

# 量子エラー訂正技術の動向



# 量子エラー訂正技術の動向

## Agenda

はじめに

Part 1 量子エラー訂正技術の基礎

Part 2 Shor が考えたこと

Part 3 Stabilizer -- もう一つの観測演算子

# Part 1

## 量子エラー訂正技術の基礎

1. qubitの状態の表現 – ベクトル表現
2. qubitの状態の変化 – ユニタリ変換
3. qubitの状態の観測 – Bornルール
4. Observableの固有値と固有ベクトル

# Part 2

## Shor が考えたこと

1. Shor code
2. 観測と射影演算子
3. 射影演算子でのbit flipの検出

## Part 3

# Stabilizer -- もう一つの観測演算子

1. Stabilizerとは何か？
2. bit flipの検出
3. Phase flipの検出
4. まとめ

はじめに



# はじめに

2023年、2024年と、量子エラー訂正技術の分野で大きな前進がありました。代表的なものを二つ紹介します。

一つは、量子エラー訂正にsurface codeの手法を導入し、量子エラー訂正技術のマイルストーンを超えたと評された、2023年のGoogleのNature誌への投稿論文です。

もう一つは、2024年のMicrosoftとQuantinuumの論文で、論理qubitのエラーを物理qubitのエラーの800分の1にしたとして注目を集めました。

[nature](#) > [articles](#) > article

Article | [Open access](#) | Published: 22 February 2023

## Suppressing quantum errors by scaling a surface code logical qubit

[Google Quantum AI](#)

*Nature* **614**, 676–681 (2023) | [Cite this article](#)

**148k** Accesses | **402** Citations | **975** Altmetric | [Metrics](#)

### Abstract

---

Practical quantum computing will require error rates well below those achievable with physical qubits. Quantum error correction<sup>1,2</sup> offers a path to algorithmically relevant error

<https://www.nature.com/articles/s41586-022-05434-1>

## Demonstration of logical qubits and repeated error correction with better-than-physical error rates

A. Paetznick,<sup>1</sup> M. P. da Silva,<sup>1</sup> C. Ryan-Anderson,<sup>2</sup> J. M. Bello-Rivas,<sup>1</sup> J. P. Campora III,<sup>2</sup>  
A. Chernoguzov,<sup>2</sup> J. M. Dreiling,<sup>2</sup> C. Foltz,<sup>2</sup> F. Frachon,<sup>1</sup> J. P. Gaebler,<sup>2</sup> T. M. Gatterman,<sup>2</sup>  
L. Grans-Samuelsson,<sup>1</sup> D. Gresh,<sup>2</sup> D. Hayes,<sup>2</sup> N. Hewitt,<sup>2</sup> C. Holliman,<sup>2</sup> C. V. Horst,<sup>2</sup> J. Johansen,<sup>2</sup>  
D. Lucchetti,<sup>2</sup> Y. Matsuoka,<sup>2</sup> M. Mills,<sup>2</sup> S. A. Moses,<sup>2</sup> B. Neyenhuis,<sup>2</sup> A. Paz,<sup>1</sup> J. Pino,<sup>2</sup>  
P. Siegfried,<sup>2</sup> A. Sundaram,<sup>1</sup> D. Tom,<sup>1</sup> S. J. Wernli,<sup>1</sup> M. Zanner,<sup>1</sup> R. P. Stutz,<sup>2</sup> and K. M. Svore<sup>1</sup>

<sup>1</sup>*Microsoft Azure Quantum*

<sup>2</sup>*Quantinuum*

(Dated: November 19, 2024)

The promise of quantum computers hinges on the ability to scale to large system sizes, e.g., to run quantum computations consisting of more than 100 million operations fault-tolerantly. This in turn requires suppressing errors to levels inversely proportional to the size of the computation. As a step towards this ambitious goal, we present experiments on a trapped-ion QCCD processor where, through the use of fault-tolerant encoding and error correction, we are able to suppress logical error rates to levels below the physical error rates. In particular, we entangled logical qubits encoded in the  $[[7, 1, 3]]$  code with error rates  $9.8\times$  to  $500\times$  lower than at the physical level, and entangled logical qubits encoded in a  $[[12, 2, 4]]$  code based on Knill's C4/C6 scheme with error rates  $4.7\times$  to  $800\times$  lower than at the physical level, depending on the judicious use of post-selection. Moreover, we demonstrate repeated error correction with the  $[[12, 2, 4]]$  code, with logical error rates below physical circuit baselines corresponding to repeated CNOTs, and show evidence that the error rate per error correction cycle, which consists of over 100 physical CNOTs, approaches the error rate of two physical CNOTs. These results signify a transition from noisy intermediate scale quantum computing to reliable quantum computing, and demonstrate advanced capabilities toward large-scale fault-tolerant quantum computing.

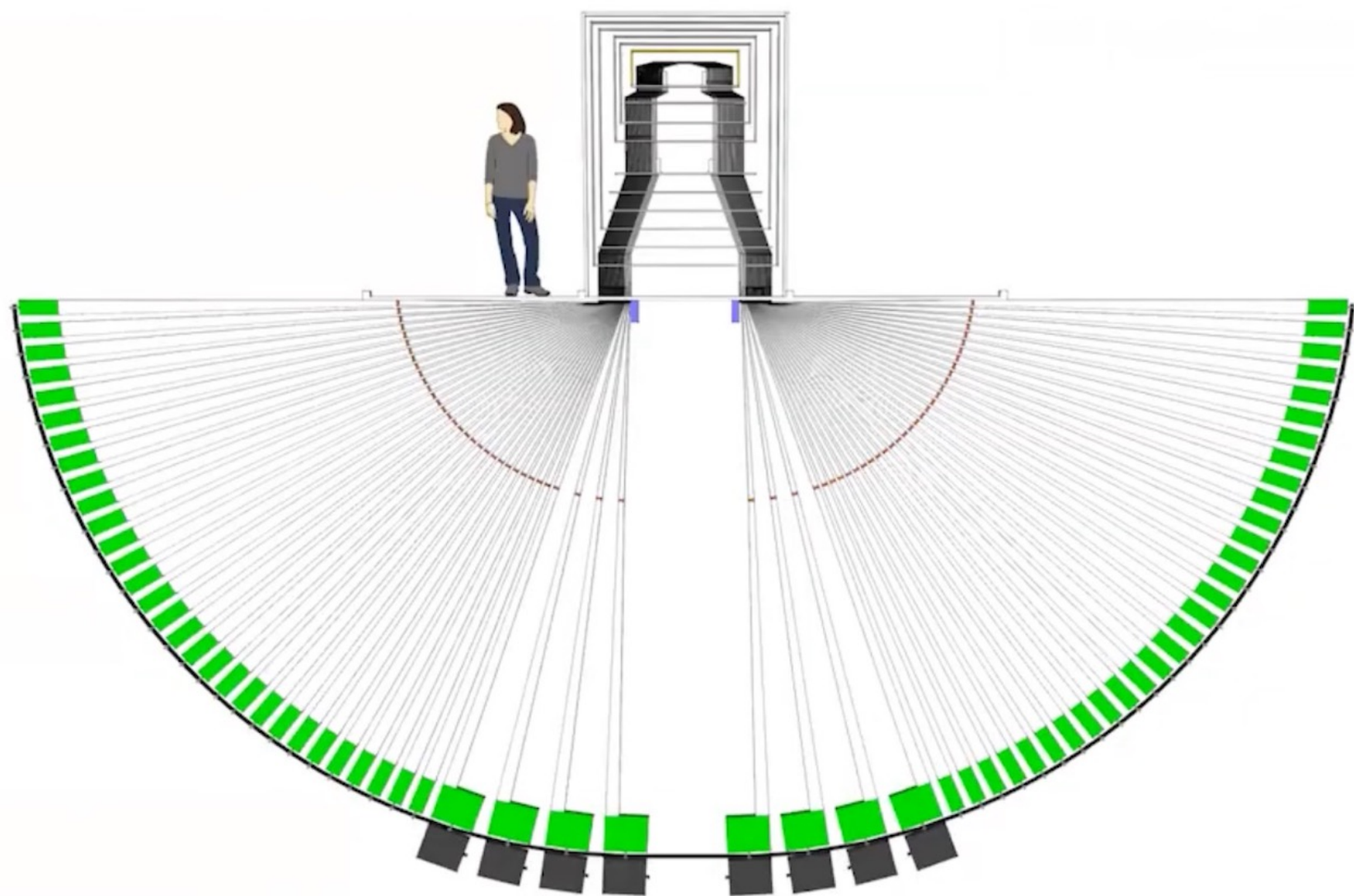
<https://arxiv.org/pdf/2404.02280>

# NISQの時代から、ノイズ耐性を持つ Resilientコンピューティングの時代へ

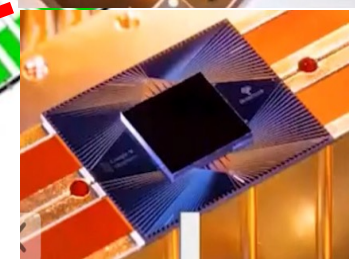
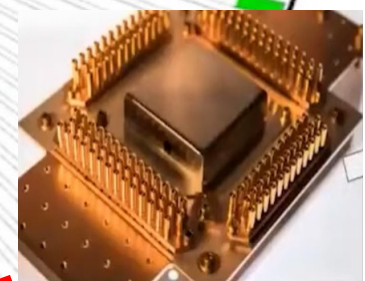
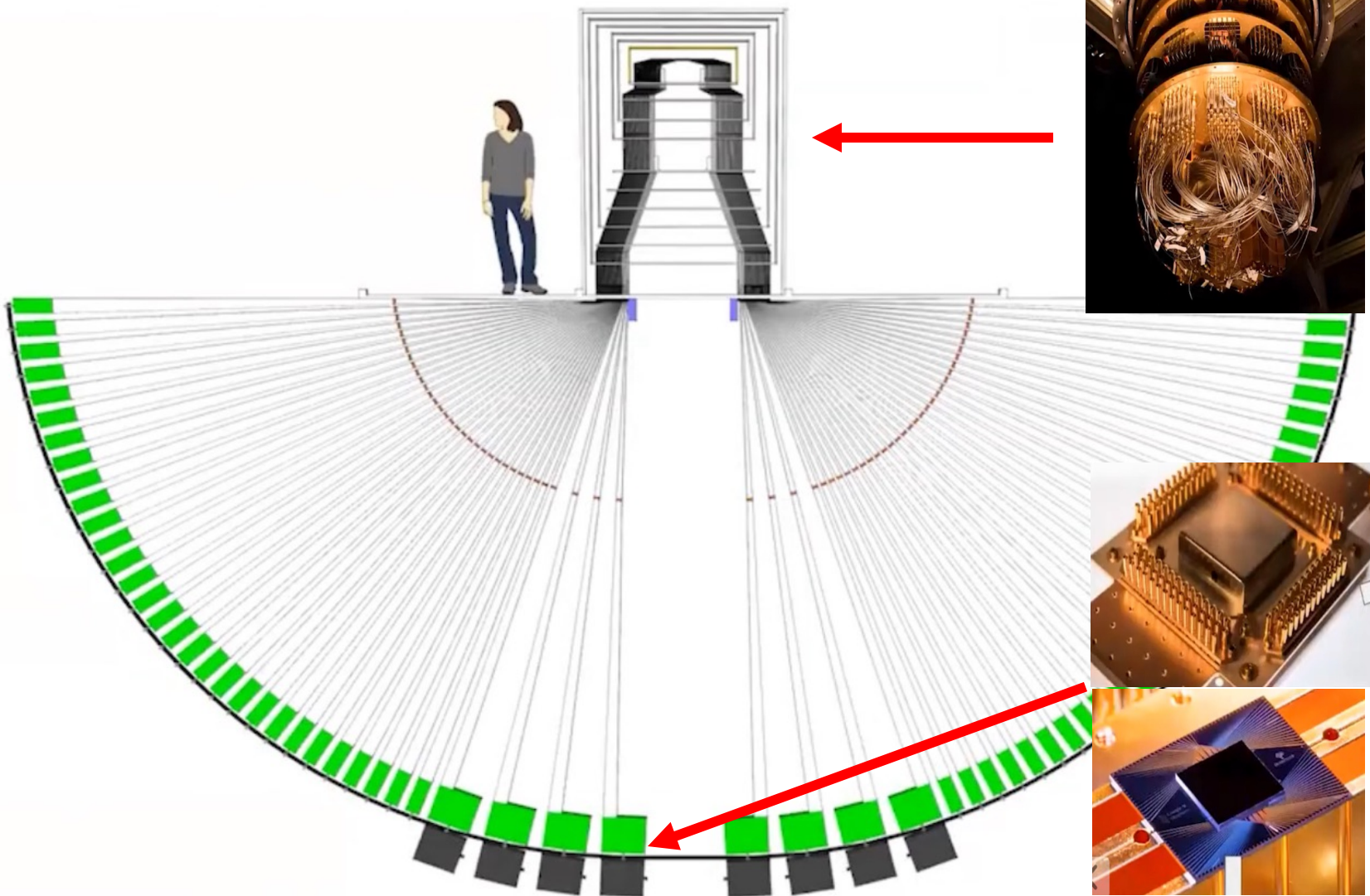
こうした量子エラー訂正技術の大きな前進は、かつて(2017年に) Preskill が、量子コンピューティングの現状を、「NISQ (Noisy Intermediate-Scale Quantum)の時代」と特徴づけた時代から、次の時代、すなわち、ノイズ耐性を持つResilient量子コンピューティングの時代にさし掛かりつつあると考えることができることを意味します。

現代の研究者たちは、「NISQの時代」「Resilientの時代」のさらにその先を見据えて、量子コンピューティングの「Scalingの時代」を展望しようとしています。

# 10k scalable qubits

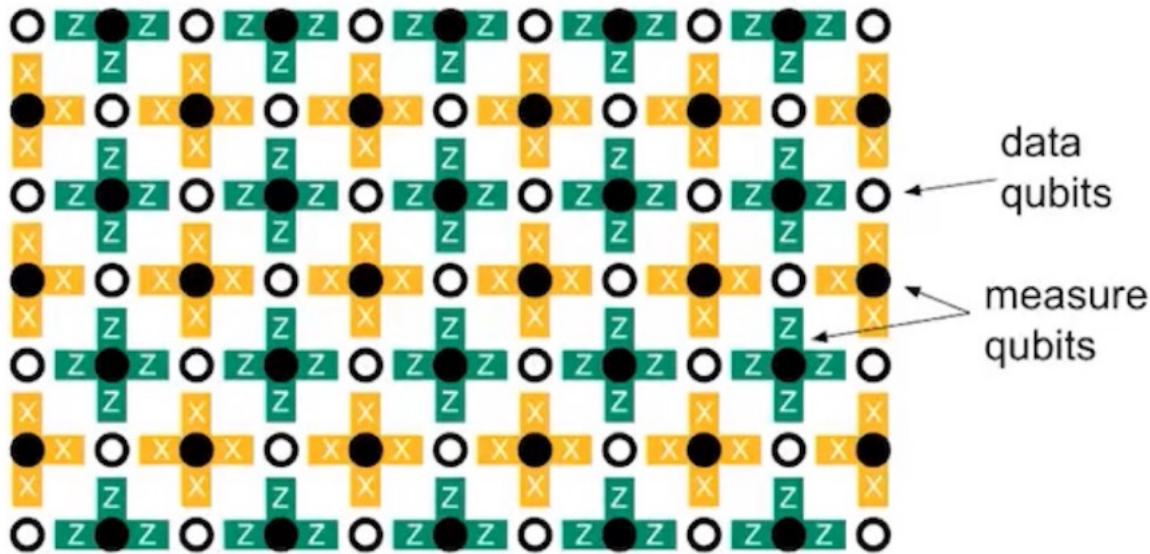


# 10k scalable qubits





# data qubit と measure qubit

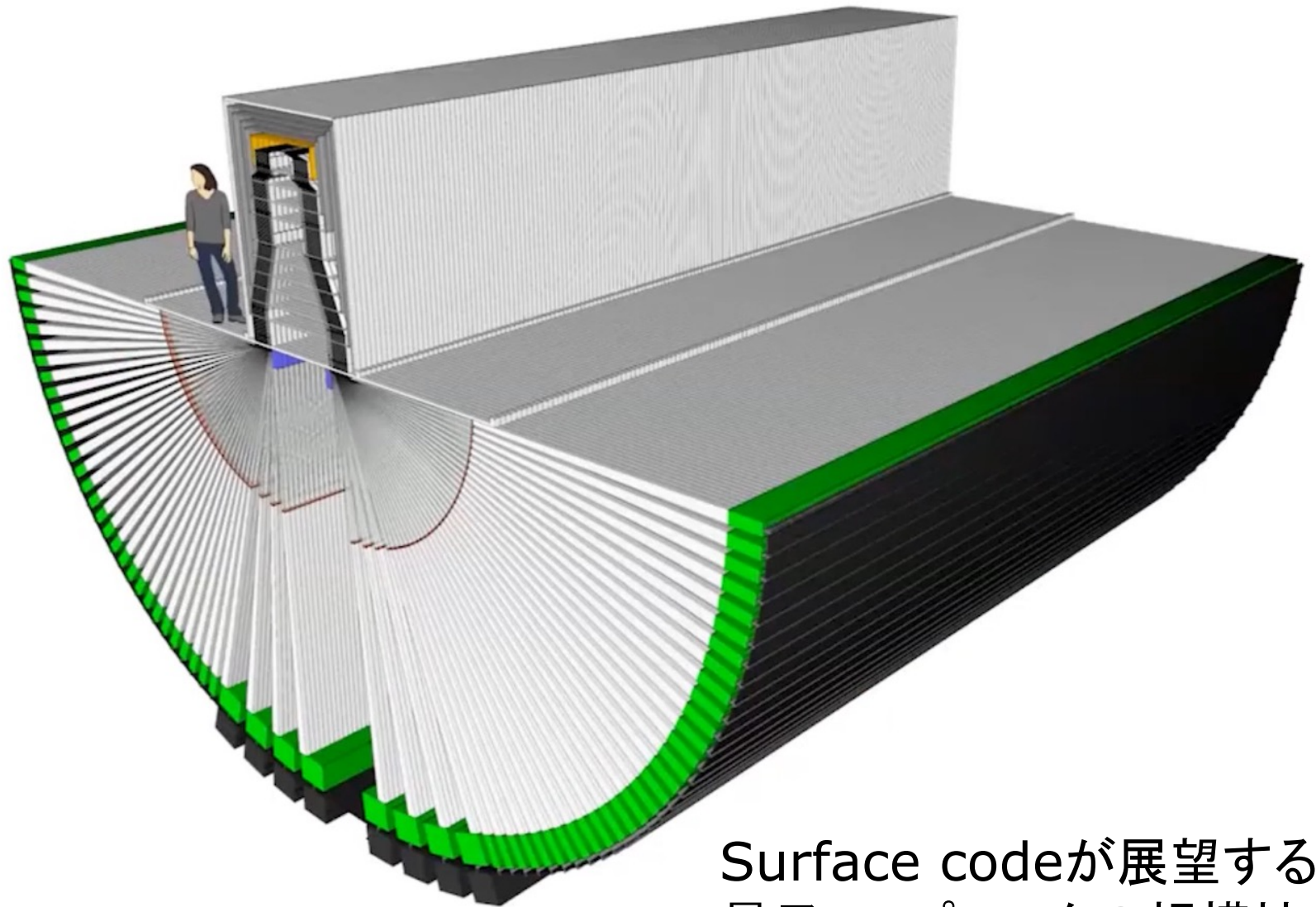


配置されているqubit  
には、二つの種類があ  
ります。

一つは、データを保持  
する**data qubit**で、図  
では白丸で表されてい  
ます。

もう一つは、周囲の  
data qubitの状態を  
監視する**measure  
qubit**です。図では、  
黒丸で表されています。

1M qubits



Surface codeが展望する  
量子コンピュータの規模拡大

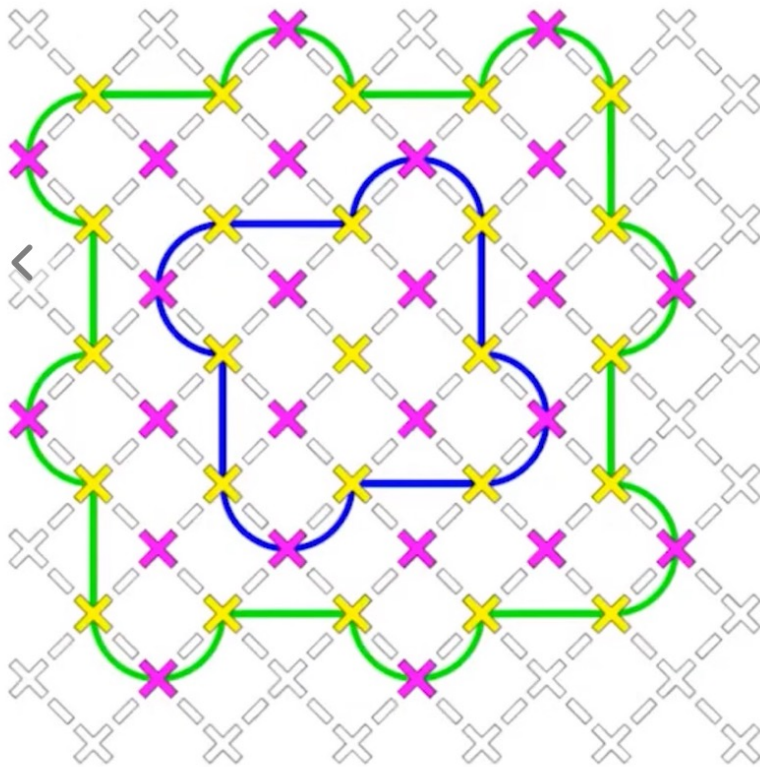
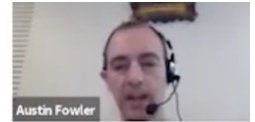
## 物理qubitから「論理qubit」へ

「Scalingの時代」を展望する、量子コンピューティング技術の中で、注目すべき動きがあります。それは、物理的なqubitの上に、量子アルゴリズムごとに物理的に直接量子回路を実装していくというこれまでの素朴なスタイルを見直そうという動きです。

それは、量子コンピュータの「仮想化」と呼ばれることもありますが、現在の技術的な中心は、複数の物理的なqubitから一つの「論理qubit」を構成しようという動きです。

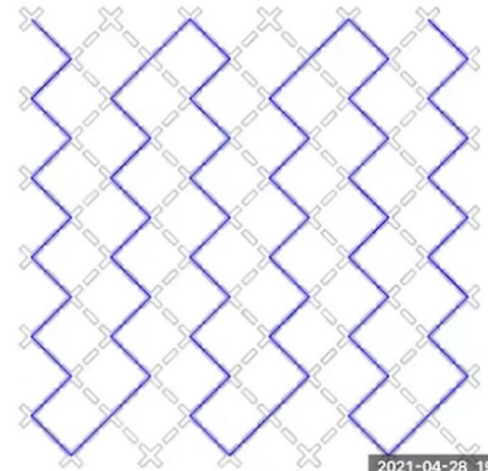
「論理qubit」は、発生史的には、一つのqubitをエラー訂正回路のqubitで取り囲んだものを起源とすると考えることもできるのですが、現在の「論理qubit」の構成は、そうしたプリミティブなものから大きく脱皮しようとしています。

# Device and experiment



- Show  $d=5$  better than  $d=3$
- Continuous running
- Real-time decoding
- $d=34$  extreme exponential suppression ( $d=11$  arXiv:2102.06132)

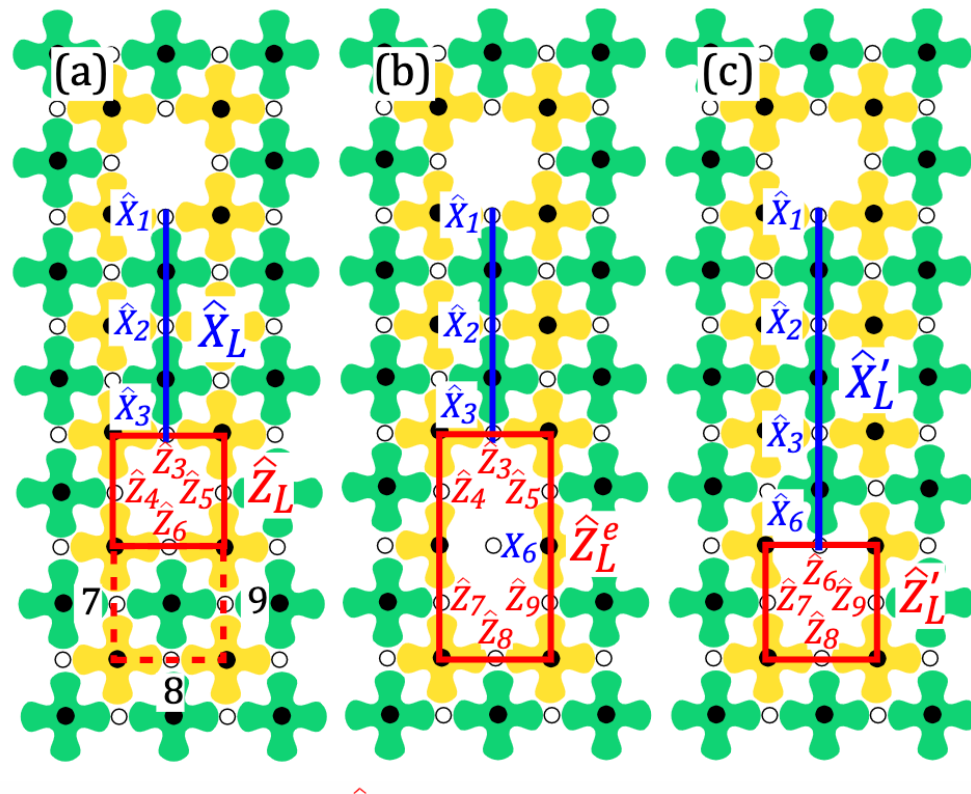
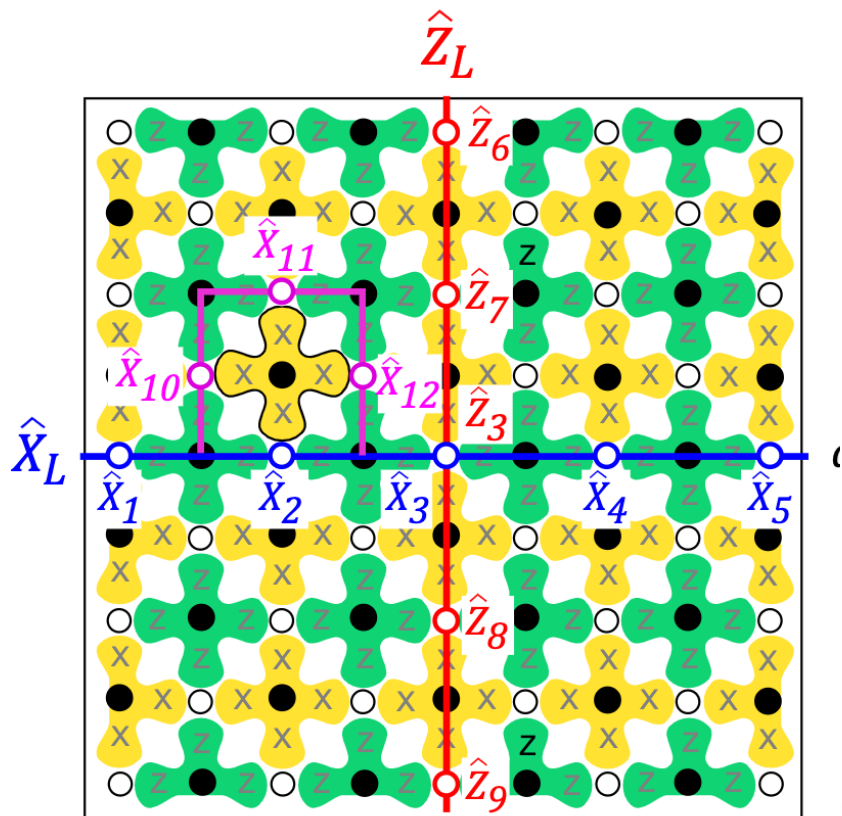
- × Not used
- × Data qubit
- × Measure qubit



2021-04-26 19:10:42

41個の物理data qubit  
からなる論理qubit

論理qubitの移動



$\hat{X}_L, \hat{Z}_L$  は stabilizer

# Stabilizer と Surface code

重要なことは、こうした現在の「論理qubit」概念の成立を支えているのが、量子エラー訂正技術に起源を持つ「Stabilizer」という技術だということです。

Surface code は、このStabilizer 技術を駆使して、多数の物理的qubitという同一の物理的基盤の上に、高いエラー耐性を持つだけでなく、自由に柔軟な量子回路を論理的に構成する可能性を持った、最新のアプローチです。

# Surface codes: Towards practical large-scale quantum computation

Austin G. Fowler

*Centre for Quantum Computation and Communication Technology,  
School of Physics, The University of Melbourne, Victoria 3010, Australia*

Matteo Mariantoni

*Department of Physics, University of California, Santa Barbara, CA 93106-9530, USA and  
California Nanosystems Institute, University of California, Santa Barbara, CA 93106-9530, USA*

John M. Martinis and Andrew N. Cleland

*California Nanosystems Institute, University of California, Santa Barbara, CA 93106-9530, USA*  
(Dated: October 26, 2012)

This article provides an introduction to surface code quantum computing. We first estimate the size and speed of a surface code quantum computer. We then introduce the concept of the stabilizer, using two qubits, and extend this concept to stabilizers acting on a two-dimensional array of physical qubits, on which we implement the surface code. We next describe how logical qubits are formed in the surface code array and give numerical estimates of their fault-tolerance. We outline how logical qubits are physically moved on the array, how qubit braid transformations are constructed, and how a braid between two logical qubits is equivalent to a controlled-NOT. We then describe the single-qubit Hadamard,  $\hat{S}$  and  $\hat{T}$  operators, completing the set of required gates for a universal quantum computer. We conclude by briefly discussing physical implementations of the surface code. We include a number of appendices in which we provide supplementary information to the main text.

<https://arxiv.org/pdf/1208.0928>

# Abstract

This article provides an introduction to surface code quantum computing. We first estimate the size and speed of a surface code quantum computer. We then introduce the concept of the stabilizer, using two qubits, and extend this concept to stabilizers acting on a two-dimensional array of physical qubits, on which we implement the surface code. We next describe how logical qubits are formed in the surface code array and give numerical estimates of their fault-tolerance. We outline how logical qubits are physically moved on the array, how qubit braid transformations are constructed, and how a braid between two logical qubits is equivalent to a controlled-NOT.

## Larger Logical Qubits

- Presents a method to construct larger distance logical qubits
- Notes the relationship between qubit size and error tolerance

## Moving Qubits

- Presents the method used to move logical qubit holes in the 2D array
- Introduces the Heisenberg representation for operators, and describes how to transform logical operators when moving qubit holes
- Presents the concept of byproduct operators

## The Braiding Transformation and the Logical CNOT

- Describes the braid transformation and its effects on logical operators
- Shows that braiding a Z-cut through an X-cut qubit is equivalent to a CNOT with the Z-cut as control

# 今回のセミナーで期待していること

量子エラー訂正技術が、高いエラー耐性を実現したことによって評価されるのは当然ですが、その取り組みの中で生まれた Stabilizer や Surface code といった新しい技術に興味を持ってもらえたらいいと考えています。

今回のセミナーが、Surface code を中心とする新しい量子コンピュータ技術理解の最初の一歩となることを期待しています。





# Part 1

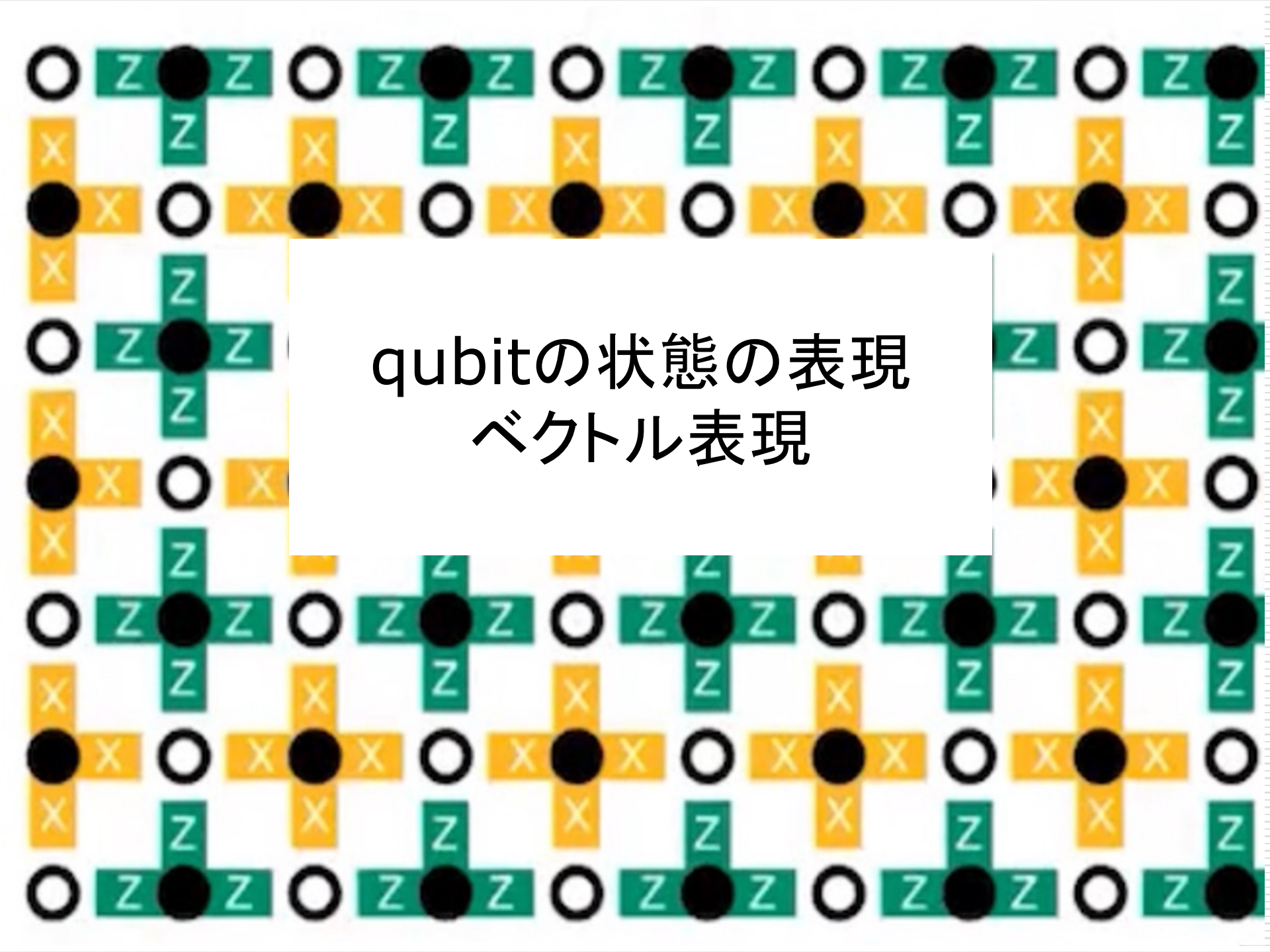
## 量子エラー訂正技術の基礎



# Part 1

## 量子エラー訂正技術の基礎

1. qubitの状態の表現 – ベクトル表現
2. qubitの状態の変化 – ユニタリ変換
3. qubitの状態の観測 – Bornルール
4. Observableの固有値と固有ベクトル

The image features a 10x10 grid of qubit states. Each cell contains a circle (white or black) and a cross (white or black). The circles are labeled 'Z' and the crosses are labeled 'X'. The labels are placed on the horizontal and vertical bars extending from the circles and crosses. The grid is symmetric about a central white box containing the text 'qubitの状態の表現 ベクトル表現'.

qubitの状態の表現  
ベクトル表現

# 1つのqubitの状態

1-qubitの状態 $|Q\rangle$ は、二つのベクトル  $a|0\rangle$  と  $b|1\rangle$  の和重ね合わせ」として次の形のベクトルで表すことができます。

$$\text{qubitの状態 } |Q\rangle = a|0\rangle + b|1\rangle$$

ただし、 $a, b$ は複素数で、 $|a|^2 + |b|^2 = 1$

ここで、 $|0\rangle$ ,  $|1\rangle$ は、次の成分を持つ列ベクトルを表しています。こうした表現をベクトルの「**ケット表記**」といいます。

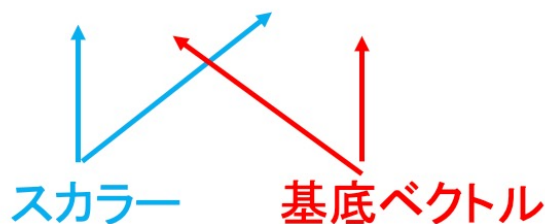
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

二次元のベクトル空間上のすべてのベクトル  $v$  が、二つのスカラー  $a, b$  を用いて、 $v = a|0\rangle + b|1\rangle$  の形で表現できるとき、 $|0\rangle, |1\rangle$  を、この二次元ベクトル空間の「基底」と呼びます。

$|0\rangle, |1\rangle$  は、1-qubit の状態を表す二次元ベクトル空間の基底の一つです。

qubitの状態を成分で表すと、次のようになります。

$$|Q\rangle = a|0\rangle + b|1\rangle = a \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$



もっと一般に、 $n$  個の量子の状態  $|\psi\rangle$  は、  
 $n$  個の状態の重ね合わせとして表現される

$n$ 次元複素ベクトル空間の基底を  $|0\rangle, |1\rangle, |2\rangle, \dots, |n-1\rangle$   
とすると、

$$\text{量子の状態 } |\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} \text{ は、}$$

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_{n-1}|n-1\rangle$$

と表現されます。ただし、 $|c_0|^2 + |c_1|^2 + \dots + |c_{n-1}|^2 = 1$

n 個の量子の状態  $|\psi\rangle$  を、 $\Sigma$  を使って表す

総和の記号 $\Sigma$ を使うと、n 個の量子の状態を見やすく表現できます。

n 個の量子の状態  $|\psi\rangle$  の表現

$$\begin{aligned} |\psi\rangle &= c_0|0\rangle + c_1|1\rangle + \dots + c_{n-1}|n-1\rangle \\ &= \sum_{k=0}^{n-1} c_k|k\rangle \end{aligned}$$

ただし、
$$\sum_{k=0}^{n-1} |c_k|^2 = 1$$

## 二つのqubitの状態をテンソル積で表す

$|P\rangle, |Q\rangle$  で表される状態を持つ二つのqubitの状態は、二つの状態のテンソル積  $|P\rangle \otimes |Q\rangle$  で表すことができます。

$$\begin{aligned} |P\rangle &= a_p|0\rangle + b_p|1\rangle, \\ |Q\rangle &= a_q|0\rangle + b_q|1\rangle \end{aligned}$$

だとすると、 $|P\rangle \otimes |Q\rangle$  の計算は、係数のたすき掛けで次のように行います。

$$\begin{aligned} &|P\rangle \otimes |Q\rangle \\ &= (a_p|0\rangle + b_p|1\rangle) \otimes (a_q|0\rangle + b_q|1\rangle) \\ &= a_p a_q |0\rangle \otimes |0\rangle + a_p b_q |0\rangle \otimes |1\rangle \\ &\quad + b_p a_q |1\rangle \otimes |0\rangle + b_p b_q |1\rangle \otimes |1\rangle \end{aligned}$$

この形は、二つのqubitの状態  $|P\rangle, |Q\rangle$  の状態  $|P\rangle \otimes |Q\rangle$  を表すベクトル空間の基底ベクトルが、

$$\begin{aligned} &|0\rangle \otimes |0\rangle \\ &|0\rangle \otimes |1\rangle \\ &|1\rangle \otimes |0\rangle \\ &|1\rangle \otimes |1\rangle \end{aligned}$$

であることを示しています。

これらの基底を次のように表します。

$$\begin{aligned} &|0\rangle \otimes |0\rangle \quad \dashrightarrow \quad |00\rangle \\ &|0\rangle \otimes |1\rangle \quad \dashrightarrow \quad |01\rangle \\ &|1\rangle \otimes |0\rangle \quad \dashrightarrow \quad |10\rangle \\ &|1\rangle \otimes |1\rangle \quad \dashrightarrow \quad |11\rangle \end{aligned}$$

この基底の表現を使うと、2個のqubitの状態のテンソル積で表される状態 $|\psi\rangle$ は、次のよう形で表現できます。

$$|\psi\rangle = c_{00} |00\rangle + c_{01} |01\rangle + c_{10} |10\rangle + c_{11} |11\rangle$$

$$\text{ただし、 } |c_{00}|^2 + |c_{01}|^2 + |c_{10}|^2 + |c_{11}|^2 = 1$$

## 三つのqubitの状態の基底

先と同様の計算を繰り返すと、三つのqubitの状態の基底は、次のような形になることがわかります。

$$\begin{array}{l} |0\rangle \otimes |0\rangle \otimes |0\rangle \longrightarrow |000\rangle \\ |0\rangle \otimes |0\rangle \otimes |1\rangle \longrightarrow |001\rangle \\ |0\rangle \otimes |1\rangle \otimes |0\rangle \longrightarrow |010\rangle \\ |0\rangle \otimes |1\rangle \otimes |1\rangle \longrightarrow |011\rangle \\ |1\rangle \otimes |0\rangle \otimes |0\rangle \longrightarrow |100\rangle \\ |1\rangle \otimes |0\rangle \otimes |1\rangle \longrightarrow |101\rangle \\ |1\rangle \otimes |1\rangle \otimes |0\rangle \longrightarrow |110\rangle \\ |1\rangle \otimes |1\rangle \otimes |1\rangle \longrightarrow |111\rangle \end{array}$$

# n個のqubitの状態の基底

一般に、n個のqubitからなる系の状態の基底は、次のような形をしていて、 $2^n$ 個あることがわかります。

$$\begin{array}{l} |0 \dots 000 \rangle \\ |0 \dots 001 \rangle \\ |0 \dots 010 \rangle \\ |0 \dots 011 \rangle \\ |0 \dots 100 \rangle \\ \dots \\ \dots \\ |1 \dots 111 \rangle \end{array} \left. \vphantom{\begin{array}{l} |0 \dots 000 \rangle \\ |0 \dots 001 \rangle \\ |0 \dots 010 \rangle \\ |0 \dots 011 \rangle \\ |0 \dots 100 \rangle \\ \dots \\ \dots \\ |1 \dots 111 \rangle \end{array}} \right\} 2^n$$

$\underbrace{\hspace{10em}}_n$

## ベクトルの内積と外積

ベクトル $|i\rangle$ とベクトル $|j\rangle$ の**内積**は、ケット記法では、 $\langle i|j\rangle$ の形で表される**スカラー**です。

ベクトル $|i\rangle$ とベクトル $|j\rangle$ に対して、ケット記法で  $|i\rangle\langle j|$  の形で表される積を、ベクトル $|i\rangle$ とベクトル $|j\rangle$ の**外積**と言います。

## 外積の働き

外積  $|i\rangle\langle j|$  は、ベクトル  $|k\rangle$  に作用して、ベクトル  $a|i\rangle$  (ただし、 $a$ は、内積  $\langle j|k\rangle$  で定義されるスカラー値)を生み出す演算子であることは、次のような計算でわかります。

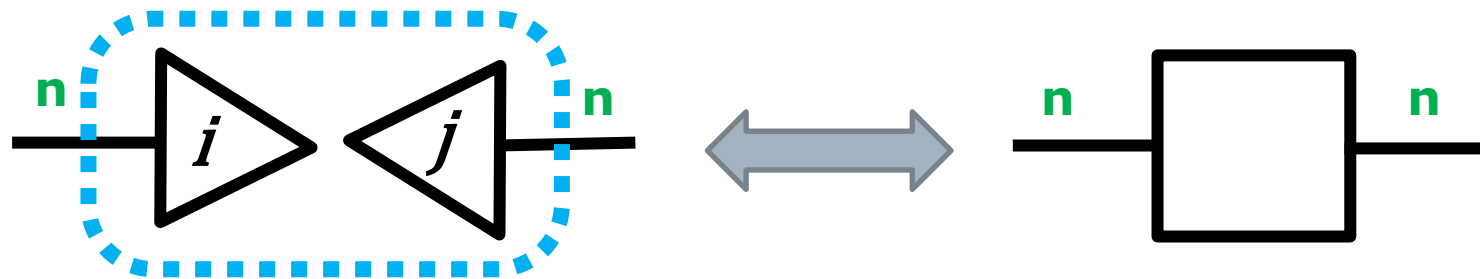
$$\begin{aligned} (|i\rangle\langle j|) (|k\rangle) &= \\ |i\rangle\langle j||k\rangle &= |i\rangle (\langle j|k\rangle) = (\langle j|k\rangle) |i\rangle = a|i\rangle \end{aligned}$$

これは、内積なので  
スカラーになる

ただし、 $a = \langle j|k\rangle$   
なるスカラーである

外積  $|i\rangle\langle j|$  は、ベクトル  $|k\rangle$  に作用して、ベクトル  $a|i\rangle$  を生み出すのですから、外積  $|i\rangle\langle j|$  は、行列であることがわかります。

ベクトル  $|i\rangle, |j\rangle$  が、ともに  $n$ 次元のベクトルであるとき、外積  $|i\rangle\langle j|$  は  $n \times n$  の行列になります。



左右に腕の出た行列である

# エンタングルメント

これまで、 $|P\rangle, |Q\rangle$  で表される状態を持つ二つのqubitの状態は、**二つの状態のテンソル積  $|P\rangle \otimes |Q\rangle$**  で表すことができることを見てきました。

ただ、二つのqubit からなる状態でありながら、二つのqubitの状態のテンソル積  $|P\rangle \otimes |Q\rangle$  では表されない状態が存在します。

こうした状態を、「エンタングルメント」状態と呼びます。

# 基本的な2-qubitのエンタングルメント状態

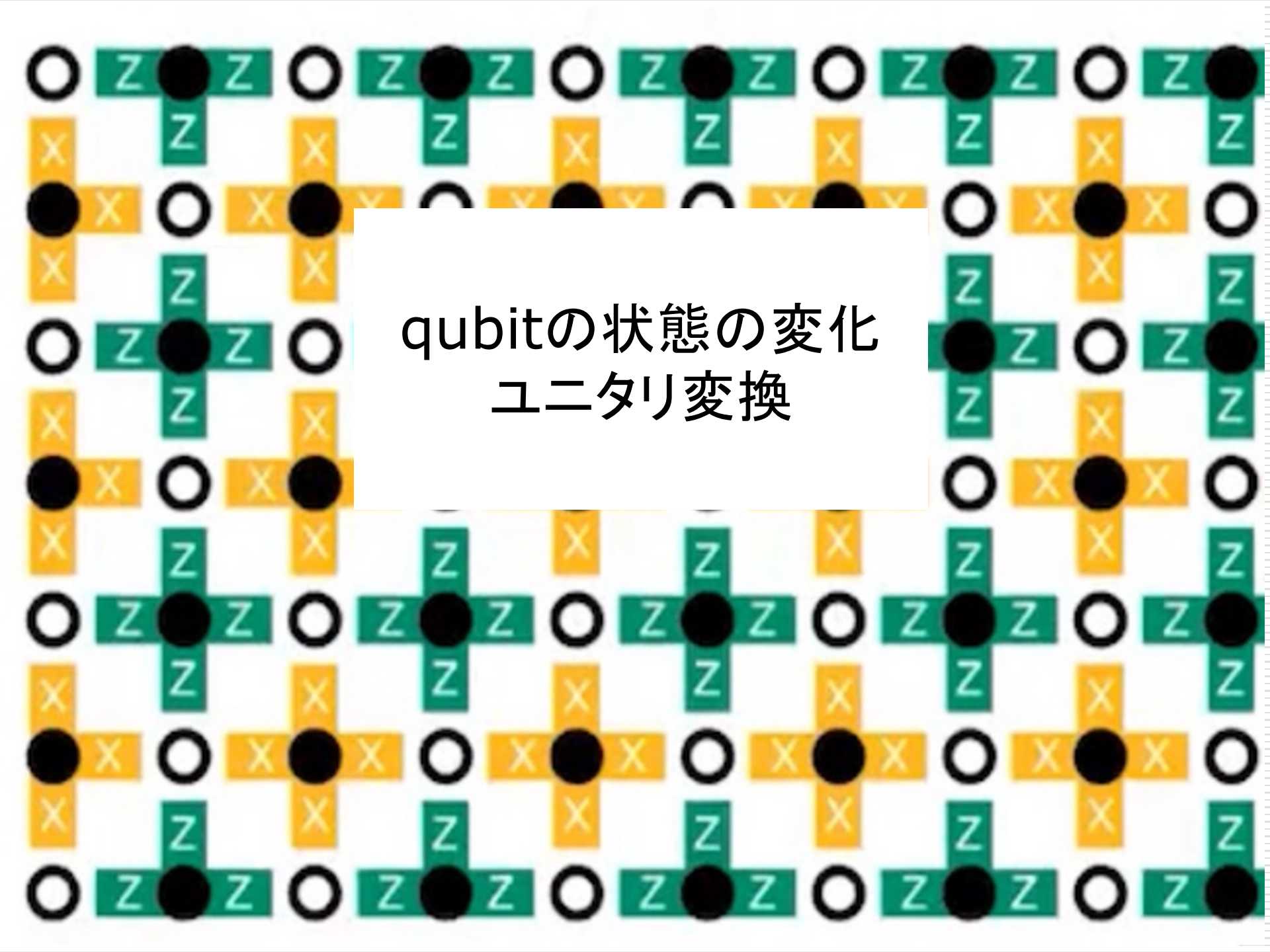
最も基本的な2-qubitのエンタングルメント状態を **Bell State** と呼びます。次の四つがあります。

$$(|00\rangle + |11\rangle)/\sqrt{2} = |\Phi^+\rangle$$

$$(|00\rangle - |11\rangle)/\sqrt{2} = |\Phi^-\rangle$$

$$(|01\rangle + |10\rangle)/\sqrt{2} = |\Psi^+\rangle$$

$$(|01\rangle - |10\rangle)/\sqrt{2} = |\Psi^-\rangle$$

The image features a 10x10 grid of qubit states. Each cell contains a black circle (representing state 0) or a white circle (representing state 1). The circles are arranged in a checkerboard pattern. Labels 'Z' and 'X' are placed on the grid. 'Z' labels are in green boxes and are located at the top and bottom of each column, and in the middle of each row. 'X' labels are in yellow boxes and are located on the left and right sides of each row, and in the middle of each column. The labels are positioned such that they appear to be associated with the qubit states in the grid.

qubitの状態の変化  
ユニタリ変換

# qubitの状態の変化

このセッションではqubitの状態の変化について考えます。

量子コンピューティングは、ある量子アルゴリズムに基づいて我々が意図したようにqubitの状態を操作することで、計算の目的を達成しようとしています。

ただ、qubitの状態は非常に不安定なので、現実には、我々の意図とは無関係に、qubitの状態が変化することが起こりえます。

我々は、そうしたタイプの量子状態の変化を「量子エラー」とよびます。

# 量子の状態はベクトルで表現される

量子の状態がベクトルで表されるのなら、量子の状態の変化は、あるベクトルから他のベクトルへの変化として表されます。

あるベクトルから他のベクトルへの変換は、行列とベクトルの積で表現される。 $n \times n$  の行列に  $n$ 次元のベクトルを掛けると  $n$ 次元のベクトルが生まれます。

量子の状態の変化は、  
行列とベクトルの積で表現される。

ある行列Uと $|a|^2 + |b|^2 = 1$  という条件を満たすベクトル $\begin{pmatrix} a \\ b \end{pmatrix}$ の積が

$$U \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}$$

になったとしましょう。

このベクトル $\begin{pmatrix} c \\ d \end{pmatrix}$ が、qubitの状態を表しているためには、

$|c|^2 + |d|^2 = 1$  という条件が必要です。

こうした変換の条件を満たす行列Uをユニタリ行列と言います。

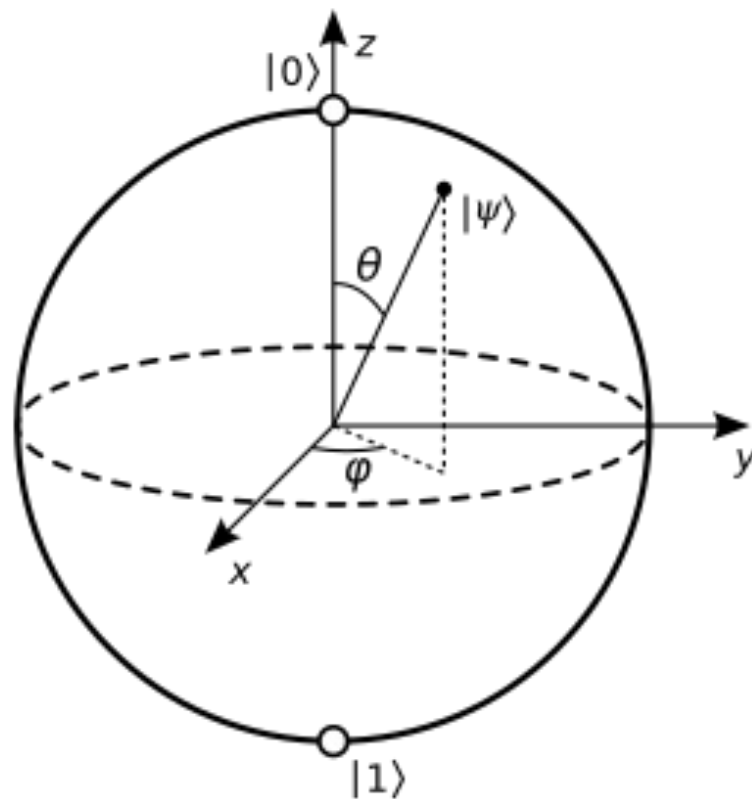
量子の状態の変化は、  
ユニタリ行列とベクトルの積で表現される。

## ユニタリ行列は「回転」を表す

先の例は、二次元の実数平面の話でしたが、ユニタリ行列が「回転」を表すなら、複素数値の係数を持つ純粋状態のqubitも、**回転によって不変な原点から1の距離をなす球面上の点をさすベクトル $|\psi\rangle$** として表現することが可能です。

右のBloch sphere は、そうしたものです。

ここでは、qubitの状態は、この球面上の点として表現されています。



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

# エルミート行列とユニタリ行列

行列Mの行と列を入れ替え、その成分の複素共役をとったものを、行列Mのエルミート共役といい、 $M^\dagger$ と表します。

$$M^\dagger = M$$

が成り立つ時、Mをエルミート行列と言います。

また、

$$UU^\dagger = U^\dagger U = I \text{ (単位行列)}$$

が成り立つ時、Uをユニタリ行列と言います。

# ユニタリ行列は、qubitの状態を qubitの状態に変換する演算子

ユニタリ行列 $U$ は、 $U$ とベクトル $v$ の積を取ることで、qubitの状態を表すベクトル $v$ をqubitの状態を表すベクトル $w$ に変換する。

$$Uv = w$$

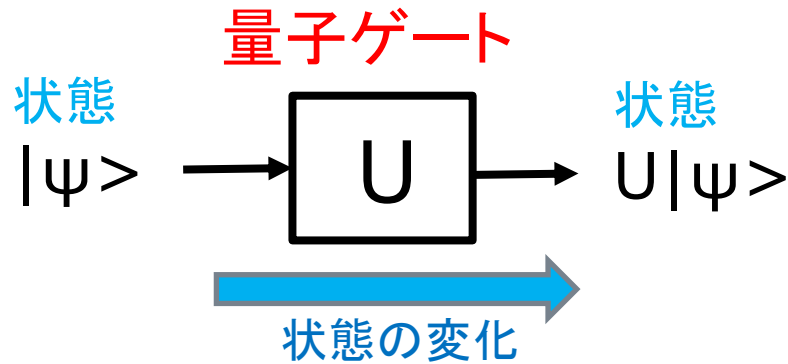
ベクトル空間に作用する行列 $U$ を「演算子」と呼ぶ。ユニタリ行列は、qubitの状態をqubitの状態に変換する演算子である。

先の関係はケット記法で書くと、 $v = |\psi\rangle$ なら、 $w = U|\psi\rangle$ となる。


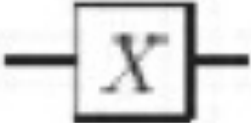
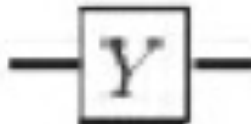

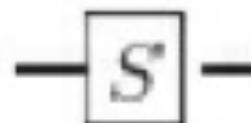
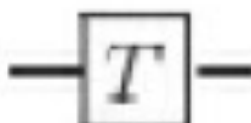
これは、数学的な関係である。

# 「量子ゲート」としてのユニタリ行列

量子コンピュータのような、時間の中の現実的な過程では、次のような図式で、ユニタリ行列の働きを、量子状態の変化を引き起こす「量子ゲート」として捉えることができる。



# 1-qubitの量子ゲートの例とそれに対応する行列

Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

# 代表的なユニタリ行列

以下では、代表的なユニタリ行列＝量子演算子として、次のものを取り上げ、その性質を見ていく。

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

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

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

## 行列 $X, Z, H$ の性質

$$X^\dagger = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^\dagger = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = X$$

$$Z^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z$$

$$H^\dagger = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^\dagger = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H$$

$$X^\dagger = X, \quad Z^\dagger = Z, \quad H^\dagger = H$$

$X, Z, H$  はエルミート行列である。

## $X^2, Z^2, H^2$

$$X^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

$$Z^2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

$$H^2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = I$$

$$X^2 = Z^2 = H^2 = I$$

行列  $X, Z, H$  はユニタリ行列である

$X^\dagger = X, Z^\dagger = Z, H^\dagger = H$  と  $X^2 = Z^2 = H^2 = I$  より

$$X^\dagger X = X X^\dagger = X^2 = I$$

$$Z^\dagger Z = Z Z^\dagger = Z^2 = I$$

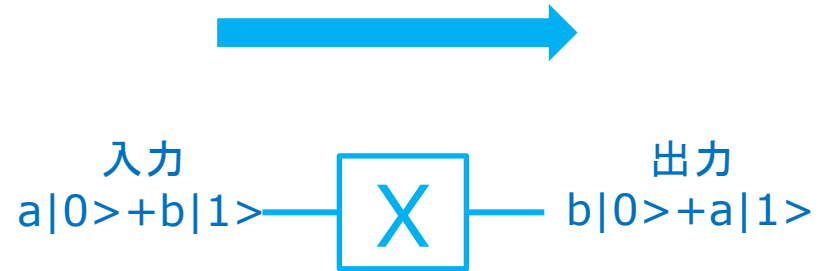
$$H^\dagger H = H H^\dagger = H^2 = I$$

行列  $X, Z, H$  はユニタリ行列である

# X, Z, H の行列と対応する量子ゲートの働き

$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$  なので、

$$X \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix}$$



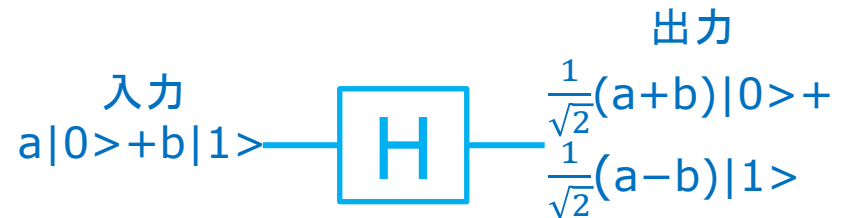
$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$  なので、

$$Z \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ -b \end{pmatrix}$$



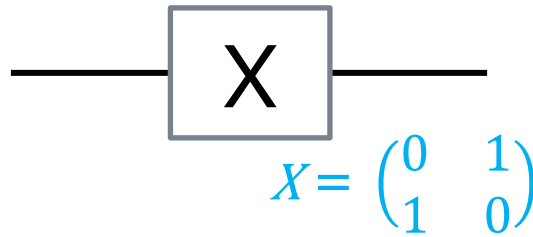
$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  なので、

$$H \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} a + b \\ a - b \end{pmatrix}$$



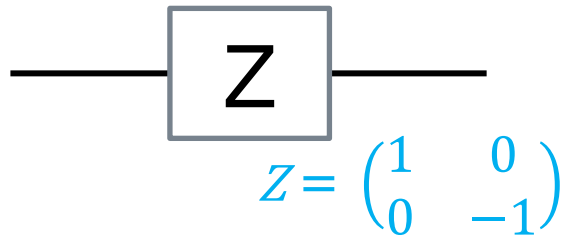
# Bit Flipper, Phase Flipper 基底変換器

入力  出力



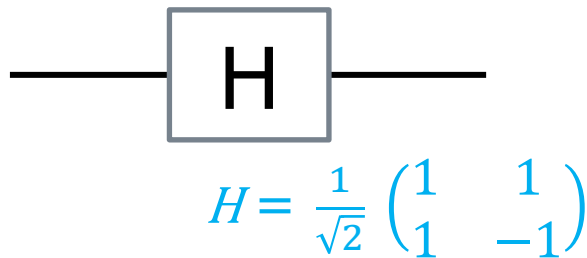
## Bit Flipper

$$a|0\rangle + b|1\rangle \rightarrow b|0\rangle + a|1\rangle$$



## Phase Flipper

$$a|0\rangle + b|1\rangle \rightarrow a|0\rangle - b|1\rangle$$



## Hadamard

$$a|0\rangle + b|1\rangle \rightarrow \frac{1}{\sqrt{2}}(a+b)|+\rangle + \frac{1}{\sqrt{2}}(a-b)|-\rangle$$

$|0\rangle, |1\rangle$  基底から

$|+\rangle, |-\rangle$  基底への変換

$$XZ = -ZX$$

$$X(a|0\rangle + b|1\rangle) = b|0\rangle + a|1\rangle$$

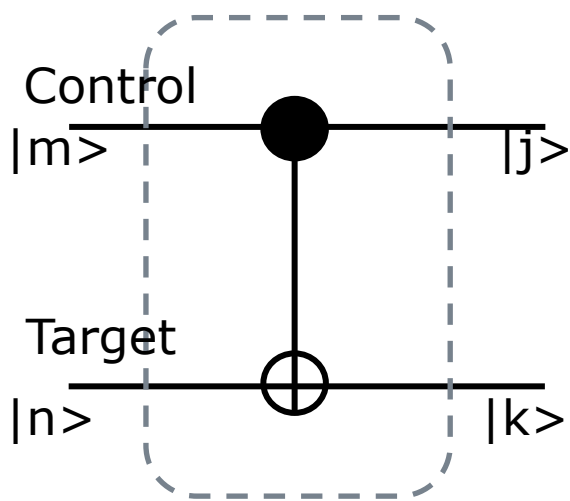
$$Z(a|0\rangle + b|1\rangle) = a|0\rangle - b|1\rangle$$

$$XZ(a|0\rangle + b|1\rangle) = X(a|0\rangle - b|1\rangle) = -b|0\rangle + a|1\rangle$$

$$ZX(a|0\rangle + b|1\rangle) = Z(b|0\rangle + a|1\rangle) = b|0\rangle - a|1\rangle$$

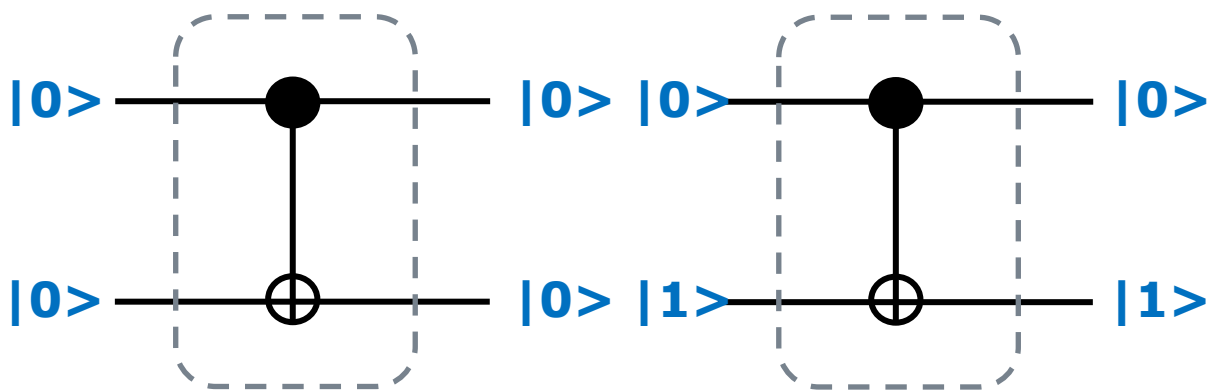
# 2-qubitのゲート CNOT (Control-NOT)

Control が  $|1\rangle$  の時  
Target のNOTをとる

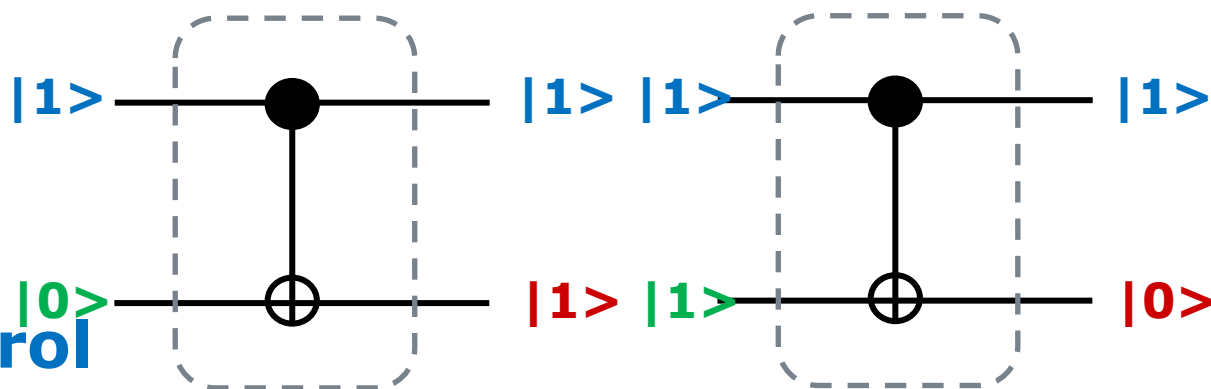


$|mn\rangle$  の  
第一bit  $|m\rangle$  がControl  
第二bit  $|n\rangle$  がTarget

Controlが  $|0\rangle$  なら何もしない

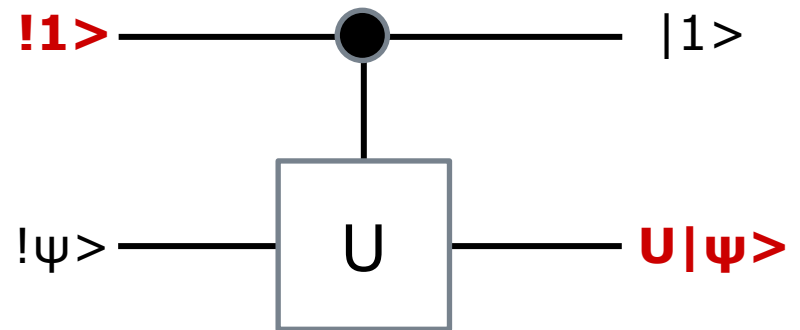
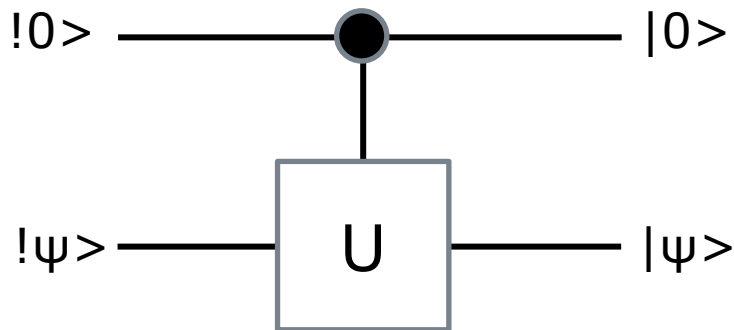


Controlが  $|1\rangle$  ならNOT操作



## 2-qubitのゲート **Control-U**

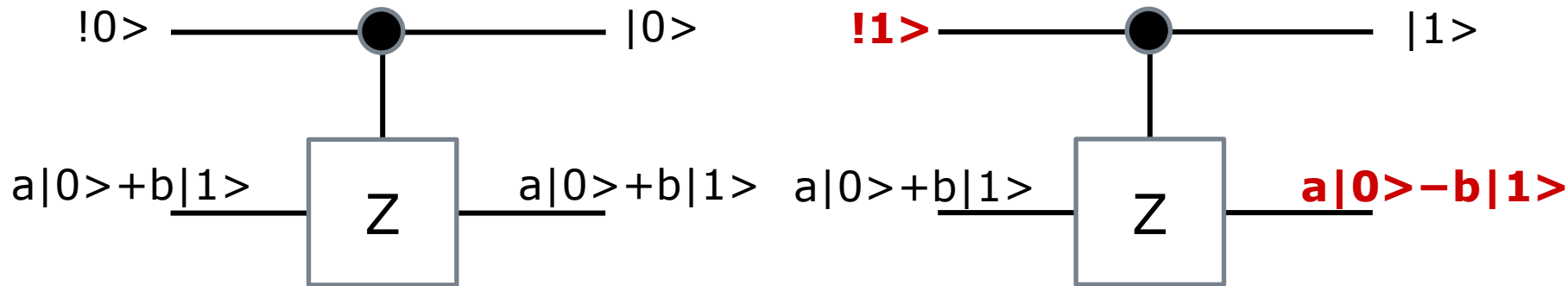
コントロール qubit が、 $|0\rangle$  の時には、何もせず、(左図)  
コントロール qubit が、 $|1\rangle$  の時には、ターゲットのqubit  
 $|\psi\rangle$  にユニタリ演算子  $U$  を適用した  $U|\psi\rangle$  を出力する(右図)  
ゲートを、**Control-U** ゲートという。



# 2-qubitのゲート Control-U の例 (1)

## Control-Z

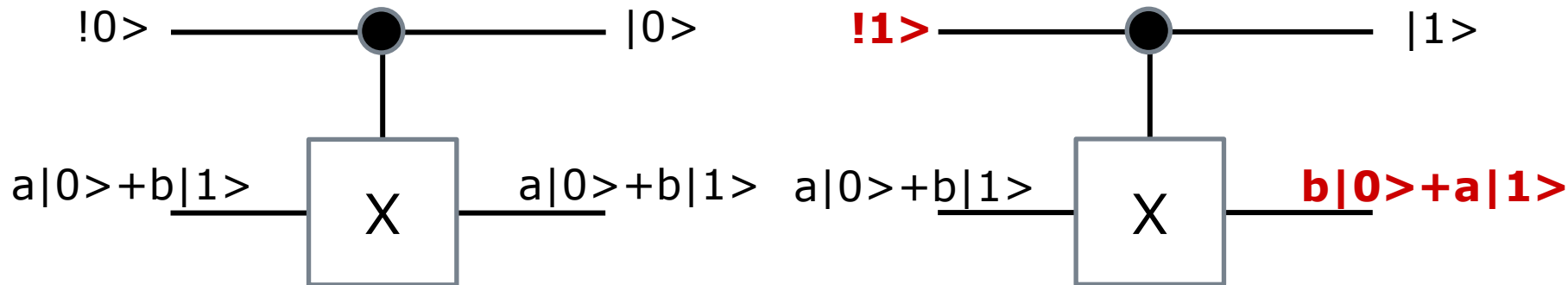
コントロール qubit が、 $|0\rangle$  の時には、何もせず、(左図)  
コントロール qubit が、 $|1\rangle$  の時には、ターゲットのqubit  
 $|\psi\rangle$  にユニタリ演算子  $Z$  を適用した  $Z|\psi\rangle$  を出力する(右図)



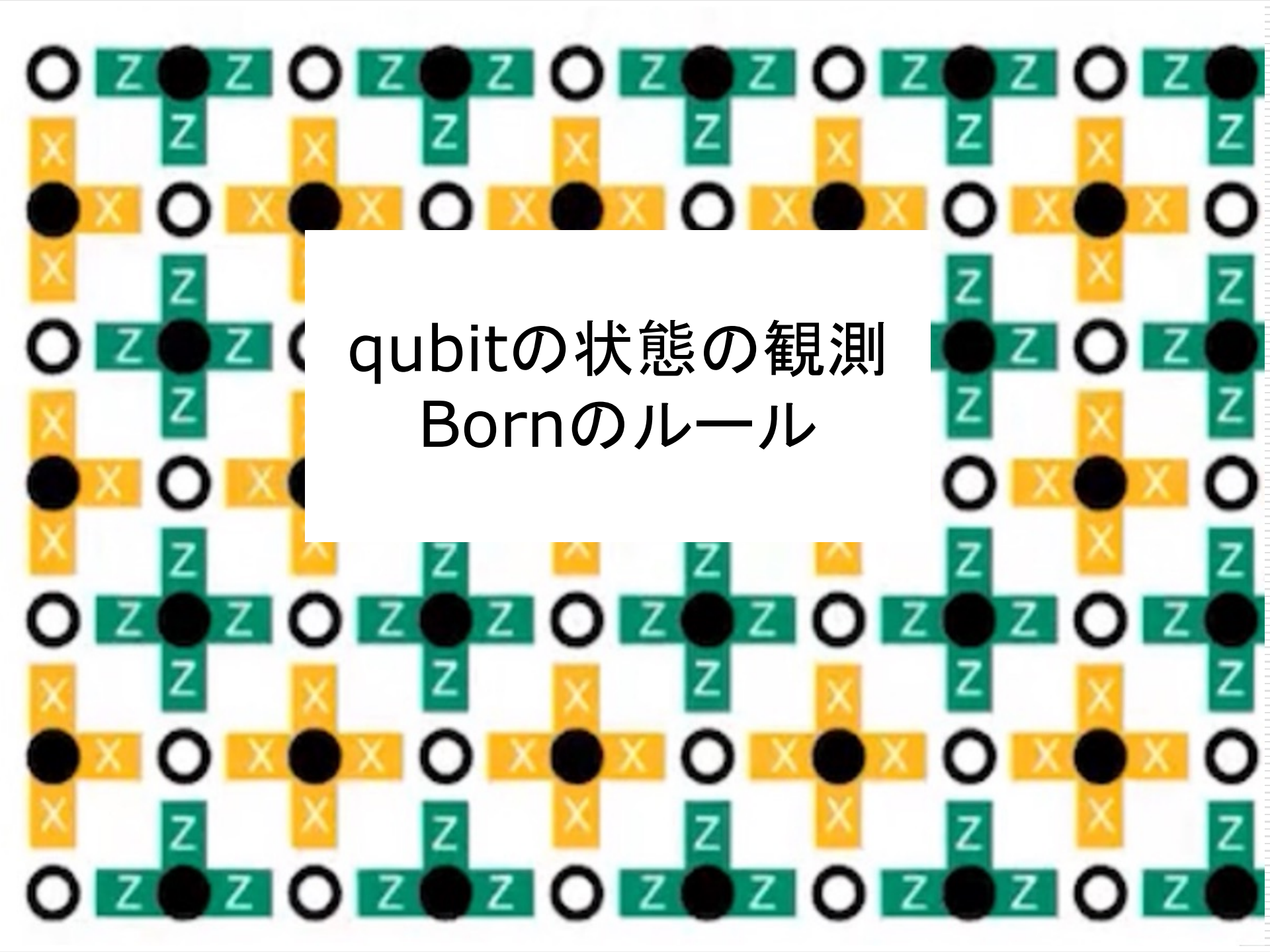
## 2-qubitのゲート Control-U の例 (2)

### Control-X

コントロール qubit が、 $|0\rangle$  の時には、何もせず、(左図)  
コントロール qubit が、 $|1\rangle$  の時には、ターゲットのqubit  
 $|\psi\rangle$  にユニタリ演算子  $X$  を適用した  $X|\psi\rangle$  を出力する(右図)



Control-Xは、CNOTと等しいことを確かめよ

The image features a 10x10 grid of qubit states. Each cell in the grid contains a pair of symbols: a white circle and a black circle. The white circles are arranged in a checkerboard pattern, with white circles at (row, col) where both row and col are even (assuming 0-indexing from top-left). The black circles are at (row, col) where both row and col are odd. The symbols are surrounded by green and yellow rectangular blocks. Green blocks contain the letter 'Z' and yellow blocks contain the letter 'X'. The blocks are arranged in a grid that is offset from the qubit symbols. A central white box with a black border contains the text 'qubitの状態の観測' and 'Bornのルール' in black Japanese characters.

qubitの状態の観測  
Bornのルール

# qubitの状態の観測

このセッションでは、qubitの状態の観測の問題を扱います。

ただ、このセッションのコンテンツの中心は、量子論の基本的な原理の一つである「観測の原理」の初等的な説明です。

# qubitの状態の観測の奇妙さ

qubitの状態を観測しようとする、我々はさまざまな奇妙な現象に直面します。量子論の奇妙さの多くは、この観測の問題に起因しています。

qubitの状態を観測するというのは、どういうことでしょうか？

$a|0\rangle + b|1\rangle$  という状態をもつqubitを観測して、我々が知りたいのは、 $a$ ,  $b$  という二つの値です。

ところが、我々は qubitを観測することでは、 $a$ ,  $b$  の値を知ることができないのです。

# qubitの状態

qubitの状態は、  
状態  $|0\rangle$  と状態  $|1\rangle$  の  
重ね合わせの状態を  
取っている

qubitの状態は、  
 $a|0\rangle + b|1\rangle$  という形で  
二つの数字  $a, b$  で表される。

$$|\text{qubit}\rangle = a|0\rangle + b|1\rangle$$

$$|a|^2 + |b|^2 = 1 ; a, b \in \mathbb{C}$$

# qubitの状態

観測

$$|\text{qubit}\rangle = a|0\rangle + b|1\rangle$$

$$|a|^2 + |b|^2 = 1 ; a, b \in \mathbb{C}$$

## qubitの状態

観測

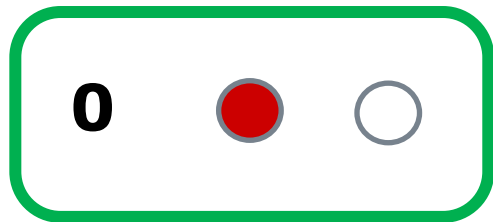
$a|0\rangle + b|1\rangle$   
の重ね合わせの  
状態は、失われる

観測を行うと、qubitの  
重ね合わせの状態は  
失われる。

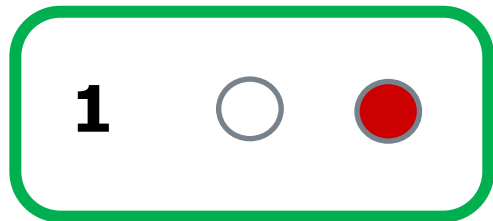
## qubitの状態



$a|0\rangle + b|1\rangle$   
の重ね合わせの  
状態は、失われる



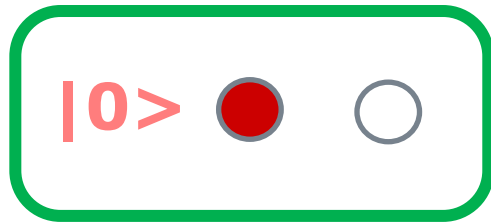
あるいは



観測を行うと、qubitの重ね合わせの状態は失われ、0か1かの情報が返る。  
一つの数字のみが返る。

qubitは、観測すると古典ビットのようになる

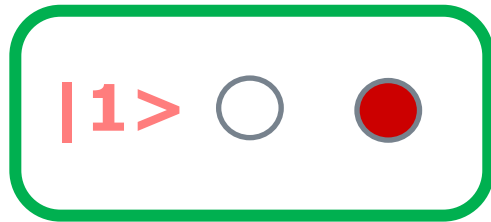
$$|a|^2 + |b|^2 = 1 ; a, b \in \mathbb{C}$$



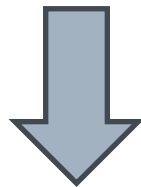
$|0\rangle$ が残る  
確率は  $|a|^2$

$a|0\rangle + b|1\rangle$   
の重ね合わせの  
状態は、失われる

あるいは



$|1\rangle$ が残る  
確率は  $|b|^2$

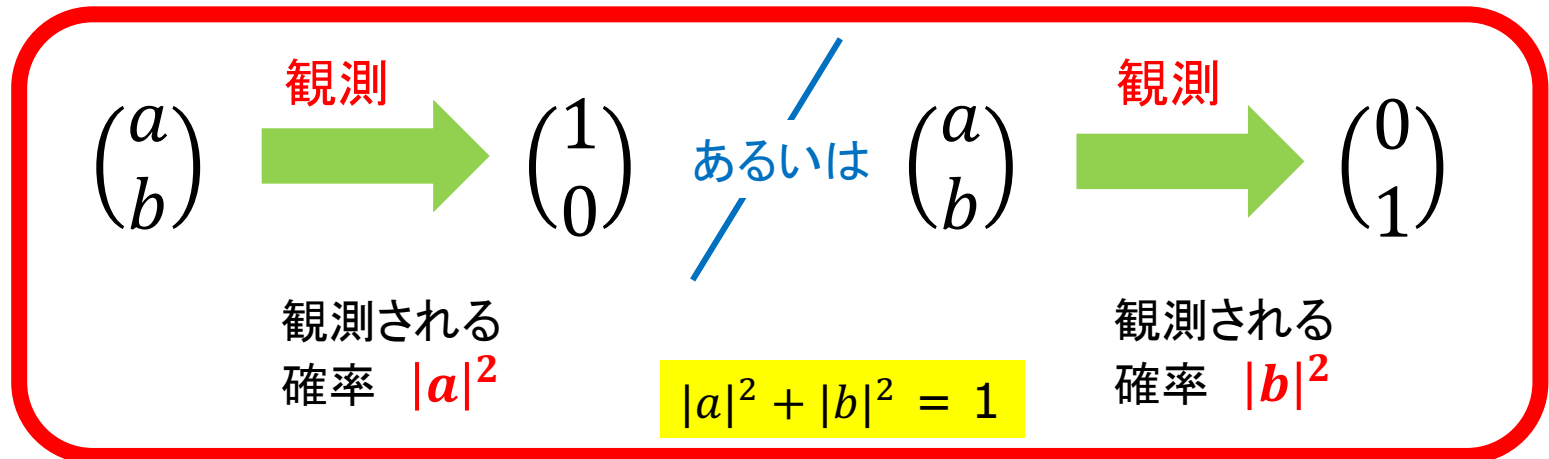


これをBornのルールと呼ぶ

# 観測の原理

観測を行うと、qubitの重ね合わせの状態は失われ、0か1かの情報が返る。(  $|0\rangle$  または  $|1\rangle$  が観測される。)

この時0を得る確率は  $|a|^2$  で、1を得る確率は  $|b|^2$  で、与えられる。



# 一般の量子状態の観測

一般の量子状態  $|\psi\rangle = \sum_{k=0}^{n-1} c_k |k\rangle$  が与えられた時、

状態  $|0\rangle$  が観測される確率は、 $|c_0|^2$  で与えられる。  
観測前の状態  $|\psi\rangle$  は、新しい状態  $|0\rangle$  に変わる。

状態  $|1\rangle$  が観測される確率は、 $|c_1|^2$  で与えられる。  
観測前の状態  $|\psi\rangle$  は、新しい状態  $|1\rangle$  に変わる。

...

状態  $|k\rangle$  が観測される確率は、 $|c_k|^2$  で与えられる。  
観測前の状態  $|\psi\rangle$  は、新しい状態  $|k\rangle$  に変わる。

# qubitの状態の観測の例

ここでは、qubitの状態の観測の例を紹介します。

一つは、アダマール・ゲートの出力の観測例、  
もう一つは、エンタングルした状態の観測例です。

## $|+\rangle$ の観測

$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$  を観測してみよう

Bornのルールから、

0 が観測される確率は、 $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$

1 が観測される確率は、 $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$

だから、0と1は、同じ確率で観測される。

状態  $|+\rangle$  を、「量子コイン」と呼ぶ。

## $|-\rangle$ の観測

$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$  を観測してみよう

Bornのルールから、

$$0 \text{ が観測される確率は、} \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}$$

$$1 \text{ が観測される確率は、} \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}$$

だから、0と1は、同じ確率で観測される。

状態  $|-\rangle$  も、「量子コイン」として振る舞う。

また、観測では、状態  $|+\rangle$  と状態  $|-\rangle$  との区別がつかないことがわかる。

# エンタングルしたqubitの状態の観測

すべての2-qubitsの状態が、二つの1-qubitのテンソル積に分解できるとは限りません。

二つの1-qubitの状態のテンソル積に分解できない2-qubitの状態を、**エンタングルメント** といいます。

## EPRペア

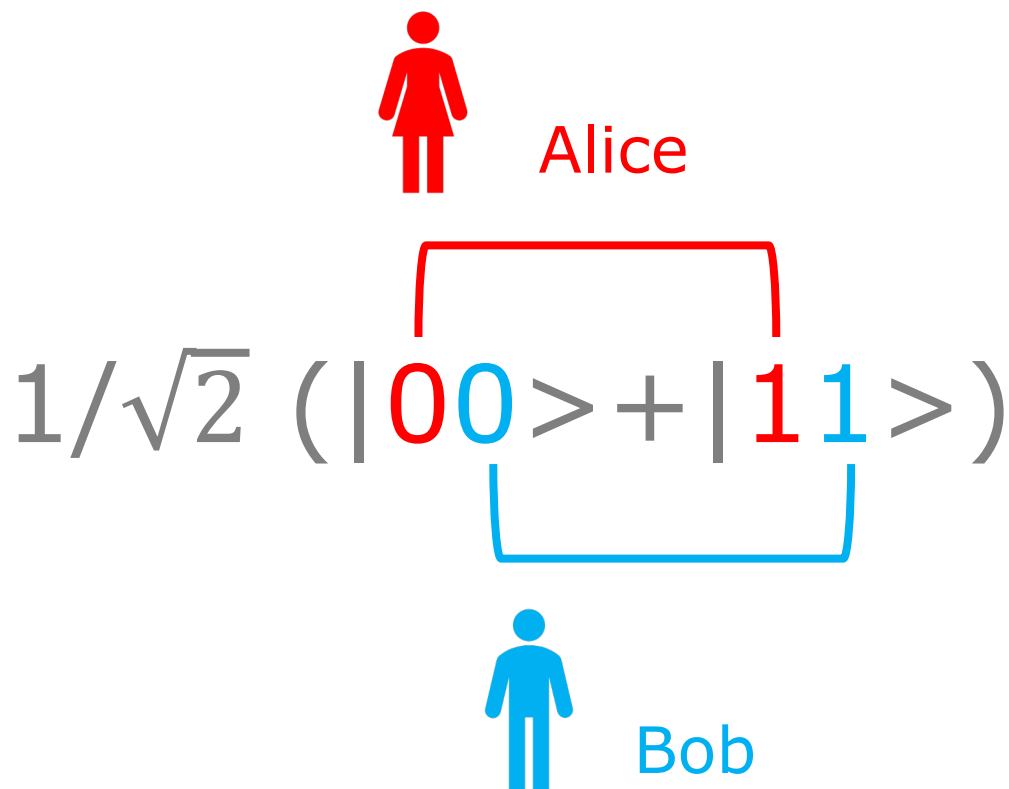
### 二つのqubitの状態が一つの式で表される

通常は、二つのqubitの状態は、それぞれのqubitの状態を表す二つの式で表されるのですが、EPRペアの場合、二つのqubitの状態が一つの式で表されています。

一つの式で表されているのですが、この一つの式は二つのqubitの状態を表しています。ですから、一方のqubitをAliceが、他方のqubitをBobが持つことができます。

独立したAliceとBobが独立して持てるのですから、「この二つのqubitは、独立している」と考えたくなりますが、実は二つのqubitは、独立していないんです。

# EPRペア: エンタングルした二つのqubit



$1/\sqrt{2} (|00\rangle + |11\rangle)$  で表される状態は、二つのqubitの状態である。  
一方のqubitをAliceが、他方のqubitをBobが持つことができる。

# EPRペア:もつれ合った二つのqubit



Alice

$$1/\sqrt{2} (|00\rangle + |11\rangle)$$

$$1/\sqrt{2} (|00\rangle + |11\rangle)$$



Bob

$$1/\sqrt{2} (|00\rangle + |11\rangle)$$

Aliceが観察できるのは、 $|00\rangle + |11\rangle$ の**第一bit**で、  
Bobが観察できるのは、 $|00\rangle + |11\rangle$ の**第二bit**である。  
この関係は、両者がどんなに離れていても変わらない。

## EPRペアを観測する

AliceとBobがEPRペア  $1/\sqrt{2} (|00\rangle + |11\rangle)$  で表される状態のqubitを持っているとしましょう。

Aliceが持っているqubitは、EPRペアの片割れで、その状態は、次の赤字で示す状態に対応しています。

$$1/\sqrt{2} (|00\rangle + |11\rangle)$$

Bobが持っているqubitは、EPRペアの片割れで、その状態は、次の青字で示す状態に対応しています。

$$1/\sqrt{2} (|00\rangle + |11\rangle)$$

## AliceがEPRペアを観測する

AliceがEPRペア  $1/\sqrt{2} (|00\rangle + |11\rangle)$ を観測したとしましょう。  
Aliceが観測できるのは、自分が持つqubitの状態だけです。

それが $|0\rangle$ である確率は、 $1/\sqrt{2}(|00\rangle + |11\rangle)$ から、 $|01\rangle$ の係数  $1/\sqrt{2}$  の絶対値の二乗で $1/2$ となります。

同様に、

それが $|1\rangle$ である確率は、 $1/\sqrt{2}(|00\rangle + |11\rangle)$ から、 $|11\rangle$ の係数  $1/\sqrt{2}$  の絶対値の二乗で $1/2$ となります。

## Aliceが $|0\rangle$ を観測した場合

Aliceが、EPRペア  $1/\sqrt{2} (|00\rangle + |11\rangle)$  を観測して、その結果が $|0\rangle$ であったとしましょう。

$$1/\sqrt{2} (|00\rangle + \cancel{|11\rangle})$$

この観測の結果、新しい状態は、 $|00\rangle$ に変わります。

それは、第二ビットの観測が、 $|0\rangle$ である確率が 1 であることを意味します。100% の確率で、Bobの持つ第二ビットの状態が0であることがわかるのです。

すなわち、Aliceが自分のqubitで状態 $|0\rangle$ を観測するとすぐに、遠く離れたBobの持つqubitの状態が $|0\rangle$ であることがわかることになります。

## Aliceが $|1\rangle$ を観測した場合

Aliceが、EPRペア  $1/\sqrt{2} (|00\rangle + |11\rangle)$  を観測して、その結果が $|1\rangle$ であったとしましょう。

$$1/\sqrt{2} (\cancel{|00\rangle} + |11\rangle)$$

この観測の結果、新しい状態は、 $|11\rangle$ に変わります。

それは、第二ビットの観測が、 $|1\rangle$ である確率が 1 であることを意味します。100% の確率で、Bobの持つ第二ビットの状態が $|1\rangle$ であることがわかるのです。

すなわち、Aliceが自分のqubitで状態 $|1\rangle$ を観測するとすぐに、遠く離れたBobの持つqubitの状態が $|1\rangle$ であることがわかることになります。

## 「馬鹿げた遠隔作用」？

Aliceの観測結果をまとめると、次のようになります。

- Aliceが自分のqubitで状態 $|0\rangle$ を観測すると、すぐに、遠く離れたBobの持つqubitの状態が $|0\rangle$ であることがわかる。
- Aliceが自分のqubitで状態 $|1\rangle$ を観測すると、すぐに、遠く離れたBobの持つqubitの状態が $|1\rangle$ であることがわかる。

Aliceの観測結果が、瞬時に、遠く離れたBobの観測結果に影響を与える？ これは、光のスピード以上で情報が伝わらないとする物理法則に矛盾しないか？

実際、アインシュタインは、こうした現象は「馬鹿げた遠隔作用」と言いました。



観測とObservableの  
固有値と固有ベクトル

# Observableの固有値と固有状態

量子の状態を表すベクトル $|\psi\rangle$ に作用する演算子 $O$ について、次の式を満たす実数 $r$ が存在する時、実数 $r$ を演算子 $O$ の固有値、状態 $|\psi\rangle$ を演算子 $O$ の固有状態と呼びます。

$$O|\psi\rangle = r|\psi\rangle$$

観測について言えば、観測によって実際に得られる観測値は、Observable  $O$  の固有値  $r$  で、観測後の状態は、固有状態 $|\psi\rangle$ に変化します。

観測によって引き起こされる、もとの空間からその部分空間である固有状態への変化を、「射影projection」と言います。

演算子Zの固有値と固有状態を考えてみましょう。

$$Z|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = +1|0\rangle$$
$$Z|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -\begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1|1\rangle$$

Observable Zは、二つの固有値 +1と-1を持ち、その固有状態は、 $|0\rangle$ と $|1\rangle$ であることがわかります。

我々が普段 qubitの状態に計算に使っている標準基底 $|0\rangle$ と $|1\rangle$ は、Zの固有状態にほかなりません。

演算子Xの固有値と固有状態を考えてみましょう。

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \text{ とする}$$

$$X|+\rangle = X\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle) = |+\rangle$$

$$X|-\rangle = X\left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -|-\rangle$$

$$X|+\rangle = +1|+\rangle$$

$$X|-\rangle = -1|-\rangle$$

Observable Xは、二つの固有値 +1と-1を持ち、その固有状態は、 $|+\rangle$ と $|-\rangle$ であることがわかります。

Hadamard基底 $|+\rangle$ と $|-\rangle$ は、Xの固有状態にほかなりません。





## Part 2

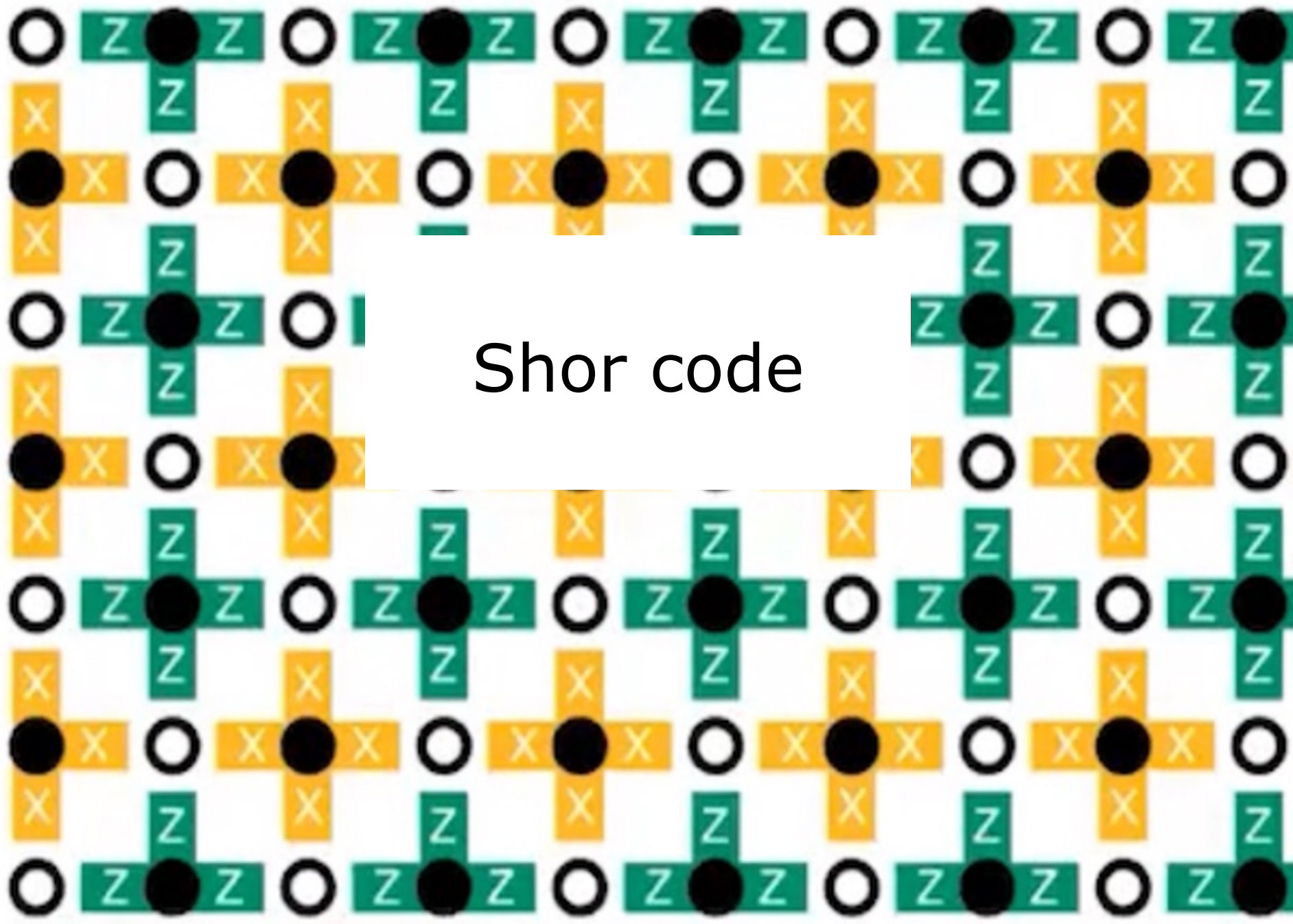
# Shor が考えたこと



## Part 2

### Shor が考えたこと

1. Shor code
2. 観測と射影演算子
3. 射影演算子でのbit flipの検出



Shor code

# Shor code

このPart 2では、現代のstabilizer codeの先駆としてのShor codeを紹介します。

Shorは、量子コンピュータでは、古典コンピュータと比較して、素因数分解が指数関数的に高速化でき、多項式時間で計算可能であることを示した「Shorのアルゴリズム」のShorです。

Shorは、量子エラー訂正技術においても、重要な貢献をしています。

# Shorの9-qubit code

Shorの量子エラー訂正コードは、9-qubit code と呼ばれることがあります。

それは、ひとつのqubitに起こりうる、bit反転、phase反転、あるいはその両方の組み合わせといった量子エラーを検出し訂正するためには、9つのqubitがあれば対応できるというものだったからです。

# 現代のStabilizer codeとShor code

現代のStabilizer code = surface code は、物理qubit の階層の上に、「論理qubit」の階層を構成することを最大の特徴としています。

この論理層の上に、あらためて「論理的な量子ゲート」、「論理的な測定」を定義・構成していきます。そこでもstabilizerが大きな役割を果たします。

そうした点では、Shor codeは、物理qubit上の量子エラー訂正技術に見えるかもしれませんが、しかし、重要なことは、そうした違いがありつつも、Shor codeは、stabilizer codeの本質を的確に捉えています。

# Shorが考えたこと 量子エラーのモデル

今日では自明なことと思われるかもしれませんが、Shorが初めて明確にしたことがあります。

量子エラーが発生するには、さまざまな原因があります。基本的には量子回路を構成するqubitの状態は脆弱なもので、外部環境との相互作用で容易にdecoherenceの状態になります。

Shorは、どんな量子エラーも、それが量子状態の変化である以上、最初の変化はあるユニタリ演算子の作用によって引き起こされるはずだという量子エラーのモデルを提案します。

bitの反転なら、 $a|0\rangle + b|1\rangle$ の状態を $b|0\rangle + a|1\rangle$ に変えるユニタリ演算子 $X$ が作用しているはずでず。

phaseの反転なら、 $a|0\rangle + b|1\rangle$ の状態を $a|0\rangle - b|1\rangle$ の状態に変えるユニタリ演算子 $Z$ が作用しているはずでず。

このエラーモデルは、エラー訂正の方法も同時に示すことになります。

エラーの箇所とその種類が特定できたら、 $X^2 = Z^2 = I$  ですので、bitの反転なら演算子 $X$ を、phaseの反転なら演算子 $Z$ を適用すれば、反転は反転し元に戻ってエラーは訂正されます。

## Shorの卓見 -- Bell基底の採用

二つのqubit  $a, b$  からなる系の状態  $|\psi_{ab}\rangle$  は、一般に次の形で表すことができます。

$$|\psi_{ab}\rangle = A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle$$

この状態をObservable  $O$ で観測すると、観測は $O$ の固有値を観測値として返して、元の状態はさまざまな形で $O$ の固有状態に射映され変化します。

ShorのShor codeのアイデアで素晴らしいのは、観測によって射映され変化して空間の基底に、思いがけないものを提案したことです。

Shorは、射映された固有状態の基底に、エンタングルした二つの qubitの状態を選びます。それは、以前にEPRペアとして紹介した次の四つです。

このERPペアの状態を **Bell State** とも言います。

- $\Phi^+$  :  $1/\sqrt{2} (|00\rangle + |11\rangle)$
- $\Phi^-$  :  $1/\sqrt{2} (|00\rangle - |11\rangle)$
- $\Psi^+$  :  $1/\sqrt{2} (|01\rangle + |10\rangle)$
- $\Psi^-$  :  $1/\sqrt{2} (|01\rangle - |10\rangle)$

この時、興味深いことが起きます。二つのqubit  $a, b$  を Observable  $X$ で観測する $X_a X_b$ と、同じ $a, b$ をObservable  $Z$ で観測する $Z_a Z_b$  が、同じ固有状態を持つのです。

## エンタングルメントを利用して、 エンタングルメントをコントロールする

一つのqubit を対象とした観測では、 $X$ による観測と $Z$ による観測は、違った結果をもたらします。また、 $XZ \neq ZX$ ですから、観測の順番で結果も変わります。そういうことが、 $X_a X_b$ と $Z_a Z_b$ の間では起きないのです。

Preskillは、こうしたShor codeの働きを、「エンタングルメントを利用して、エンタングルメントをコントロールする」と評しました。

こうした枠組みは、現代のstabilizer code にも、そのまま受け継がれています。

# Shor codeと古典的なエラー訂正技術

こうしたことは、Shor code での観測が、観測によって観測の対象の状態が変わることがない古典的な観測と共通の性質を持つことを意味しています。

実際、Shor code は、古典的なエラー対策である情報の冗長化の手法の量子版として解釈できます。

ビット '0' を、'1' を送るのに同じ情報を繰り返して '000', '111' を送るように、Shor code では、次のような冗長化が行われます。

$$|0\rangle \rightarrow |000\rangle$$

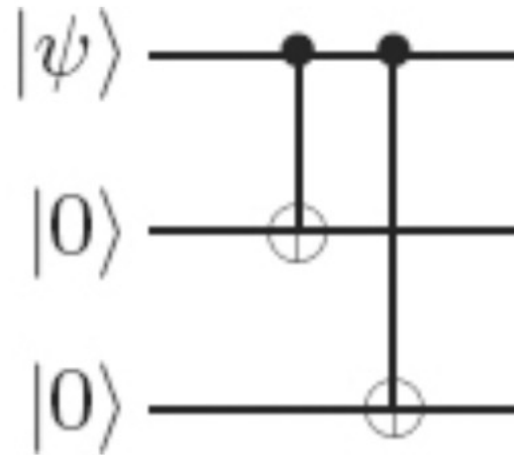
$$|1\rangle \rightarrow |111\rangle$$

# 冗長化の量子回路

次の回路が、冗長化を可能にすることを確かめてください。

$$|0\rangle \rightarrow |000\rangle$$

$$|1\rangle \rightarrow |111\rangle$$



# 観測と射影演算子



# 観測を表現する演算子

qubitの状態  $|\psi\rangle$  は、ユニタリ演算子  $U$  の作用で新しい状態  $U|\psi\rangle$  に変化します。

$$|\psi\rangle \rightarrow U|\psi\rangle$$

ただし、それはqubitのシステムが外部からの影響を受けない純粋な状態の場合だけです。

観測は、外部からqubitのシステムを見ることです。qubitのシステムは観測という外部からの干渉で、新しい状態  $P|\psi\rangle$  に変化します。

$$|\psi\rangle \rightarrow P|\psi\rangle$$

基本的には、新しい状態は、元の状態  $|\psi\rangle$  が属していた空間の部分空間への「射影」なのですが、そうした観測を表現する演算子について、このセッションでは考えます。

# 射影演算子

n次元のベクトル空間の基底  $|i\rangle$  に対して外積  $|i\rangle\langle i|$  で定義される演算子を射影演算子と言います。

一つのqubitの状態  $Q$  は、二つの基底  $|0\rangle, |1\rangle$  で張られる二次元のベクトルとして、次のように表現されます。

$$|Q\rangle = a|0\rangle + \beta|1\rangle$$

二次元のベクトル空間には、二つの射影演算子が存在します。

$$P_0 = |0\rangle\langle 0|$$

$$P_1 = |1\rangle\langle 1|$$

## 一つのqubitの状態 $|Q\rangle$ に対する 射影演算子 $P_0, P_1$ の働き

$|Q\rangle = a|0\rangle + \beta|1\rangle$  に対して  
 $P_0|Q\rangle, P_1|Q\rangle$  を計算してみよう。

$$\begin{aligned} P_0|Q\rangle &= (|0\rangle\langle 0|) (a|0\rangle + \beta|1\rangle) \\ &= |0\rangle (a\langle 0|0\rangle + \beta\langle 0|1\rangle) = |0\rangle (a \times 1 + \beta \times 0) \\ &= a|0\rangle = |0\rangle \quad (|\alpha|^2 + |\beta|^2 = 1 \text{ から } \alpha = 1) \end{aligned}$$

$$\begin{aligned} P_1|Q\rangle &= (|1\rangle\langle 1|) (a|0\rangle + \beta|1\rangle) \\ &= \beta|1\rangle = |1\rangle \quad (|\alpha|^2 + |\beta|^2 = 1 \text{ から } \beta = 1) \end{aligned}$$

## 射影演算子は観測演算子

$$P_0|Q\rangle = (|0\rangle\langle 0|) (a|0\rangle + \beta|1\rangle) = |0\rangle$$

$$P_1|Q\rangle = (|1\rangle\langle 1|) (a|0\rangle + \beta|1\rangle) = |1\rangle$$

$P_0$  を  $a|0\rangle + \beta|1\rangle$  に適用すると、 $\beta|1\rangle$  の項は失われ、 $|0\rangle$  が残り

$P_1$  を  $a|0\rangle + \beta|1\rangle$  に適用すると、 $a|0\rangle$  の項は失われ、 $|1\rangle$  が残ります。

「射影」というのは、そういう意味です。

射影演算子 $P_0, P_1$ によって、重ね合わせの状態は失われ、一つの状態のみが観測後には残ることが表現されています。

射影演算子を、観測演算子なのです。

先に触れたように、 $P_0|Q\rangle = \alpha|0\rangle$  ,  $P_1|Q\rangle = \beta|1\rangle$  ですが、 $|\alpha|^2 + |\beta|^2 = 1$  からそれぞれ  $\alpha = 1$  ,  $\beta = 1$  であることが分かります。観測後に残る状態は、 $|0\rangle$  ,  $|1\rangle$  となります。

ですので、観測後の状態は、 $P_0, P_1$ を用いて、

$$P_0|Q\rangle$$

$$P_1|Q\rangle$$

と表せることとなります。

ただ、繰り返しになりますが、観測前の  $\alpha$ 、 $\beta$  の値は、観測では得られるわけではないことに注意してください。

## 射影演算子で観測の確率を表す

qubitの元の状態の、基底の成分 $\alpha, \beta$ は「確率振幅」で、Bornのルールで、その絶対値の二乗がその基底が観測される確率を表しています。

この $\alpha, \beta$ を、次のようにすると、射影演算子 $P_0, P_1$ を使って表現できます。

$P_0 = |0\rangle\langle 0|$ ,  $P_1 = |1\rangle\langle 1|$  として、qubit  $|Q\rangle = \alpha|0\rangle + \beta|1\rangle$  に対して  $\langle Q|P_0|Q\rangle$  を計算してみましょう。

$$\begin{aligned}\langle Q|P_0|Q\rangle &= (\langle 0|a^* + \langle 1|\beta^*)(|0\rangle\langle 0|)(a|0\rangle + \beta|1\rangle) \\ &= (\langle 0|a^* + \langle 1|\beta^*)(a|0\rangle) = a^*a = |a|^2\end{aligned}$$

同様に、

$$\langle Q|P_1|Q\rangle = \beta^*\beta = |\beta|^2 \text{ となります。}$$

状態 $|Q\rangle$  で

$$|0\rangle \text{ が観測される確率 } p_0 = \langle Q|P_0|Q\rangle = \langle Q|0\rangle\langle 0|Q\rangle$$

$$|1\rangle \text{ が観測される確率 } P_1 = \langle Q|P_1|Q\rangle = \langle Q|1\rangle\langle 1|Q\rangle$$

これは、次のような内積による、  
観測確率の表現とも一致する。

$|A\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$  を観測した時、

0が観測される確率

$$p(0) = \alpha_0^* \alpha_0 = \underbrace{\langle A|0\rangle}_{\alpha_0^*} \underbrace{\langle 0|A\rangle}_{\alpha_0}$$

1が観測される確率

$$p(1) = \alpha_1^* \alpha_1 = \underbrace{\langle A|1\rangle}_{\alpha_1^*} \underbrace{\langle 1|A\rangle}_{\alpha_1}$$

$$p(0) = \langle A|0\rangle \langle 0|A\rangle$$

$$p(0) = \boxed{\langle A|0\rangle} \boxed{\langle 0|A\rangle}$$

内積による表現

$$p(0) = \langle A| \boxed{0} \langle A|$$

射影演算子による表現

$|\psi\rangle$  を観測して、状態  $|k\rangle$  が観測される確率

$$\langle \psi | P_k | \psi \rangle$$

$$|\psi\rangle = \sum_i c_i |i\rangle, \quad \langle \psi| = \sum_j \langle j| c_j^*$$

$$\begin{aligned} & \langle \psi | P_k | \psi \rangle \\ &= \langle \psi | k \rangle \langle k | \psi \rangle \\ &= \sum_j \langle j | c_j^* | k \rangle \langle k | \sum_i c_i | i \rangle \\ &= \sum_{i,j} c_j^* c_i \langle j | k \rangle \langle k | i \rangle \\ &= c_k^* c_k = |c_k|^2 \end{aligned}$$

## 観測値の期待値

ある量子系のシステムの状態を  $|A\rangle$  とする。

系の観測可能量は、あるエルミートな演算子  $L$  で表現されるのだが、観測値は、この  $L$  の固有値  $\lambda_i$  である。

また、 $L$  の固有ベクトル  $|\lambda_i\rangle$  は、この系の基底である。だから、 $|A\rangle$  は、次のように表現できる。

$$|A\rangle = \sum_i \alpha_i |\lambda_i\rangle$$

観測値  $\lambda_i$  が観測される確率  $P(\lambda_i) = \alpha_i^* \alpha_i$  である。

# 観測値の期待値

ある状態  $|A\rangle$  をとる系の、Observable  $L$  の観測値の期待値は、 $L$  を  $A$  でサンドイッチした  $\langle L \rangle = \langle A|L|A \rangle$  で求まる。

一般に、値  $\lambda_i$  が確率  $P(\lambda_i)$  で観測される時、その観測  $L$  の期待値を  $\langle L \rangle$  で表せば、 $\langle L \rangle$  は、次の式で与えられる。

$$\langle L \rangle = \sum_i \lambda_i P(\lambda_i)$$

# 射影演算子 $P_i$ の基本的性質

$|\psi\rangle$  を観測して、状態  $|i\rangle$  が観測される確率  $p(i)$

$$p(i) = \langle \psi | P_i | \psi \rangle$$

$$\sum_i P_i = 1$$

例えば、

$$|0\rangle\langle 0| + |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

$$P_0 + P_1 \equiv 1$$

$$P_i^2 = P_i$$

$$\because P_i^2 = |i\rangle\langle i||i\rangle\langle i| = |i\rangle\langle i| = P_i$$

$$P_i^\dagger = P_i \quad P_i \text{ はエルミートである}$$

$$P_i^\dagger = (|i\rangle\langle i|)^\dagger = (\langle i|)^\dagger (|i\rangle)^\dagger = |i\rangle\langle i| = P_i$$

# 射影演算子による観測

射影演算子による観測は、Observable  $M$ によって記述される。

Observable  $M$ は、観測される状態の空間に作用するエルミート演算子で、次の形で表現される。ここで、 $P_m$ は固有値 $m$ を持つ $M$ の固有状態への射影演算子である。

$$M = \sum m P_m$$

$m$ は、Observable  $M$ の固有値で、観測によって得られる可能な値は、 $m$ に対応している。

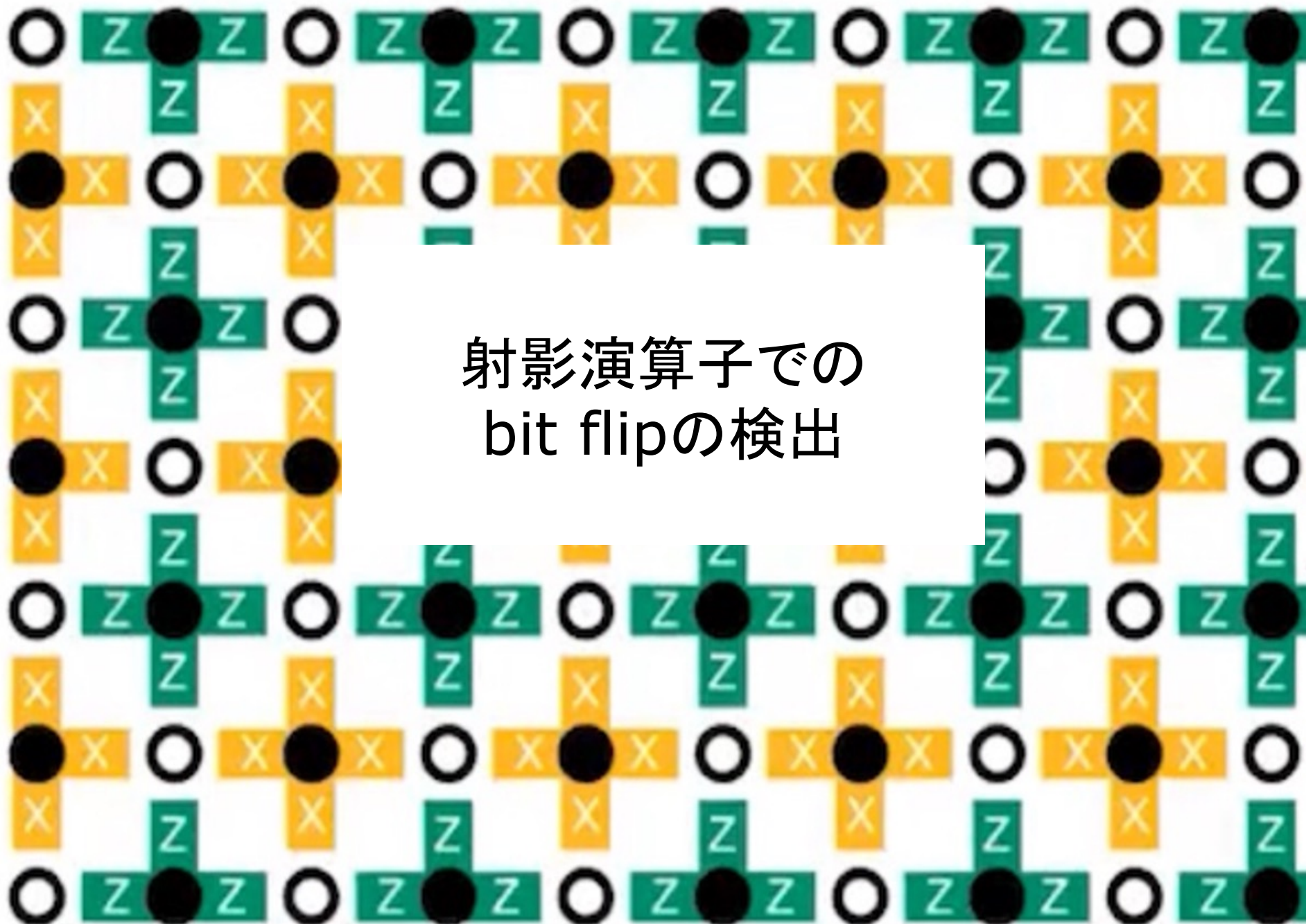
状態 $|\psi\rangle$ の観測で、値 $m$ を得られる確率  $p(m)$ は、次の式で与えられる。

$$p(m) = \langle \psi | P_m | \psi \rangle$$

値 $m$ が観測されたとき、システムの状態は、ただちに次の状態に変化する。

$$\frac{P_m |\psi\rangle}{\sqrt{p(m)}}$$

射影演算子での  
bit flipの検出



# Syndrome Measurement

## エラーの検出

このセッションでは、Shor codeでは bit flipエラーの検出をどのように行なっているかを紹介しようと思います。

エラーの検出はエラーの観測に他なりません。ですので、エラーの検出には、それに対応した観測演算子＝射影演算子が存在するはずです。

エラーを検出する観測を、**Syndrome Measurement** と言います。

まずは、その観測演算子を見つけましょう。

# 何がエラーなのかを考える

量子エラーには、さまざまな形があります。そのエラー全部をカバーする観測演算子を考えるのは難しそうです。

ここでは、対象を三つのqubitからなるシステムで起こりうるエラーに限って、かつ、エラーの種類をビットが反転するエラーに絞って、それを観測する観測演算子を考えることにしましょう。

三つのqubitからなるシステムは、次の8つの基底を持っています。

$|000\rangle, |001\rangle, |010\rangle, |011\rangle$   
 $|100\rangle, |101\rangle, |110\rangle, |111\rangle$

これらの3-qubitのシステムの基底は、

$$|000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle$$

$$|111\rangle = |1\rangle \otimes |1\rangle \otimes |1\rangle$$

...

のように、1-qubitの基底  $|0\rangle$  と  $|1\rangle$  のテンソル積で定義されています。

エラーを検出するためにはその前提として、三つのqubitのシステムが取りうる、この8つの基底から構成される多数の状態の中から、どれかを「エラー状態」とみなすべきか、あらかじめ決めておく必要があります。

ある状態は、あるコンテキスト、あるシナリオのもとで、はじめて「エラー状態」とみなされます。

# エラーの通信チャンネル・モデル

## 基本的シナリオ

エラー検出の背景となっているシナリオは、次のようなものです。

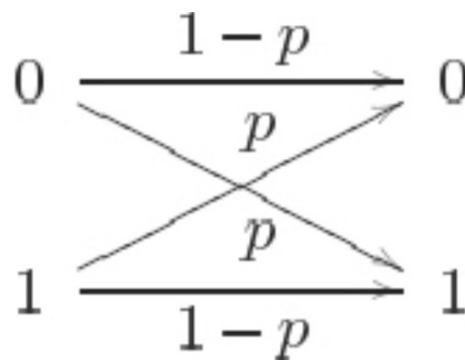
基本的には、一つのqubitの状態を、ノイズのある通信路を通じて送信して、それを受信したらどうなるかというシナリオを考えます。

通信路にノイズがあると、ビット 0 がビット 1 に変わったり、ビット 1 がビット 0 に変わったりする変化が起こり得ます。

これは、通信というコンテキストから考えれば、ビット反転のエラーとみなされます。

ビット 0が ビット 1に反転するエラー確率と、ビット 1が ビット 0に反転するエラー確率が等しく  $p$ だとします。

この時、エラーが起こらず、ビット 0が ビット 0に、ビット 1が ビット 1に、正しく送られる確率は、 $1 - p$ になります。



こうした通信路を、「二値対称チャンネル」と呼びます。

# Shor codeのシナリオ

今回紹介するShor codeが想定しているシナリオの基本的な目標は、送信側の一つのqubitの情報  $a|0\rangle + b|1\rangle$  を、できる限り正確に受信側に送ることです。ただ、その手順は少し複雑な次のようなステップを踏みます。

1. 送信者は、qubitの情報を冗長化してencodeする。
2. 送信者は、encodeした情報をチャンネルに送る。
3. チャンネルを通過し、情報はノイズの影響を受ける。
4. 受信者は、チャンネルから情報を受け取る。
5. 受信者は、情報からノイズで生まれたエラーを検出する。
6. 受信者は、検出されたエラーを訂正する。
7. 受信者は、訂正された情報をdecodeする、

Shor codeの想定しているステップを、少し詳しく見てみましょう。

Step 1 ですが、Shor codeは次のような冗長化を行います。

$$\begin{aligned} |0\rangle &\rightarrow |000\rangle \\ |1\rangle &\rightarrow |111\rangle \end{aligned}$$

Step2 では、Step 1 のencodeの下で、1-qubitの情報  $a|0\rangle + b|1\rangle$  は、3-qubitの情報である  $a|000\rangle + b|111\rangle$  という形で、チャンネルに送り出されます。

$$a|0\rangle + b|1\rangle \rightarrow a|000\rangle + b|111\rangle$$

Shor codeの冗長化とは、1 qubitの情報を3 qubitの情報に冗長化することに他なりません。

Step 5. 6 は、エラーの検出・訂正というエラー処理の中心部分ですので、その働きは後でさらに見ることにします。

Step 6 でエラー訂正された 3-qubitの状態は、Step 2でチャンネルに送信された  $a|000\rangle + b|111\rangle$  と同じ状態になっているはずで

Step 7 では、この3-qubitの状態  $a|000\rangle + b|111\rangle$  は、1-qubitの状態である  $a|0\rangle + b|1\rangle$  にdecodeされます。

$$a|000\rangle + b|111\rangle \rightarrow a|0\rangle + b|1\rangle$$

$$a|0\rangle + b|1\rangle$$



encode  $|0\rangle \rightarrow |000\rangle$  ,  $|1\rangle \rightarrow |111\rangle$

$$a|000\rangle + b|111\rangle$$



ノイズによるエラー発生



エラー検出・訂正



$$a|000\rangle + b|111\rangle$$



decode  $|000\rangle \rightarrow |0\rangle$  ,  $|111\rangle \rightarrow |1\rangle$

$$a|0\rangle + b|1\rangle$$

# Shor codeのシナリオ

$$a|0\rangle + b|1\rangle$$

↓ Step 1

encode  $|0\rangle \rightarrow |000\rangle$  ,  $|1\rangle \rightarrow |111\rangle$

$$a|000\rangle + b|111\rangle$$

↓ Step 2 Step 3

ノイズによるエラー発生

↓ Step 4 Step 5

エラー検出・訂正

↓ Step 6

$$a|000\rangle + b|111\rangle$$

↓ Step 7

decode  $|000\rangle \rightarrow |0\rangle$  ,  $|111\rangle \rightarrow |1\rangle$

$$a|0\rangle + b|1\rangle$$

## 3-qubit のシステムのbit-flipエラーを考える

チャンネルに送り出された 3-qubitの情報が、Step 3でどのようにノイズから影響を受けるかを考えてみましょう。

ここでもノイズによって引き起こされるエラーを全て考えるのは難しいので、3-qubit のシステムの中で一つqubitに起こるbit-flipのみを、まず、考えることにします。

1-qubit システムの bit flip は、次のようなものです。

$$a|0\rangle + b|1\rangle \rightarrow a|1\rangle + b|0\rangle$$

ここでは、 $|0\rangle \rightarrow |1\rangle$  と  $|1\rangle \rightarrow |0\rangle$  という二つの変化が同時に起きています。

状態  $a|000\rangle + b|111\rangle$  で、一つのbit flip が起きたとすれば、そのbit flipは、三つのqubitの中の一つで起きたはずで、qubitに名前をつけて、それぞれ  $q1, q2, q3$  としましょう。

何もflipが起きないとすると、状態の変化は、次のように表せます。

$$a|000\rangle + b|111\rangle \rightarrow a|000\rangle + b|111\rangle$$

$q1$ にbit flip  $|0\rangle \rightarrow |1\rangle$  かつ  $|1\rangle \rightarrow |0\rangle$  が起きると、

$$a|000\rangle + b|111\rangle \rightarrow a|100\rangle + b|011\rangle$$

$q2$ にbit flip  $|0\rangle \rightarrow |1\rangle$  かつ  $|1\rangle \rightarrow |0\rangle$  が起きると、

$$a|000\rangle + b|111\rangle \rightarrow a|010\rangle + b|101\rangle$$

$q3$ にbit flip  $|0\rangle \rightarrow |1\rangle$  かつ  $|1\rangle \rightarrow |0\rangle$  が起きると、

$$a|000\rangle + b|111\rangle \rightarrow a|001\rangle + b|110\rangle$$

## このコンテキストの下で、 3-qubitの一つのbit flipを検出する

このコンテキストの下ではというのは、

$$a|000\rangle + b|111\rangle \rightarrow a|000\rangle + b|111\rangle$$

をエラーのない遷移として、それ以外の遷移の発生をエラーとみなすということです。

この時、3-qubit のシステムに起きる、一つのbit flip エラーを検知する観測演算子は、次の四つ射影演算子として構成できます。

# 3-qubitの一つのbit flipエラーを検出する 4つの観測演算子

$$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$$

$$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$$

$$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$$

$$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$$

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$  の性質

$$\begin{aligned} P_0|000\rangle &= (|000\rangle\langle 000| + |111\rangle\langle 111|) |000\rangle \\ &= |000\rangle\langle 000|000\rangle + |111\rangle\langle 111|000\rangle = |000\rangle \end{aligned}$$

$$\begin{aligned} P_0|111\rangle &= (|000\rangle\langle 000| + |111\rangle\langle 111|) |111\rangle \\ &= |000\rangle\langle 000|111\rangle + |111\rangle\langle 111|111\rangle = |111\rangle \end{aligned}$$

$P_0$ の固有値は 1, 固有状態は  $|000\rangle$  と  $|111\rangle$

$$\langle 000|P_0|000\rangle = \langle 111|P_0|111\rangle = 1$$

$$P_0|(a|000\rangle + b|111\rangle) = a|000\rangle + b|111\rangle$$

$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$  の性質

$$\begin{aligned} P_1|100\rangle &= (|100\rangle\langle 100| + |011\rangle\langle 011|) |100\rangle \\ &= |100\rangle\langle 100|100\rangle + |011\rangle\langle 011|100\rangle = |100\rangle \end{aligned}$$

$$\begin{aligned} P_1|011\rangle &= (|100\rangle\langle 100| + |011\rangle\langle 011|) |011\rangle \\ &= |100\rangle\langle 100|011\rangle + |011\rangle\langle 011|011\rangle = |011\rangle \end{aligned}$$

$P_1$ の固有値は 1, 固有状態は  $|100\rangle$  と  $|011\rangle$

$$\langle 100|P_1|100\rangle = \langle 011|P_1|011\rangle = 1$$

$$P_1|(a|100\rangle + b|011\rangle) = a|100\rangle + b|011\rangle$$

$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$  の性質

$$\begin{aligned} P_2|010\rangle &= (|010\rangle\langle 010| + |101\rangle\langle 101|) |010\rangle \\ &= |010\rangle\langle 010|010\rangle + |101\rangle\langle 101|010\rangle = |010\rangle \end{aligned}$$

$$\begin{aligned} P_2|101\rangle &= (|010\rangle\langle 010| + |101\rangle\langle 101|) |101\rangle \\ &= |010\rangle\langle 010|101\rangle + |101\rangle\langle 101|101\rangle = |101\rangle \end{aligned}$$

$P_2$ の固有値は 1, 固有状態は  $|010\rangle$  と  $|101\rangle$

$$\langle 010|P_2|010\rangle = \langle 101|P_2|101\rangle = 1$$

$$P_2|(a|010\rangle + b|101\rangle) = a|010\rangle + b|101\rangle$$

$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$  の性質

$$\begin{aligned} P_3|001\rangle &= (|001\rangle\langle 001| + |110\rangle\langle 110|) |001\rangle \\ &= |001\rangle\langle 001|001\rangle + |110\rangle\langle 110|001\rangle = |001\rangle \end{aligned}$$

$$\begin{aligned} P_3|110\rangle &= (|001\rangle\langle 001| + |110\rangle\langle 110|) |110\rangle \\ &= |001\rangle\langle 001|110\rangle + |110\rangle\langle 110|110\rangle = |110\rangle \end{aligned}$$

$P_3$ の固有値は 1, 固有状態は  $|001\rangle$  と  $|110\rangle$

$$\langle 001|P_3|001\rangle = \langle 110|P_3|110\rangle = 1$$

$$P_3|(a|001\rangle + b|110\rangle) = a|001\rangle + b|110\rangle$$

$$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$$

この観測演算子は、何も変化がないことを検出します。

$$a|000\rangle + b|111\rangle \rightarrow a|000\rangle + b|111\rangle$$

$$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$$

この演算子は、一つ目のqubitにbit flipが起きたことを検出します。

$$a|000\rangle + b|111\rangle \rightarrow a|100\rangle + b|011\rangle$$

$$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$$

この演算子は、二つ目のqubitにbit flipが起きたことを検出します。

$$a|000\rangle + b|111\rangle \rightarrow a|010\rangle + b|101\rangle$$

$$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$$

この演算子は、三つ目のqubitにbit flipが起きたことを検出します。

$$a|000\rangle + b|111\rangle \rightarrow a|001\rangle + b|110\rangle$$





## Part 3

# Stabilizer -- もう一つの観測演算子



## Part 3

# Stabilizer -- もう一つの観測演算子

1. Stabilizerとは何か？
2. bit flipの検出
3. Phase flipの検出
4. まとめ



Stabilizerとは何か？

# Stabilizerとは何か？

このセッションから、Stabilizer を紹介する Part 3 に入ります。

今回は、Stabilizer とは何かを、手短かに概説しようと思います。

# Stabilizerは量子エラー訂正技術

stabilizerは、量子エラー訂正技術の一つです。

その技術は、現在もっとも注目すべき量子コンピュータ技術である surface code に全面的に採用されています。

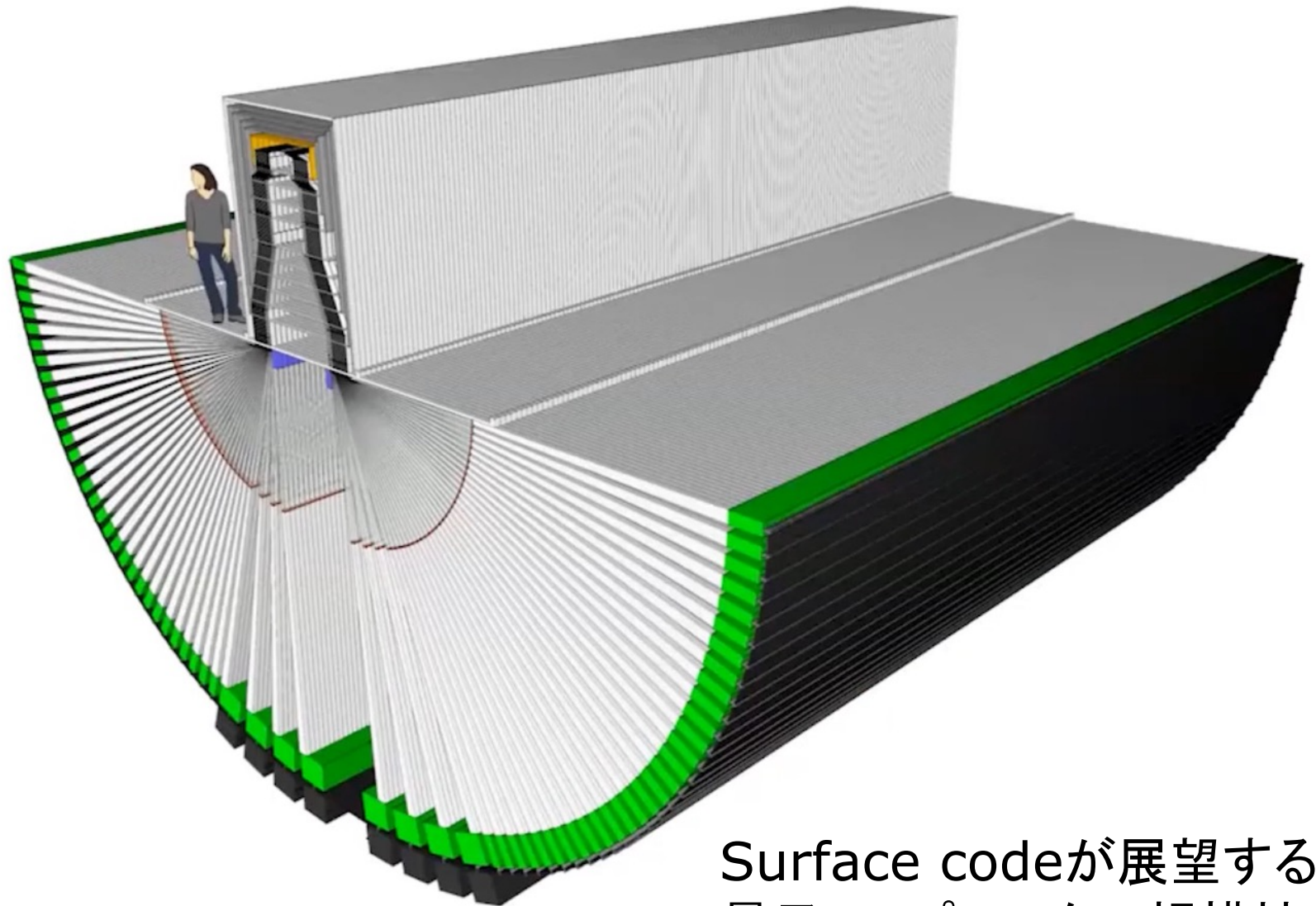
量子エラーに対する耐性を飛躍的に高めた surface code の理解には、その量子エラー訂正技術としての stabilizer の理解が不可欠です。

# Surface code と Stabilizer は、 量子コンピュータのスケール拡大を可能にする

量子エラー耐性の向上は、量子コンピュータのスケール拡大を可能にします。

現在 surface code は、主に 53 量子 bit の Sycamore 量子プロセッサに実装されているのですが、そのスケール拡大の展望は、興味深いものです。

1M qubits



Surface codeが展望する  
量子コンピュータの規模拡大

# Stabilizerは、「論理qubit」を可能にする

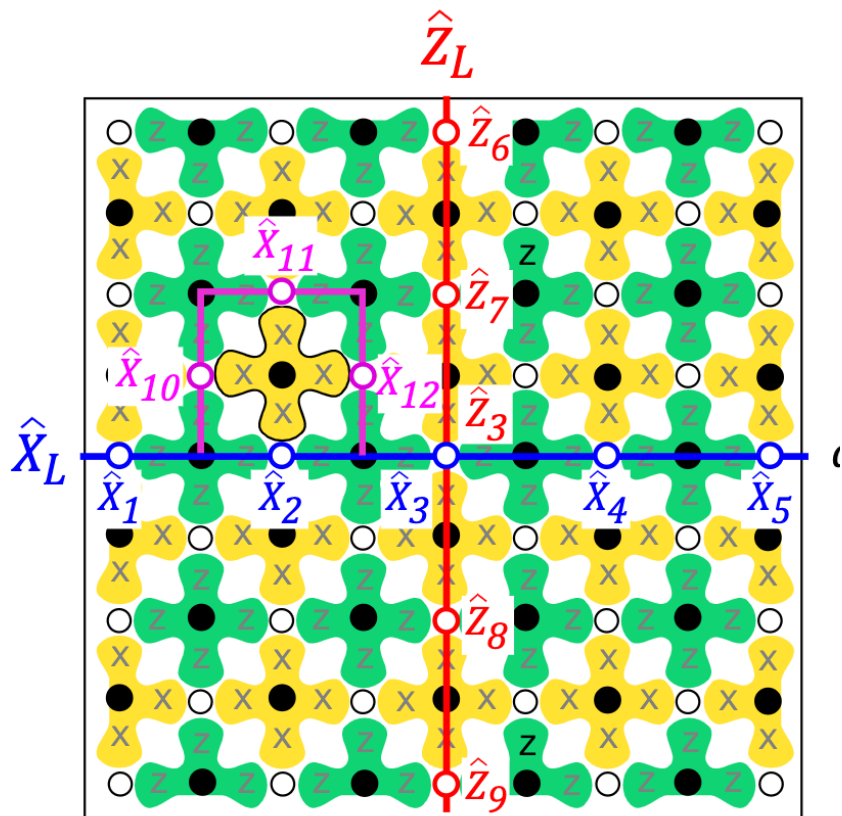
surface code のアーキテクチャーが革新的なのは、その量子エラー耐性の高さだけではありません。

surface code では、平面上に配置された多数の物理的なqubit から、一つの「論理qubit」を構成することが可能になります。「論理qubit」は、平面上を移動したり、大きさを拡大することが自由にできるようになります。

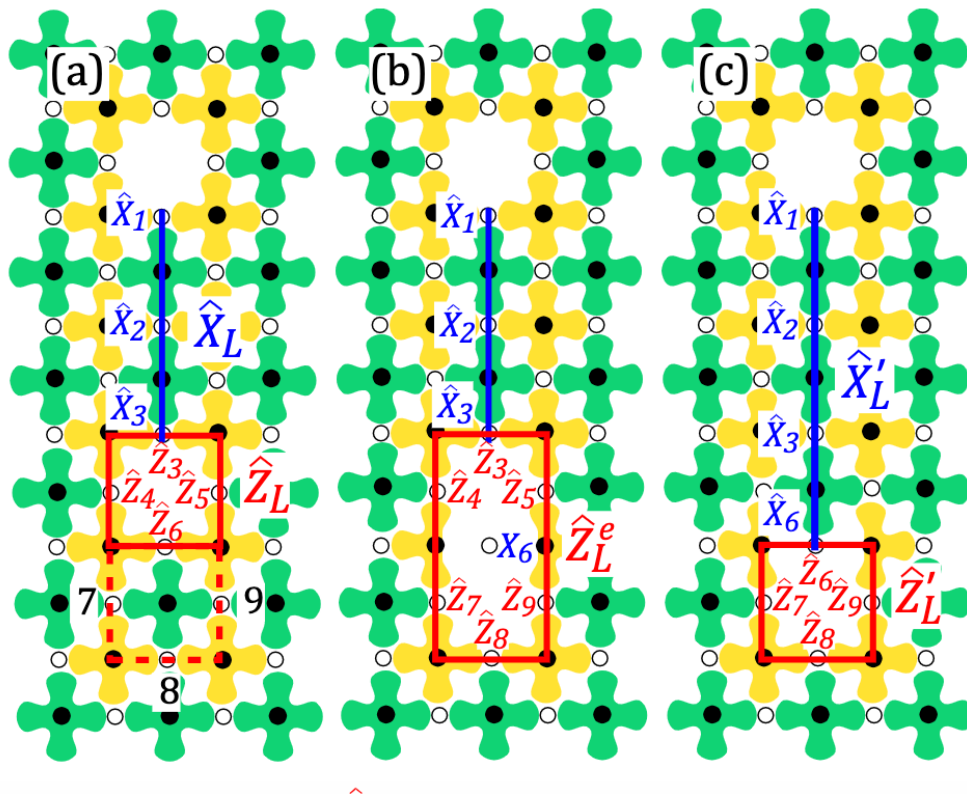
「論理qubit」の構成でも、Stabilizerが本質的な役割を果たします。

また、Stabilizerを利用した同様の手法で、物理的な量子ゲートに対応する「論理量子ゲート」を構成することができます。

41個の物理data qubit  
からなる論理qubit

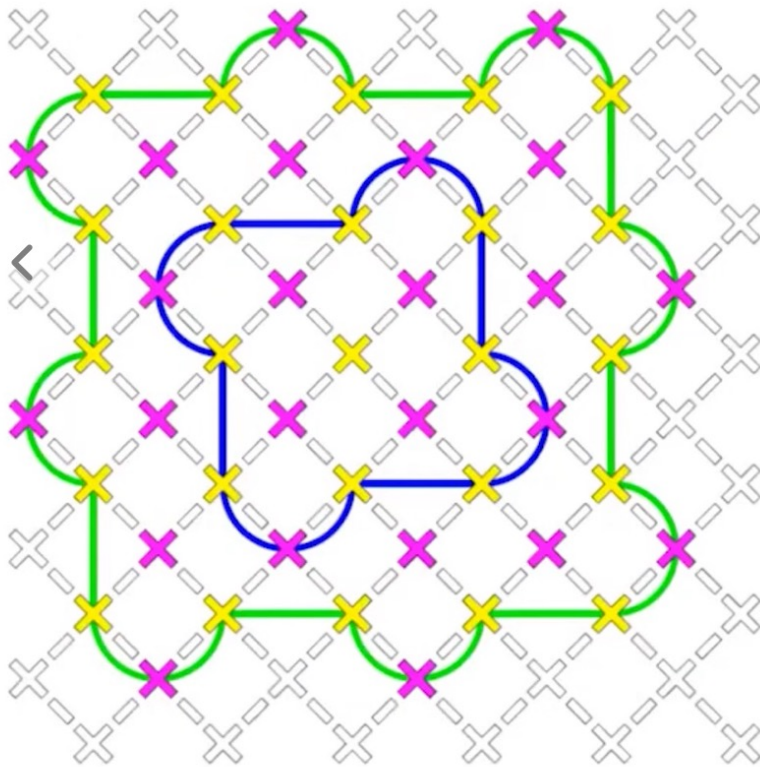
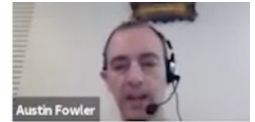


論理qubitの移動



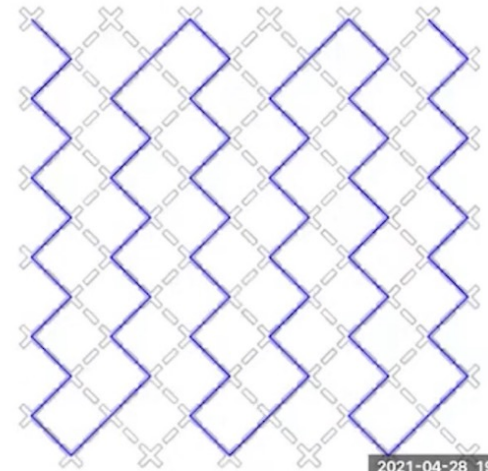
$\hat{X}_L, \hat{Z}_L$  は stabilizer

# Device and experiment



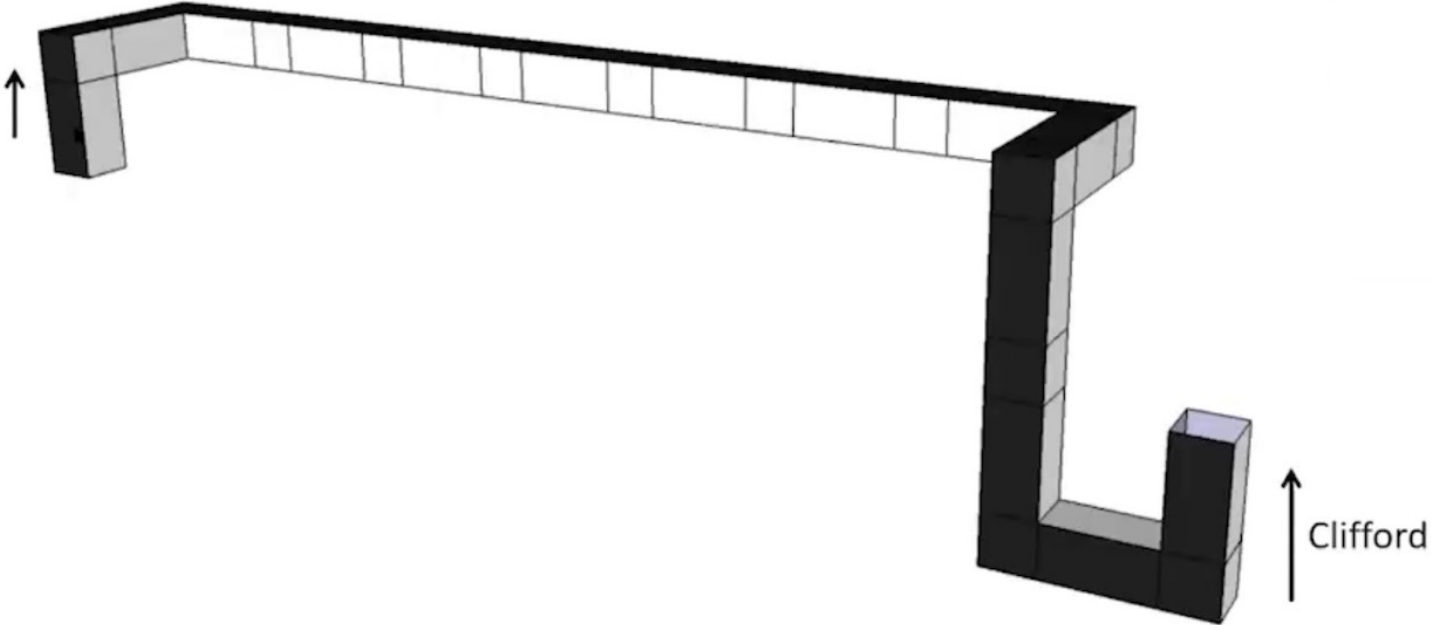
- Show  $d=5$  better than  $d=3$
- Continuous running
- Real-time decoding
- $d=34$  extreme exponential suppression ( $d=11$  arXiv:2102.06132)

- ⊗ Not used
- ⊗ Data qubit
- ⊗ Measure qubit



2021-04-26 19:10:42

# Logical move



Can move anywhere, even back in time, up to Pauli operators.

# stabilizer は、エラーを測定する

stabilizerを、量子エラー訂正技術として紹介してきましたが、一つ注意すべきことがあります。

それは、基本的には、エラーを訂正することではなく、エラーを測定することにフォーカスした回路です。測定回路自体にエラーが起きる場合以外は、エラーの訂正を行おうとはしません。

一つのstabilizer は、一つのqubit ではなく、  
複数のqubitの状態を同時に観測する

もう一つ、stabilizerによる測定には、大きな特徴があります。

それは、stabilizerは、一つのqubit ではなく、複数のqubitの状態を同時に測定することです。すぐ後で見える $Z_1Z_2$  や $X_1X_2$  は二つのqubit  $Q_1, Q_2$ を同時に観測します。

一つのqubitの観測でも、さまざまな問題(基本的なものは、観測すること自体によって、もとのqubitの状態が変化することです)があるのに、複数のqubitを同時に観測して大丈夫なののでしょうか？

stabilizerによる測定は、測定の対象のqubitの状態を変えないのです。それがstabilizerの最大の特徴であり、名前の由来にもなっています。



もう一つの観測演算子  
bit flipの検出

## もう一つの観測演算子 $Z_1Z_2$ と $Z_2Z_3$

Part 2では、3-qubit のシステムの bit flipエラーを検出する、4つの観測演算子  $P_0, P_1, P_3, P_4$  を紹介してきました。

このPart 3では、3-qubit のシステムの bit flipエラーを検出するという働きでは同じですが、ただし違うタイプのもう一つの観測演算子を紹介しようと思います。

それがObservable  $Z$ を用いて、次のように定義される  $Z_1Z_2$  と  $Z_2Z_3$  です。

$$\begin{aligned}Z_1Z_2 &= Z \otimes Z \otimes I \\Z_2Z_3 &= I \otimes Z \otimes Z\end{aligned}$$

## $Z_1Z_2$ と $Z_2Z_3$ という名前

$Z_1Z_2$  と  $Z_2Z_3$  の名前とその添え字は、この二つの演算子が3つの qubit の Observable  $Z$  による観測であることに関係しています。

この演算子が観測する三つの qubit を  $Q_1, Q_2, Q_3$  としましょう。  
三つの  $Q_1, Q_2, Q_3$  の状態  $|q_1q_2q_3\rangle = |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$  は Observable  $Z_1Z_2$  では、次のように観測されます。

$$\begin{aligned} & Z_1Z_2|q_1q_2q_3\rangle \\ &= (Z \otimes Z \otimes I)(|q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle) \\ &= Z|q_1\rangle \otimes Z|q_2\rangle \otimes I|q_3\rangle \end{aligned}$$

Iで観測するという事は、なんの変化も起こりませんので、観測しないことと同じことです。

$$\begin{aligned} & Z_1 Z_2 |q_1 q_2 q_3\rangle \\ &= (Z \otimes Z \otimes I) |q_1 q_2 q_3\rangle \\ &= (Z \otimes Z) |q_1 q_2\rangle \otimes |q_3\rangle \end{aligned}$$

となつて、 $Z_1 Z_2$ の観測に関わる部分は、Observable  $Z \otimes Z$ で、三つのQubitのうちの $Q_1, Q_2$ と状態 $|q_1 q_2\rangle$ を観測する部分になります。

$$\begin{aligned} & Z_2 Z_3 |q_1 q_2 q_3\rangle \\ &= (I \otimes Z \otimes Z) |q_1 q_2 q_3\rangle \\ &= |q_1\rangle \otimes (Z \otimes Z) |q_2 q_3\rangle \end{aligned}$$

こうして、次のような対応のもとで、

$$\begin{array}{ccccc} & Q_1 & Q_2 & Q_3 & \\ & \downarrow & \downarrow & \downarrow & \\ Z_1 Z_2 & = & Z \otimes Z & \otimes I & \\ Z_2 Z_3 & = & I \otimes Z & \otimes Z & \\ & \uparrow & \uparrow & \uparrow & \\ & Q_1 & Q_2 & Q_3 & \end{array}$$

$Z_1 Z_2 = Z \otimes Z \otimes I$  は、qubit  $Q_1$  と  $Q_2$  の状態を観測し、  
 $Z_2 Z_3 = I \otimes Z \otimes Z$  は、qubit  $Q_2$  と  $Q_3$  の状態を観測することがわかります。

## 2-qubitのObservable $Z \otimes Z$ の行列を計算する

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ から、}$$

$$Z \otimes Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} \boxed{1} & \boxed{0} & 0 & 0 \\ \boxed{0} & \boxed{-1} & 0 & 0 \\ 0 & 0 & \boxed{-1} & \boxed{0} \\ 0 & 0 & \boxed{0} & \boxed{1} \end{pmatrix}$$

2-qubitのObservable  $Z \otimes Z$  の行列から、  
その外積での表示を求める

$$Z \otimes Z = \begin{matrix} & \begin{matrix} 00 & 01 & 10 & 11 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \end{matrix}$$

$$Z \otimes Z =$$

$$\begin{aligned} & |00\rangle\langle 00| - |01\rangle\langle 01| - |10\rangle\langle 10| + |11\rangle\langle 11| = \\ & |00\rangle\langle 00| + |11\rangle\langle 11| - (|01\rangle\langle 01| + |10\rangle\langle 10|) \end{aligned}$$

## $Z \otimes Z$ での2-qubitの基底の観測

$$Z \otimes Z = (|00\rangle\langle 00| + |11\rangle\langle 11|) \\ - (|01\rangle\langle 01| + |10\rangle\langle 10|)$$

$Z \otimes Z$ で、2-qubitの基底  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ ,  $|11\rangle$ を観測する。

$$(Z \otimes Z)|\mathbf{00}\rangle \\ = |00\rangle\langle 00|\mathbf{00}\rangle + |11\rangle\langle 11|\mathbf{00}\rangle \\ - (|01\rangle\langle 01|\mathbf{00}\rangle + |10\rangle\langle 10|\mathbf{00}\rangle) \\ = |00\rangle$$

$$\begin{aligned}
& (Z \otimes Z)|\mathbf{01}\rangle \\
&= |00\rangle \langle 00|\mathbf{01}\rangle + |11\rangle \langle 11|\mathbf{01}\rangle \\
&\quad - (|01\rangle \langle 01|\mathbf{01}\rangle + |10\rangle \langle 10|\mathbf{01}\rangle) \\
&= -|01\rangle
\end{aligned}$$

$$\begin{aligned}
& (Z \otimes Z)|\mathbf{10}\rangle \\
&= |00\rangle \langle 00|\mathbf{10}\rangle + |11\rangle \langle 11|\mathbf{10}\rangle \\
&\quad - (|01\rangle \langle 01|\mathbf{10}\rangle + |10\rangle \langle 10|\mathbf{10}\rangle) \\
&= -|10\rangle
\end{aligned}$$

$$\begin{aligned}
& (Z \otimes Z)|\mathbf{11}\rangle \\
&= |00\rangle \langle 00|\mathbf{11}\rangle + |11\rangle \langle 11|\mathbf{11}\rangle \\
&\quad - (|01\rangle \langle 01|\mathbf{11}\rangle + |10\rangle \langle 10|\mathbf{11}\rangle) \\
&= |\mathbf{11}\rangle
\end{aligned}$$

## $Z \otimes Z$ の性質

$$(Z \otimes Z) |00\rangle = +1 |00\rangle$$

$$(Z \otimes Z) |01\rangle = -1 |01\rangle$$

$$(Z \otimes Z) |10\rangle = -1 |10\rangle$$

$$(Z \otimes Z) |11\rangle = +1 |11\rangle$$

$Z \otimes Z$ の固有値  $\pm 1$

固有値  $+1$  の時 固有状態  $|00\rangle, |11\rangle$

$$|q_i\rangle = |q_j\rangle \quad (i \neq j)$$

固有値  $-1$  の時 固有状態  $|01\rangle, |10\rangle$

$$|q_i\rangle \neq |q_j\rangle \quad (i \neq j)$$

## $Z_1Z_2$ と $Z_2Z_3$ の働きを再確認

$$\begin{array}{ccccc} & Q_1 & Q_2 & Q_3 & \\ & \downarrow & \downarrow & \downarrow & \\ Z_1Z_2 & = & Z \otimes Z & \otimes I & \\ Z_2Z_3 & = & I \otimes Z & \otimes Z & \\ & \uparrow & \uparrow & \uparrow & \\ & Q_1 & Q_2 & Q_3 & \end{array}$$

$Z_1Z_2 = Z \otimes Z \otimes I$  は、qubit  $Q_1$  と  $Q_2$  の状態を観測し、  
 $Z_2Z_3 = I \otimes Z \otimes Z$  は、qubit  $Q_2$  と  $Q_3$  の状態を観測することがわかります。

$Z_1 Z_2$  で観測 (q1とq2を観測)

観測値 +1       $q1 = q2$

観測値 -1       $q1 \neq q2$

$Z_2 Z_3$  で観測 (q2とq3を観測)

観測値 +1       $q2 = q3$

観測値 -1       $q2 \neq q3$

## $Z_1Z_2, Z_2Z_3$ を使って3qubit システムの bit flip エラーを検出する

$Z_1Z_2$	$Z_2Z_3$		
+1	+1	$q1 = q2 = q3$	No error
+1	-1	$q1 = q2 \neq q3$	q3 flipped
-1	+1	$q1 \neq q2 = q3$	q1 flipped
-1	-1	$q1 \neq q2 \neq q3$	q2 flipped

前回見た、4つの観測演算子  $P_0, P_1, P_3, P_4$  と同じ機能を、  
二つの観測演算子  $Z_1Z_2, Z_2Z_3$ を使って実行できる。

## $Z_1Z_2, Z_2Z_3$ の他の性質

$$\begin{aligned} Z_1Z_2 & |(A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle) \\ &= A|00\rangle - B|01\rangle - C|10\rangle + D|11\rangle \end{aligned}$$

$$\begin{aligned} Z_2Z_3 & |(A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle) \\ &= A|00\rangle - B|01\rangle - C|10\rangle + D|11\rangle \end{aligned}$$

$$\begin{aligned} Z_1Z_2 \circ Z_2Z_3 & (A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle) = \\ Z_2Z_3 \circ Z_1Z_2 & (A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle) = \\ & A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle \end{aligned}$$

## 同じ固有値を持つ固有ベクトルの線型結合

ある演算子Mの同じ固有値mを持つ固有ベクトルの線型結合を考えよう。ベクトル $|a\rangle$ とベクトル $|b\rangle$ が共に演算子Mの固有値mを持つ固有ベクトルだとしよう。

$$M|a\rangle = m|a\rangle$$

$$M|b\rangle = m|b\rangle$$

両辺を加えて

$$M(|a\rangle + |b\rangle) = m(|a\rangle + |b\rangle)$$

この式は、同じ固有値mを持つ固有ベクトルの線型結合 $(|a\rangle + |b\rangle)$ は、再び、元の演算子Mの固有値mの固有ベクトルであることを示す。

## $Z_1 Z_2$ でBell Stateを観測する

エンタングルした二つのqubitの状態を、「EPRペア」あるいは、「Bell State」といいます。この状態を $Z_1 Z_2$ で観測してみましょう。(正確には、 $Z_1 Z_2 = Z \otimes Z \otimes I \otimes I$ の $\otimes I$ の寄与を無視し、正規化因子も省略しています)

以前に見たように、それには、次の4種類があります。これを、 $\Phi^+$ 、 $\Phi^-$ 、 $\Psi^+$ 、 $\Psi^-$  と呼ぶことがあります。

- $|\Phi^+\rangle : \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$
- $|\Phi^-\rangle : \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle)$
- $|\Psi^+\rangle : \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle)$
- $|\Psi^-\rangle : \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle)$

$|\Phi^+\rangle$ :  $1/\sqrt{2} ( |00\rangle + |11\rangle )$  の観測

$$(Z \otimes Z) |\Phi^+\rangle$$

$$= (|00\rangle\langle 00| + |11\rangle\langle 11| - |01\rangle\langle 01| - |10\rangle\langle 10|) (|00\rangle + |11\rangle)$$

$$= |00\rangle\langle 00|00\rangle + |11\rangle\langle 11|00\rangle - |01\rangle\langle 01|00\rangle - |10\rangle\langle 10|00\rangle$$

$$+ |00\rangle\langle 00|11\rangle + |11\rangle\langle 11|11\rangle - |01\rangle\langle 01|11\rangle - |10\rangle\langle 10|11\rangle$$

$$= |00\rangle + |11\rangle = |\Phi^+\rangle$$

$|\Phi^-\rangle$  :  $1/\sqrt{2} ( |00\rangle - |11\rangle )$  の観測

$$(Z \otimes Z) |\Phi^-\rangle$$

$$= (|00\rangle\langle 00| + |11\rangle\langle 11| - |01\rangle\langle 01| - |10\rangle\langle 10|) (|00\rangle - |11\rangle)$$

$$= |00\rangle\langle 00|00\rangle + |11\rangle\langle 11|00\rangle - |01\rangle\langle 01|00\rangle - |10\rangle\langle 10|00\rangle$$

$$|00\rangle\langle 00|11\rangle - |11\rangle\langle 11|11\rangle + |01\rangle\langle 01|11\rangle + |10\rangle\langle 10|11\rangle$$

$$= |00\rangle - |11\rangle = |\Phi^-\rangle$$

$|\Psi^+\rangle : \frac{1}{\sqrt{2}} ( |01\rangle + |10\rangle )$  の観測

$$(Z \otimes Z)|\Psi^+\rangle$$

$$=(|00\rangle\langle 00| + |11\rangle\langle 11| - |01\rangle\langle 01| - |10\rangle\langle 10|) ( |01\rangle + |10\rangle )$$

$$= |00\rangle\langle 00|01\rangle + |11\rangle\langle 11|01\rangle - |01\rangle\langle 01|01\rangle - |10\rangle\langle 10|01\rangle$$

$$+ |00\rangle\langle 00|10\rangle + |11\rangle\langle 11|10\rangle - |01\rangle\langle 01|10\rangle - |10\rangle\langle 10|10\rangle$$

$$= -( |01\rangle + |10\rangle ) = -|\Psi^+\rangle$$

$|\Psi^-\rangle$  :  $1/\sqrt{2} ( |01\rangle - |10\rangle )$  の観測

$$(Z \otimes Z)|\Psi^-\rangle$$

$$=(|00\rangle\langle 00| + |11\rangle\langle 11| - |01\rangle\langle 01| - |10\rangle\langle 10|) ( |01\rangle - |10\rangle )$$

$$= |00\rangle\langle 00|01\rangle + |11\rangle\langle 11|01\rangle - |01\rangle\langle 01|01\rangle - |10\rangle\langle 10|01\rangle$$

$$|00\rangle\langle 00|10\rangle + |11\rangle\langle 11|10\rangle + |01\rangle\langle 01|10\rangle + |10\rangle\langle 10|10\rangle$$

$$= -( |01\rangle - |10\rangle ) = -|\Psi^-\rangle$$

Bell Stateは、 $Z_1Z_2$ の固有状態である

$$Z_1Z_2|\Phi^+\rangle = +1|\Phi^+\rangle$$

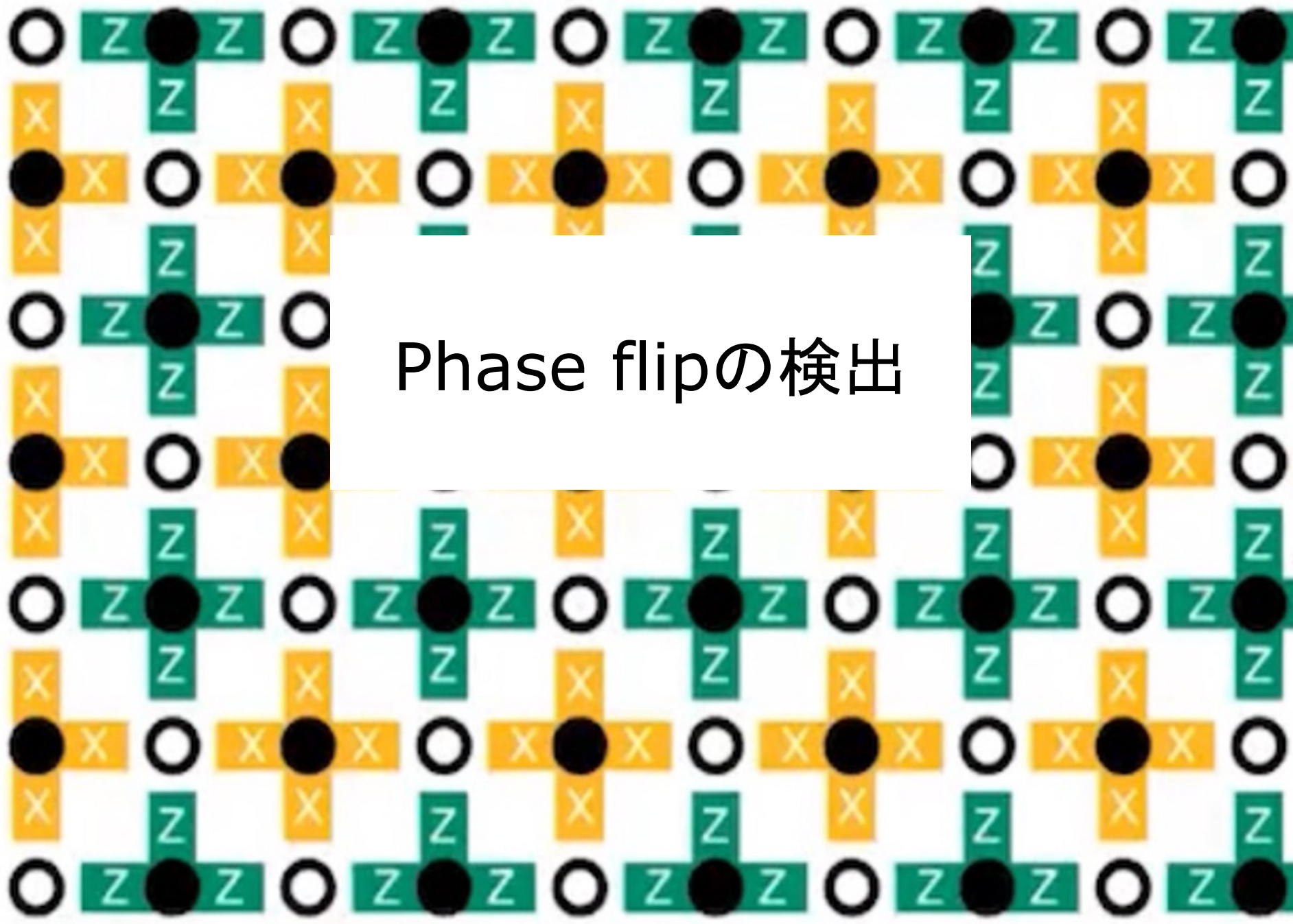
$$Z_1Z_2|\Phi^-\rangle = +1|\Phi^-\rangle$$

$$Z_1Z_2|\Psi^+\rangle = -1|\Psi^+\rangle$$

$$Z_1Z_2|\Psi^-\rangle = -1|\Psi^-\rangle$$

## $Z_1 Z_2$ の固有値と固有状態

固有値	固有状態
+1	$1/\sqrt{2} ( 00\rangle +  11\rangle)$
+1	$1/\sqrt{2} ( 00\rangle -  11\rangle)$
-1	$1/\sqrt{2} ( 01\rangle +  10\rangle)$
-1	$1/\sqrt{2} ( 01\rangle -  10\rangle)$



Phase flipの検出

## 先のbit flip検出の手法では phase flipは検出できない

Part 2 で見た bit flipの検出の方法は、次のようなものでした。通信チャンネルに送り出される直前のencodeで、三つのqubitに多重化された $|\psi\rangle$ は、次のような形をしています。

$$\begin{aligned} |0\rangle &\rightarrow |000\rangle, |1\rangle \rightarrow |111\rangle \\ a|0\rangle + b|1\rangle &\rightarrow |\psi\rangle = a|000\rangle + b|111\rangle \end{aligned}$$

$|\psi\rangle$ がチャンネルの中で、ノイズの影響を受けて $|\psi'\rangle$ に変化したとします。

$$|\psi'\rangle = a|q_1q_2q_3\rangle + b|q_1q_2q_3\rangle$$

先のbit flipの検出では、三つのqubitの中で、一つのqubitにだけエラーが起きると想定しています。ですので、 $Z_1Z_2$ で $q_1$ と $q_2$ を比較し、 $Z_2Z_3$ で $q_2$ と $q_3$ を比較すれば、一つのqubitにだけ起きる変化は検出できます。

ただ、問題はそこにあるわけではありません。チャンネルで起きるphase flipは、次の形をしています。

$$|\psi''\rangle = a|q_1q_2q_3\rangle - b|q_1q_2q_3\rangle$$

これを

$$|\psi'\rangle = a|q_1q_2q_3\rangle + b|q_1q_2q_3\rangle$$

というbit flipの検出で利用した形と区別できるかが問題です。

$q_1$ と $q_2$ を比較し、 $q_2$ と $q_3$ を比較するだけでは、この符号の違いに気づくことはできません。

## phase flipをbit flipに変換する

ただ、うまい方法があるのです。それは、Hadamard演算子とHadamard基底を使うという方法です。

$|0\rangle, |1\rangle$ 基底でのphase flipは、  
 $|+\rangle, |-\rangle$ 基底でのbit flipの形に変換できるのです。

少し、Hadamard演算子  $H$  とHadamard基底  $|+\rangle, |-\rangle$ の性質を振り返ってみましょう。

# Hadamard演算子とHadamard基底

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

$$H(a|0\rangle + b|1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} a + b \\ a - b \end{pmatrix}$$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

$|0\rangle, |1\rangle$ 基底 と  $|+\rangle, |-\rangle$ 基底

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$
$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

とすると。

$$|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$$
$$|1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$$

となります。

## phase flipは、bit flipに変換できる

これから、 $|0\rangle$ ,  $|1\rangle$ 基底でのphase flipは、次のように  
 $|+\rangle$ ,  $|-\rangle$ 基底でのbit flipの形に変換できることが分かります。

$$\begin{aligned} & a|0\rangle - b|1\rangle \\ = & \frac{a}{\sqrt{2}}(|+\rangle + |-\rangle) - \frac{b}{\sqrt{2}}(|+\rangle - |-\rangle) \\ = & \frac{a-b}{\sqrt{2}}|+\rangle + \frac{a+b}{\sqrt{2}}|-\rangle \\ = & a'|+\rangle + b'|-\rangle \end{aligned}$$

# phase flip検出回路を考える

phase flip の検出回路を考えます。

$$\begin{aligned} H|0\rangle &= |+\rangle \\ H|1\rangle &= |-\rangle \end{aligned}$$

から、次の図のようなencoderを考えることができます。

$HH=I$  から、

$$\begin{aligned} H|+\rangle &= |0\rangle \\ H|-\rangle &= |1\rangle \end{aligned}$$

であることも分かります。

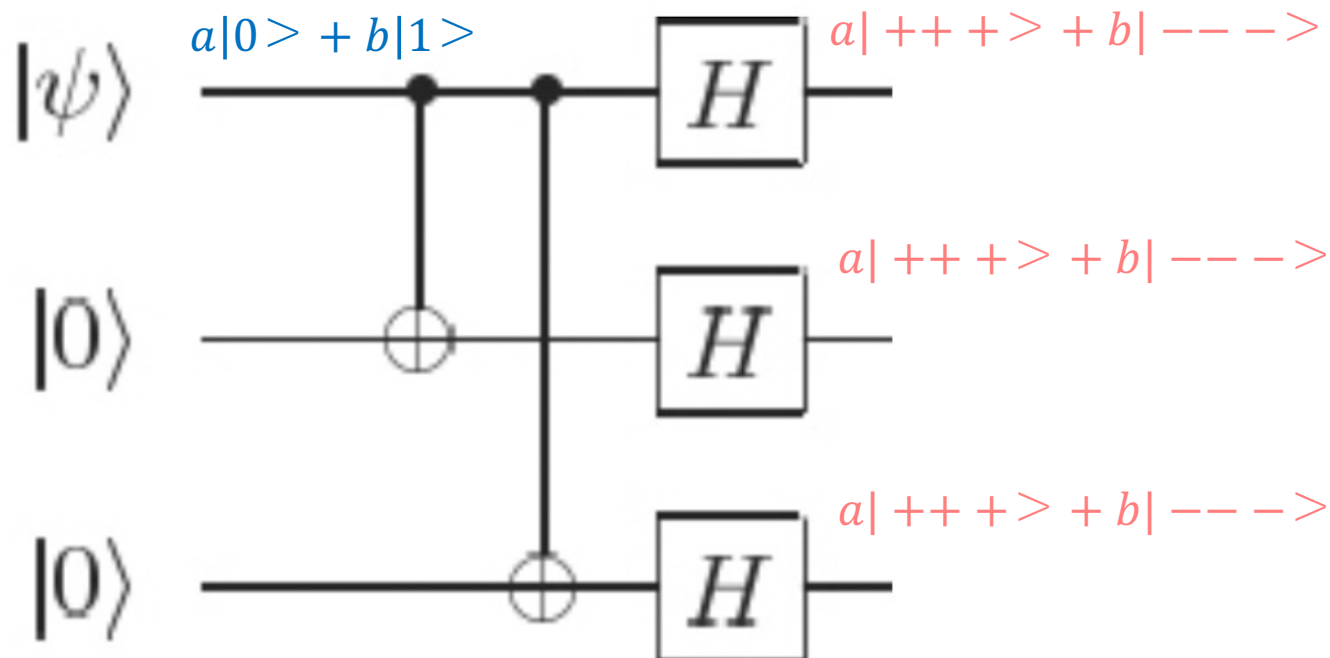
この関係は、decoder に使えます。

# Phase flip 検出のencode回路

この回路は、 $|0\rangle \rightarrow |+++ \rangle$ ,  $|1\rangle \rightarrow |--- \rangle$  と基底を変換して、

$$a|0\rangle + b|1\rangle \rightarrow a|+++ \rangle + b|--- \rangle$$

を、チャンネルに送ります。

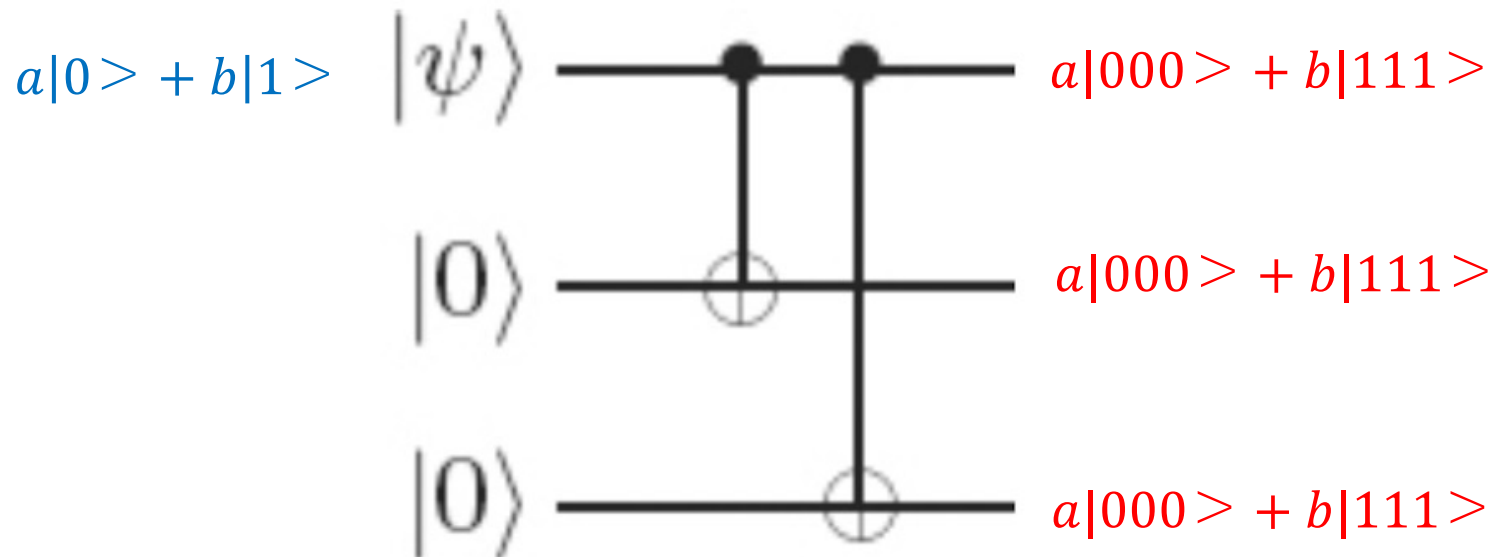


# Hadamard transformation

ただ、Hadamard 付きのencoderの実際の働きは、

$$|0\rangle \rightarrow |000\rangle, |1\rangle \rightarrow |111\rangle \text{ と基底を変換して、}$$
$$a|0\rangle + b|1\rangle \rightarrow a|000\rangle + b|111\rangle$$

をチャンネルに送りだす、以前に見た。次のような回路よりは、もっと複雑なものです。



先の Phase flip 検出の encode 回路の説明では、三つの状態  $a|+++> + b|--->$  が、独立のラインでチャンネルに送り込まれるような印象を与えたと思いますが、実際には、違います。

一般には、並行に配置され結合された  $n$  個の Hadamard 回路に、例えば  $|0>$  の状態を送りこめば、その回路の出力側に現れるのは、 $|0>$  から  $|2^{n-1}>$  までの  $2^n$  個の基底の重ね合わせの状態です。

この状態を、 $H^{\otimes n}|0>$  と表します。

ただし、この状態にあらためて  $H^{\otimes n}$  を適用すると、元の状態  $|0>$  が現れます。

$$|0> \xrightarrow{H^{\otimes n}} H^{\otimes n}|0> \xrightarrow{H^{\otimes n}} |0>$$

phaze flipの検出 =  
 $|+\rangle, |-\rangle$ 基底での bit flipの検出

この回路に続く回路は、phaze flipの検出を行います。

ただ、それは、 $|+\rangle, |-\rangle$ 基底での bit flipの検出なので、前回見たおなじ手法、二つの演算子 $Z_1Z_2$ と $Z_2Z_3$ を使った手法が利用できます。

まず、 $|0\rangle, |1\rangle$ 基底での  $Z \otimes Z$  の外積表記

$$Z \otimes Z = (|00\rangle\langle 00| + |11\rangle\langle 11|) \\ - (|01\rangle\langle 01| + |10\rangle\langle 10|)$$

を、 $|+\rangle, |-\rangle$ を基底とする外積表記に変えます。

$$Z \otimes Z = (|++\rangle\langle++| + |--\rangle\langle--|) \\ -(|+-\rangle\langle+-| + |-+\rangle\langle-+|)$$

$Z \otimes Z$  で、2-qubitの基底  $|++\rangle, |+-\rangle, |-+\rangle, |--\rangle$  を観測します。

例えば、次のように。

$$(Z \otimes Z)|++\rangle \\ = |++\rangle\langle++|++\rangle + |--\rangle\langle--|++\rangle \\ -(|+-\rangle\langle+-|++\rangle + |-+\rangle\langle-+|++\rangle) \\ = |++\rangle$$

その結果は、前回見たものと同じものです。前回の計算式中の 0, 1 が+, - に変わったただけですので。

## 二つのbit flip チェックの比較 前回の場合

ただ、前回紹介したbit flipチェックと、今回、phase flip検出を目的として導入された今回のbit flip チェックは、全く同じものではありません。

前回のbit flip チェックでは、受信側が受け取った情報は、次の二つのObservable で観測されます。

1.  $Z_1 Z_2 = Z \otimes Z \otimes I$  (第一qubitと第二qubitの比較)
2.  $Z_2 Z_3 = I \otimes Z \otimes Z$  (第二qubitと第三qubitの比較)

## 二つのbit flip チェックの比較 今回の場合

ただ、今回のbit flipチェックでは、最終のdecoderで、3-qubitの情報を1-qubitの情報にする以前に、次のような処理が行われています。

1.  $H^{\otimes 3}$ の適用 ( $|0\rangle, |1\rangle$ 基底から $|+\rangle, |-\rangle$ 基底への変換)
2.  $Z_1Z_2 = Z \otimes Z \otimes I$  (第一qubitと第二qubitの比較)
3.  $Z_2Z_3 = I \otimes Z \otimes Z$  (第二qubitと第三qubitの比較)
4.  $H^{\otimes 3}$ の適用 ( $|+\rangle, |-\rangle$ 基底から $|0\rangle, |1\rangle$ 基底への変換)

$$H^{\otimes 3} Z_1 Z_2 H^{\otimes 3} = X_1 X_2$$

こうして、bit flipに置き換えられたphase flipを検出するための処理は、次の二つからなることが分かります。

1.  $H^{\otimes 3} Z_1 Z_2 H^{\otimes 3}$

2.  $H^{\otimes 3} Z_2 Z_3 H^{\otimes 3}$

ところが、興味深いことに、この二つの処理は、Observable  $X$  を使うと、次の形に変形できます。

1.  $H^{\otimes 3} Z_1 Z_2 H^{\otimes 3} = X_1 X_2$

2.  $H^{\otimes 3} Z_2 Z_3 H^{\otimes 3} = X_2 X_3$

$Z_1Z_2, Z_2Z_3$ が

$$Z_1Z_2 = Z \otimes Z \otimes I,$$

$$Z_2Z_3 = I \otimes Z \otimes Z$$

の形をしていたように、

ここで、 $X_1X_2, X_2X_3$ は、

$$X_1X_2 = X \otimes X \otimes I,$$

$$X_2X_3 = I \otimes X \otimes X$$

をしています。

これらについて、

$$H^{\otimes 3}Z_1Z_2H^{\otimes 3} = X_1X_2$$

$$H^{\otimes 3}Z_2Z_3H^{\otimes 3} = X_2X_3$$

が成り立つのです。

$$HZH = X$$

今回、先の式が成り立つことの証明は割愛しますが、その基礎となるのは、H,Z,Xについて、次のような関係が成り立つことです。

$$HZH = X$$

$HZH(a|0\rangle + b|1\rangle)$ を計算してみましょう。(正規化因子を省略しています。)

$$\begin{aligned} HZH(a|0\rangle + b|1\rangle) &= HZH \begin{pmatrix} a \\ b \end{pmatrix} = HZ \begin{pmatrix} a + b \\ a - b \end{pmatrix} \\ &= H \begin{pmatrix} a + b \\ -a + b \end{pmatrix} = \begin{pmatrix} a + b + (-a + b) \\ a + b - (-a + b) \end{pmatrix} = \begin{pmatrix} 2b \\ 2a \end{pmatrix} \end{aligned}$$

$$HZH \begin{pmatrix} a \\ b \end{pmatrix} = X \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} \quad \text{これから } HZH = X \text{ が言えます。}$$

Observable  $X_1X_2$ ,  $X_2X_3$  による観測は、それぞれ一番目と二番目、二番目と三番目のqubitのパリティのサインを比較することに対応しています。

例えば、 $X_1X_2$ による観測は、 $|+\rangle|+\rangle \otimes (\cdot)$  あるいは  $|-\rangle|-\rangle \otimes (\cdot)$  の場合には、**+1**を返し、 $|+\rangle|-\rangle \otimes (\cdot)$  あるいは  $|-\rangle|+\rangle \otimes (\cdot)$  の場合には、**-1**を返します。

## 2-qubitのObservable $X \otimes X$ の行列を計算する

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ から、}$$

$$X \otimes X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

2-qubitのObservable  $X \otimes X$  の行列から、  
その外積での表示を求める

$$X \otimes X = \begin{matrix} & \begin{matrix} 00 & 01 & 10 & 11 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} & \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \end{matrix}$$

$$X \otimes X =$$

$$|00\rangle\langle 11| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 00|$$

## $X_1 X_2$ での2-qubitのBell Stateの観測

$$X \otimes X = |00\rangle\langle 11| + |01\rangle\langle 10| \\ + |10\rangle\langle 01| + |11\rangle\langle 00|$$

$X_1 X_2$  で、2-qubitの四つのBell State

$$|00\rangle + |11\rangle, |00\rangle - |11\rangle, \\ |01\rangle + |10\rangle, |01\rangle - |10\rangle$$

を観測してみましょう。

$(X \otimes X)(|00\rangle + |11\rangle)$  の場合

$$\begin{aligned} & (X \otimes X)(|00\rangle + |11\rangle) \\ = & |00\rangle \langle 11|00\rangle + |01\rangle \langle 10|00\rangle \\ & + |10\rangle \langle 01|00\rangle + |11\rangle \langle 00|00\rangle \\ & + |00\rangle \langle 11|11\rangle + |01\rangle \langle 10|11\rangle \\ & + |10\rangle \langle 01|11\rangle + |11\rangle \langle 00|11\rangle \end{aligned}$$

$$(X \otimes X)(|00\rangle + |11\rangle) = |11\rangle + |00\rangle$$

$(X \otimes X)(|00\rangle - |11\rangle)$  の場合

$$\begin{aligned} & (X \otimes X)(|00\rangle - |11\rangle) \\ = & |00\rangle \langle 11|00\rangle + |01\rangle \langle 10|00\rangle \\ & + |10\rangle \langle 01|00\rangle + |11\rangle \langle 00|00\rangle \\ - & |00\rangle \langle 11|11\rangle - |01\rangle \langle 10|11\rangle \\ - & |10\rangle \langle 01|11\rangle - |11\rangle \langle 00|11\rangle \end{aligned}$$

$$(X \otimes X)(|00\rangle - |11\rangle) = |11\rangle - |00\rangle$$

$(X \otimes X)(|01\rangle + |01\rangle)$ の場合

$$\begin{aligned} & (X \otimes X)(|01\rangle + |10\rangle) \\ = & |00\rangle \langle 11|01\rangle + |01\rangle \langle 10|10\rangle \\ & + |10\rangle \langle 01|01\rangle + |11\rangle \langle 00|01\rangle \\ & + |00\rangle \langle 11|10\rangle + |01\rangle \langle 10|10\rangle \\ & + |10\rangle \langle 01|10\rangle + |11\rangle \langle 00|10\rangle \end{aligned}$$

$$(X \otimes X)(|01\rangle + |10\rangle) = |10\rangle + |01\rangle$$

$(X \otimes X)(|01\rangle - |10\rangle)$  の場合

$$\begin{aligned} & (X \otimes X)(|01\rangle - |10\rangle) \\ = & |00\rangle \langle 11|01\rangle + |01\rangle \langle 10|10\rangle \\ & + |10\rangle \langle 01|01\rangle + |11\rangle \langle 00|01\rangle \\ & - |00\rangle \langle 11|10\rangle - |01\rangle \langle 10|10\rangle \\ & - |10\rangle \langle 01|10\rangle - |11\rangle \langle 00|10\rangle \end{aligned}$$

$$(X \otimes X)(|01\rangle - |10\rangle) = |10\rangle - |01\rangle$$

## 整理すると

$$(X \otimes X)(|00\rangle + |11\rangle) = +\mathbf{1}(|00\rangle + |11\rangle)$$

$$(X \otimes X)(|00\rangle - |11\rangle) = -\mathbf{1}(|00\rangle - |11\rangle)$$

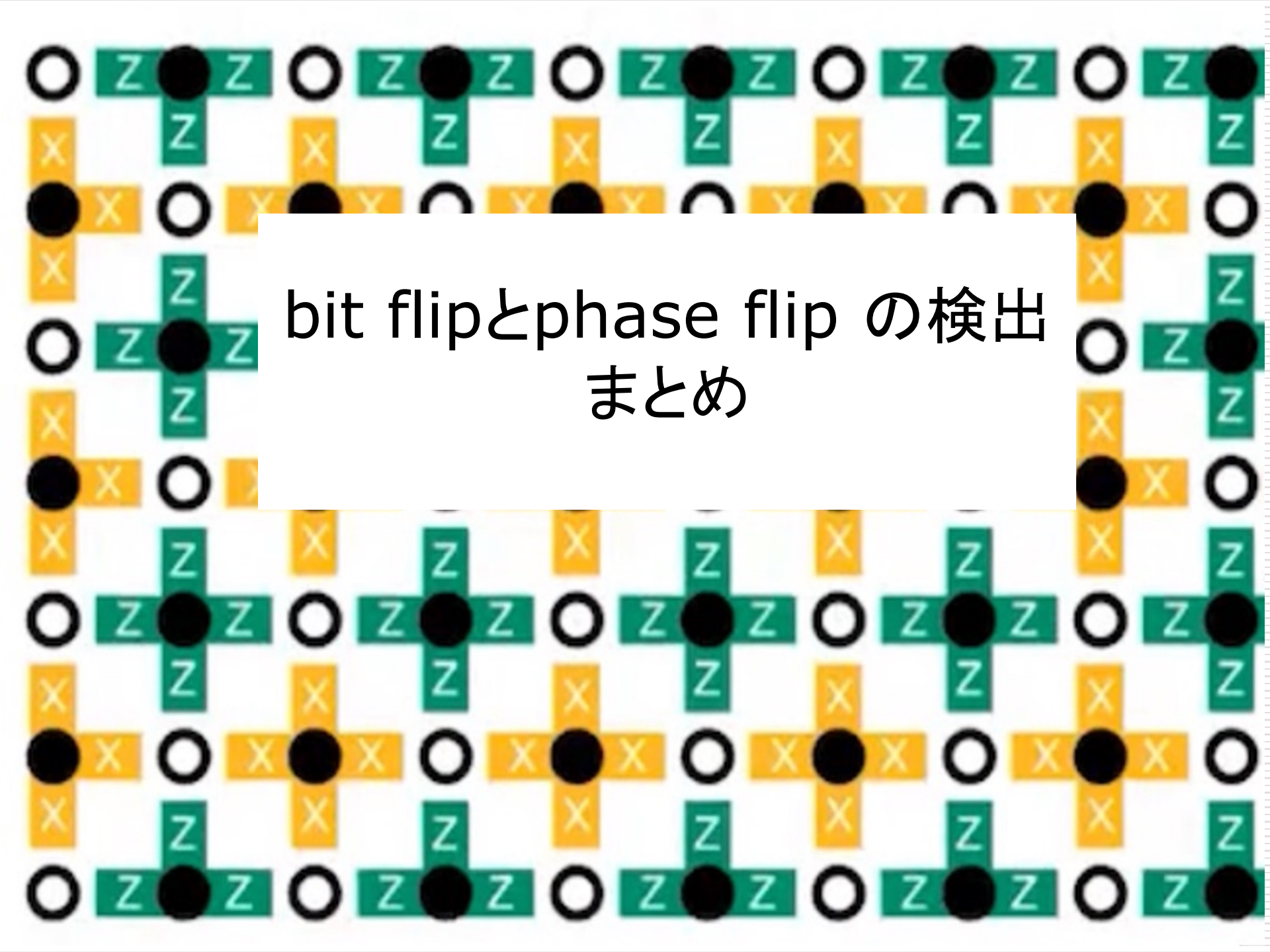
$$(X \otimes X)(|01\rangle + |10\rangle) = +\mathbf{1}(|01\rangle + |10\rangle)$$

$$(X \otimes X)(|01\rangle - |10\rangle) = -\mathbf{1}(|01\rangle - |10\rangle)$$

$X_1X_2$  の固有値は、+1 と -1 である。  
Bell State は、 $X_1X_2$  の固有状態である

## $X_1 X_2$ の固有値と固有状態

固有値	固有状態
+1	$1/\sqrt{2} ( 00\rangle +  11\rangle)$
-1	$1/\sqrt{2} ( 00\rangle -  11\rangle)$
+1	$1/\sqrt{2} ( 01\rangle +  10\rangle)$
-1	$1/\sqrt{2} ( 01\rangle -  10\rangle)$

The image features a repeating pattern of quantum circuit symbols. It consists of a grid of circles, squares, and crosses. The circles are arranged in a checkerboard pattern, alternating between white and black. The squares are arranged in a grid, with some containing the letter 'Z' and others containing the letter 'X'. The crosses are arranged in a grid, with some containing the letter 'Z' and others containing the letter 'X'. The symbols are colored in green, yellow, and black. The text "bit flipとphase flip の検出" and "まとめ" is overlaid on the grid.

bit flipとphase flip の検出  
まとめ

## $Z \otimes Z$ はパリティをチェックする

$$(Z \otimes Z) |00\rangle = +\mathbf{1}|00\rangle$$

$$(Z \otimes Z) |01\rangle = -\mathbf{1}|01\rangle$$

$$(Z \otimes Z) |10\rangle = -\mathbf{1}|10\rangle$$

$$(Z \otimes Z) |11\rangle = +\mathbf{1}|11\rangle$$

$Z \otimes Z$  の固有値  $\pm 1$

固有値  $+1$  の時 固有状態  $|00\rangle, |11\rangle$

$$|q_i\rangle = |q_j\rangle \quad (i \neq j)$$

固有値  $-1$  の時 固有状態  $|01\rangle, |10\rangle$

$$|q_i\rangle \neq |q_j\rangle \quad (i \neq j)$$

次のような対応のもとで、

$$\begin{array}{ccccc} & & Q_1 & Q_2 & Q_3 \\ & & \downarrow & \downarrow & \downarrow \\ Z_1 Z_2 & = & Z \otimes Z & \otimes I \\ Z_2 Z_3 & = & I \otimes Z \otimes Z \\ & & \uparrow & \uparrow & \uparrow \\ & & Q_1 & Q_2 & Q_3 \end{array}$$

$Z_1 Z_2 = Z \otimes Z \otimes I$  は、qubit  $Q_1$  と  $Q_2$  の状態を観測し、

$Z_2 Z_3 = I \otimes Z \otimes Z$  は、qubit  $Q_2$  と  $Q_3$  の状態を観測する。

$Z_1 Z_2$  で観測 (q1とq2を観測)

観測値 +1       $q_1 = q_2$

観測値 -1       $q_1 \neq q_2$

$Z_2 Z_3$  で観測 (q2とq3を観測)

観測値 +1       $q_2 = q_3$

観測値 -1       $q_2 \neq q_3$

## bit flip の検出

$Z_1Z_2$	$Z_2Z_3$		
+1	+1	$q1 = q2 = q3$	No flip
+1	-1	$q1 = q2 \neq q3$	q3 flipped
-1	+1	$q1 \neq q2 = q3$	q1 flipped
-1	-1	$q1 \neq q2 \neq q3$	q2 flipped

## $X \otimes X$ はパリティをチェックする

$$(X \otimes X) |++\rangle = +\mathbf{1}|++\rangle$$

$$(X \otimes X) |+-\rangle = -\mathbf{1}|+-\rangle$$

$$(X \otimes X) |-+\rangle = -\mathbf{1}|-+\rangle$$

$$(X \otimes X) |--\rangle = +\mathbf{1} |--\rangle$$

$X \otimes X$  の固有値  $\pm 1$

固有値  $+1$  の時 固有状態  $|++\rangle, |--\rangle$

$$|q_i\rangle = |q_j\rangle \quad (i \neq j)$$

固有値  $-1$  の時 固有状態  $|+-\rangle, |-+\rangle$

$$|q_i\rangle \neq |q_j\rangle \quad (i \neq j)$$

次のような対応のもとで、

$$\begin{array}{ccccc} & & Q_1 & Q_2 & Q_3 \\ & & \downarrow & \downarrow & \downarrow \\ X_1 X_2 & = & X \otimes X & \otimes I & \\ X_2 X_3 & = & I \otimes X & \otimes X & \\ & & \uparrow & \uparrow & \uparrow \\ & & Q_1 & Q_2 & Q_3 \end{array}$$

$X_1 X_2 = X \otimes X \otimes I$  は、qubit  $Q_1$  と  $Q_2$  の状態を観測し、  
 $X_2 X_3 = I \otimes X \otimes X$  は、qubit  $Q_2$  と  $Q_3$  の状態を観測する。

$X_1 X_2$  で観測 (q1とq2を観測)

観測値 +1       $q1 = q2$

観測値 -1       $q1 \neq q2$

$X_2 X_3$  で観測 (q2とq3を観測)

観測値 +1       $q2 = q3$

観測値 -1       $q2 \neq q3$

# phase flip の検出

$X_1X_2$	$X_2X_3$		
+1	+1	$q1 = q2 = q3$	No flip
+1	-1	$q1 = q2 \neq q3$	q3 flipped
-1	+1	$q1 \neq q2 = q3$	q1 flipped
-1	-1	$q1 \neq q2 \neq q3$	q2 flipped

## bit flip, phase flip 検出のまとめ

$Z_1Z_2$	$Z_2Z_3$	bit flip検出
+1	+1	No flip
+1	-1	q3 flipped
-1	+1	q1 flipped
-1	-1	q2 flipped

$X_1X_2$	$X_2X_3$	phase flip検出
+1	+1	No flip
+1	-1	q3 flipped
-1	+1	q1 flipped
-1	-1	q2 flipped

## $Z_1Z_2$ と $X_1X_2$ の固有値と固有状態

固有値

固有状態

$Z_1Z_2$	$X_1X_2$	
+1	+1	$1/\sqrt{2} ( 00\rangle +  11\rangle)$
+1	-1	$1/\sqrt{2} ( 00\rangle -  11\rangle)$
-1	+1	$1/\sqrt{2} ( 01\rangle +  10\rangle)$
-1	-1	$1/\sqrt{2} ( 01\rangle -  10\rangle)$



