Exercises
3. Multi-Qubit Gates
  - 3.1 The CNOT-gate
  - 3.2 Entangled States
  - 3.3 Exercises

## 2.2.1 Representing Multi-Qubit States

### 複数量子ビット状態の表現

# 量子ビットの状態ベクトル表現

- 1量子ビットの状態ベクトル

$$|a\rangle = a_0|0\rangle + a_1|1\rangle = a_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

$$|a_0|^2 + |a_1|^2 = 1$$

- 2量子ビットの状態ベクトル

$$|a\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle = a_{00} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + a_{01} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + a_{10} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + a_{11} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix}$$

$$|a_{00}|^2 + |a_{01}|^2 + |a_{10}|^2 + |a_{11}|^2 = 1$$

$\begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow \\ \uparrow & \uparrow & \uparrow & \uparrow \\ \uparrow & \uparrow & \uparrow & \uparrow \\ \uparrow & \uparrow & \uparrow & \uparrow \\ \uparrow & \uparrow & \uparrow & \uparrow \\ \uparrow & \uparrow & \uparrow & \uparrow \end{matrix}$

観測した結果11が得られる確率  
 観測した結果10が得られる確率  
 観測した結果01が得られる確率  
 観測した結果00が得られる確率

# 状態ベクトルのテンソル積

- 2量子ビットの状態ベクトルは各量子ビットの状態ベクトルのテンソル積で表現できる

$$|a\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}, \quad |b\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$|ab\rangle = |a\rangle \otimes |b\rangle = \begin{bmatrix} a_0 \times \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ a_1 \times \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

# 状態ベクトルの次元

• 1量子ビット： 2次元

• 2量子ビット： 4次元

• 3量子ビット： 8次元

...

• n量子ビット：  $2^n$ 次元

• 20量子ビット：  $2^{20} = 1,048,576$ 個の状態について計算が必要

→ この規模ならPCでも計算可能

• 100量子ビット：  $2^{100} = 1.27 \times 10^{30}$ 個の状態について計算が必要

→スーパーコンピュータでも計算が困難

$$|abc\rangle = \begin{bmatrix} a_0b_0c_0 \\ a_0b_0c_1 \\ a_0b_1c_0 \\ a_0b_1c_1 \\ a_1b_0c_0 \\ a_1b_0c_1 \\ a_1b_1c_0 \\ a_1b_1c_1 \end{bmatrix}$$

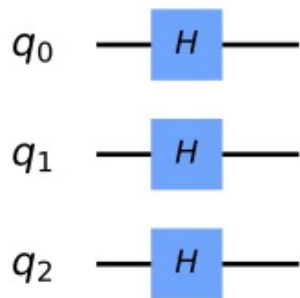
例：3量子ビットの場合

接頭辞	記号	$1000^m$	$10^n$
ヨタ (yotta)	Y	$1000^8$	$10^{24}$
ゼタ (zetta)	Z	$1000^7$	$10^{21}$
エクサ (exa)	E	$1000^6$	$10^{18}$
ペタ (peta)	P	$1000^5$	$10^{15}$
テラ (tera)	T	$1000^4$	$10^{12}$
ギガ (giga)	G	$1000^3$	$10^9$
メガ (mega)	M	$1000^2$	$10^6$
キロ (kilo)	k	$1000^1$	$10^3$
ヘクト (hecto)	h		$10^2$
デカ (deca)	da		$10^1$

Wikipedia

<https://ja.wikipedia.org/wiki/ペタ>

# 3量子ビットの各ビットを アダマール・ゲートで変換した場合



$$|q_0\rangle = |q_1\rangle = |q_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$|q_0 q_1\rangle = |q_0\rangle \otimes |q_1\rangle = \frac{1}{2} \begin{bmatrix} 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$|a\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}, \quad |b\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$|ab\rangle = |a\rangle \otimes |b\rangle = \begin{bmatrix} a_0 \times \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ a_1 \times \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

$$|q_0 q_1 q_2\rangle = |q_0 q_1\rangle \otimes |q_2\rangle = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

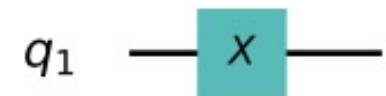
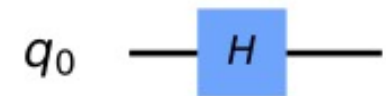
$\leftarrow q_0=0, q_1=0, q_2=0$ の確率  
 $\leftarrow q_0=0, q_1=0, q_2=1$ の確率  
 $\leftarrow q_0=0, q_1=1, q_2=0$ の確率  
 $\leftarrow q_0=0, q_1=1, q_2=1$ の確率  
 $\leftarrow q_0=1, q_1=0, q_2=0$ の確率  
 $\leftarrow q_0=1, q_1=0, q_2=1$ の確率  
 $\leftarrow q_0=1, q_1=1, q_2=0$ の確率  
 $\leftarrow q_0=1, q_1=1, q_2=1$ の確率

## 2.2.2 Single Qubit Gates on Multi-Qubit Statevectors

複数量子ビット状態ベクトル上の  
単一量子ビットゲート

# 2つの量子ビットに同時にゲートを適用する場合（例：XゲートとHゲート）

- 2つの量子ビットへのゲートの同時適用は、各ゲートのテンソル積で表現できる



$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$X|q_1\rangle \otimes H|q_0\rangle = (X \otimes H)|q_1 q_0\rangle$$

$$X \otimes H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 1 \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 0 \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$


$$X \otimes H = \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix}$$



## 2つの量子ビットのうち1つの量子ビットに対してのみゲートを適用する場合

- 2つの量子ビットのうち1つの量子ビットに対してのみゲートを適用する場合は、Iゲートとのテンソル積で表現できる

$q_0$  —————

$q_1$  —  —

$$X|q_1\rangle \otimes I|q_0\rangle = (X \otimes I)|q_1 q_0\rangle$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

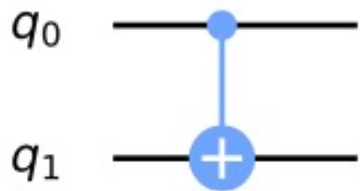
$$X \otimes I = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 1 \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 0 \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$$

## 2.2.3 Multi-Qubit Gates

### 複数量子ビットゲート

# C-NOTゲートの行列表現

- $q_0$ が0の時は $q_1$ はそのまま、 $q_0$ が1の時は $q_1$ を反転させるゲート
- $q_0$ を制御ビット、 $q_1$ を標的ビットと呼ぶ
- CNOTゲートの行列表現は状態ベクトルにてどの順番で $q_0$ 、 $q_1$ の状態を記述するかに依存



$q_0$	$q_1$	CNOT	$q_0'$	$q_1'$
$ 0\rangle$	$ 0\rangle$	$\rightarrow$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$\rightarrow$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$\rightarrow$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$\rightarrow$	$ 1\rangle$	$ 0\rangle$

}  $q_1$  (標的ビット) が反転

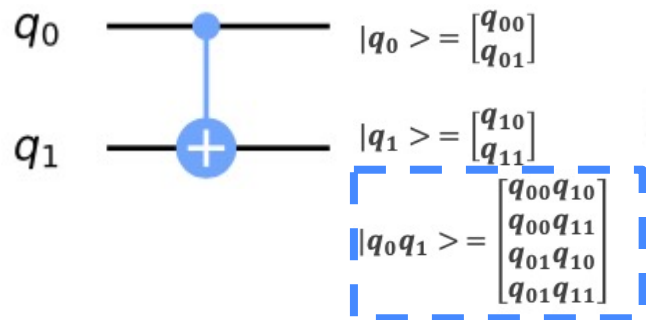
$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

or

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

# C-NOTゲート (CNOT $|q_0q_1\rangle$ の場合)

- 状態ベクトルを $|q_0q_1\rangle$ と記述した場合のCNOTゲートの行列表現は以下のとおり



$$CNOT|q_0q_1\rangle = CNOT \begin{bmatrix} q_{00}q_{10} \\ q_{00}q_{11} \\ q_{01}q_{10} \\ q_{01}q_{11} \end{bmatrix} = \begin{bmatrix} q_{00}q_{10} \\ q_{00}q_{11} \\ q_{01}q_{11} \\ q_{01}q_{10} \end{bmatrix}$$

$\leftarrow q_0=0, q_1=0$ の確率  
 $\leftarrow q_0=0, q_1=1$ の確率  
 $\leftarrow q_0=1, q_1=0$ の確率  
 $\leftarrow q_0=1, q_1=1$ の確率

上記を満たすCNOTは以下の通り

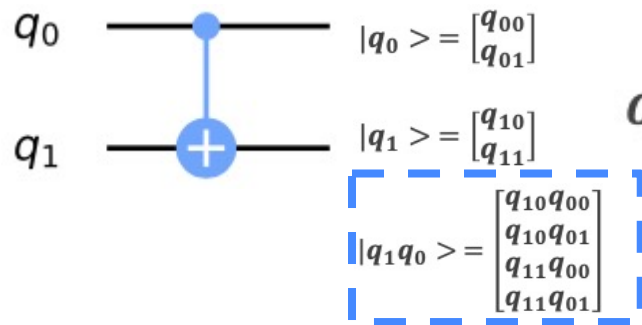
$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

q0	q1	CNOT	q0'	q1'
0>	0>	→	0>	0>
0>	1>	→	0>	1>
1>	0>	→	1>	1>
1>	1>	→	1>	0>

# C-NOTゲート (CNOT $|q_1q_0\rangle$ の場合)

※Qiskit Textbookはこちらを使用

- 状態ベクトルを $|q_1q_0\rangle$ と記述した場合のCNOTゲートの行列表現は以下のとおり



$$CNOT|q_1q_0\rangle = CNOT \begin{bmatrix} q_{10}q_{00} \\ q_{10}q_{01} \\ q_{11}q_{00} \\ q_{11}q_{01} \end{bmatrix} = \begin{bmatrix} q_{10}q_{00} \\ q_{11}q_{00} \\ q_{10}q_{01} \\ q_{11}q_{01} \end{bmatrix}$$

$\leftarrow q_1=0, q_0=0$ の確率  
 $\leftarrow q_1=0, q_0=1$ の確率  
 $\leftarrow q_1=1, q_0=0$ の確率  
 $\leftarrow q_1=1, q_0=1$ の確率

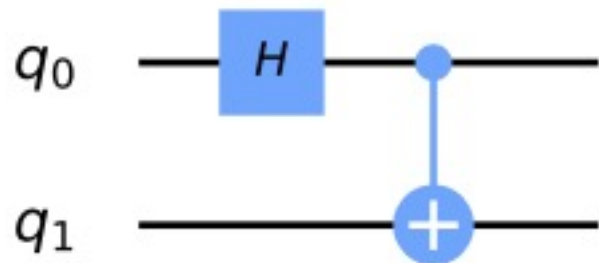
上記を満たすCNOTは以下の通り

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

q0	q1	CNOT	q0'	q1'
$ 0\rangle$	$ 0\rangle$	$\rightarrow$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$\rightarrow$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$\rightarrow$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$\rightarrow$	$ 1\rangle$	$ 0\rangle$

# 重ね合わせ状態にC-NOTゲートを適用すると どうなる？

- $q_0$ にHゲート（アダマール・ゲート）を適用し重ね合わせ状態にする
- 重ね合わせ状態の $q_0$ を制御ビットとして $q_0$ と $q_1$ のCNOTを取る



$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{CNOT} |q_1 q_0\rangle = \text{CNOT} |0+\rangle = \text{CNOT} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \text{CNOT} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

↑  
もつれ状態

# もつれ状態

- ベル状態
- 1つの量子を観測すると、もう1つの量子に関する状態を知ることができ、同時に重ね合わせ状態は消滅する。
- 不気味な遠隔作用
- もつれ状態を応用することで光速を超える速度での通信が可能？  
→ 観測結果はランダムで事前に知ることができないため不可能（通信不可定理）

$$CNOT |q_1 q_0\rangle = CNOT |0+\rangle = CNOT \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} CNOT \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \xrightarrow{\text{ベル状態}} CNOT |0+\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

- ・ 観測した結果 $q_0=0$ だった場合、必ず $q_1=0$
- ・ 観測した結果 $q_0=1$ だった場合、必ず $q_1=1$
- ・ 観測した結果 $q_1=0$ だった場合、必ず $q_0=0$
- ・ 観測した結果 $q_1=1$ だった場合、必ず $q_0=1$

- $q_1=0, q_0=0$ が観測される確率 $1/2$
- $q_1=0, q_0=1$ が観測される確率 $0$
- $q_1=1, q_0=0$ が観測される確率 $0$
- $q_1=1, q_0=1$ が観測される確率 $1/2$

# 参考情報

- 古典プログラマ向け 量子プログラミング入門 - 量子ゲート編 -
- 有限会社ラング・エッジ
- 宮地直人
- <https://www.ossal.org/qc/Sal-QC-Prog-1st.pdf>