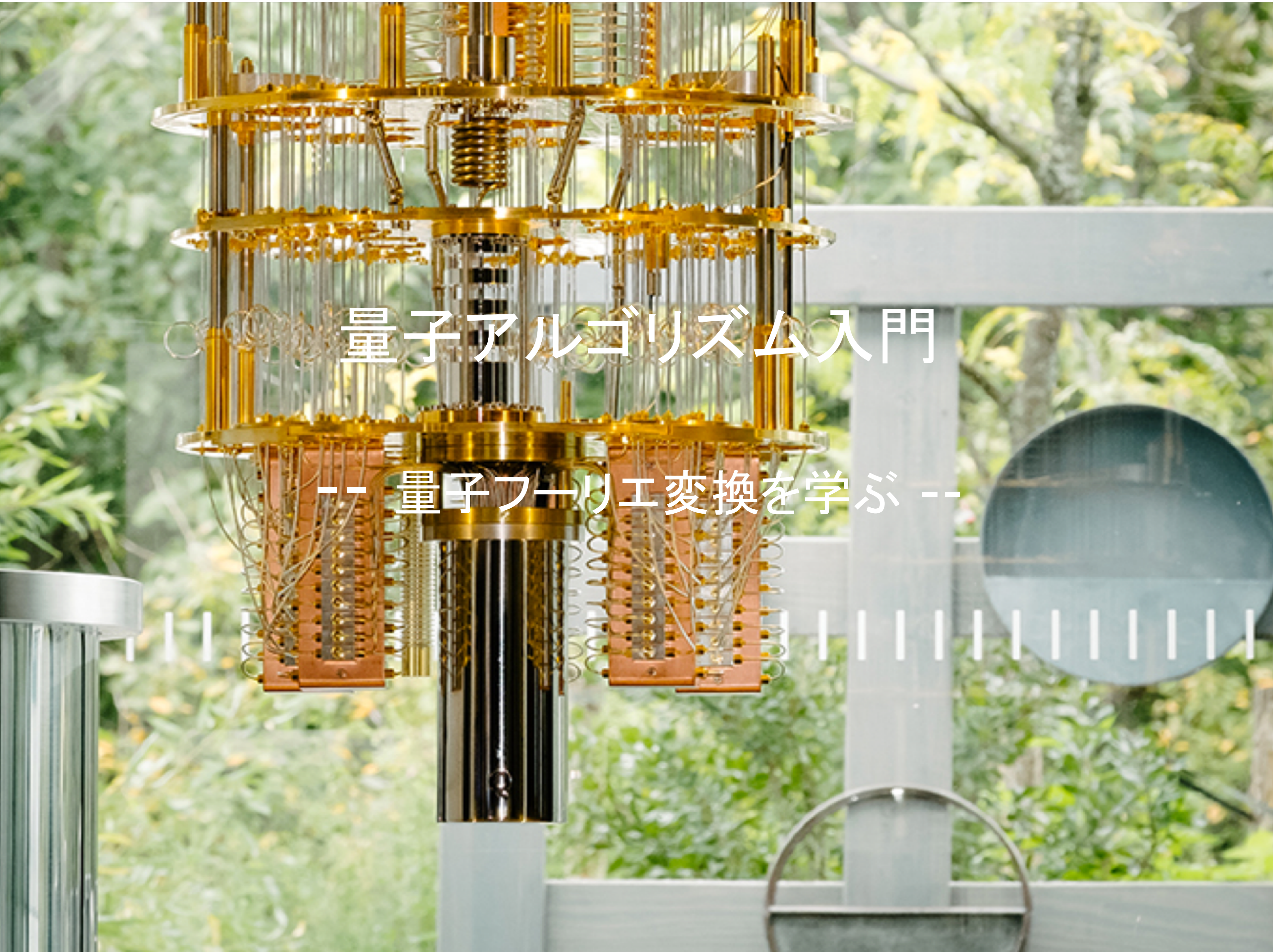


量子アルゴリズム入門

-- 量子フーリエ変換を学ぶ --

はじめに

- この間の「紙と鉛筆で学ぶ量子情報理論基礎演習 I」では、量子コンピュータの基礎である「量子ゲート」の働きを学んできた。（いわば、コンピュータのハードウェアの働きとして）
- 今回は、量子ゲートで構成される回路がどのようなアルゴリズムを実現するのかを見る。ゲート型量子コンピュータにおいては、「量子アルゴリズム」とは、そのアルゴリズムを実行する量子回路を構成することである。（残念ながら、現在の量子コンピュータでは、ハードとソフトの分離は、できていないのだ。）
- 本セミナーは、代表的な「量子アルゴリズム」の一つである「Shorのアルゴリズム」で中心的な役割を果たす「量子フーリエ変換」 のアルゴリズムの解説を、量子情報理論の初心者を対象に行うことを一つの目標としている。

はじめに

- 小論の前半では、次のような量子アルゴリズム・量子通信の基本を学ぶ。これらの内容は、「紙と鉛筆 ...」の内容のすぐ後に続くものである。
  - ノー・クローニング定理 (No Cloning Theorem)
  - エンタングルメント (Entanglement)
  - スーパー・デンス・コーディング (Superdense Coding)
  - 量子テレポーテーション (Quantum Teleportation)
  - 量子並列計算 (Quantum Parallelism)

はじめに

- 後半では、量子フーリエ変換(QFT)を取り上げる。最初の二節は少し数学的だが、DFT/FFTについては、技術者には理解者も多いと考えている。
  - 古典的フーリエ級数/変換と複素表示
  - ディジタル・フーリエ変換と高速フーリエ変換 （DFTとFFT）
  - 量子フーリエ変換 (QFT)
- 来年からは、これらの内容にいくつかのアルゴリズムを追加して演習形式で、「量子情報理論基礎演習 II – 量子アルゴリズム編」を開始できたらと考えている。
- 本セミナーを、こうした演習コースに接続する、量子アルゴリズム学習の最初の入門講座として利用されることを期待している。

Agenda Part I
量子アルゴリズムの基礎

- 量子情報理論の基礎
  - 三つの原理
  - テンソル積
  - 量子ゲート
- 量子アルゴリズムの基礎
  - No Cloning Theorem
  - Entanglement
  - Super Dense Coding
  - Quantum Teleportation
  - Quantum Parallelism

Agenda Part II
量子フーリエ変換

- Quantum Fourier Transform
  - 古典的フーリエ変換
  - Digital Fourier Transform
  - Fast Fourier Transform
  - Quantum Fourier Transform

Appendix

- Period Finding
- Phase Estimation

量子情報理論の基礎

量子力学と量子情報理論

- 量子力学は、物理学の一分野である。それは、自然の実在的な物質を対象として、物質の運動法則の解明を目指す。
- 量子ゲート型の量子コンピュータの基礎理論に限るならば、それが主要な対象としているのは、二つの状態の重ね合わせであるqubitから構成される抽象的な系である。その理論構造は、比較的、単純なものである。小論が扱うのは、その範囲である。
- 量子情報理論は、量子論から派生したものである。ただ、量子情報理論の応用は、量子ゲート型の量子コンピュータの基礎に限られているわけではない。
- 近年の物理学では、ブラックホールの情報問題や、エンタングルメントのエントロピー論が、大きな関心を集めている。それは、量子情報理論を基礎理論として、物理学を再構成しようとする流れが生まれているように見える。

qubitを理解する三つの原理

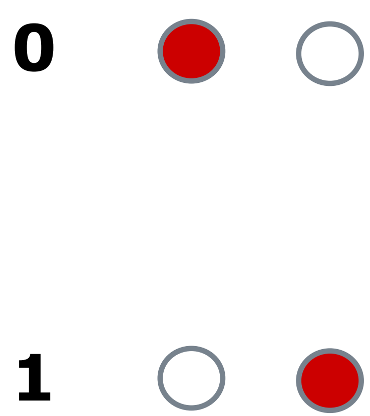
ここでは、問題を大幅に単純化して、次の三つの原理で、量子コンピュータの基礎を述べて見ようと思う。

1. 重ね合わせの原理
2. 観測の原理
3. ユニタリー発展の原理

重ね合わせの原理

古典Bit

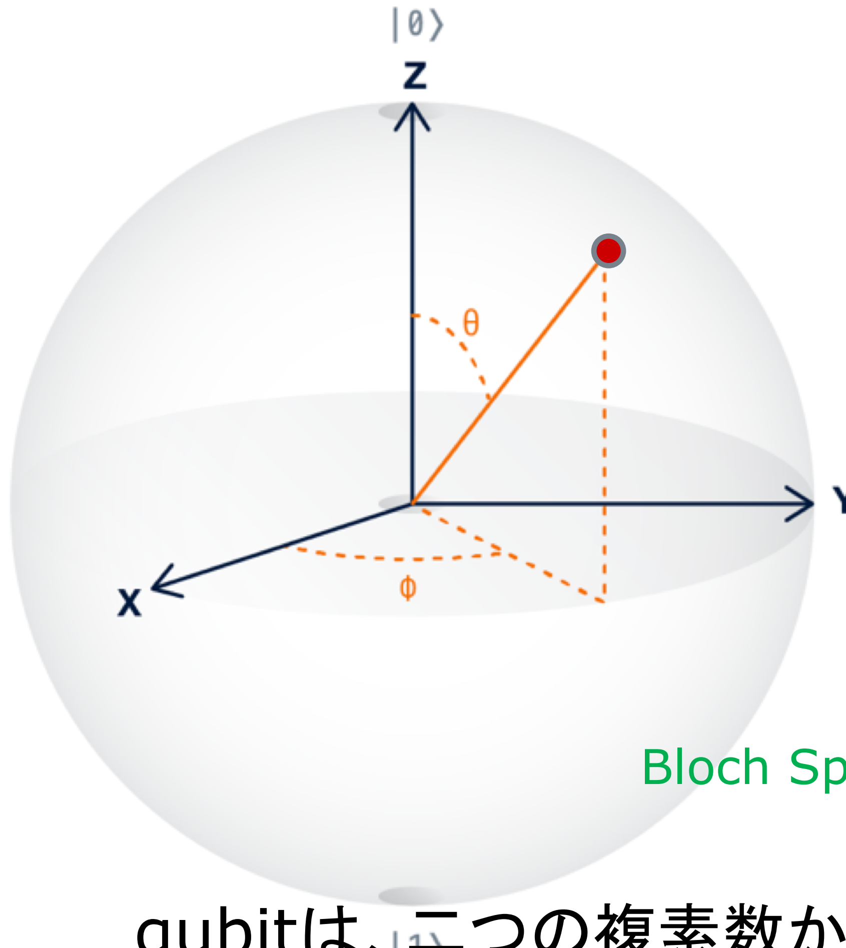
0か1



Bitは、0または1
どちらかの状態を取る

bitは、整数値を取り、離散的

量子Qubit



Bloch Sphere

qubitは、二つの複素数から
なり、連続的

qubitを決める複素数c_0,c_1は、それぞれ2つの実数からなるが、$|c_0|^2+|c_1|^2=1$という条件があるので、自由度は一つ減って、qubitを決めるのは、3つの実数である。

bitの状態は、0か1で表される
qubitの状態は、2次元の複素数ベクトルで表現される

bitの状態 ＝ 0か1

$$\text{qubitの状態} = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$$

ただし、c_0, c_1は複素数で、$|c_0|^2 + |c_1|^2 = 1$

一つのbitは、0か1の二つの状態しかとらないが、
一つのqubitは、無限に多くの状態を取りうる

もっと一般に、量子の状態は、
n次元の複素数ベクトルで表現される

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

ただし、$c_0, c_1, \ldots, c_{n-1}$は複素数

こうした複素数ベクトル空間をヒルベルト空間という

重ね合わせ

qubitは、二つの状態の和として表される

$$\text{qubitの状態} = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = c_0\begin{pmatrix} 1 \\ 0 \end{pmatrix} + c1\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

ベクトル空間の「基底」

ここで、$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ を |0>, $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ を |1> で表すと

$$\text{qubitの状態} = c_0|0> + c_1|1>$$

ただし、 $|c_0|^2+|c_1|^2=1$

と表現できる。

もっと一般に、量子の状態は、
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>,|1>,|2>,...,|n-1>
とすると、

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

$$c_0|0\rangle + c_1|1\rangle + \ldots + c_{n-1}|n\text{-}1\rangle$$

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

量子の状態を、Σ を使って表す

総和の記号Σ を使うと、量子の状態を見やすく表現できる。

量子の状態 $= c_0|0> + c_1|1> + \dots + c_{n-1}|n-1>$

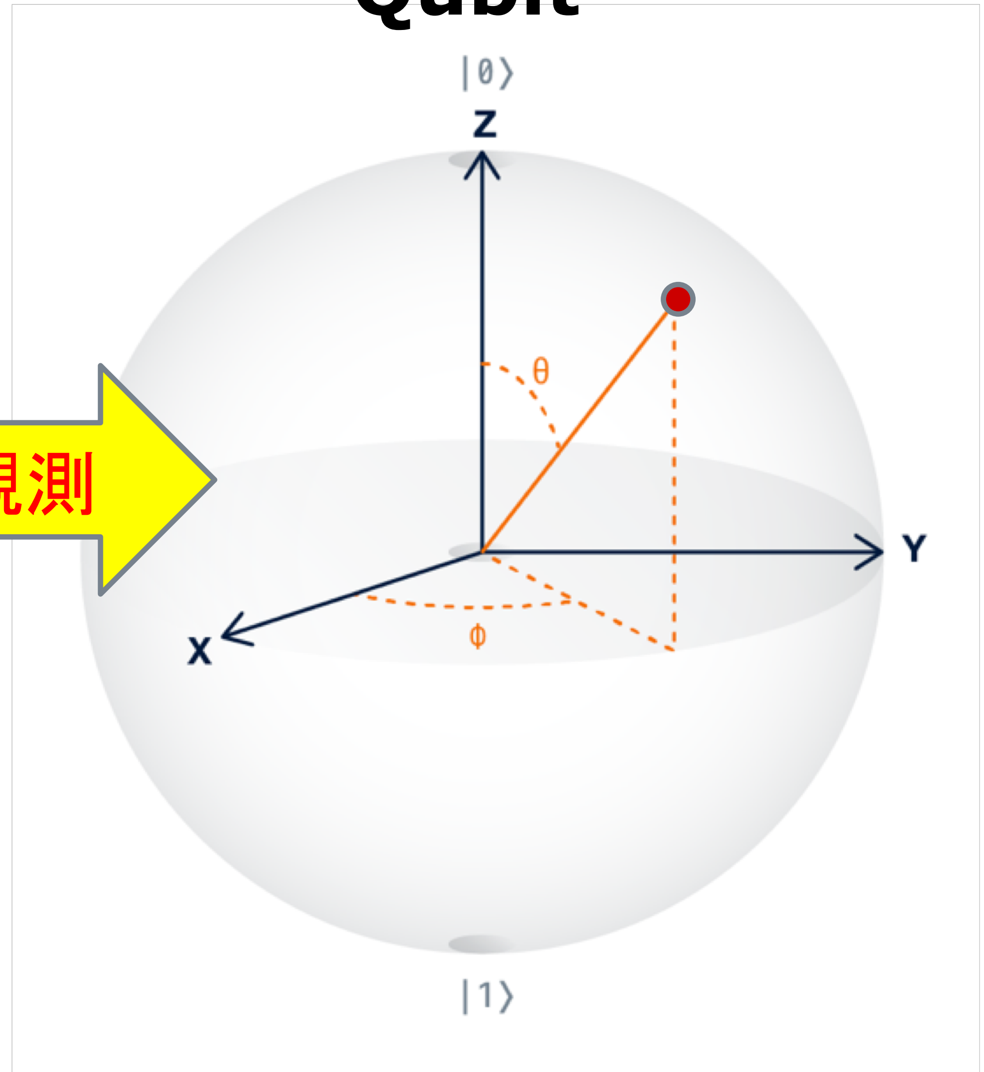
$$= \sum_{k=0}^{n-1} c_k|k>$$

ただし、$\sum_{k=0}^{n-1} |c_k|^2 = 1$

観測の原理

Qubit

観測

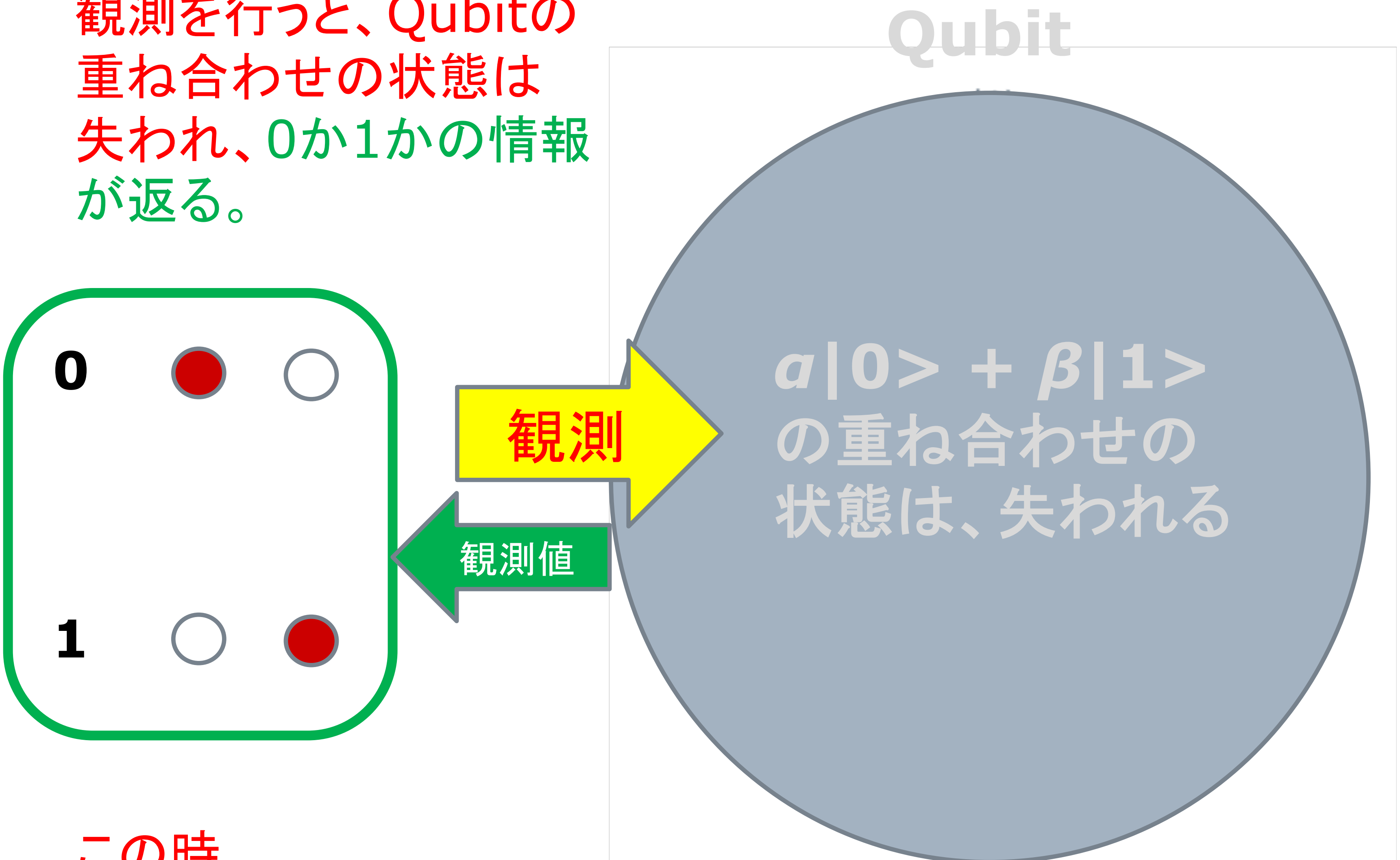


Qubitは、状態 |0>と状態 |1>の**重ね合わせ**の状態を取る

$$|\mathrm{Qubit}> = \alpha|0> + \beta|1>$$

$$|\alpha|^2 + |\beta|^2 = 1\ ;\ \ \alpha, \beta \in \mathbb{C}$$

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。



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

$$\alpha^*\alpha + \beta^*\beta = 1\ ;\ \ \alpha, \beta \in \mathbb{C}$$

qubitの観測

qubitの状態 $c_0|0> +c_1|1>$ を観測したとする。この時、

状態 $|0>$ が観測される確率は、$|c_0|^2$ で与えられる。
観測前の状態$c_0|0> +c_1|1>$は、新しい状態 $|0>$に変わる。

状態 $|1>$ が観測される確率は、$|c_1|^2$ で与えられる。
観測前の状態$c_0|0> +c_1|1>$は、新しい状態 $|1>$に変わる。

これをBornのルールと呼ぶ、

観測によって、重ね合わせの状態は破られる。

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。

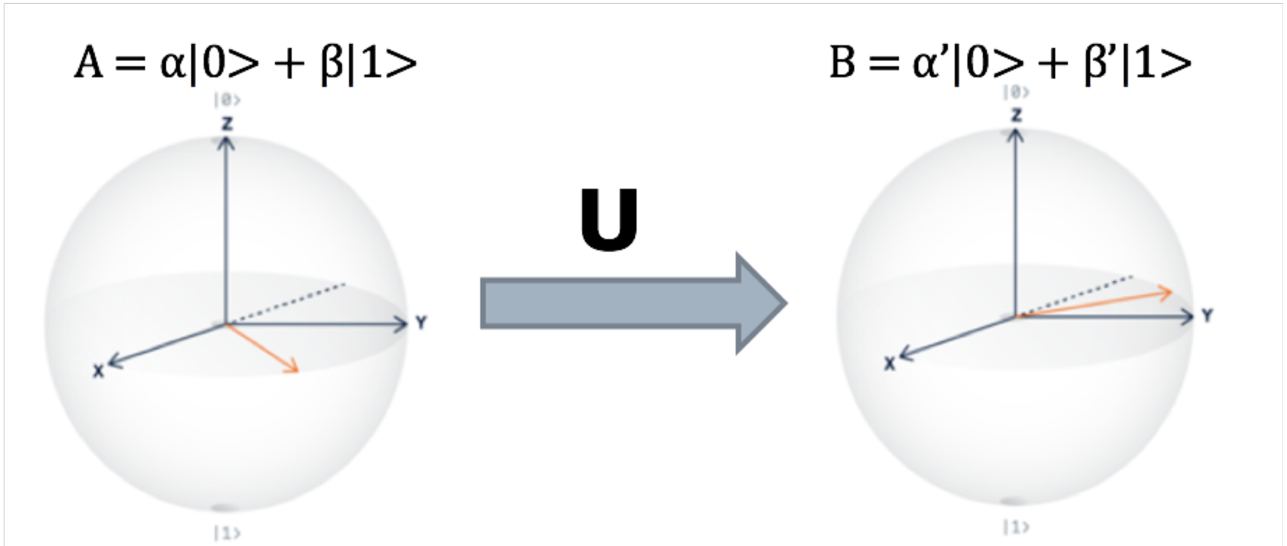
ユニタリ発展の原理

ユニタリ変換の原理

ある量子の状態は、他の状態に変化することができる。
その変化は、「ユニタリ変換」という、ルールに従う。

量子の状態ベクトル |A>は、このユニタリ変換Uによって、他の状態 |B> に変換される。

$$U\,|A\rangle = |B\rangle$$

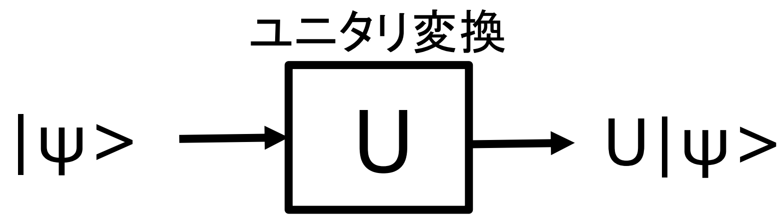


ユニタリ演算子　Unitary Operator

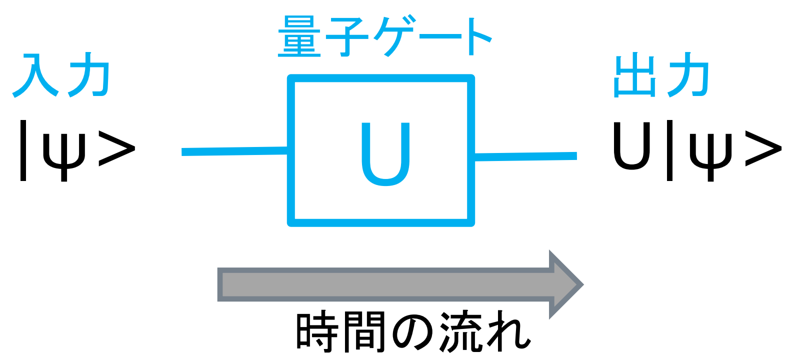
- $UU^\dagger = U^\dagger U = I$ (単位行列)を満たす行列を、ユニタリ行列と呼ぶ。
- ユニタリ変換Uは、幾何的には、ベクトルの長さを変えないベクトルの回転である。
- ユニタリ演算子を適用しても、二つのベクトルの内積は変化しない。ユニタリ演算子は、内積を保存する。

量子ゲート

- 量子の状態 |ψ>は、ユニタリ変換Uの作用を受けて、状態 U|ψ> に変化する。
- この変化 |ψ> → U|ψ>を、次のように表そう。



- この時、Uを、|ψ> を入力、U|ψ> を出力とする回路と考えることができる。これを、「量子ゲート」と呼ぶ。

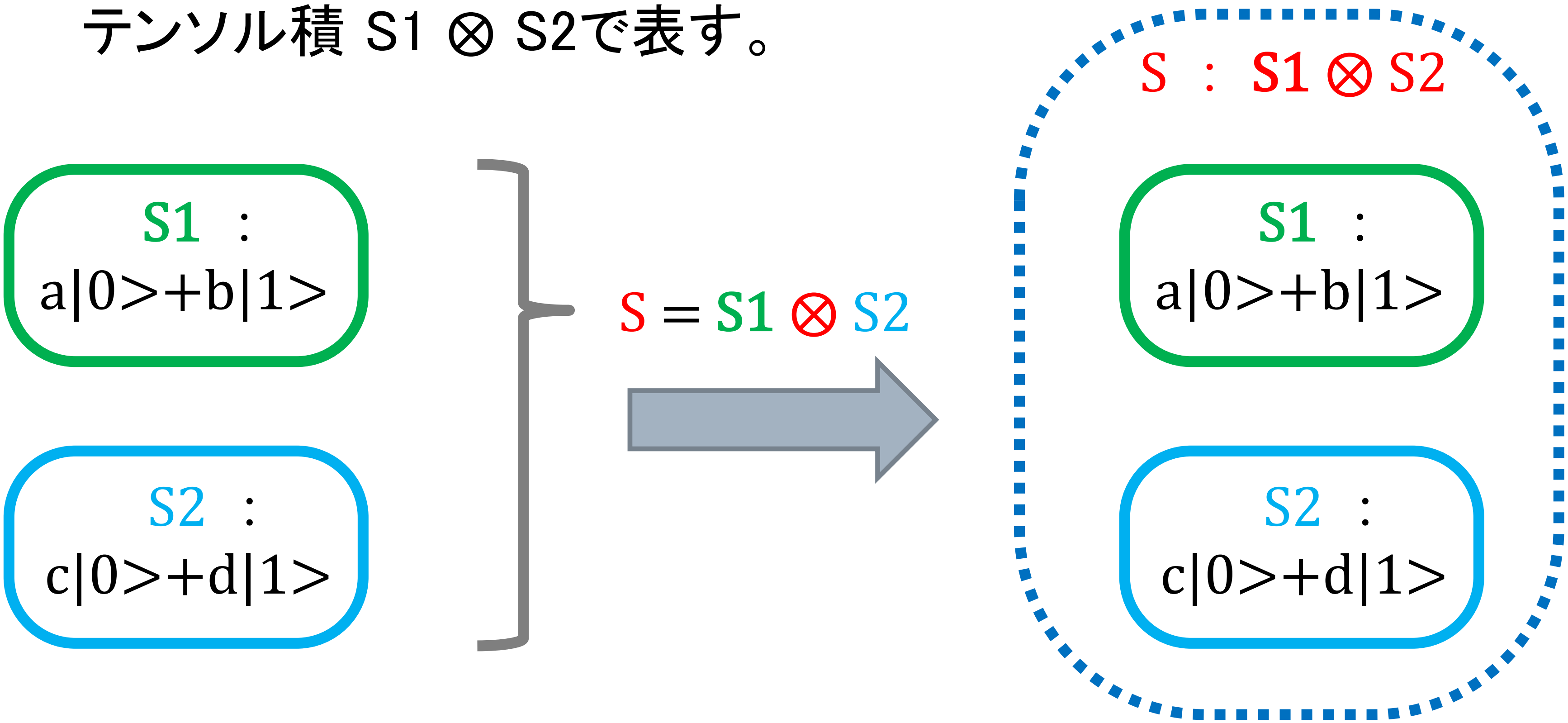


テンソル積

テンソル積
複数のシステムを一つのものと考える

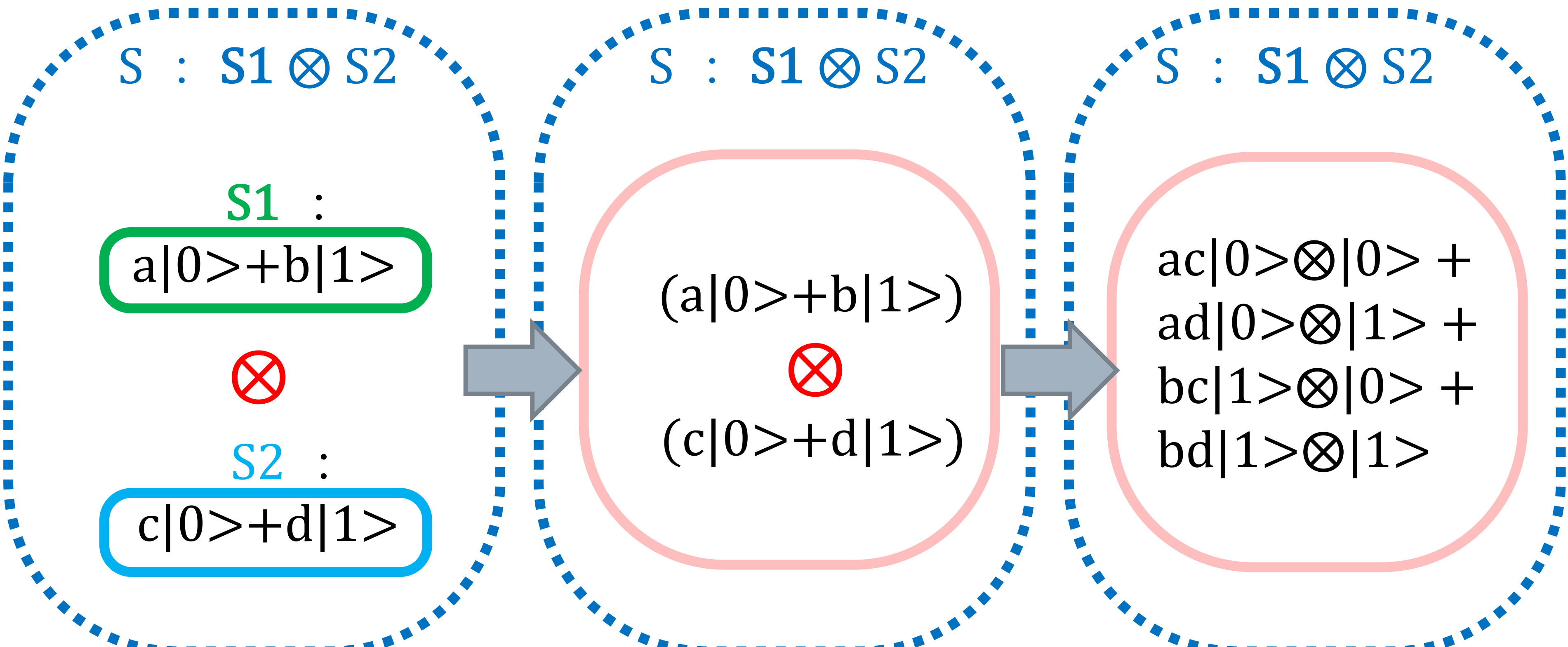
テンソル積

- qubit a|0>+b|1> のみを含むシステムS1、
  qubit c|0>+d|1> のみを含むシステムS2
  があるとしよう。
  この二つを一緒にしたシステムSを
  テンソル積 S1 ⊗ S2で表す。



テンソル積

- この時、次のような計算で、Sの状態を計算する。
  $S = S1 \otimes S2 = (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$
  $= ac|0\rangle \otimes |0\rangle + ad|0\rangle \otimes |1\rangle + bc|1\rangle \otimes |0\rangle + bd|1\rangle \otimes |1\rangle$



テンソル積

- |a>⊗|b>を |ab>と表すことにする。
  - |0>⊗|0> -> |00>, |0>⊗|1> -> |01>,
  - |1>⊗|0> -> |10>, |1>⊗|1> -> |11>

S : S1 ⊗ S2

ac|0>⊗|0> +
ad|0>⊗|1> +
bc|1>⊗|0> +
bd|1>⊗|1>

S : S1 ⊗ S2

ac|00> +
ad|01> +
bc|10> +
bd|11>

ただし、
$|ac|^2 + |ad|^2 + |bc|^2 + |bd|^2 = 1$

3つのqubitからなるシステム

- 先のSに、もう一つqubit e|0>+f|1> のみを含むシステムS3を追加して、3つのqubitからなるシステムを考えよう。改めて、これもSとすると

$$
\begin{aligned}
S &= S1 \otimes S2 \otimes S3 \\
&= (a|0\rangle+b|1\rangle) \otimes (c|0\rangle+d|1\rangle) \otimes (c|0\rangle+d|1\rangle) \\
&= \cdots \\
&= A|000\rangle+B|001\rangle+C|010\rangle+D|011\rangle \\
&\quad + E|100\rangle+F|101\rangle+G|110\rangle+H|111\rangle
\end{aligned}
$$

  という 8つの状態の重ね合わせの形になる。

Two Qubitのテンソル積の計算例

- 第一のQubitが、3/5|0>+4/5|1> で、
  第二のQubitが、$1/\sqrt{2}|0>-1/\sqrt{2}|0>$としよう。

- この二つのQubitが結合したシステムの状態は、
  $(3/5|0>+4/5|1>) \otimes (1/\sqrt{2}|0>-1/\sqrt{2}|1>)$
  $= 3/5\sqrt{2}|00>-3/5\sqrt{2}|01>+4/5\sqrt{2}|10>-4/5\sqrt{2}|11>$
  のように計算される。

- ⊗ を、普通の掛け算のように計算している。
  その上で、|0>⊗|0> -> |00>, |0>⊗|1> -> |01>,
  |1>⊗|0> -> |10>, |1>⊗|1> -> |11> と置き換えればいい。

Two Qubitsのシステムの観測

- 二つのQubitのシステムは、|00>, |01>, |10>, |11>の四つの状態を持つので、次のように表現される。

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

ここで、 $\alpha_{ij} \leq \mathbb{C}, \sum_{ij} |\alpha_{ij}|^2 = 1.$ である。

$$|\psi> = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$ である。

- この時、|00> が観測される確率は、$|\ \alpha_{00}\ |^2$で、観測後の状態は、|00>に変わる。

Two Qubits　部分的な観測

- 先の $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$ で、最初の1ビットのみを観測して、それが0である確率は、どうなるであろうか？
- 次のように考える。

$$\Pr\{\text{1st bit } = 0\} = \Pr\{00\} + \Pr\{01\} = |\alpha_{00}|^2 + |\alpha_{01}|^2.$$

- それでは、このとき、2-Qubitsのシステムの状態は、どういう状態になるのだろうか？　それは、この観測に矛盾する全ての項（最初のビットが1の項）を消して、残った項の重ね合わせの状態になる。ただし、それが、単位ベクトルになるように、正規化されねばならない。新しい状態は、次のようになる。

$$|\phi_{\text{new}}\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

行列のテンソル積

行列のテンソル積

行列のテンソル積を次のように定義する。（2x2行列で例示）

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B \\ A_{21}B & A_{22}B \end{pmatrix}$$

Bの成分も書くと、

$$A \otimes B = \begin{pmatrix} A_{11}B_{11} & A_{11}B_{12} & A_{12}B_{11} & A_{12}B_{12} \\ A_{11}B_{21} & A_{11}B_{22} & A_{12}B_{21} & A_{12}B_{22} \\ A_{21}B_{11} & A_{21}B_{12} & A_{22}B_{11} & A_{22}B_{12} \\ A_{21}B_{21} & A_{21}B_{22} & A_{22}B_{21} & A_{22}B_{22} \end{pmatrix}.$$

行列のテンソル積の例 (1)

$A=\begin{pmatrix}1 & -1\\0 & 2\end{pmatrix}$, $B=\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}$ としよう。

$$A\otimes B=\begin{pmatrix}\mathbf{1} & \mathbf{-1}\\\mathbf{0} & \mathbf{2}\end{pmatrix}\otimes\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}=\begin{pmatrix}\mathbf{1}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} & \mathbf{-1}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\\\mathbf{0}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} & \mathbf{2}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\end{pmatrix}$$

$$=\begin{pmatrix}1 & 2 & -1 & -2\\3 & 4 & -3 & -4\\0 & 0 & 2 & 4\\0 & 0 & 6 & 8\end{pmatrix}$$

行列のテンソル積の例 (2)

A$=\begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix}$, B$=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ としよう。

$$B\otimes A=\begin{pmatrix}\mathbf{1} & \mathbf{2}\\ \mathbf{3} & \mathbf{4}\end{pmatrix}\otimes\begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix}=\begin{pmatrix}\mathbf{1}\begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix} & \mathbf{2}\begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix}\\ \mathbf{3}\begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix} & \mathbf{4}\begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix}\end{pmatrix}$$

$$=\begin{pmatrix}1 & -1 & 2 & -2\\ 0 & 2 & 0 & 4\\ 3 & -3 & 4 & -4\\ 0 & 6 & 0 & 8\end{pmatrix}$$

テンソル積での内積の定義

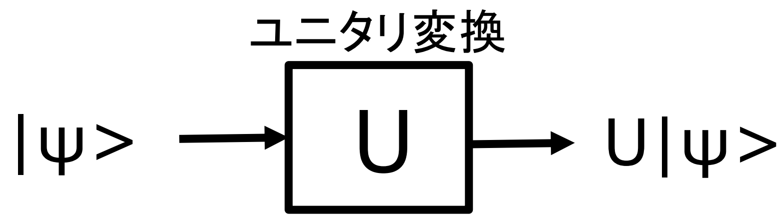
$$\left(\sum_i a_i|v_i\rangle \otimes |w_i\rangle, \sum_j b_j|v'_j\rangle \otimes |w'_j\rangle\right) \equiv \sum_{ij} a_i^* b_j \langle v_i|v'_j\rangle\langle w_i|w'_j\rangle.$$

量子ゲート

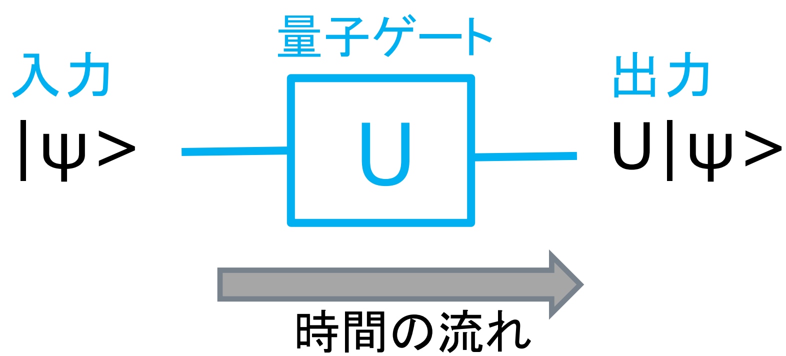
1-qubit ゲート

量子ゲート

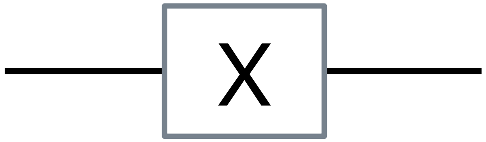
- 量子の状態 |ψ>は、ユニタリ変換Uの作用を受けて、状態 U|ψ> に変化する。
- この変化 |ψ> → U|ψ>を、次のように表そう。



- この時、Uを、|ψ> を入力、U|ψ> を出力とする回路と考えることができる。これを、「量子ゲート」と呼ぶ。



代表的な1-qubitのゲート



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

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

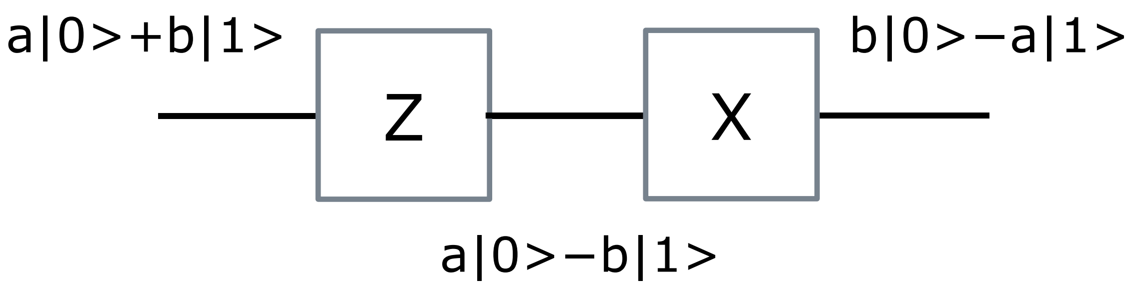
$\frac{1}{\sqrt{2}}(a+b)|0\rangle + \frac{1}{\sqrt{2}}(a-b)|1\rangle$

演習課題 2-3

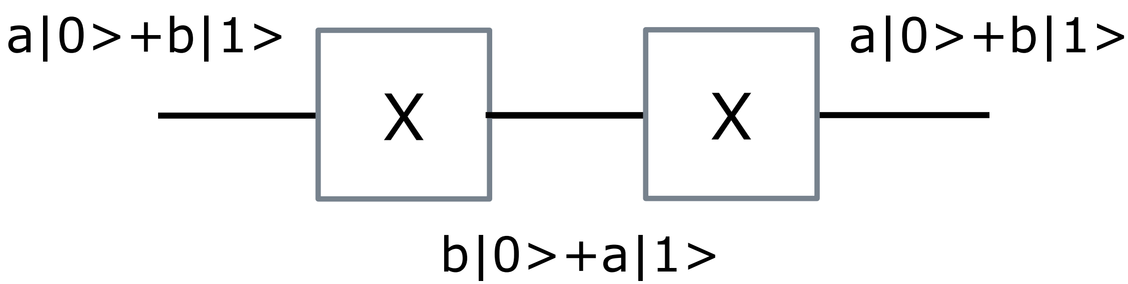
|0>,|1>基底から

|+>,|->基底への変換

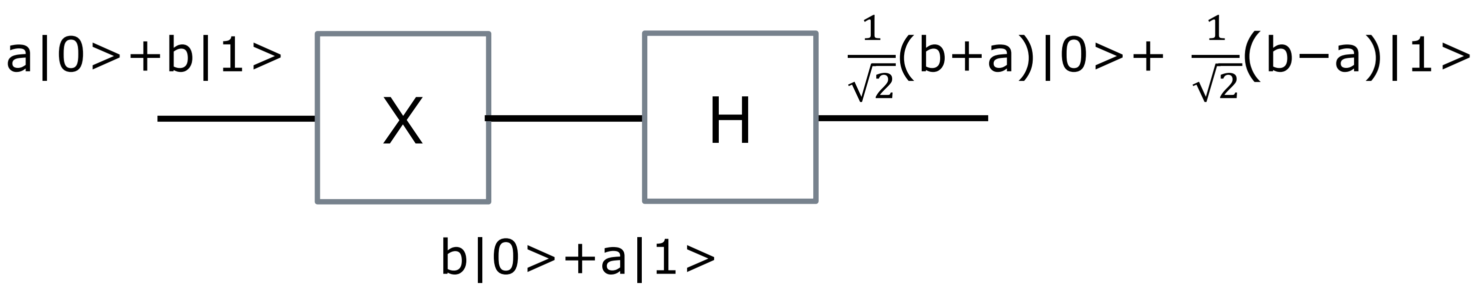
1-qubitのゲートを、直列に組み合わせる (1)



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



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



1-qubitのゲートを、直列に組み合わせる (2)

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$ とする。

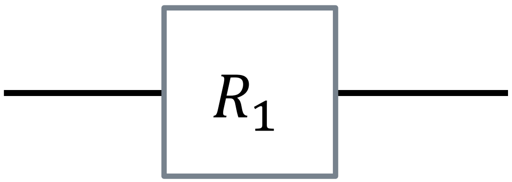
$$R_1 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

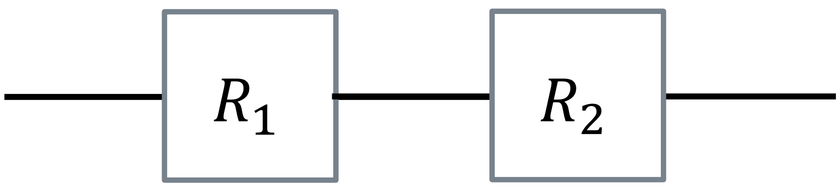
$$R_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} i \end{pmatrix}$$

$$R_4 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^4} \end{pmatrix} = \cdots$$

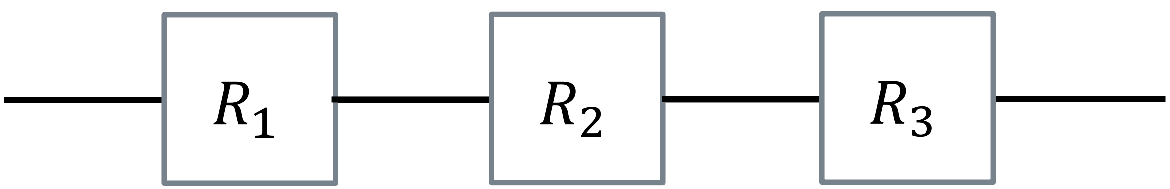
1-qubitのゲートを、直列に組み合わせる (2)



$$\underset{R_1}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^1} \end{pmatrix}}|\psi>$$



$$\underset{R_2}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}}\underset{R_1}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^1} \end{pmatrix}}|\psi>$$

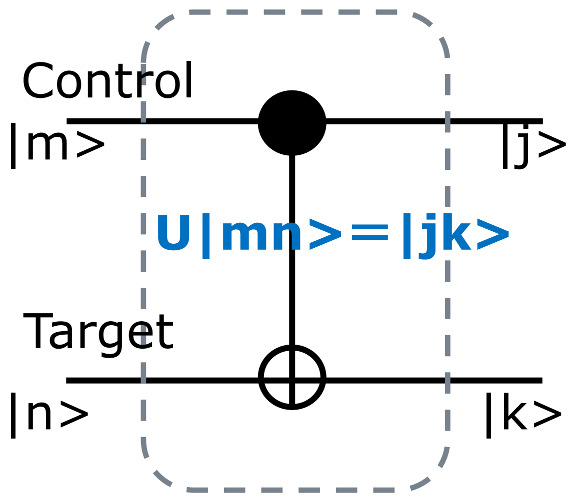


$$\underset{R_3}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix}}\underset{R_2}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}}\underset{R_1}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^1} \end{pmatrix}}|\psi>$$

Control-NOT

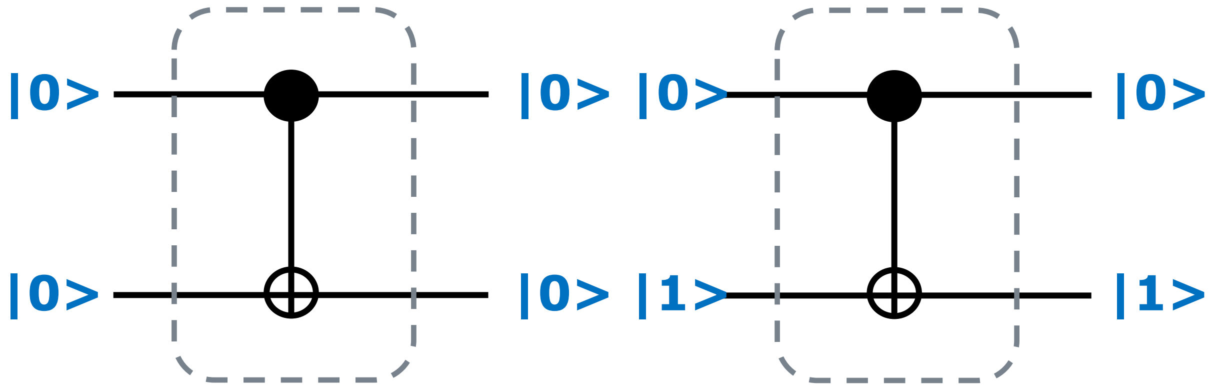
2-qubitのゲート
CNOT (Control-NOT)

Control が |1> の時
Target のNOTをとる

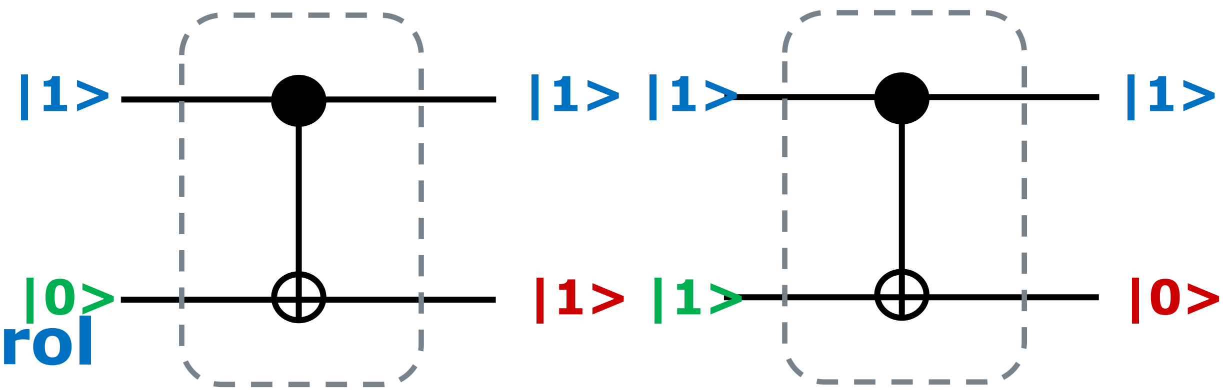


|mn>の
第一bit |m>がControl
第二bit |n>がTarget

Controlが |0>なら何もしない

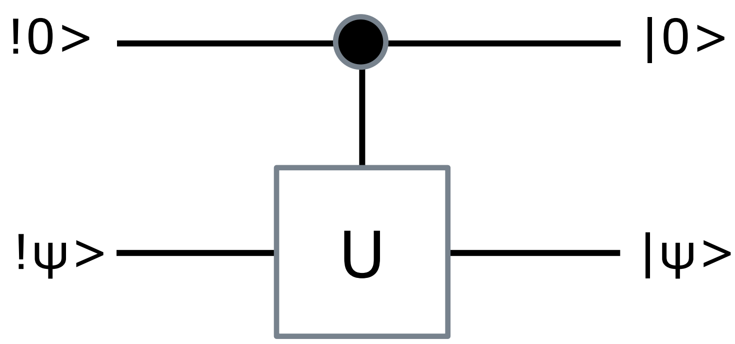


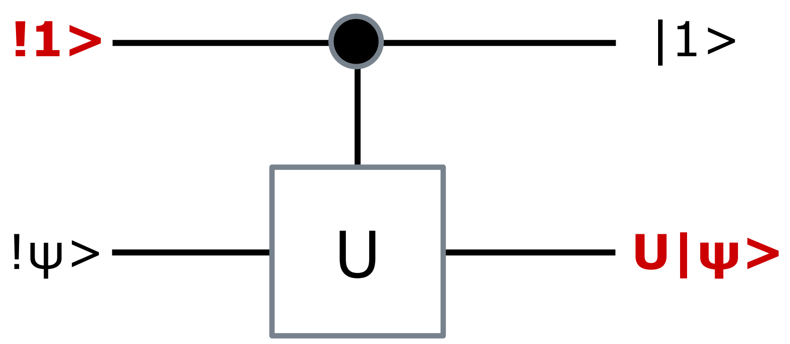
Controlが |1>ならNOT操作



2-qubitのゲート Control-U

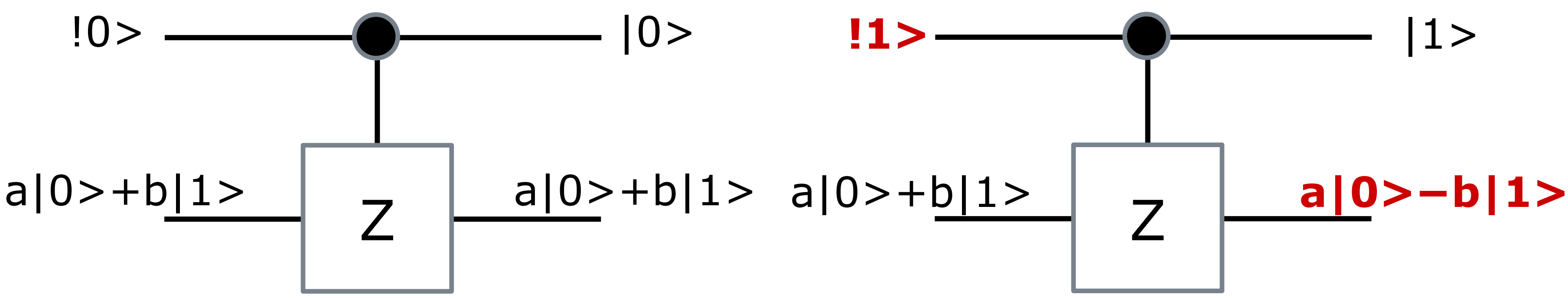
コントロール qubit が、|0> の時には、何もせず、（左図）
コントロール qubit が、|1> の時には、ターゲットのqubit |ψ> にユニタリ演算子 U を適用した U|ψ>を出力する（右図）
ゲートを、**Control-U ゲート**という。





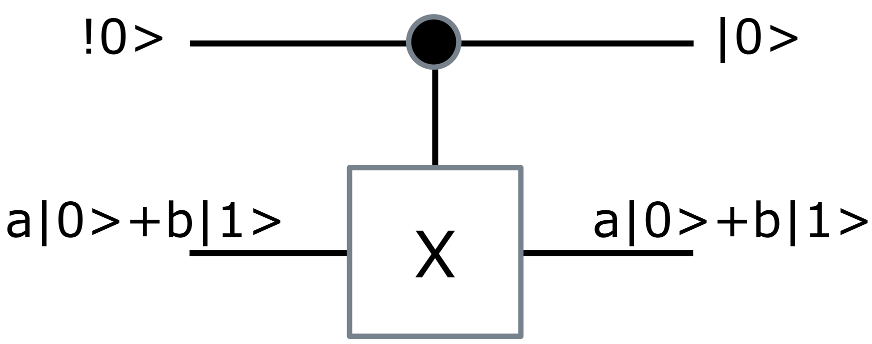
2-qubitのゲート Control-U の例 (1)
Control-Z

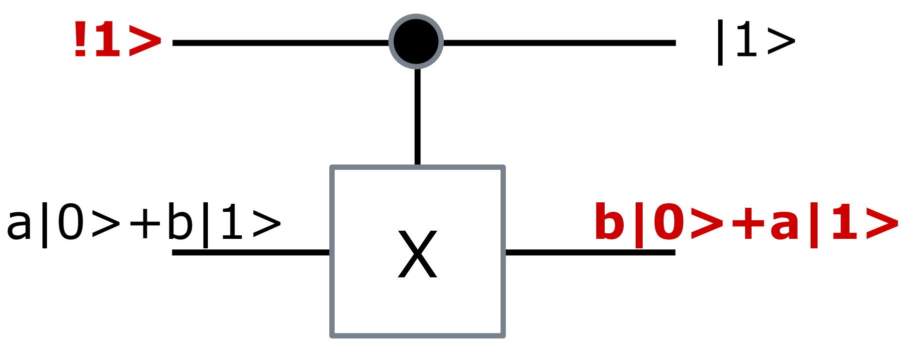
コントロール qubit が、|0> の時には、何もせず、（左図）
コントロール qubit が、|1> の時には、ターゲットのqubit |ψ> にユニタリ演算子 Z を適用した Z|ψ>を出力する（右図）



2-qubitのゲート Control-U の例 (2)
Control-X

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



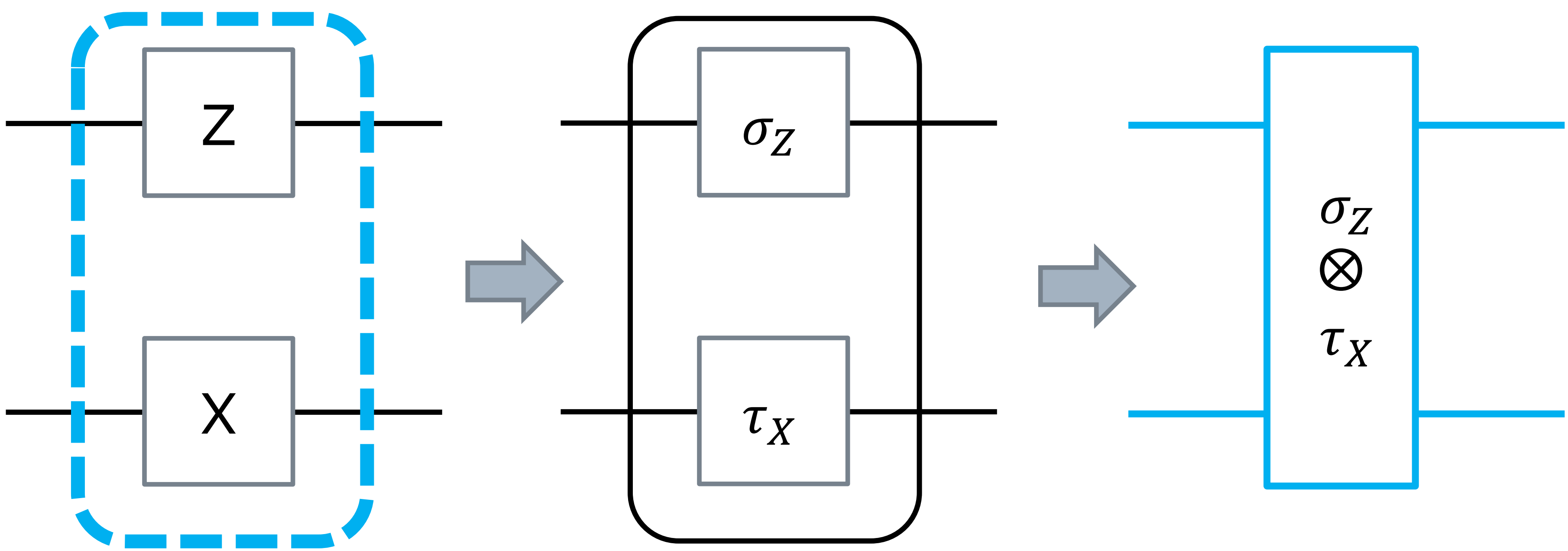


Control-Xは、CNOTと等しいことを確かめよ

ゲートを並列に組み合わせる

1-qubitのゲートを、並列に組み合わせる

二つの 1-qubit ゲートがある時、それらを並列に組み合わせて、2-qubitの量子ゲートを構成することができる。この量子ゲートは、それぞれの量子ゲートのユニタリ行列のテンソル積として作用する。



1-qubitのゲートを、並列に組み合わせる

2-qubitのシステムで、最初のqubitに、演算子 X, Y, Zを適用する演算子を、σ_x , σ_y , σ_z とし、二つ目のqubitに、演算子 X, Y, Zを適用する演算子を、τ_x , τ_y , τ_z とする。

$$\sigma_z \otimes \tau_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

$$\sigma_x \otimes \tau_z = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

1-qubitのゲートを、並列に組み合わせる

最初のqubitにZを作用させる

$(\sigma_z \otimes \tau_x)$ |01> = |00>

二つ目のqubitにZを作用させる

Z|0>=|0>, X|1>=|0>

結果は|0>⊗|0>=|00>

$$\begin{array}{cccc} & \sigma_Z \otimes \tau_X & |01\rangle & & |00\rangle \\ (\sigma_z \otimes \tau_x)\,|01\rangle = & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & = & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

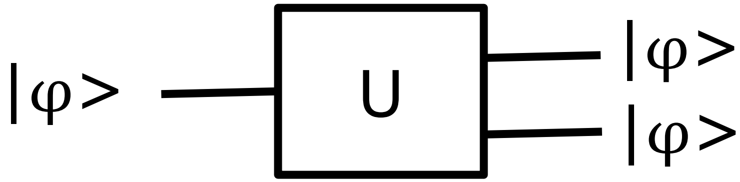
量子アルゴリズムの基礎

No Cloning 定理

未知の量子の状態を複数コピーすることはできない

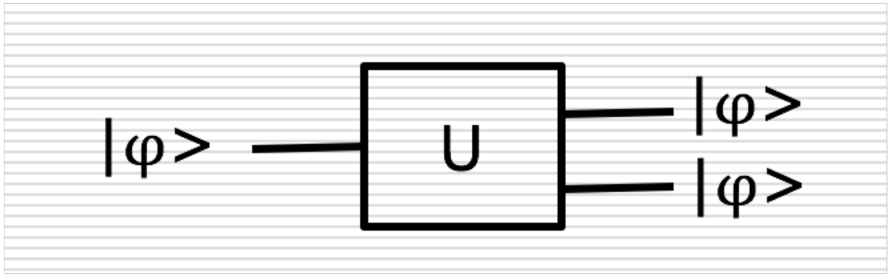
No Cloning 定理

- ある状態 $|\varphi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$に対して、$|\varphi\rangle \otimes |\varphi\rangle$ すなわち自己のコピーをもう一つ生成する回路を考えよう。実は、量子ゲートの世界では、こうした基本的な操作ができないのだ。これを、No Cloning 定理と呼ぶ。以下、それを説明しよう。



こうした回路は存在しない！

No Cloning 定理の証明



- 先のような回路(ユニタリ変換U)が存在したとする。
  その回路は、一般のφだけでなく、|0>, |1> に対しても働くので、

$$U|0\rangle = |00\rangle$$
$$U|1\rangle = |11\rangle$$

- $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ とすれば、

$$U|\psi\rangle = \alpha U|0\rangle + \beta U|1\rangle = \boxed{\alpha|00\rangle + \beta|11\rangle}$$

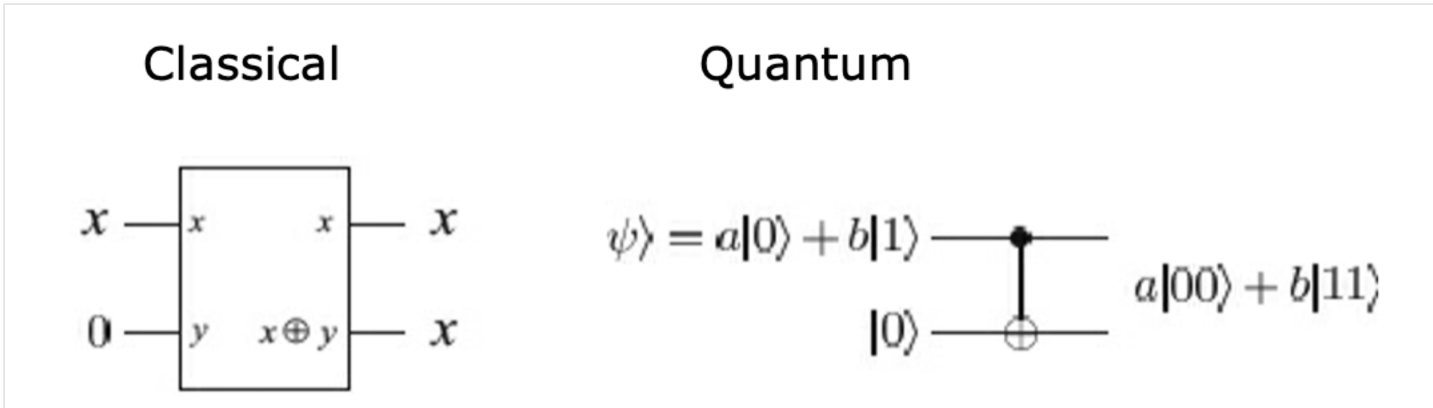
- Uは、|ψ>もコピーできるはずなので、

$$U(|\psi\rangle) = |\psi\rangle|\psi\rangle = \boxed{\alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle}$$

- しかし、この二つの条件を満たすα, βは、存在しない。

CNOTとコピー回路

CNOTに相当する古典回路は、古典情報をコピーする。



未知の状態 $|\Psi\rangle = a|0\rangle + b|1\rangle$ のあるqubitを、CNOTゲートを使ってコピーしようとしたとしよう。二つのqubitの入力の状態は、次のようにかける。

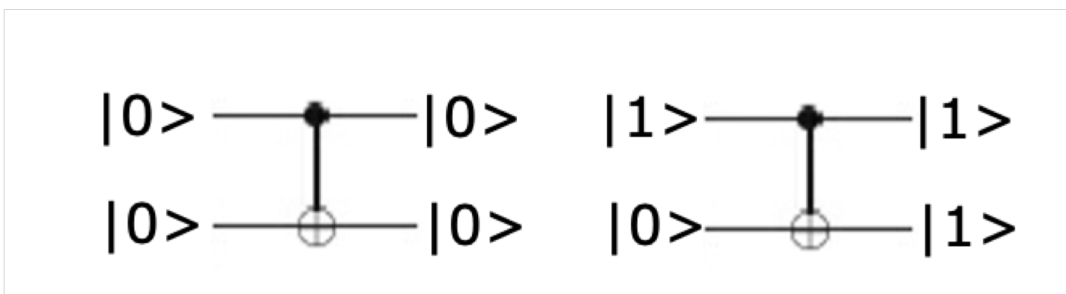
$$[a|0>+b|1>]\otimes|0> = a|00>+b|10>$$

CNOTの機能は、第一qubitが1の時、第二のqubitを否定することだから、この出力は、単純で、$a|00\rangle + b|11\rangle$となる。

$$\mathrm{CNOT}(a|00>+b|10>) = a|00>+b|11>$$

CNOTとコピー回路

この時、首尾よく |Ψ⟩をコピーできたであろうか？ すなわち、状態|Ψ⟩|Ψ⟩を生み出すことができたであろうか？
確かに、 |Ψ⟩ ＝ |0⟩ あるいは |Ψ⟩ ＝ |1⟩ の時には、この回路は、まさにそれをしている。CNOT回路を、|0> あるいは|1>にエンコードされた古典情報をコピーするのに利用することは可能なのだ。



CNOTとコピー回路

しかし、もっと一般的な状態 $|\Psi\rangle$ については、$|\Psi\rangle|\Psi\rangle$は次のようになる。

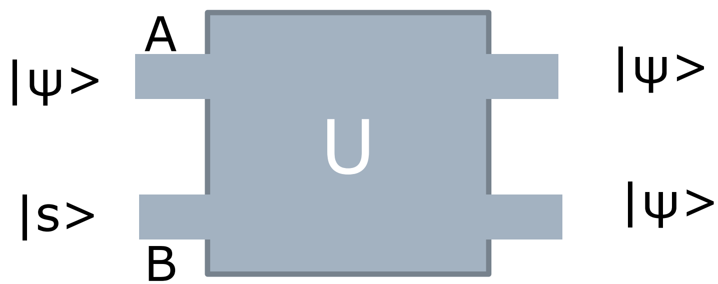
$$|\psi\rangle|\psi\rangle = a^2|00\rangle + ab|01\rangle + ab|10\rangle + b^2|11\rangle.$$

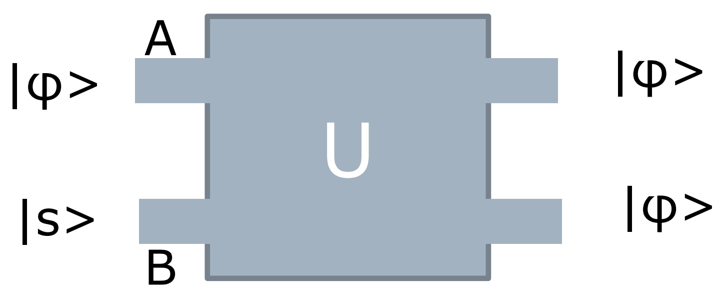
$a|00\rangle + b|11\rangle$ と比較すると、ab=0でなければ、CNOT回路は、入力の量子状態をコピーしていないことがわかる。

実際、未知の量子状態をコピーすることは、不可能だということがわかる。qubitはコピーされないというこの性質は、No-cloning定理として知られている、量子情報と古典情報の主要な違いの一つである。

もう少し、きちんとした証明

二つのスロットA, Bを持つ量子回路を考えよう。スロットAはデータスロットで、最初の状態では、未知のピュアな状態|φ> が与えられている。それが、ターゲットスロットのBにコピーされるとしよう。





もう少し、きちんとした証明

この時、次の式が成り立つ。

$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle$

$U(|\varphi\rangle \otimes |s\rangle) = |\varphi\rangle \otimes |\varphi\rangle$

両辺の内積を取る。

左辺は、$(\ U(|\psi\rangle \otimes |s\rangle), U(|\varphi\rangle \otimes |s\rangle)\ ) =$ Uは内積を保存するから

$= (\ |\psi\rangle \otimes |s\rangle, |\varphi\rangle \otimes |s\rangle) = \langle\psi|\varphi\rangle\langle s|s\rangle = \langle\psi|\varphi\rangle$

右辺は、$(\ |\psi\rangle \otimes |\psi\rangle, |\varphi\rangle \otimes |\varphi\rangle) = \langle\psi|\varphi\rangle\ \langle\psi|\varphi\rangle =$

$= (\langle\psi|\varphi\rangle)^2$

よって、 $\langle\psi|\varphi\rangle == (\langle\psi|\varphi\rangle)^2$

テンソル積の内積の定義

$$\left(\sum_i a_i |v_i\rangle \otimes |w_i\rangle, \sum_j b_j |v'_j\rangle \otimes |w'_j\rangle\right) \equiv \sum_{ij} a_i^* b_j \langle v_i|v'_j\rangle \langle w_i|w'_j\rangle.$$

もう少し、きちんとした証明

ただ、$<\psi|\varphi>=(<\psi|\varphi>)^2$ ということだが、$x = x^2$ には、二つの解しかない。$x = 0$ と $x = 1$である。だから、$<\psi|\varphi>=0$ であるか、$<\psi|\varphi>=1$のいずれかということになる。

$|\psi> = |\varphi>$ であるか、 $|\psi>$ と $|\varphi>$ が直交するかのどちらかである。こうして、クローニング回路は、もし存在するとしても、お互いに直交する状態しかコピーできない。

よって、一般的な量子状態をコピーするマシンは、不可能である。

同時にこのことは、なぜ、0と1の古典ビットからなる古典情報がコピー可能なのかの説明にもなっている。

At first sight the no-cloning theorem appears rather peculiar. After all, isn’t classical physics a special case of quantum mechanics? How is it possible that we can copy classical information if we can’t copy quantum states? The answer is that the no-cloning theorem does not prevent all quantum states from being copied, it simply says that nonorthogonal quantum states cannot be copied. More precisely, suppose and are two non-orthogonal quantum states. Then the no-cloning theorem implies that it is not possible to build a quantum device that, when input with $|\psi>$ or $|\varphi>$ will output two copies of the input state, $|\psi\psi>$ or $|\varphi\varphi>$.

On the other hand, if $|\psi>$ and $|\varphi>$ are orthogonal, then the no-cloning theorem doesn't prohibit their cloning. Indeed, it is rather easy to design quantum circuits which copy such states!

This observation resolves the apparent contradiction between the no-cloning theorem and the ability to copy classical information, for the different states of classical information can be thought of merely as orthogonal quantum states.

Nielsen, Michael A..
Quantum Computation and Quantum Information (p.531).
Cambridge University Press.

Entanglement

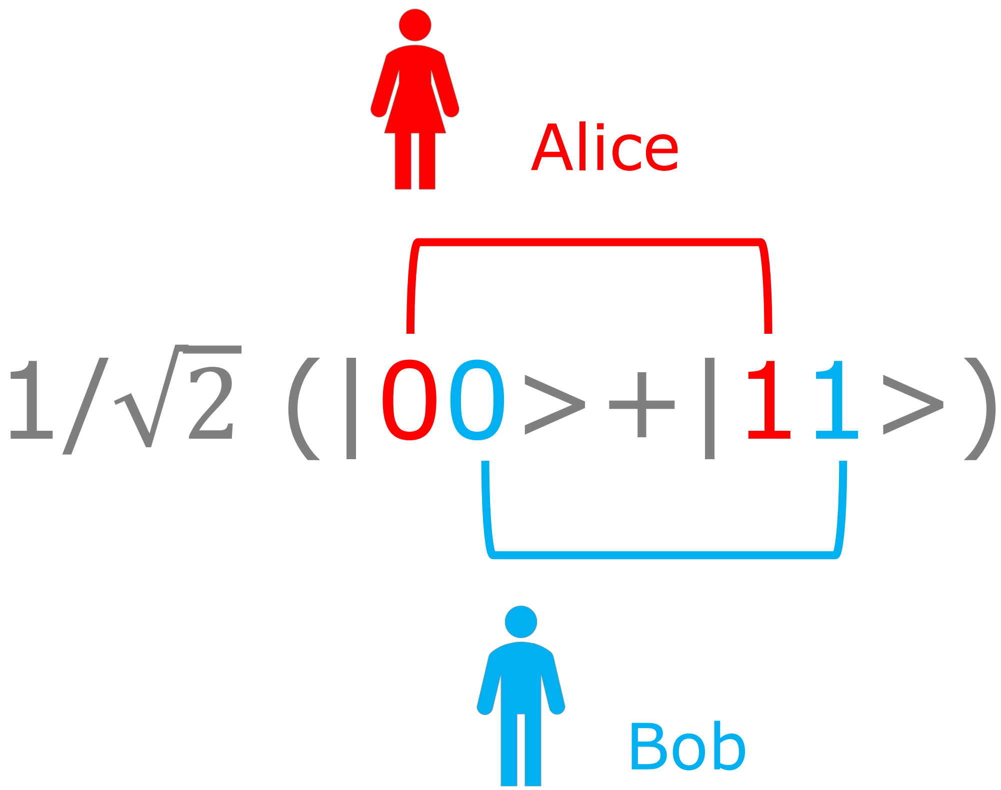
二つの量子の不思議な絡み合い

エンタングルメント：
二つの状態のテンソル積に表せない状態

Entanglement もつれ合い
二つの状態のテンソル積に分解できない状態

- ところで、全ての Two qubitsの状態は、二つのqubitの状態のテンソル積に、分解できるだろうか？
- 次のTwo qubitの状態を考えよう。
  $1/\sqrt{2}\,(|00\rangle+|11\rangle)$
  ところが、この状態は、二つのqubitのテンソル積では表現できないことが、次のようにしてわかる。
- $1/\sqrt{2}\,(|00\rangle+|11\rangle)$ が、$(a|0\rangle+b|1\rangle)\otimes(c|0\rangle+d|1\rangle)$ と二つの状態のテンソル積に分解できたとしよう。
  $$(a|0\rangle+b|1\rangle)\otimes(c|0\rangle+d|1\rangle)$$
  $$= ac|00\rangle+ad|01\rangle+bc|10\rangle+bd|11\rangle$$
- 両辺の係数を比較すると、$ad=bc=0$となって、
  $ac=bd=1/\sqrt{2}$ となる a,b,c,d が存在しないことがわかる。
  こうした 2-qubitの状態を、「もつれ合い」と呼ぶ。

EPRペア：もつれ合った二つのqubit

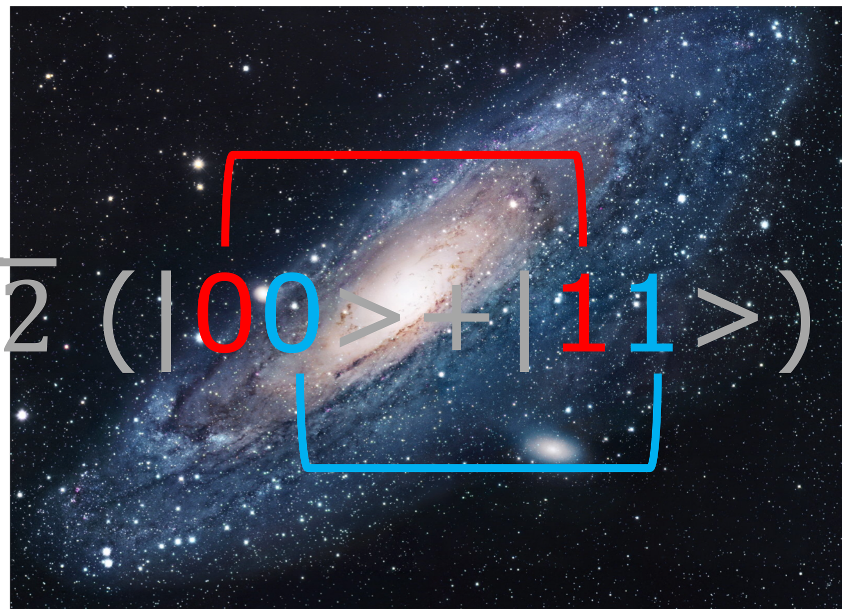


$1/\sqrt{2}$ (|00>+|11>) で表される状態は、二つのqubitの状態である。
一方のqubitをAliceが、他方のqubitをBobが持つことができる。
こうした、もつれ合った二つのqubitを、EPRペアと呼ぶ。

EPRペア：もつれ合った二つのqubit

Alice

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



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

Bob

Aliceが観察できるのは、$|00>+|11>$の第一bitで、
Bobが観察できるのは、$|00>+|11>$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りない。

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

EPRペアのビットを観測する

Aliceが自分の持つEPRペア、$1/\sqrt{2}$ (|00>+|11>)の最初のビットを観測したとしよう。それが1である確率は1/2で、0である確率は1/2である。

今、その結果が0であったとしよう。この観測の結果、新しい状態は、|00>に変わる。それは、第二ビットの観測が、0である確率が 1 であることを意味する。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。

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

最初の観測結果が1であったとしても、今度は、新しい状態が、|11>になるので、第二ビットの観察は、100%の確率で、1を返すことになる。

EPRペアのビットを観測する

Aliceの最初の観測結果が1であったとしよう。今度は、新しい状態が、|11>になるので、第二ビットの観察は、100%の確率で、1を返すことになる。

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

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？

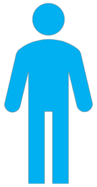
実際、アインシュタインは、こうした現象は「馬鹿げた遠隔作用」だと言った。

EPRペア：もつれ合った二つのqubit

Φ^+

Alice

$$1/\sqrt{2}\ (|{\color{red}0}{\color{cyan}0}> + |{\color{red}1}{\color{cyan}1}>)$$



Bob

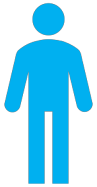
こうした性質を持つペアは、$1/\sqrt{2}\ (|00>+|11>)$ だけではない。

EPRペア：もつれ合った二つのqubit

Φ^-

Alice

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



Bob

こうした性質を持つペアは、$1/\sqrt{2}\ (|00\rangle + |11\rangle)$ だけではない。

EPRペア：もつれ合った二つのqubit

Ψ^+

Alice

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

Bob

こうした性質を持つペアは、$1/\sqrt{2}\ (|00\rangle + |11\rangle)$ だけではない。

EPRペア：もつれ合った二つのqubit

Ψ^-

Alice

$$1/\sqrt{2}\,(|01\rangle - |10\rangle)$$

Bob

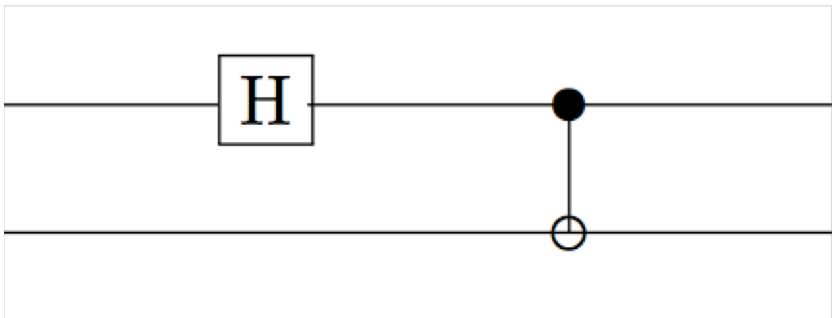
こうした性質を持つペアは、$1/\sqrt{2}\,(|00\rangle+|11\rangle)$ だけではない。

Bell State ゲート

奇妙に見えるEPRペアだが、それらは二つの量子ゲートで、簡単に生成できる！

Bell States ゲート

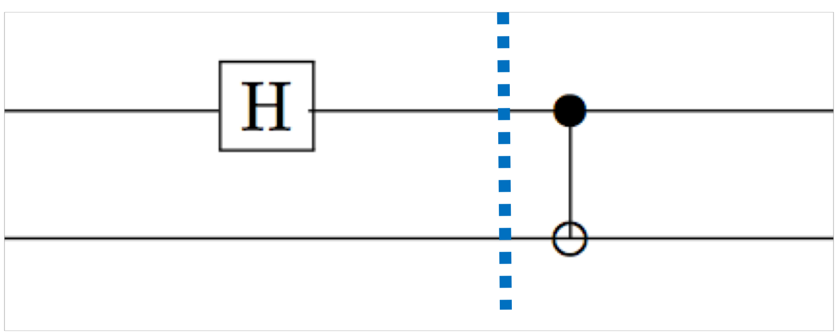
- 次のような回路を考える。



- この回路に |00>を入力として与える。点線のところまで考える

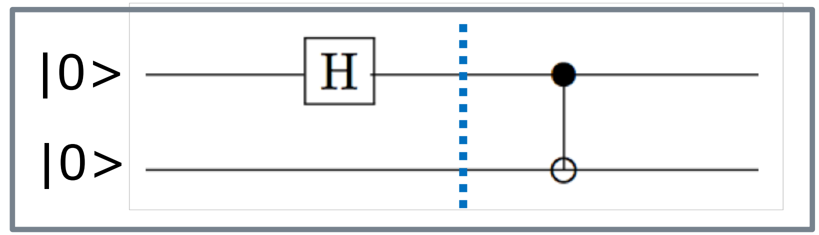
第一qubit |0>

第二qubit |0>



|0> ⊗ |0> = |00>

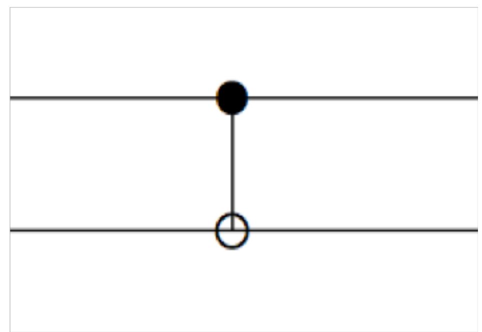
Bell States ゲート
入力が|00>の場合



□ Hは、第一qubitの |0> を $1/\sqrt{2}$ (|0>+|1>)に変える。

第一qubit $1/\sqrt{2}$ (|0>+|1>)

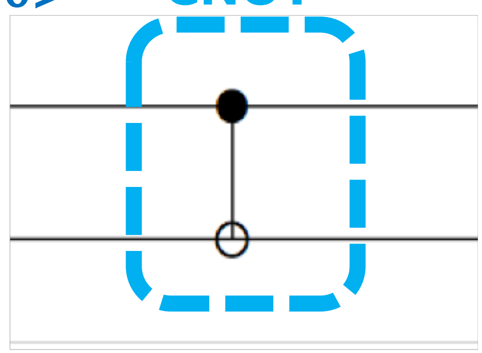
第二qubit |0>



(|0>+|1>) ⊗|0> =|00>+|10>

CNOT

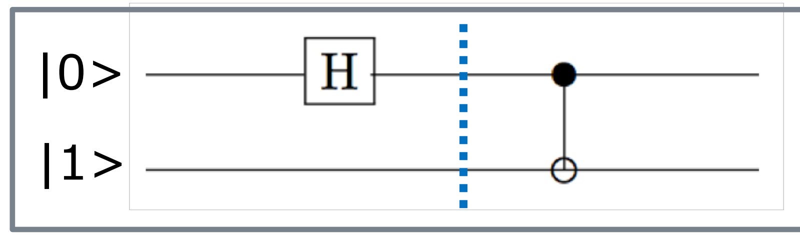
$1/\sqrt{2}$ (|00>+|10>)



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

Φ^{+}

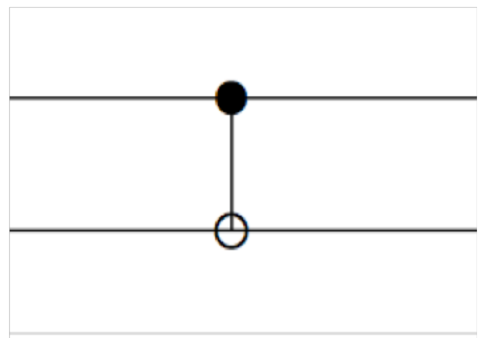
Bell States ゲート
入力が|01>の場合



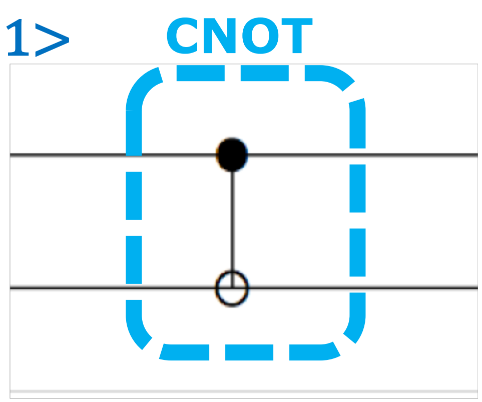
□ Hは、第一qubitの |0> を $1/\sqrt{2}$ (|0>+|1>)に変える。

第一qubit $1/\sqrt{2}$ (|0>+|1>)

第二qubit |1>



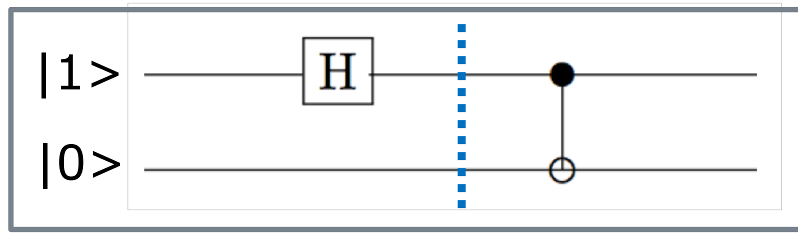
(|0>+|1>) ⊗|1> =|01>+|11>



$1/\sqrt{2}$ (|01>+|11>) → $1/\sqrt{2}$ (|01>+|10>)

Ψ^+

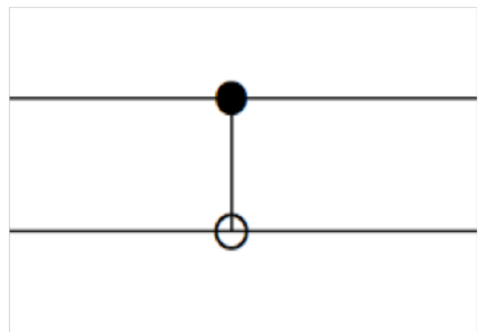
Bell States ゲート
入力が|10>の場合



□ Hは、第一qubitの |1> を $1/\sqrt{2}$ (|0>−|1>)に変える。

第一qubit $1/\sqrt{2}$ (|0>−|1>)

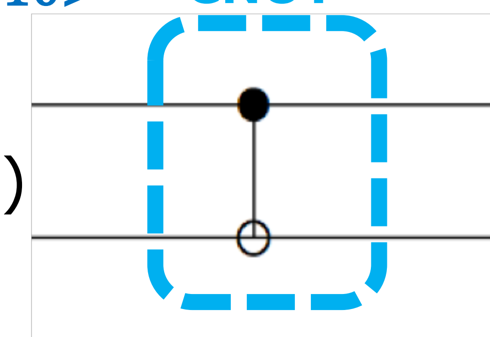
第二qubit |0>



(|0>−|1>) ⊗|0> =|00>−|10>

CNOT

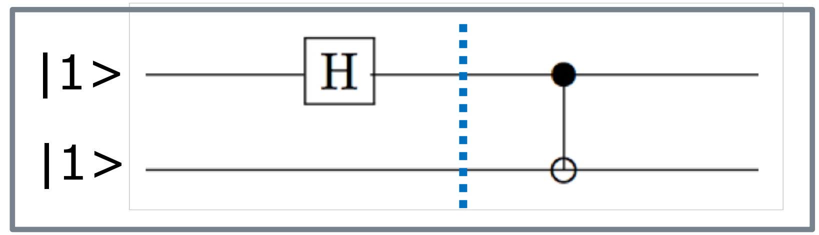
$1/\sqrt{2}$ (|00>−|10>)



$1/\sqrt{2}$ (|00>−|11>)

Φ^{-}

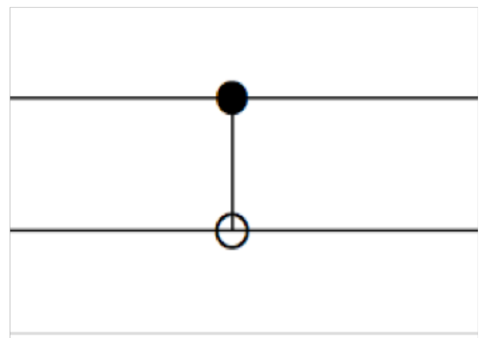
Bell States ゲート
入力が|11>の場合



□ Hは、第一qubitの |1> を $1/\sqrt{2}$ (|0>−|1>)に変える。

第一qubit $1/\sqrt{2}$ (|0>−|1>)

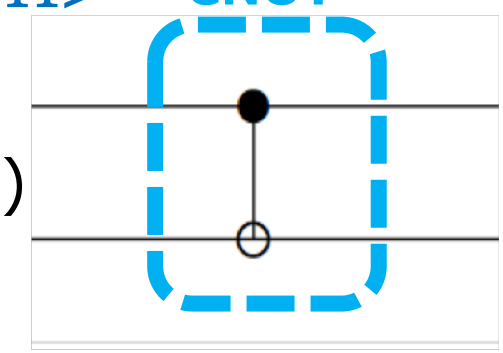
第二qubit |1>



(|0>−|1>) ⊗|1> =|01>−|11>

CNOT

$1/\sqrt{2}$ (|00>−|11>)



$1/\sqrt{2}$ (|01>−|10>)

Ψ^-

Bell State Gateから出力される次の四つの状態は、直交正規基底である。

$$\left|\Phi^{\pm}\right\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right)$$
$$\left|\Psi^{\pm}\right\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle \pm |10\rangle\right)$$

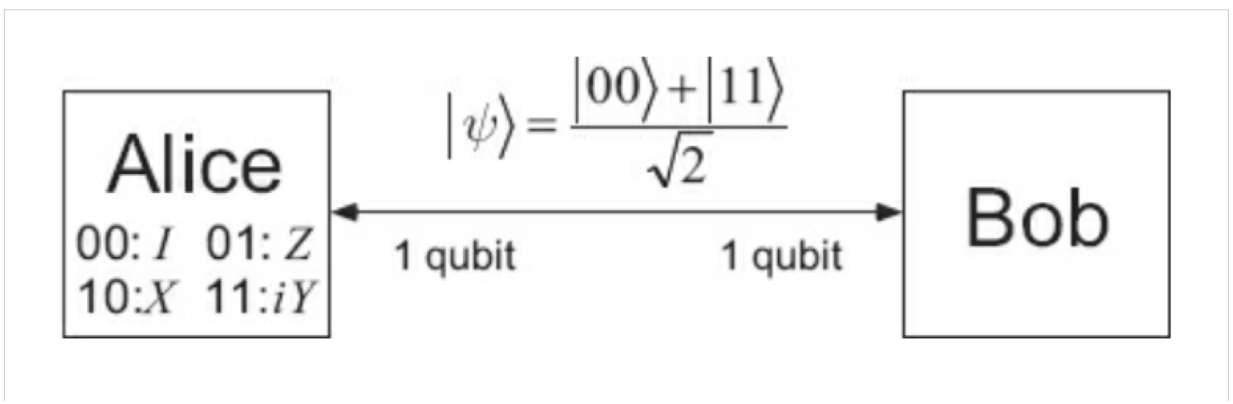
$$< \Phi^{\pm} | \Phi^{\pm} > = \frac{1}{2}(<00| \pm <11|)\cdot(|00> \pm |11>$$
$$= \frac{1}{2}(<00|00> \pm <00|11> \pm <11|00> + <11|11>$$
$$= \frac{1}{2}(\ 1 \pm 0 \pm 0 + 1) = 1$$
$$< \Phi^{\pm} | \Phi^{\mp} > = \frac{1}{2}(<00| \pm <11|)\cdot(|00> \mp |11>$$
$$= \frac{1}{2}(<00|00> \mp <00|11> \pm <11|00> - <11|11>$$
$$= \frac{1}{2}(\ 1 \mp 0 \pm 0 - 1) = 0$$

Super Dense Coding

一つのqubitで、二つの古典ビットをコードして送ることができる

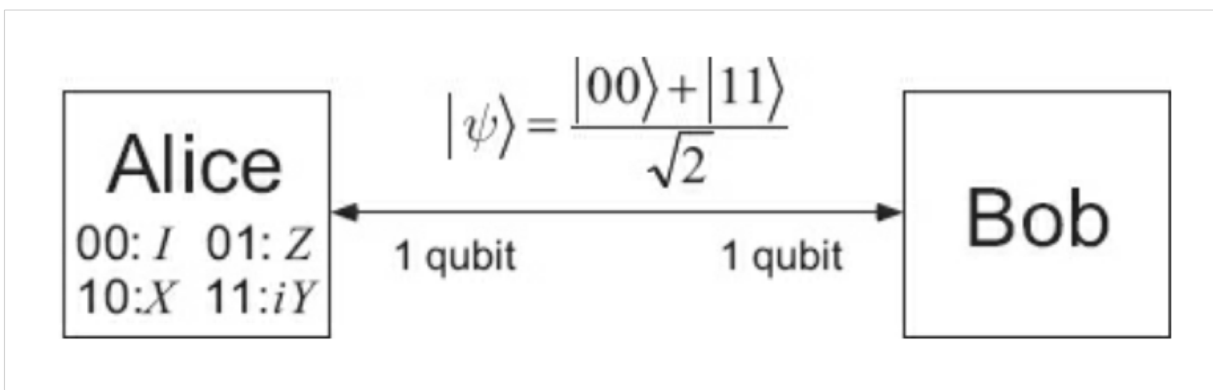
Superdense coding

- Superdense codingというのは、あらかじめ用意されたエンタングルメントを利用して、AliceからBobに、一つのqubitを送るだけで、二つの古典ビットを送ることができることをいう。



- AliceとBobは、次のようなエンタングルした状態を共有しているとする。第一ビットがAliceのqubit, 第二qubitがBobのqubitとしよう。

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$



- もし、ビット列′00′を送りたいのなら何もしない。もし、′01′を送りたいのなら、自分のqubitにフェーズ・フリップのZを適用する。′10′だったら、量子NOTゲートのXを、′11′だったら、iYを適用する。結果は次のようになる。

I $\quad 00 : |\psi\rangle \rightarrow \dfrac{|00\rangle + |11\rangle}{\sqrt{2}}$ $\quad \Phi^+$

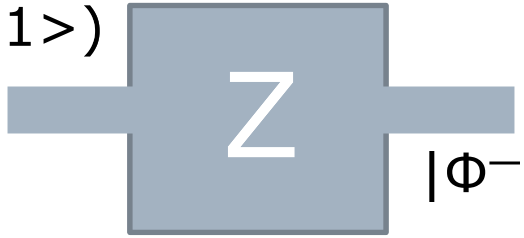
Z $\quad 01 : |\psi\rangle \rightarrow \dfrac{|00\rangle - |11\rangle}{\sqrt{2}}$ $\quad \Phi^-$

X $\quad 10 : |\psi\rangle \rightarrow \dfrac{|10\rangle + |01\rangle}{\sqrt{2}}$ $\quad \Psi^+$

iY $\quad 11 : |\psi\rangle \rightarrow \dfrac{|01\rangle - |10\rangle}{\sqrt{2}}.$ $\quad \Psi^-$

$Z|\Phi^{+}> \rightarrow \Phi^{-}$

$|\Phi^{+}> = 1/\sqrt{2}\ (|00> + |11>)$

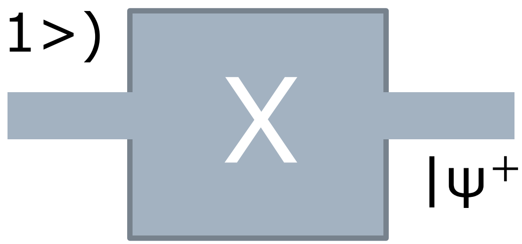


$|\Phi^{-}> = 1/\sqrt{2}\ (|00> - |11>)$

Phase Flip

$X|\Phi^{+}> \rightarrow \psi^{+}$

$|\Phi^{+}> = 1/\sqrt{2}\ (|00> + |11>)$

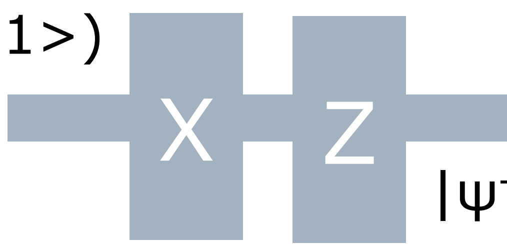


$|\psi^{+}> = 1/\sqrt{2}\ (|10> + |01>)$

Bit Flip

$ZX|\Phi^{+}> \rightarrow \psi^{-}$

$|\Phi^{+}> = 1/\sqrt{2}\ (|00> + |11>)$



$|\psi^{-}> = 1/\sqrt{2}\ (|10> - |01>)$

- この四つの状態は、Bell基底として知られているものである。AliceがBobにこの手順でqubitを送ったとすれば、Bobは、Bell基底で測定すれば、Aliceが送った四つの可能なビット列を決定できる。

- まとめれば、Aliceは、ただ一つのqubitと相互作用するだけで、Bobに2ビットの情報を送ることができる。もちろん、このプロトコルでは、二つのqubitが利用されているのだが、Aliceは、二番目のqubitとは、相互作用する必要はない。

- Even more interesting are the results in distributed quantum computation. Imagine you have two computers networked, trying to solve a particular problem. How much communication is required to solve the problem? Recently it has been shown that quantum computers can require exponentially less communication to solve certain problems than would be required if the networked computers were classical!

- Unfortunately, as yet these problems are not especially important in a practical setting, and suffer from some undesirable technical restrictions. A major challenge for the future of quantum computation and quantum information is to find problems of real-world importance for which distributed quantum computation offers a substantial advantage over distributed classical computation.

Nielsen, Michael A..
Quantum Computation and Quantum Information (p.9).
Cambridge University Press.

Quantum Teleportation

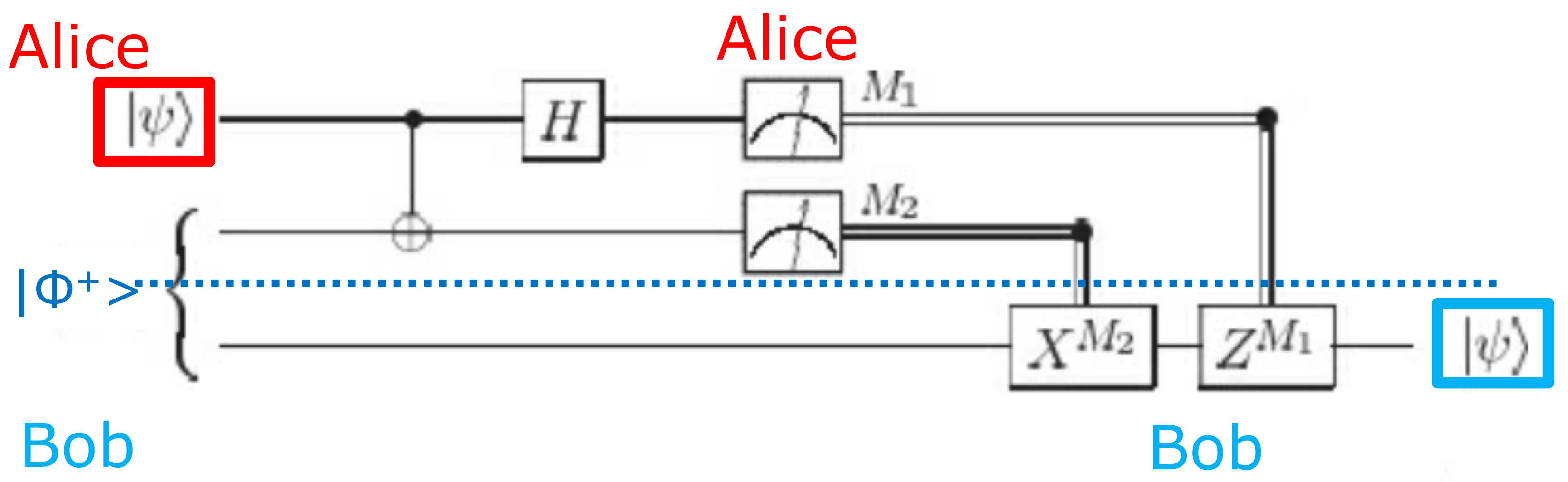
量子を、その状態を知ることなく、遠くに伝達できる

- ここでは、量子テレポーテーションがどう働くのかを見ておこう。AliceとBobは、ずいぶん昔にあっていたのだが、今は、遠くに離れて住んでいる。一緒にいた時に、EPRペアを作って、別れる時に、それぞれがEPRペアの一つのqubitを持つようにした。数年経ってから、Bobは遠くにいるのだが、Aliceのミッションは受け取ったqubit |ψ>をBobに送ることだった。彼女は、qubitの状態を知らない。さらに、Bobには、古典的な手段でしか情報を送れない。この時、Aliceは、このミッションを遂行できるだろうか？

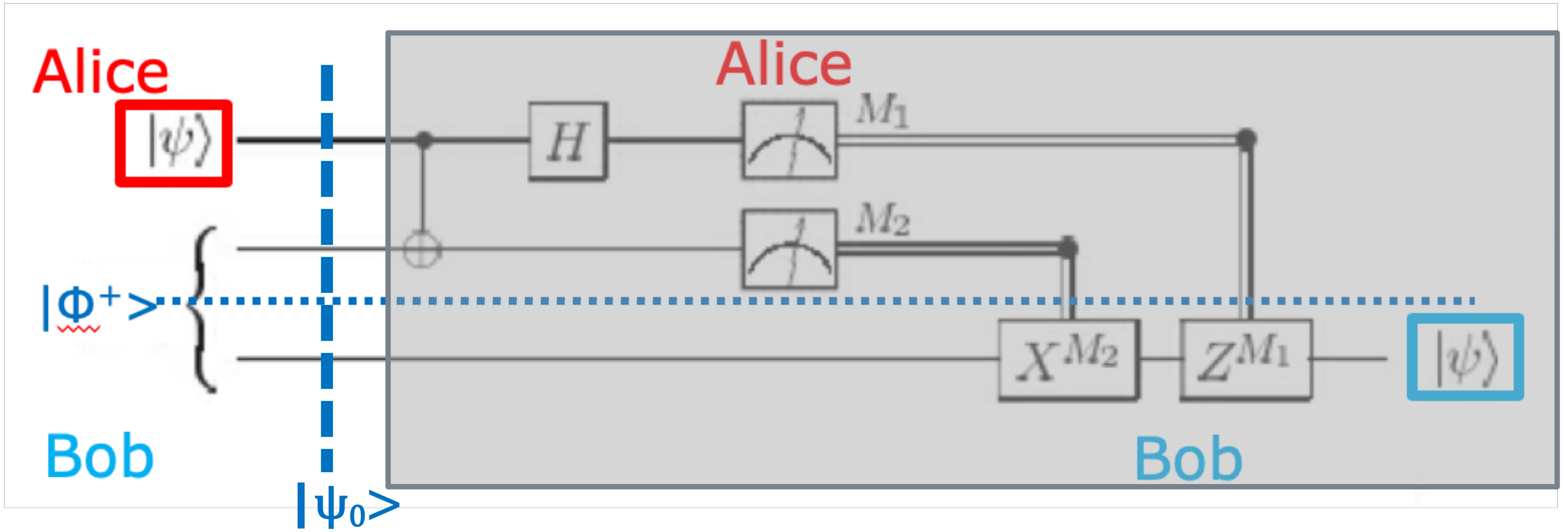
- 直感的には、Aliceはかなり分が悪いように見える彼女は、Bobに送らなければならないそのqubit $|\Psi\rangle$ の状態を知らないし、量子力学の法則は、たった一つのコピーを持っているだけの状態を決めることを妨げるだろう。もっと悪いことに、たとえ彼女が$|\Psi\rangle$の状態を知っていたとしても、$|\Psi\rangle$の状態は、連続空間上に値をとるわけで、それを正確に記述するには、無限の量の古典情報が必要になる。それをBobに伝えるには無限の時間が必要になる。
- Aliceにとって幸いなことに、量子テレポーテーションは、エンタングルしたEPRペアを利用して、ほんの少しの古典的コミュニケーションの手間だけで、Bobに $|\Psi\rangle$ の状態を送ることを可能にするのである。

- Aliceは、彼女の持つEPRペアの片方に作用して、彼女が持つ二つのqubitを測定して、四つの可能な古典的結果 00, 01, 10, 11 を得る。
- 彼女は、この情報を、古典的な手段でBobに送る。
- Aliceの古典的なメッセージに基づいて、Bobは四つのうち一つの操作を選んで彼のEPRペアの片方に適応する。
- 驚くべきことに、それだけで、Bobはオリジナルの状態 $|\Psi\rangle$ を復元できるのである。

- Aliceの出力の測定に従って、Bobのqubitはこれらの四つの可能な状態をとる。もちろん、どの状態にあるかを知るためには、BobはAliceの測定の結果を知らされていなければならない。この事実によって、情報の伝達が光より速くなることは避けられることになる。いったんBobが測定の結果を知れば、Bobはその状態を、適当な量子ゲートを適応して「修正」して|Ψ⟩を復活できる。
- 例えば、測定の結果が00であれば、Bobは何もする必要はない。測定が01の時には、Xゲートを適用すればいい。10ならZゲートを適用すればいい。もし、11ならば、最初にXゲートを、続いてZゲートを適用すればいい。

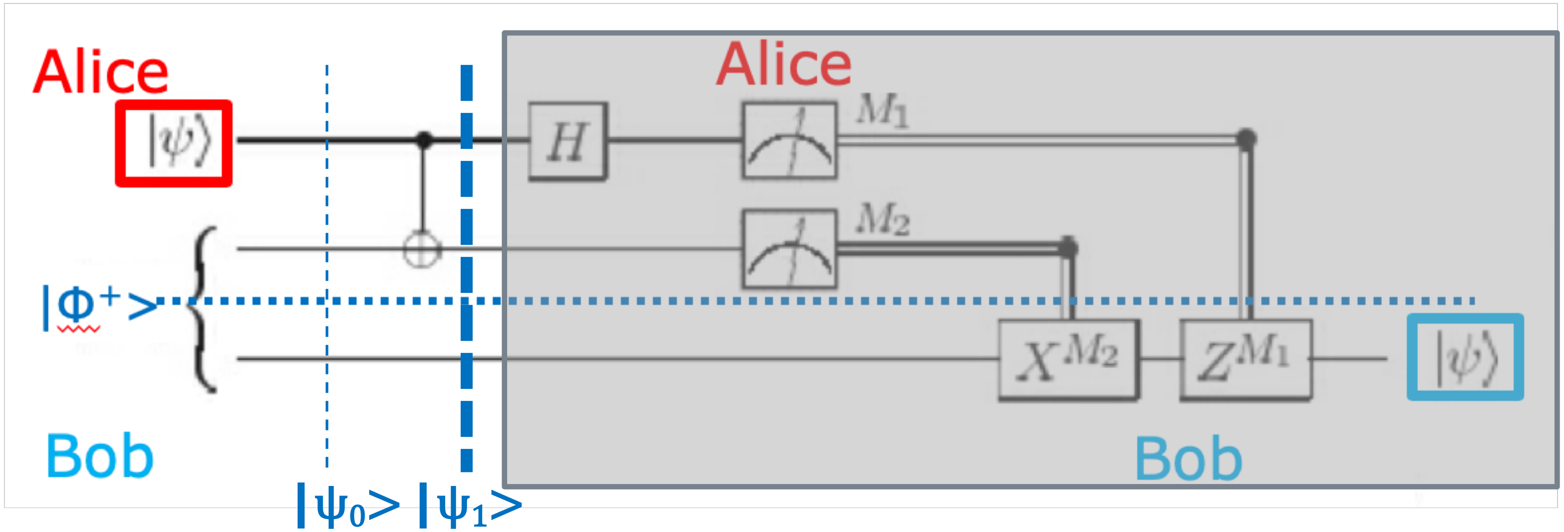


Alice
$|\psi\rangle$
H
Alice
M_1
M_2
$|\Phi^+>$
X^{M_2}
Z^{M_1}
$|\psi\rangle$
Bob
Bob



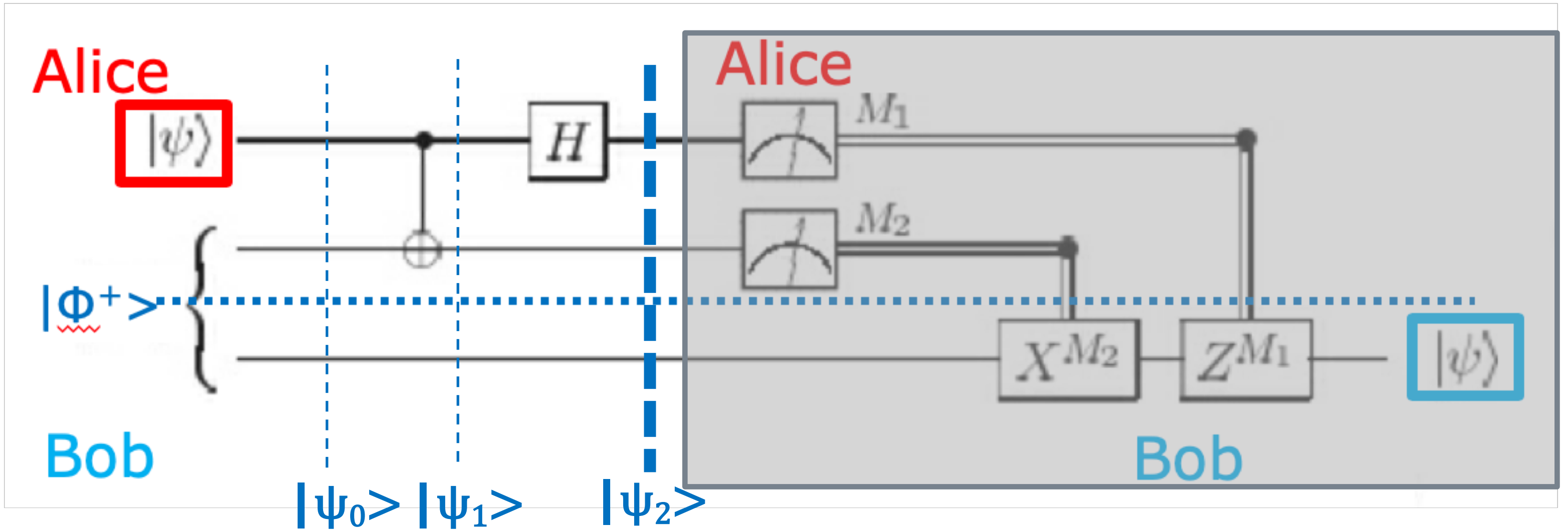
$|\psi> = \alpha|0>+\beta|1>$, $|\Phi^+> = 1/\sqrt{2}\,(|00> + |11>)$ とすると。

$$
\begin{aligned}
|\psi_0> = {} & |\psi> \otimes |\Phi^+> \\
= {} & (\alpha|0>+\beta|1>) \otimes 1/\sqrt{2}\,(|00> + |11>) \\
= {} & 1/\sqrt{2}\,(\alpha|0> \otimes(|00> +|11>)+\beta|1>\otimes(|00> + |11>))
\end{aligned}
$$



$$|\psi_0\rangle = 1/\sqrt{2}\,(\alpha|0\rangle \otimes(|00\rangle + |11\rangle)+\beta|1\rangle\otimes(|00\rangle + |11\rangle))$$

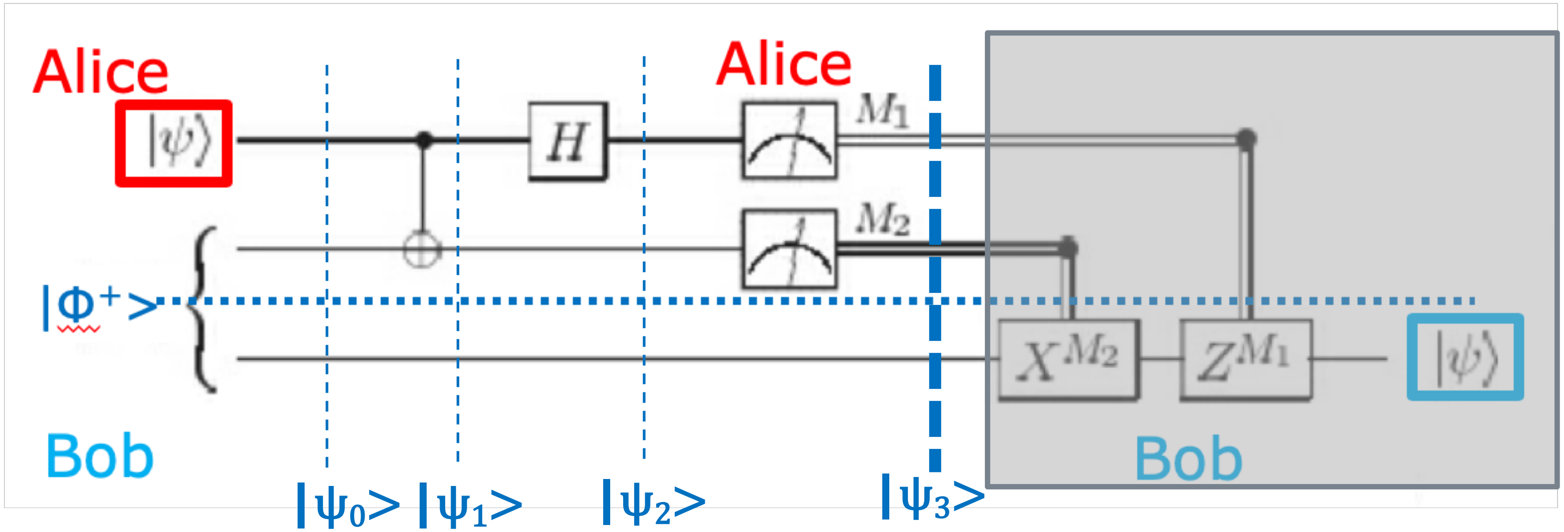
$$|\psi_1\rangle = 1/\sqrt{2}\,(\alpha|0\rangle \otimes(|00\rangle + |11\rangle)+\beta|1\rangle\otimes(|10\rangle + |01\rangle))$$



$$|\psi_0\rangle = 1/\sqrt{2}\,(\alpha|0\rangle \otimes(|00\rangle + |11\rangle) + \beta|1\rangle \otimes(|00\rangle + |11\rangle))$$

$$|\psi_1\rangle = 1/\sqrt{2}\,(\alpha|0\rangle \otimes(|00\rangle + |11\rangle) + \beta|1\rangle \otimes(|10\rangle + |01\rangle))$$

$$\begin{aligned}
|\psi_2\rangle &= 1/\sqrt{2}\,(\alpha(|0\rangle+|1\rangle)\otimes(|00\rangle + |11\rangle) + \\
&\qquad \beta(|0\rangle-|1\rangle)\otimes(|10\rangle + |01\rangle)\ ) \\
&= 1/\sqrt{2}\,(\alpha(|000\rangle+|011\rangle+|100\rangle+|111\rangle) + \\
&\qquad \beta(|010\rangle+|001\rangle-|110\rangle-|101)\ ) \\
&= 1/\sqrt{2}\,(|00\rangle(\alpha|0\rangle+\beta|1\rangle)+|01\rangle(\alpha|1\rangle+\beta|0\rangle) + \\
&\qquad |10\rangle(\alpha|0\rangle-\beta|1\rangle)+|11\rangle\,(\alpha|1\rangle-\beta|0\rangle)\ )
\end{aligned}$$

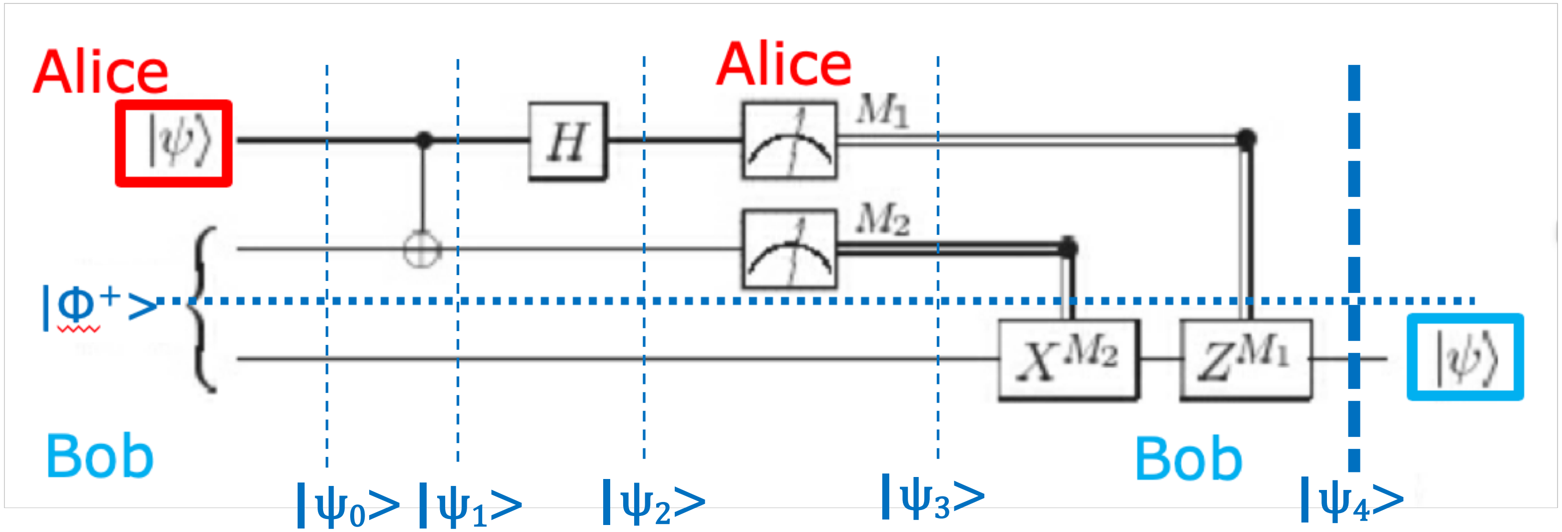


$$|\psi_0> = 1/\sqrt{2}\,(\alpha|0>\otimes(|00>+|11>)+\beta|1>\otimes(|00>+|11>))$$

$$|\psi_1> = 1/\sqrt{2}\,(\alpha|0>\otimes(|00>+|11>)+\beta|1>\otimes(|10>+|01>))$$

$$|\psi_2> = 1/\sqrt{2}\,(|00>(\alpha|0>+\beta|1>)+|01>(\alpha|1>+\beta|0>) + |10>(\alpha|0>-\beta|1>)+|11>(\alpha|1>-\beta|0>))$$

$$|\psi_3(00)> = (\alpha|0>+\beta|1>)$$
$$|\psi_3(01)> = (\alpha|1>+\beta|0>)$$
$$|\psi_3(10)> = (\alpha|0>-\beta|1>)$$
$$|\psi_3(11)> = (\alpha|1>-\beta|0>)$$



$$
\begin{aligned}
|\psi_3(00)> &= (\alpha|0>+\beta|1>) \\
|\psi_3(01)> &= (\alpha|1>+\beta|0>) \\
|\psi_3(10)> &= (\alpha|0>-\beta|1>) \\
|\psi_3(11)> &= (\alpha|1>-\beta|0>)
\end{aligned}
$$

$$
\left.
\begin{aligned}
|\psi_4(00)> &= \psi_3(00)> = (\alpha|0>+\beta|1>) \\
|\psi_4(01)> &= X\,|\psi_3(01)> = X(\alpha|1>+\beta|0>) = (\alpha|0>+\beta|1>) \\
|\psi_4(10)> &= Z|\psi_3(10)> = Z(\alpha|0>-\beta|1>) = (\alpha|0>+\beta|1>) \\
|\psi_4(11)> &= ZX|\psi_3(11)> = ZX\,(\alpha|1>-\beta|0>) = (\alpha|0>+\beta|1>)
\end{aligned}
\right\}
\begin{array}{c} |\psi> \\ = \alpha|0>+\beta|1> \end{array}
$$

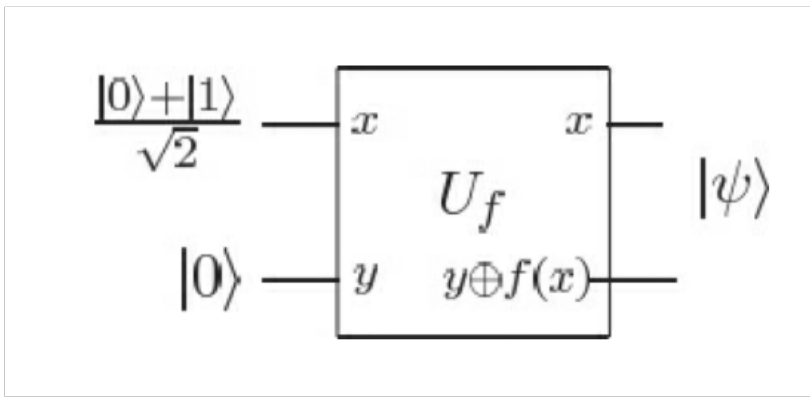
いくつかの疑問

- 第一。量子テレポーテーションは、光より早く量子状態を送ることができるのだろうか？
  これに対する答えは、明確にノーだ。テレポーテーションを実行するためには、Aliceは観測結果を、古典的な通信路で、Bobに送らなければならないのだから。

- 第二。量子テレポーテーションは、未知の量子状態のコピーを禁じたNo Cloning定理を破ることにならないか？
  これについても、答えはノーだ。Bobのもとで、量子状態は再現されるのだが、Aliceのもとにあったオリジナルの量子状態は、Aliceの観測によって、|0>か|1>かの状態に変わって、失われている。

Quantum Parallelism

- 次のような回路を考える。入力が計算基底 |0>, |1>ではなく、アダマール・ゲートを|0>に作用させた $(|0\rangle + |1\rangle)/\sqrt{2}$ だとしよう。

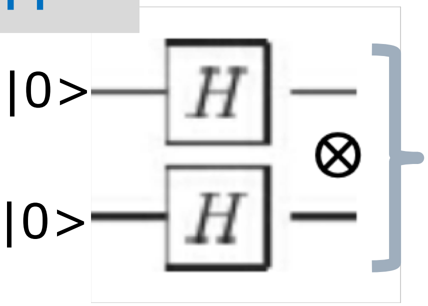


- この時、U_fを作用させれば、次の状態が得られる。

$$\frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}.$$

- 二つのアダマール・ゲートの並列動作を示すのに、$H^{\otimes 2}$ と書くことにしよう。このH$^{\otimes 2}$の作用は、次のようになる。

$H^{\otimes 2}$



$$\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\otimes\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)=\frac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{2}$$

- もっと一般的に、最初は全て|0>の状態のn個のqubitにアダマール変換を実行した結果は、

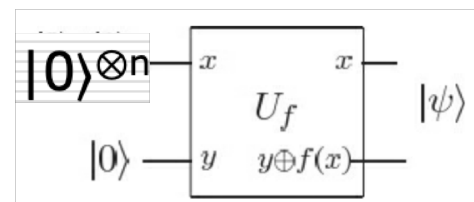
$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$$

  で表すことができる。ここに、和は、全ての可能なxの値をとったものである。この動作を$H^{\otimes n}$ で表そう。
- アダマール変換$H^{\otimes n}$は、全ての計算基底の等しい重ね合わせを生み出す。これは、n個のゲートを使って2^n個の状態の重ね合わせを生み出す非常に効率的な方法である。

量子並列評価

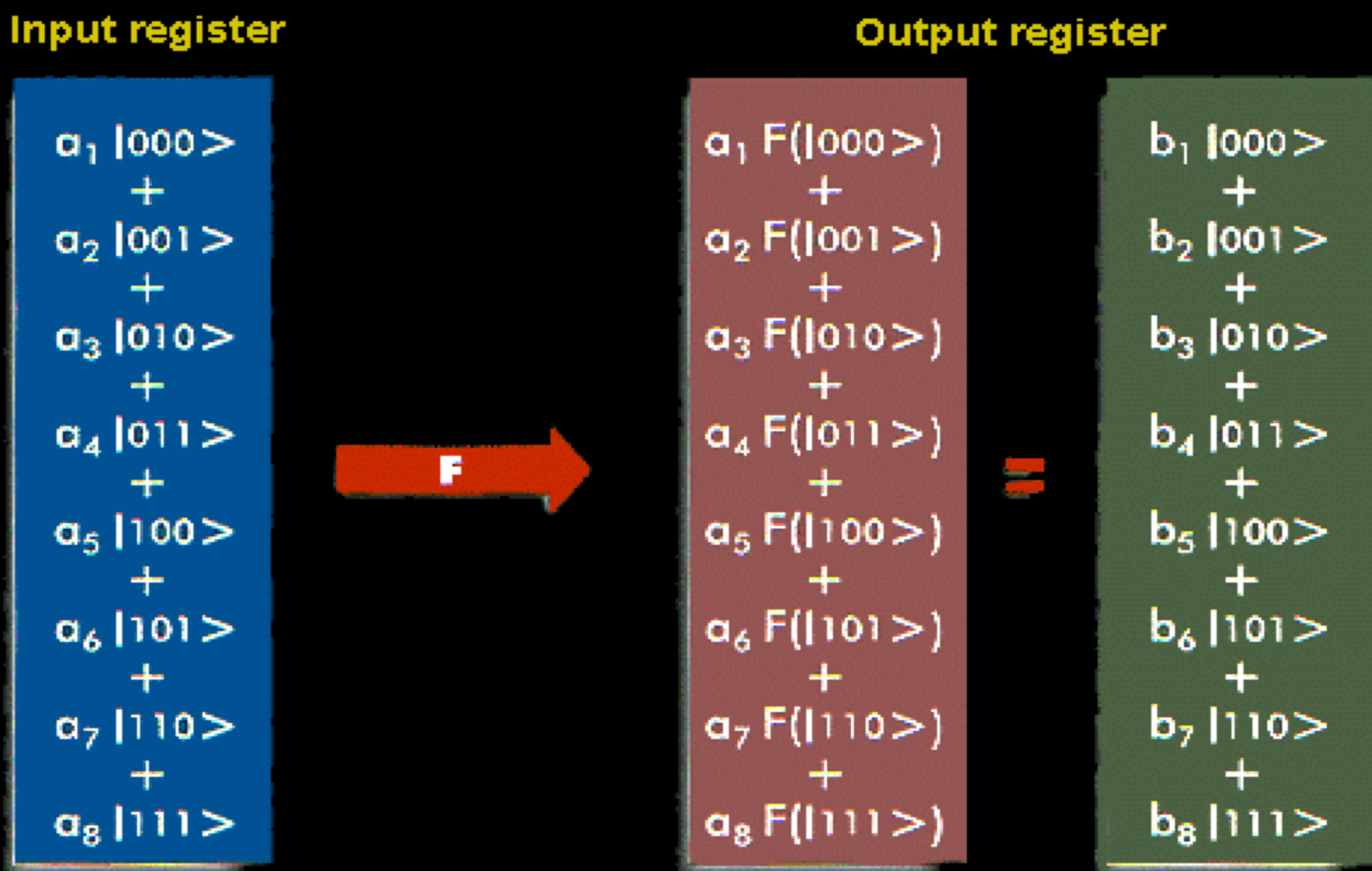
- nビットの入力xと1ビットの出力 f(x)を持つ関数の量子並列評価は、次のように実行される。
- n+1qubitの状態 $|0\rangle^{\otimes n}|0\rangle$を用意する。次に、最初のn qubitに、アダマール変換を適用する。その出力をUfの量子回路の入力に接続すると、次の状態が生み出される。

$$\boxed{\frac{1}{\sqrt{2^n}}\sum_x |x\rangle|f(x)\rangle}$$



- ある意味では、量子並列処理は、関数fの全ての可能な2^n個の値を同時に評価する。たとえ、我々は明らかにfを一回だけ評価するのであるが。

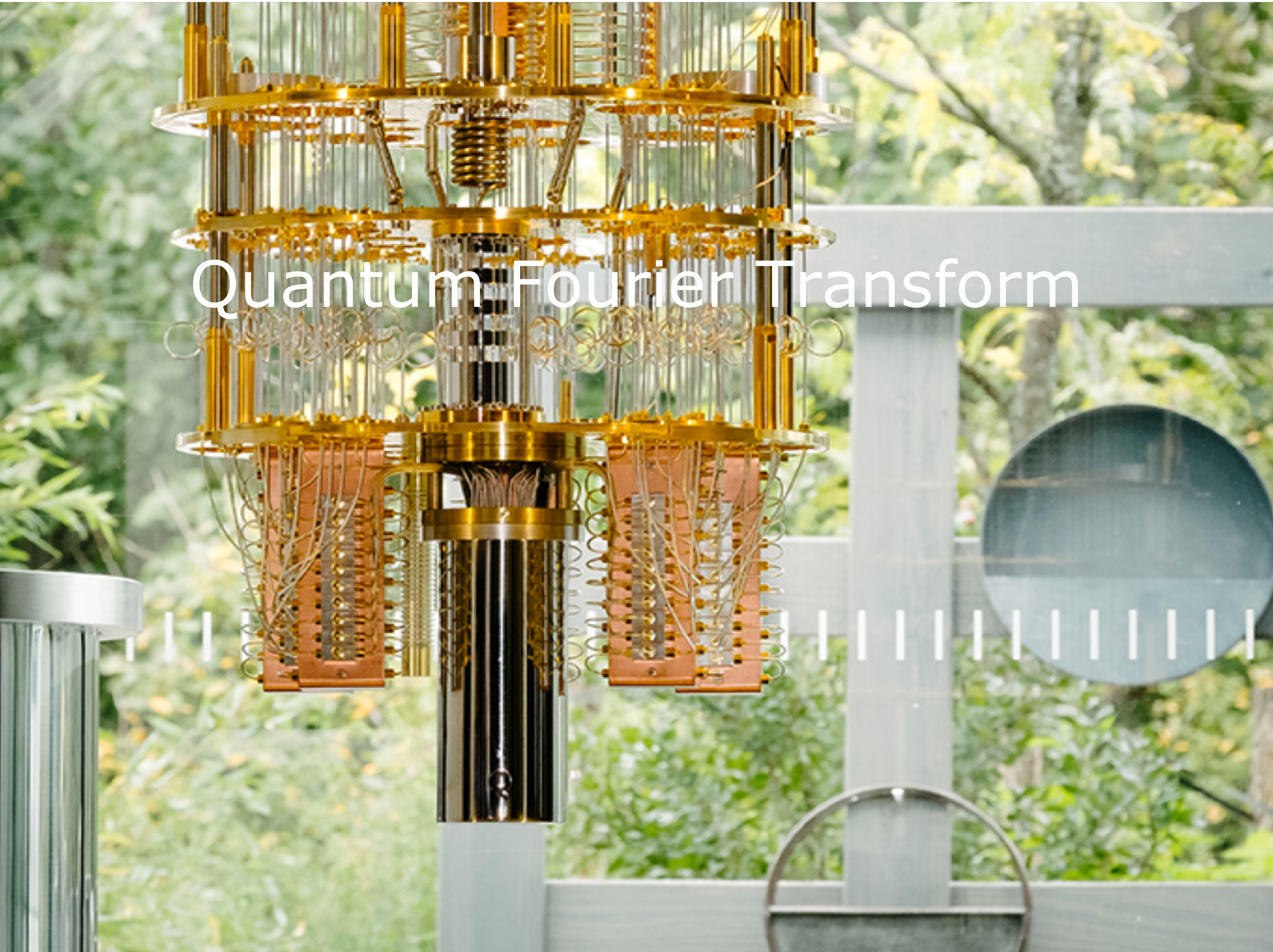
n個のqubitで、2^n個の並行計算が可能



ただし、問題がある

- しかし、この並列計算は、直ちに有用なわけではない。
  我々の最初の単一qubitのサンプル $\frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}$ では、

  この状態の測定は、$|0, f(0)\rangle$ か $|1, f(1)\rangle$の値のどれかを返すだけだ。もっと一般的に、状態 $\sum_x |x, f(x)\rangle$ の測定は、一つの値xについてのf(x)の値を返すだけだ。もちろん、古典的なコンピュータは、そうしたことを簡単にやってのける。
- 量子計算が役に立つためには、単なる量子並行計算以上の何かが必要になる。すなわち、$\sum_x |x, f(x)\rangle$ のような重ね合わせの状態から、f(x)の一つの値より多くの値の情報を引き出せるような能力が必要になる。



Quantum Fourier Transform

古典的フーリエ変換

フーリエ級数とフーリエ係数

フーリエ級数は、関数 f(x)を、三角関数 sinとcosの無限和として、次の形で表したもの。

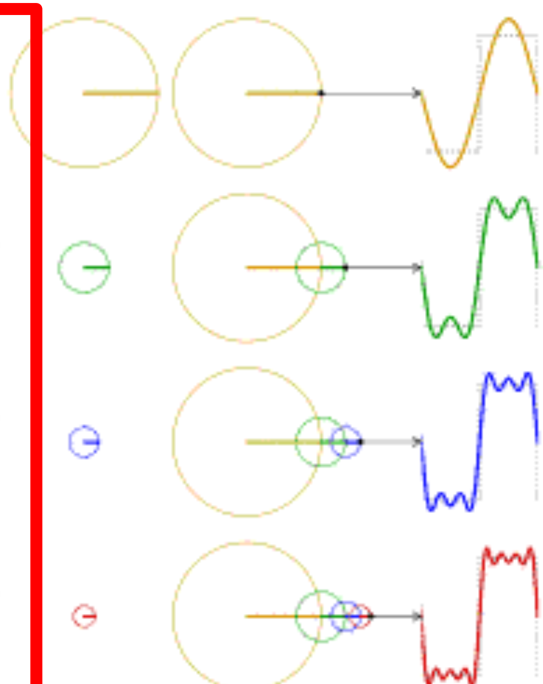
$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left(a_k \cos kx + b_k \sin kx\right)$$

この時、級数の係数 a_k, b_k は、次の式で求められる。

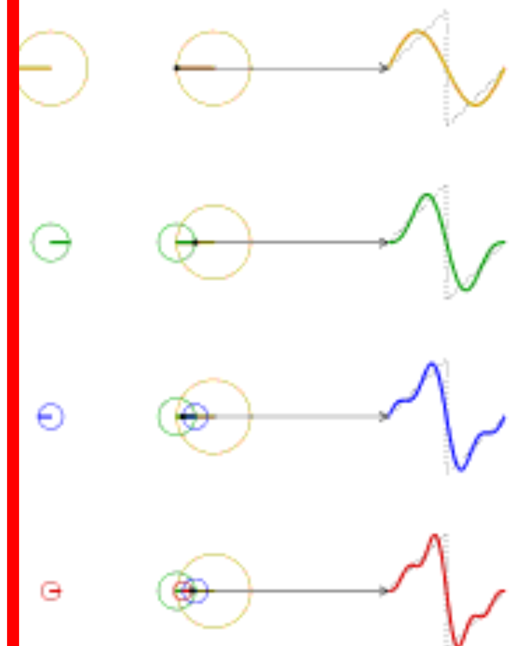
$$a_k = \frac{1}{\pi}\int_0^{2\pi} f(x) \cos kx \, \mathrm{d}x, \quad b_k = \frac{1}{\pi}\int_0^{2\pi} f(x) \sin kx \, \mathrm{d}x.$$

例えば、なめらかではない矩形波や鋸波も、次のように、sinの和として表わすことができる

$$\frac{4\sin\theta}{\pi}+\frac{4\sin 3\theta}{3\pi}+\frac{4\sin 5\theta}{5\pi}+\frac{4\sin 7\theta}{7\pi}+\cdots$$



$$\frac{2\sin\theta}{-\pi}+\frac{2\sin 2\theta}{2\pi}+\frac{2\sin 3\theta}{-3\pi}+\frac{2\sin 4\theta}{4\pi}+\cdots$$



https://goo.gl/TD11yc

フーリエ級数とフーリエ変換

関数fのフーリエ級数展開表示を、その周波数成分を取り出した周波数領域表示に変えることをフーリエ変換と呼び、$\hat{f}$で表す。

https://goo.gl/VEwWFo

最初にフーリエが考えたこと

最初にフーリエが考えたこと

$$\varphi(y) = a_0 \cos\frac{\pi y}{2} + a_1 \cos 3\frac{\pi y}{2} + a_2 \cos 5\frac{\pi y}{2} + \cdots.$$

Multiplying both sides by $\cos(2k+1)\dfrac{\pi y}{2}$, and then integrating from $y = -1$ to $y = +1$ yields:

$$a_k = \int_{-1}^{1} \varphi(y) \cos(2k+1)\frac{\pi y}{2}\, dy.$$

— Joseph Fourier (1807)
Mémoire sur la propagation de la chaleur dans les corps solides.

https://goo.gl/PHf26Y

最初にフーリエが考えたこと

$$\varphi(y) = a_0 \cos\frac{\pi y}{2} + a_1 \cos 3\frac{\pi y}{2} + a_2 \cos 5\frac{\pi y}{2} + \cdots.$$

$$\begin{aligned} a_k &= \int_{-1}^{1} \varphi(y) \cos(2k+1)\frac{\pi y}{2}\, dy \\ &= \int_{-1}^{1} \left(a_0 \cos\frac{\pi y}{2} \cos(2k+1)\frac{\pi y}{2} + a_1 \cos 3\frac{\pi y}{2} \cos(2k+1)\frac{\pi y}{2} + \cdots \right) dy \end{aligned}$$

ところで、$j \neq k$ の時 $\int_{-1}^{1} \cos(2j+1)\frac{\pi y}{2} \cos(2k+1)\frac{\pi y}{2} = 0$

である。なぜなら、

$$\boxed{\int_{-\pi}^{\pi} \cos 0x \, dx = \int_{-\pi}^{\pi} 1 \, dx = 2\pi}$$

$$\int_{-\pi}^{\pi} \cos 1x \, dx = \int_{-\pi}^{\pi} \cos 2x \, dx = \int_{-\pi}^{\pi} \cos 3x \, dx =$$

$$\int_{-\pi}^{\pi} \cos 4x \, dx = \cdots = \int_{-\pi}^{\pi} \cos nx \, dx = 0$$

$$\int_{-\pi}^{\pi} \sin 0x \, dx = \int_{-\pi}^{\pi} 0 \, dx = 0$$

$$\int_{-\pi}^{\pi} \sin 1x \, dx = \int_{-\pi}^{\pi} \sin\ 2x \, dx = \int_{-\pi}^{\pi} \sin 3x \, dx =$$

$$\int_{-\pi}^{\pi} \sin 4\mathrm{x}\, \mathrm{dx} = \cdots = \int_{-\pi}^{\pi} \sin nx \, dx = 0$$

sin も cos も周期関数で、一定区間積分すれば、打ち消しあってゼロになる

$$
\begin{aligned}
\int_{-\pi}^{\pi} \cos(mx)\ \cos(nx)\, dx &= \frac{1}{2}\int_{-\pi}^{\pi} \cos((n-m)x) + \cos((n+m)x)\, dx = \pi\delta_{mn}, \\
\int_{-\pi}^{\pi} \sin(mx)\ \sin(nx)\, dx &= \frac{1}{2}\int_{-\pi}^{\pi} \cos((n-m)x) - \cos((n+m)x)\, dx = \pi\delta_{mn},
\end{aligned}
$$

$$
\int_{-\pi}^{\pi} \cos(mx)\ \sin(nx)\, dx = \frac{1}{2}\int_{-\pi}^{\pi} \sin((n+m)x) + \sin((n-m)x)\, dx = 0;
$$

$\sin\alpha\cos\beta = \frac{1}{2}\{\sin(\alpha+\beta)+\sin(\alpha-\beta)\}$
$\cos\alpha\sin\beta = \frac{1}{2}\{\sin(\alpha+\beta)-\sin(\alpha-\beta)\}$
$\cos\alpha\cos\beta = \frac{1}{2}\{\cos(\alpha+\beta)+\cos(\alpha-\beta)\}$
$\sin\alpha\sin\beta = \frac{-1}{2}\{\cos(\alpha+\beta)-\cos(\alpha-\beta)\}$

三角関数の「積和の公式」

n=mの時 2π

$$\int_{-\pi}^{\pi} \cos(mx)\,\cos(nx)\,dx = \frac{1}{2}\int_{-\pi}^{\pi} \left(\cos((n-m)x) + \cos((n+m)x)\right) dx = \pi\delta_{mn},$$

$$\int_{-\pi}^{\pi} \sin(mx)\,\sin(nx)\,dx = \frac{1}{2}\int_{-\pi}^{\pi} \left(\cos((n-m)x) - \cos((n+m)x)\right) dx = \pi\delta_{mn},$$

残りは、全部ゼロになる

$$\int_{-\pi}^{\pi} \cos(mx)\,\sin(nx)\,dx = \frac{1}{2}\int_{-\pi}^{\pi} \left(\sin((n+m)x) + \sin((n-m)x)\right) dx = 0;$$

Sines and cosines form an orthonormal set, as illustrated above. The integral of sine, cosine and their product is zero (green and red areas are equal, and cancel out) when *m*, *n* or the functions are different, and pi only if *m* and *n* are equal, and the function used is the same.

フーリエ級数とフーリエ係数

フーリエ級数は、関数 f(x)を、三角関数 sinとcosの無限和として、次の形で表したもの。

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left(a_k \cos kx + b_k \sin kx\right)$$

この時、級数の係数 a_k, b_k は、次の式で求められる。

$$a_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos kx \,\mathrm{d}x, \quad b_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin kx \,\mathrm{d}x.$$

フーリエ変換の基本的性質

$$\mathcal{F}[a\,f + b\,g] \;=\; a\,\mathcal{F}[f] \;+\; b\,\mathcal{F}[g]$$

$$\boxed{\mathcal{F}[f(ax)](k) \;=\; \frac{1}{|a|}\mathcal{F}[f(x)]\left(\frac{k}{a}\right)}$$

$$\boxed{\mathcal{F}[f(x-a)](k) \;=\; e^{-ika}\,\mathcal{F}[f(x)](k)}$$

$$\mathcal{F}[e^{iax}f(x)](k) \;=\; \mathcal{F}[f(x)](k-a)$$

後のPeriod Finding
で利用される

http://eman-physics.net/math/fourier06.html

フーリエ級数・フーリエ変換を複素数で考える

Fourier Transform

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i x \xi}\, d\xi.$$

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx.$$

Fourier Transform

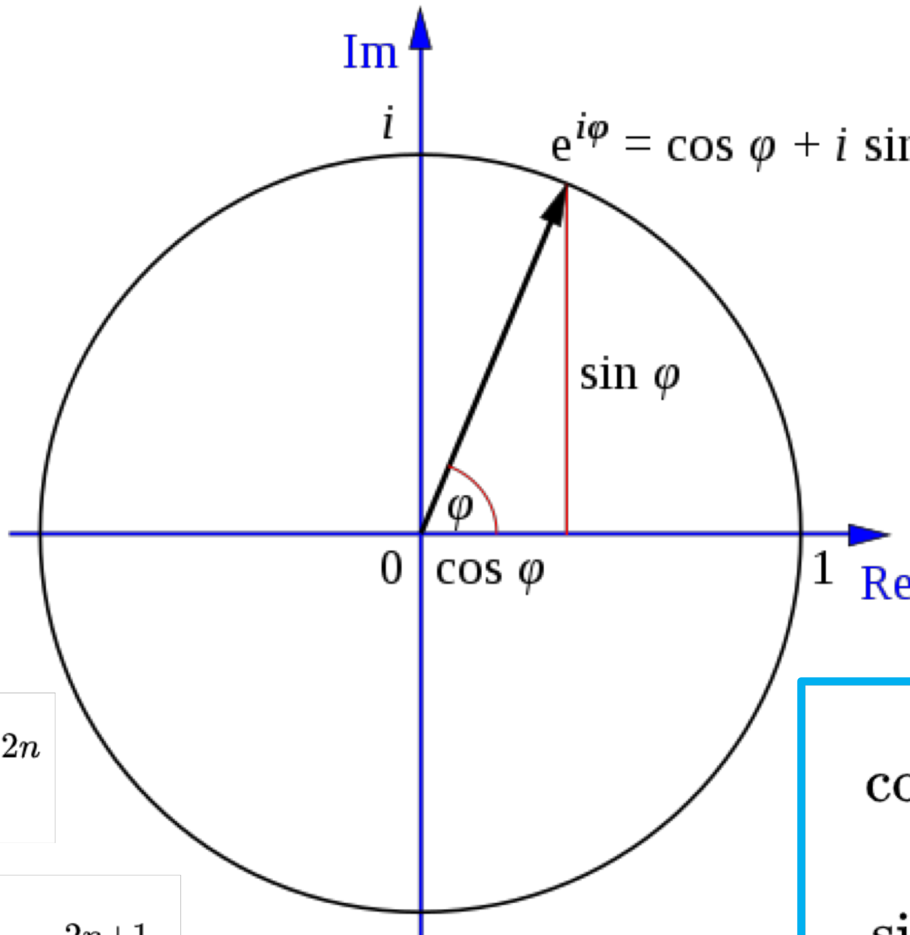
$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i x \xi}\, d\xi.$$

共役

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx.$$

オイラーの公式

$$e^{i\theta} = cos\,\theta + i\,sin\,\theta$$



$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

$$\boxed{\begin{aligned} \cos z &= \frac{e^{iz}+e^{-iz}}{2}, \\ \sin z &= \frac{e^{iz}-e^{-iz}}{2i} \end{aligned}}$$

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{2\pi}{L} nx\right) + b_n \sin\left(\frac{2\pi}{L} nx\right) \right]$$

$$a_n = \frac{2}{L} \int_0^L f(x) \cos\left(\frac{2\pi}{L} nx\right) \mathrm{d}x$$

$$b_n = \frac{2}{L} \int_0^L f(x) \sin\left(\frac{2\pi}{L} nx\right) \mathrm{d}x$$

EMAN 物理数学 **複素フーリエ級数**

http://eman-physics.net/math/fourier03.html

$$\cos z = \frac{e^{iz} + e^{-iz}}{2},$$
$$\sin z = \frac{e^{iz} - e^{-iz}}{2i}$$

$$\begin{aligned}
f(x) &= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{2\pi}{L}nx\right) + b_n \sin\left(\frac{2\pi}{L}nx\right)\right] \\
&= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left[\frac{a_n}{2}\left(e^{i\frac{2\pi}{L}nx} + e^{-i\frac{2\pi}{L}nx}\right) - i\,\frac{b_n}{2}\left(e^{i\frac{2\pi}{L}nx} - e^{-i\frac{2\pi}{L}nx}\right)\right] \\
&= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left[\frac{a_n - i\,b_n}{2}\,e^{i\frac{2\pi}{L}nx} + \frac{a_n + i\,b_n}{2}\,e^{-i\frac{2\pi}{L}nx}\right] \\
&= \frac{a_0}{2} + \sum_{n=1}^{\infty}\frac{a_n - i\,b_n}{2}\,e^{i\frac{2\pi}{L}nx} + \sum_{n=1}^{\infty}\frac{a_n + i\,b_n}{2}\,e^{-i\frac{2\pi}{L}nx} \\
&= \frac{a_0}{2} + \sum_{n=1}^{\infty}\frac{a_n - i\,b_n}{2}\,e^{i\frac{2\pi}{L}nx} + \sum_{n=-1}^{-\infty}\frac{a_{-n} + i\,b_{-n}}{2}\,e^{i\frac{2\pi}{L}nx}
\end{aligned}$$

$$
\begin{aligned}
c_n &\equiv \begin{cases} \frac{1}{2}(a_n - i\,b_n) & (n > 0) \\ \frac{1}{2}(a_{-n} + i\,b_{-n}) & (n < 0) \end{cases} \\
c_0 &\equiv \frac{a_0}{2}
\end{aligned}
$$

$$
\begin{aligned}
f(x) &= c_0 + \sum_{n=1}^{\infty} c_n\, e^{i\frac{2\pi}{L}nx} + \sum_{n=-1}^{-\infty} c_n\, e^{i\frac{2\pi}{L}nx} \\
&= \sum_{n=-\infty}^{\infty} c_n\, e^{i\frac{2\pi}{L}nx}
\end{aligned}
$$

$$
\begin{aligned}
c_n &= \frac{a_n - i\, b_n}{2} \\
&= \frac{1}{2}\,\frac{2}{L}\int_0^L f(x)\left[\cos\left(\frac{2\pi}{L}nx\right) - i\,\sin\left(\frac{2\pi}{L}nx\right)\right]\mathrm{d}x \\
&= \boxed{\frac{1}{L}\int_0^L f(x)\, e^{-i\frac{2\pi}{L}nx}\,\mathrm{d}x}
\end{aligned}
$$

$$
\begin{aligned}
c_n &= \frac{a_{-n} + i\, b_{-n}}{2} \\
&= \frac{1}{2}\,\frac{2}{L}\int_0^L f(x)\left[\cos\left(-\frac{2\pi}{L}nx\right) + i\,\sin\left(-\frac{2\pi}{L}nx\right)\right]\mathrm{d}x \\
&= \frac{1}{L}\int_0^L f(x)\left[\cos\left(\frac{2\pi}{L}nx\right) - i\,\sin\left(\frac{2\pi}{L}nx\right)\right]\mathrm{d}x \\
&= \boxed{\frac{1}{L}\int_0^L f(x)\, e^{-i\frac{2\pi}{L}nx}\,\mathrm{d}x}
\end{aligned}
$$

$$
c_0 = \frac{a_0}{2} = \frac{1}{2}\,\frac{2}{L}\int_0^L f(x)\,\mathrm{d}x = \frac{1}{L}\int_0^L f(x)\,\mathrm{d}x
$$

Fourier Transform

$$f(x) = \sum_{n=-\infty}^{\infty} c_n \, e^{i\frac{2\pi}{L}nx}$$

$$c_n = \frac{1}{L}\int_0^L f(x)\, e^{-i\frac{2\pi}{L}nx}\,\mathrm{d}x$$

Fourier Transform

$$f(x) = \sum_{n=-\infty}^{\infty} c_n \, e^{i \frac{2\pi}{L} n x}$$

共役

$$c_n = \frac{1}{L} \int_0^L f(x) \, e^{-i \frac{2\pi}{L} n x} \, \mathrm{d}x$$

Fourier Transform

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i x \xi}\, d\xi.$$

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx.$$

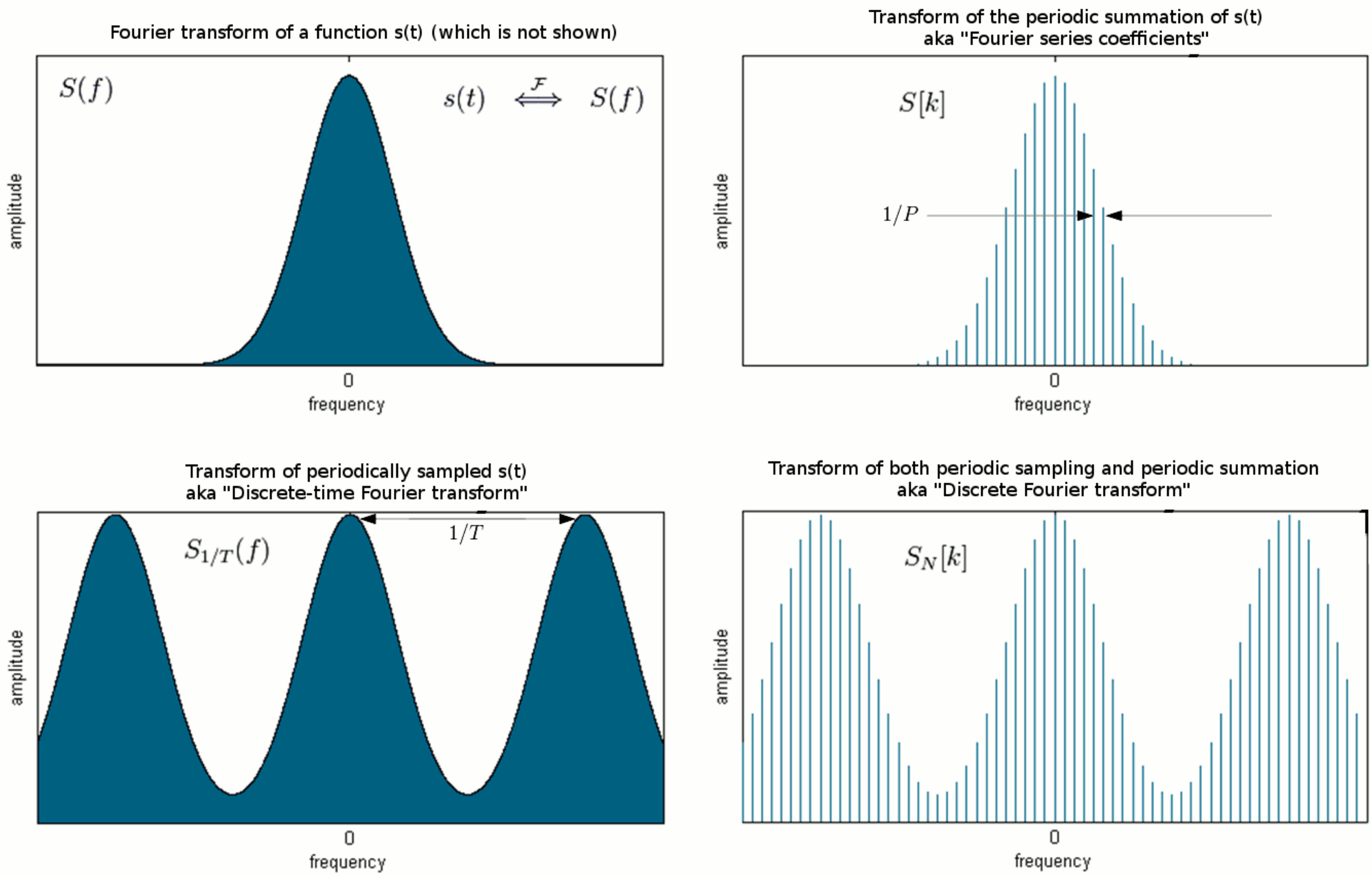
Fourier Transform

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i x \xi}\, d\xi.$$

共役

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx.$$

Digital Fourier Transform


Fourier transform of a function s(t) (which is not shown)
$S(f)$
$s(t) \overset{\mathcal{F}}{\Longleftrightarrow} S(f)$
amplitude
0
frequency
Transform of the periodic summation of s(t)
aka "Fourier series coefficients"
$S[k]$
$1/P$
amplitude
0
frequency
Transform of periodically sampled s(t)
aka "Discrete-time Fourier transform"
$S_{1/T}(f)$
$1/T$
amplitude
0
frequency
Transform of both periodic sampling and periodic summation
aka "Discrete Fourier transform"
$S_N[k]$
amplitude
0
frequency

Digital Fourier Transform

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}$$

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N} kn}$$

Digital Fourier Transform

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}$$

共役

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N}kn}$$

こんな計算になる

Let $N = 4$ and $\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2-i \\ -i \\ -1+2i \end{pmatrix}$

$$\boxed{X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N}kn}}$$

Here we demonstrate how to calculate the DFT of $\mathbf{x}$ using

$$X_0 = e^{-i2\pi 0\cdot 0/4} \cdot 1 + e^{-i2\pi 0\cdot 1/4} \cdot (2-i) + e^{-i2\pi 0\cdot 2/4} \cdot (-i) + e^{-i2\pi 0\cdot 3/4} \cdot (-1+2i) = 2$$

$$X_1 = e^{-i2\pi 1\cdot 0/4} \cdot 1 + e^{-i2\pi 1\cdot 1/4} \cdot (2-i) + e^{-i2\pi 1\cdot 2/4} \cdot (-i) + e^{-i2\pi 1\cdot 3/4} \cdot (-1+2i) = -2-2i$$

$$X_2 = e^{-i2\pi 2\cdot 0/4} \cdot 1 + e^{-i2\pi 2\cdot 1/4} \cdot (2-i) + e^{-i2\pi 2\cdot 2/4} \cdot (-i) + e^{-i2\pi 2\cdot 3/4} \cdot (-1+2i) = -2i$$

$$X_3 = e^{-i2\pi 3\cdot 0/4} \cdot 1 + e^{-i2\pi 3\cdot 1/4} \cdot (2-i) + e^{-i2\pi 3\cdot 2/4} \cdot (-i) + e^{-i2\pi 3\cdot 3/4} \cdot (-1+2i) = 4+4i$$

$$\mathbf{X} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -2-2i \\ -2i \\ 4+4i \end{pmatrix}$$

DFTの働きを行列で表現すると

$$\mathbf{F} = \begin{bmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \cdots & \omega_N^{0\cdot (N-1)} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \cdots & \omega_N^{1\cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N^{(N-1)\cdot 0} & \omega_N^{(N-1)\cdot 1} & \cdots & \omega_N^{(N-1)\cdot (N-1)} \end{bmatrix}$$

ここで

$$\boxed{\omega_N = e^{-2\pi i/N}}$$

このω_Nを、1の原始N乗根 と呼ぶ。（N乗すれば1になる）
DFT行列の要素は、ω_Nで表現される。

DFT matrix

$$W = \left(\frac{\omega^{jk}}{\sqrt{N}}\right)_{j,k=0,\ldots,N-1}$$

$$W = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix},$$

もっと具体的に言えば、DFT行列のj行k列目の要素は、ω^{jk}である。(jとkをかけて、ωをその数乗する)

ちょっと横道
量子の重ね合わせの「成分」を計算する

- 正規直交の基底を|0>, |1>, … , |n-1> とすると、任意のベクトル|A> は、次の式で表現される。 この時、α_iを、基底 |i> の成分という。

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

- この表示から、次のようにして成分を取り出すことができる。両辺に、<j| をかけると、

$$\langle j|A\rangle = \langle j| \sum_i \alpha_i |i\rangle = \underbrace{\sum_i \alpha_i \langle j|i\rangle = \alpha_j}_{\text{ここが、みそ。}}$$

$$\alpha_0 \langle j|0\rangle + \alpha_1 \langle j|1\rangle + \alpha_2 \langle j|2\rangle + \ldots + \alpha_j \langle j|j\rangle + \cdots + \alpha_{n-1} \langle j|n-1\rangle$$

$$|A> = \sum_i \alpha_i |i>$$

共役

$$\alpha_i = <i|A>$$

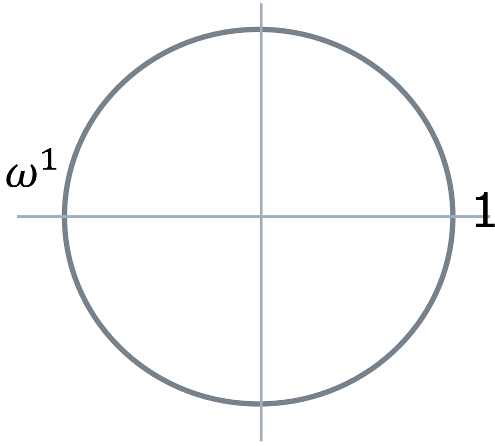
よく似た形をしている

Digital Fourier Transform
サンプル

DFT_2 , DFT_3

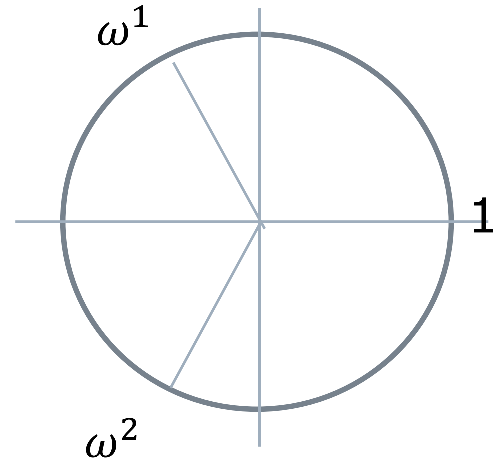
- M=2, $\omega = e^{2\pi i\, 1/2} = -1$

$$DFT_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & \omega \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



- M=3, $\omega = e^{2\pi i\, 1/3}$, $\omega^2 = e^{2\pi i\, 2/3}$, $\omega^3 = e^{2\pi i\, 3/3} = 1$

$$DFT_3 = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega^4 \end{pmatrix} = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}$$

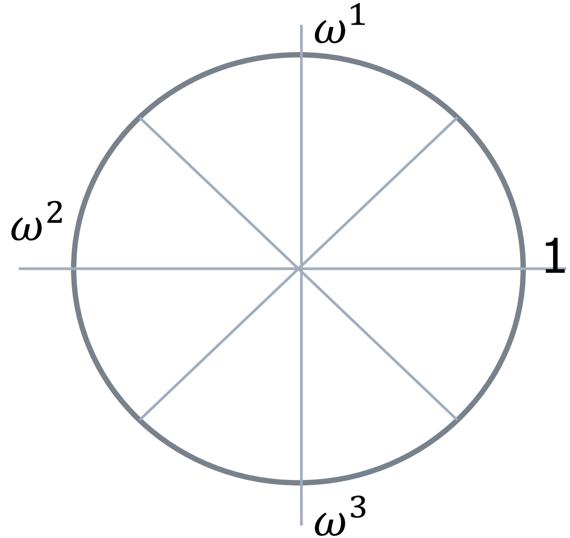


$$\phi_j = \frac{2\pi j}{M}$$

DFT_4

- M=4, $\omega = e^{2\pi i\, 1/4} = i$, $\omega^2 = e^{2\pi i\, 2/4} = -1$,
  $\omega^3 = e^{2\pi i\, 3/4} = -i$, $\omega^4 = e^{2\pi i\, 4/4} = 1$

$$DFT_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & -i \end{pmatrix}$$



実際に、DFT$_4$を適用してみる

$$\frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle) = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}; \; |g\rangle = |0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

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

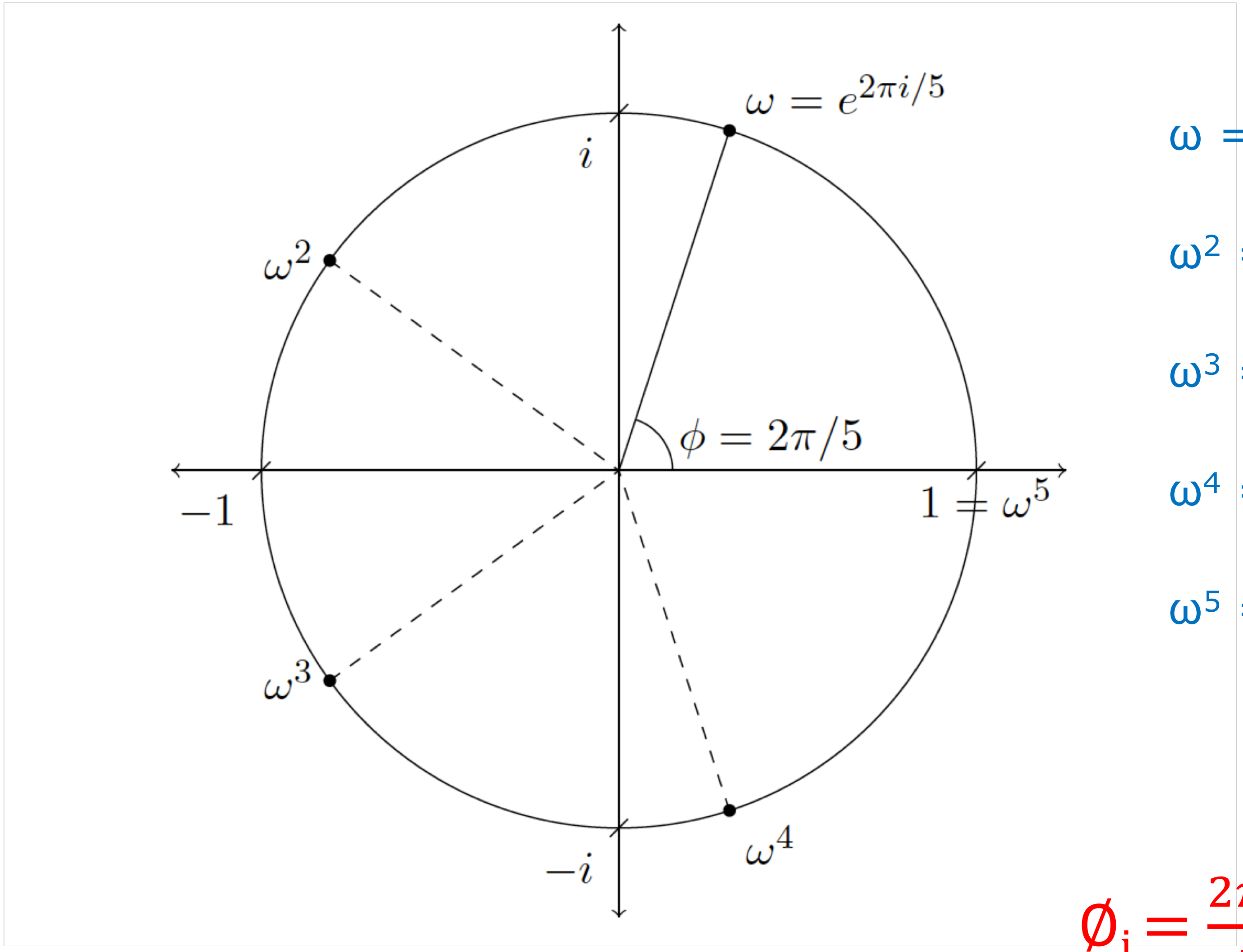
$$\left|\hat{f}\right\rangle = \frac{1}{4}\begin{pmatrix}1 & 1 & 1 & 1\\1 & i & -1 & -i\\1 & -1 & 1 & -1\\1 & -i & -1 & i\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$|\hat{g}\rangle = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & i & -1 & -i\\1 & -1 & 1 & -1\\1 & -i & -1 & i\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

実際に、DFT$_4$を適用してみる

$$\left|\hat{h}\right\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ i \\ -1 \\ -i \end{pmatrix}$$

DFT_5だったら、これを使う
1の5乗根



$\omega = e^{2\pi i/5}$

$\omega^2 = e^{2\pi i\, 2/5}$

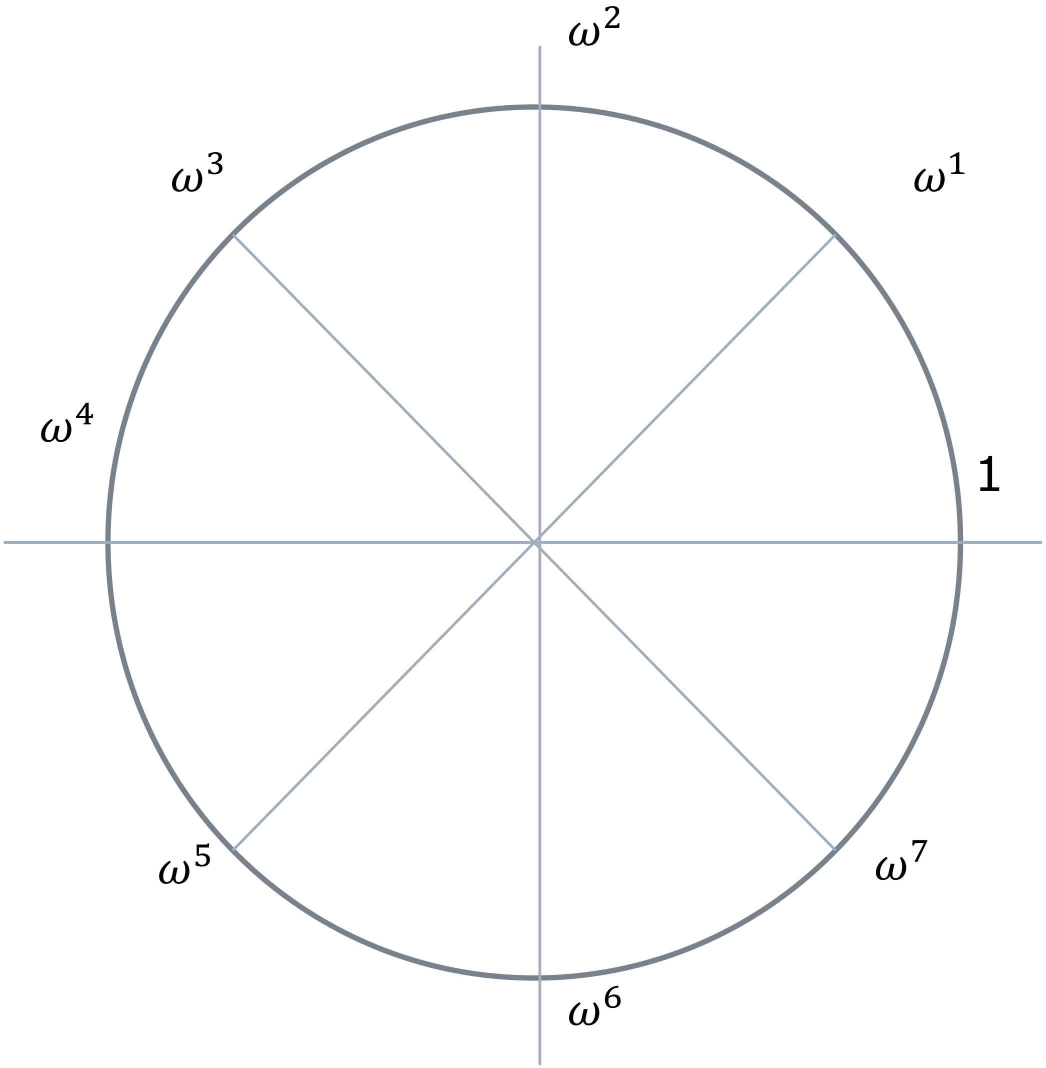
$\omega^3 = e^{2\pi i\, 3/5}$

$\omega^4 = e^{2\pi i\, 4/5}$

$\omega^5 = e^{2\pi i\, 5/5} = 1$

$$\phi_j = \frac{2\pi j}{5}$$

DFT_8だったら、これを使う
1の8乗根



$1 = -\omega^4$
$\omega^1 = -\omega^5$
$\omega^2 = -\omega^6$
$\omega^3 = -\omega^7$

$\omega^2 = i$
$\omega^4 = -1$
$\omega^6 = -i$

Fast Fourier Transform
-- DFT Matrixの変形

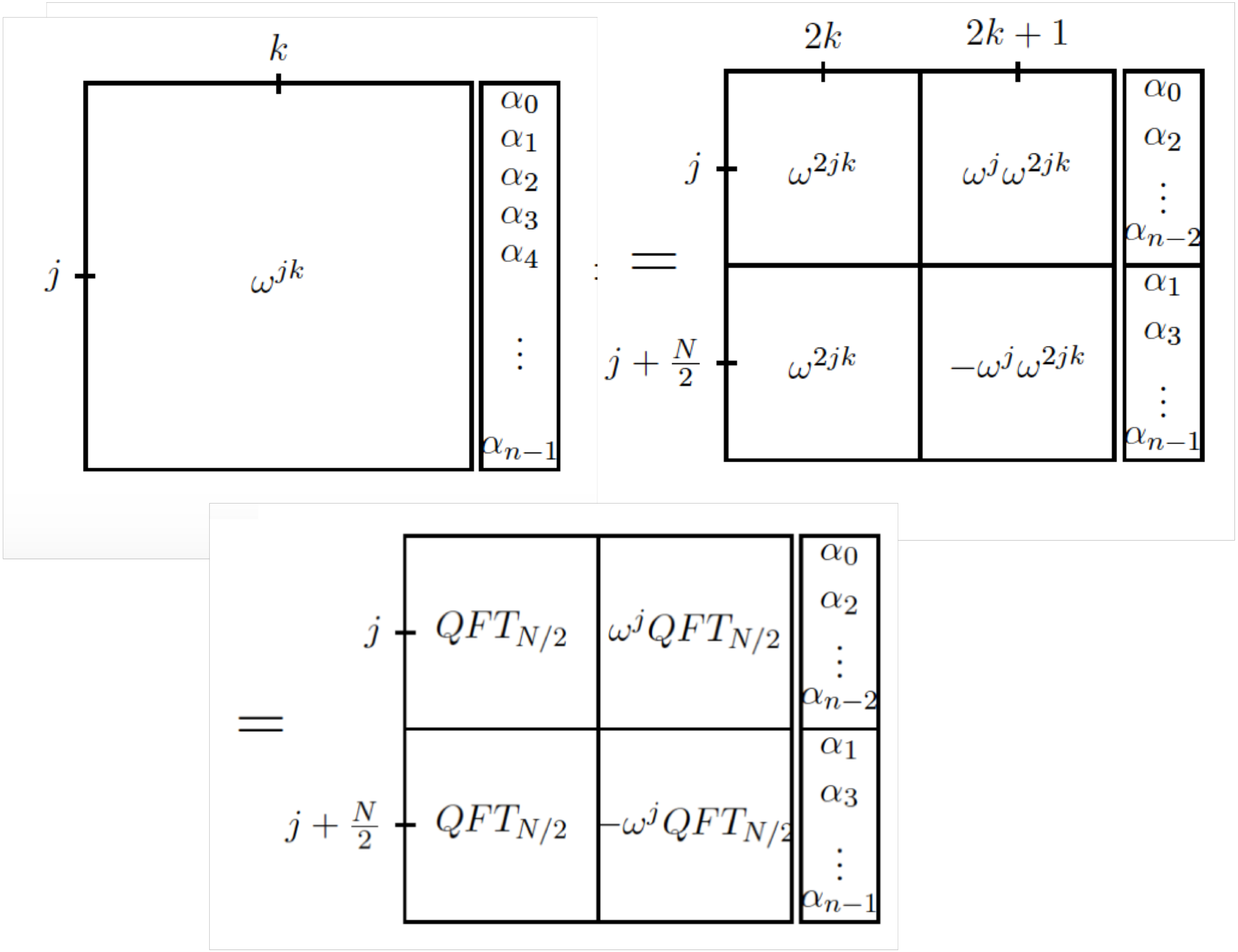
- Fast Fourier transforms are widely used for many applications in engineering, science, and mathematics. The basic ideas were popularized in 1965, but some algorithms had been derived as early as 1805.[3] In 1994, Gilbert Strang described the FFT as "the most important numerical algorithm of our lifetime"[4][5] and it was included in Top 10 Algorithms of 20th Century by the IEEE journal *Computing in Science & Engineering*.[6]

https://en.wikipedia.org/wiki/Fast_Fourier_transform

- The DFT is obtained by decomposing a sequence of values into components of different frequencies.[3] This operation is useful in many fields (see discrete Fourier transform for properties and applications of the transform) but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing the DFT of N points in the naive way, using the definition, takes $O(N^2)$ arithmetical operations, while an FFT can compute the same DFT in only $O(N \log N)$ operations. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions.

- In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to *N* / log N. This huge improvement made the calculation of the DFT practical; FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers.

次のような変形を目標とする



DFT$_8$を例に考える

奇数列

DFT 行列を奇数列と偶数列に分け、
偶数列と奇数列をまとめる

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{pmatrix}
$$

偶数列

奇数列

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
$$

偶数列

この時、次が成り立つ。

それでも、行列の積は変わらない。

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\ 1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\ 1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\ 1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\ 1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\ 1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}$$

なぜなら、

行列Mと列ベクトルAの積は、Mの列を入れ替えても変わらない

$$
\begin{pmatrix}
m_{0,0} & \cdots & m_{0,j} & \cdots & m_{0,k} & \cdots & m_{0,n-1} \\
m_{1,0} & \cdots & m_{1,j} & \cdots & m_{1,k} & \cdots & m_{1,n-1} \\
m_{2,0} & \cdots & m_{2,j} & \cdots & m_{2,k} & \cdots & m_{2,n-1} \\
\vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
\vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
m_{n-2,0} & \cdots & m_{n-2,j} & \cdots & m_{n-2,k} & \cdots & m_{n-2,n-1} \\
m_{n-1,0} & \cdots & m_{n-1,j} & \cdots & m_{n-1,k} & \cdots & m_{n-1,n-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_{n-2} \\ a_{n-1}
\end{pmatrix}
=
$$

$$
\begin{pmatrix}
m_{0,0} & \cdots & m_{0,k} & \cdots & m_{0,j} & \cdots & m_{0,n-1} \\
m_{1,0} & \cdots & m_{1,k} & \cdots & m_{1,j} & \cdots & m_{1,n-1} \\
m_{2,0} & \cdots & m_{2,k} & \cdots & m_{2,j} & \cdots & m_{2,n-1} \\
\vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
\vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
m_{n-2,0} & \cdots & m_{n-2,k} & \cdots & m_{n-2,j} & \cdots & m_{n-2,n-1} \\
m_{n-1,0} & \cdots & m_{n-1,k} & \cdots & m_{n-1,j} & \cdots & m_{n-1,n-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_{n-2} \\ a_{n-1}
\end{pmatrix}
$$

次の式が成り立つことを、実際の計算で確かめよ。

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}
=
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}
=
$$

$$
\begin{pmatrix}
1a_0 + 1a_1 + 1a_2 + 1a_3 + 1a_4 + 1a_5 + 1a_6 + 1a_7 \\
1a_0 + \omega^1 a_1 + \omega^2 a_2 + \omega^3 a_3 + \omega^4 a_4 + \omega^5 a_5 + \omega^6 a_6 + \omega^7 a_7 \\
1a_0 + \omega^2 a_1 + \omega^4 a_2 + \omega^6 a_3 + 1a_4 + \omega^2 a_5 + \omega^4 a_6 + \omega^6 a_7 \\
1a_0 + \omega^3 a_1 + \omega^6 a_2 + \omega^1 a_3 + \omega^4 a_4 + \omega^7 a_5 + \omega^2 a_6 + \omega^5 a_7 \\
1a_0 + \omega^4 a_1 + 1a_2 + \omega^4 a_3 + 1a_4 + \omega^4 a_5 + 1a_6 + \omega^4 a_7 \\
1a_0 + \omega^5 a_1 + \omega^2 a_2 + \omega^7 a_3 + \omega^4 a_4 + \omega^1 a_5 + \omega^6 a_6 + \omega^3 a_7 \\
1a_0 + \omega^6 a_1 + \omega^4 a_2 + \omega^2 a_3 + 1a_4 + \omega^6 a_5 + \omega^4 a_6 + \omega^2 a_7 \\
1a_0 + \omega^7 a_1 + \omega^6 a_2 + \omega^5 a_3 + \omega^4 a_4 + \omega^3 a_5 + \omega^2 a_6 + \omega^1 a_7
\end{pmatrix}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}
=
$$

$$
\begin{pmatrix}
1a_0 + 1a_2 + 1a_4 + 1a_6 + 1a_1 + 1a_3 + 1a_5 + 1a_7 \\
1a_0 + \omega^2 a_2 + \omega^4 a_4 + \omega^6 a_6 + \omega^1 a_1 + \omega^3 a_3 + \omega^5 a_5 + \omega^7 a_7 \\
1a_0 + \omega^4 a_2 + 1a_4 + \omega^4 a_6 + \omega^2 a_1 + \omega^6 a_3 + \omega^2 a_5 + \omega^6 a_7 \\
1a_0 + \omega^6 a_2 + \omega^4 a_4 + \omega^2 a_6 + \omega^3 a_1 + \omega^1 a_3 + \omega^7 a_5 + \omega^5 a_7 \\
1a_0 + 1a_2 + 1a_4 + 1a_6 + \omega^4 a_1 + \omega^4 a_3 + \omega^4 a_5 + \omega^4 a_7 \\
1a_0 + \omega^2 a_2 + \omega^4 a_4 + \omega^6 a_6 + \omega^5 a_1 + \omega^7 a_3 + \omega^1 a_5 + \omega^3 a_7 \\
1a_0 + \omega^4 a_2 + 1a_4 + \omega^4 a_6 + \omega^6 a_1 + \omega^2 a_3 + \omega^6 a_5 + \omega^2 a_7 \\
1a_0 + \omega^6 a_2 + \omega^4 a_4 + \omega^2 a_6 + \omega^7 a_1 + \omega^5 a_3 + \omega^3 a_5 + \omega^1 a_7
\end{pmatrix}
$$

$$\begin{pmatrix} 1a_0 + 1a_1 + 1a_2 + 1a_3 + 1a_4 + 1a_5 + 1a_6 + 1a_7 \\ 1a_0 + \omega^1 a_1 + \omega^2 a_2 + \omega^3 a_3 + \omega^4 a_4 + \omega^5 a_5 + \omega^6 a_6 + \omega^7 a_7 \\ 1a_0 + \omega^2 a_1 + \omega^4 a_2 + \omega^6 a_3 + 1a_4 + \omega^2 a_5 + \omega^4 a_6 + \omega^6 a_7 \\ 1a_0 + \omega^3 a_1 + \omega^6 a_2 + \omega^1 a_3 + \omega^4 a_4 + \omega^7 a_5 + \omega^2 a_6 + \omega^5 a_7 \\ 1a_0 + \omega^4 a_1 + 1a_2 + \omega^4 a_3 + 1a_4 + \omega^4 a_5 + 1a_6 + \omega^4 a_7 \\ 1a_0 + \omega^5 a_1 + \omega^2 a_2 + \omega^7 a_3 + \omega^4 a_4 + \omega^1 a_5 + \omega^6 a_6 + \omega^3 a_7 \\ 1a_0 + \omega^6 a_1 + \omega^4 a_2 + \omega^2 a_3 + 1a_4 + \omega^6 a_5 + \omega^4 a_6 + \omega^2 a_7 \\ 1a_0 + \omega^7 a_1 + \omega^6 a_2 + \omega^5 a_3 + \omega^4 a_4 + \omega^3 a_5 + \omega^2 a_6 + \omega^1 a_7 \end{pmatrix}$$

$$\Updownarrow$$

各行は、足し算の順序が違っている
だけなので、等しい。

$$\begin{pmatrix} 1a_0 + 1a_2 + 1a_4 + 1a_6 + 1a_1 + 1a_3 + 1a_5 + 1a_7 \\ 1a_0 + \omega^2 a_2 + \omega^4 a_4 + \omega^6 a_6 + \omega^1 a_1 + \omega^3 a_3 + \omega^5 a_5 + \omega^7 a_7 \\ 1a_0 + \omega^4 a_2 + 1a_4 + \omega^4 a_6 + \omega^2 a_1 + \omega^6 a_3 + \omega^2 a_5 + \omega^6 a_7 \\ 1a_0 + \omega^6 a_2 + \omega^4 a_4 + \omega^2 a_6 + \omega^3 a_1 + \omega^1 a_3 + \omega^7 a_5 + \omega^5 a_7 \\ 1a_0 + 1a_2 + 1a_4 + 1a_6 + \omega^4 a_1 + \omega^4 a_3 + \omega^4 a_5 + \omega^4 a_7 \\ 1a_0 + \omega^2 a_2 + \omega^4 a_4 + \omega^6 a_6 + \omega^5 a_1 + \omega^7 a_3 + \omega^1 a_5 + \omega^3 a_7 \\ 1a_0 + \omega^4 a_2 + 1a_4 + \omega^4 a_6 + \omega^6 a_1 + \omega^2 a_3 + \omega^6 a_5 + \omega^2 a_7 \\ 1a_0 + \omega^6 a_2 + \omega^4 a_4 + \omega^2 a_6 + \omega^7 a_1 + \omega^5 a_3 + \omega^3 a_5 + \omega^1 a_7 \end{pmatrix}$$

この式は、実は、次の式に等しい

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} =
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$

偶数行

奇数行

なぜなら、行列Mと列ベクトルAの積は、
Aの行を入れ替えても変わらないから

$$\begin{pmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,n-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,n-1} \\ m_{2,0} & m_{2,1} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ & & \cdots & \\ m_{n-2,0} & m_{n-2,1} & \cdots & m_{n-2,n-1} \\ m_{n-1,0} & m_{n-2,1} & \cdots & m_{n-1,n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ \boxed{a_j} \\ \vdots \\ \boxed{a_k} \\ \vdots \\ a_{n-1} \end{pmatrix} =$$

$$\begin{pmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,n-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,n-1} \\ m_{2,0} & m_{2,1} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ & & \cdots & \\ m_{n-2,0} & m_{n-2,1} & \cdots & m_{n-2,n-1} \\ m_{n-1,0} & m_{n-2,1} & \cdots & m_{n-1,n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ \boxed{a_k} \\ \vdots \\ \boxed{a_j} \\ \vdots \\ a_{n-1} \end{pmatrix}$$

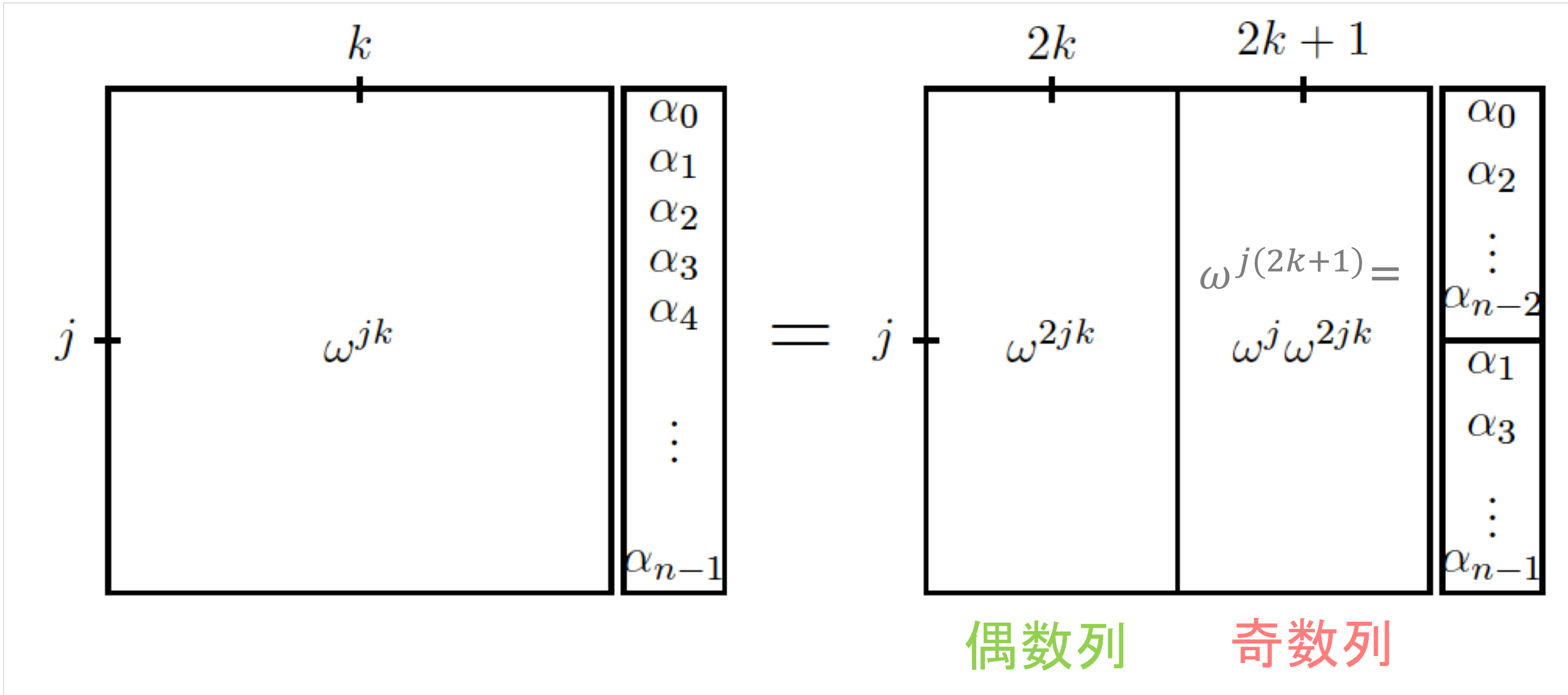
$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix}
a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7
\end{pmatrix}
=
$$

$$
\begin{pmatrix}
1a_0 + 1a_2 + 1a_4 + 1a_6 + 1a_1 + 1a_3 + 1a_5 + 1a_7 \\
1a_0 + \omega^2 a_2 + \omega^4 a_4 + \omega^6 a_6 + \omega^1 a_1 + \omega^3 a_3 + \omega^5 a_5 + \omega^7 a_7 \\
1a_0 + \omega^4 a_2 + 1a_4 + \omega^4 a_6 + \omega^2 a_1 + \omega^6 a_3 + \omega^2 a_5 + \omega^6 a_7 \\
1a_0 + \omega^6 a_2 + \omega^4 a_4 + \omega^2 a_6 + \omega^3 a_1 + \omega^1 a_3 + \omega^7 a_5 + \omega^5 a_7 \\
1a_0 + 1a_2 + 1a_4 + 1a_6 + \omega^4 a_1 + \omega^4 a_3 + \omega^4 a_5 + \omega^4 a_7 \\
1a_0 + \omega^2 a_2 + \omega^4 a_4 + \omega^6 a_6 + \omega^5 a_1 + \omega^7 a_3 + \omega^1 a_5 + \omega^3 a_7 \\
1a_0 + \omega^4 a_2 + 1a_4 + \omega^4 a_6 + \omega^6 a_1 + \omega^2 a_3 + \omega^6 a_5 + \omega^2 a_7 \\
1a_0 + \omega^6 a_2 + \omega^4 a_4 + \omega^2 a_6 + \omega^7 a_1 + \omega^5 a_3 + \omega^3 a_5 + \omega^1 a_7
\end{pmatrix}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} =
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$

ここまで出来た！

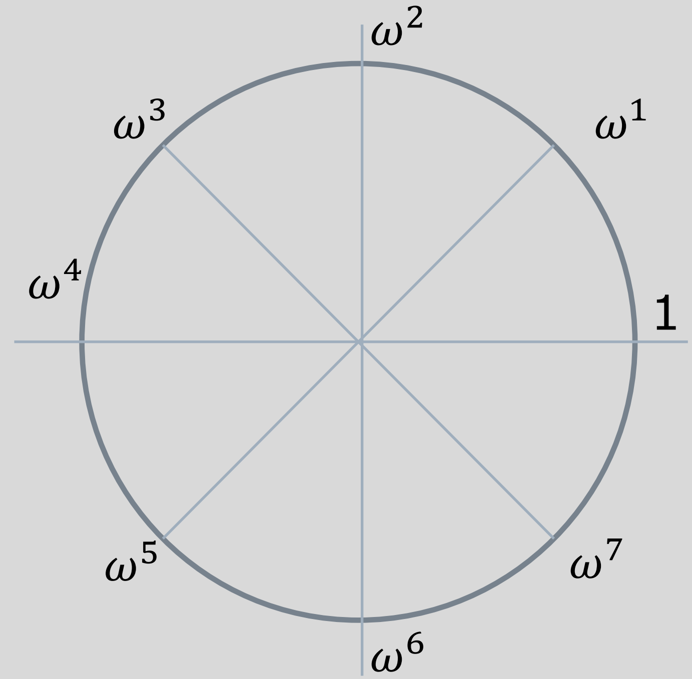


この関係は、後でQFTの導出に利用される

$$\boxed{\hat{f}(j) = \left(F_{M/2}\overrightarrow{f_{\mathrm{even}}}\right)(j) + \omega^j \left(F_{M/2}\overrightarrow{f_{\mathrm{odd}}}\right)(j)}$$

DFT$_8$での計算例

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$



$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$

$$
\begin{aligned}
1 &= -\omega^4 \\
\omega^1 &= -\omega^5 \\
\omega^2 &= -\omega^6 \\
\omega^3 &= -\omega^7
\end{aligned}
$$

$$
\begin{aligned}
\omega^2 &= i \\
\omega^4 &= -1 \\
\omega^6 &= -i
\end{aligned}
$$

DFT$_8$での計算例

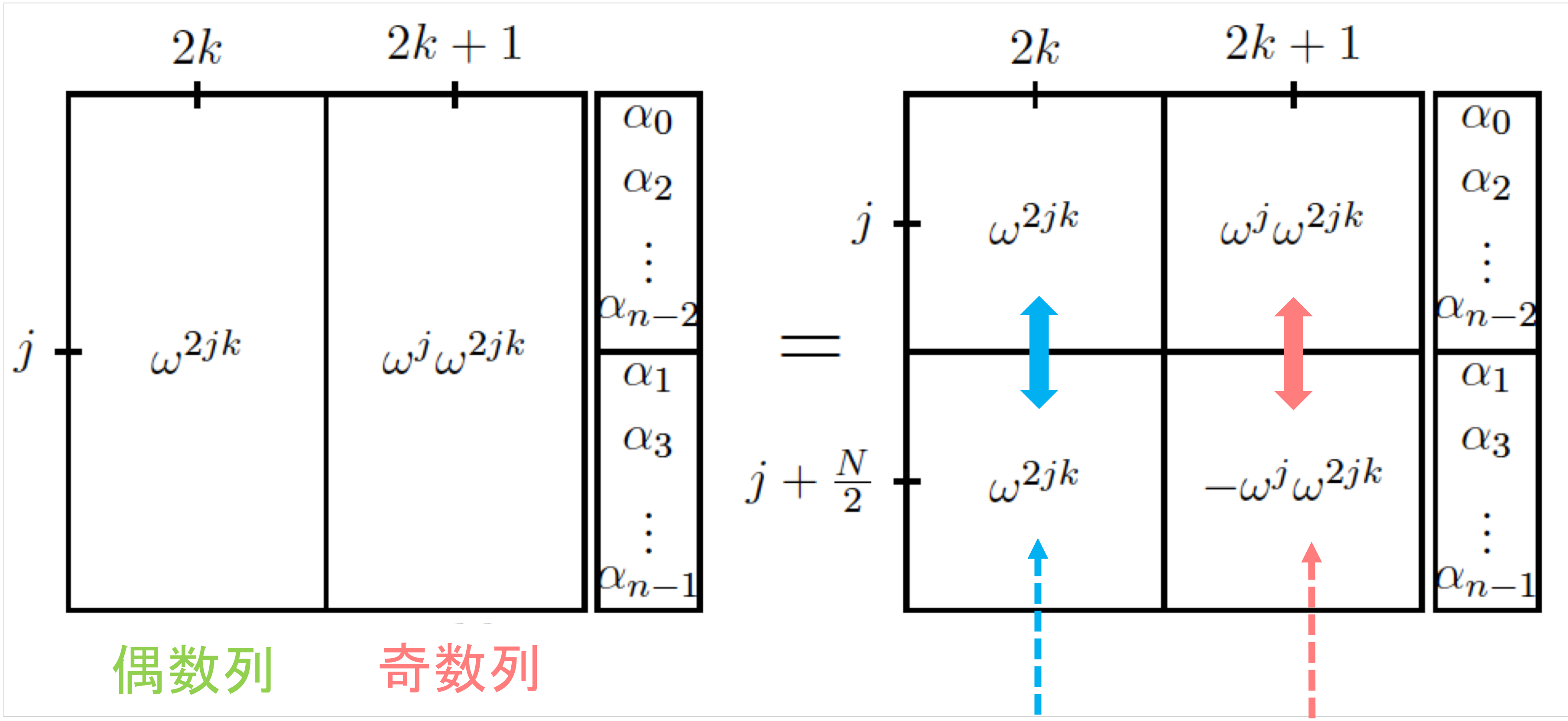
$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega & \omega^3 \\
1 & \omega^4 & 1 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$

$$
QFT_4 = \frac{1}{2}\begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & i & -1 & -i \\
1 & -1 & 1 & -1 \\
1 & -i & -1 & -i
\end{pmatrix}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 & \omega^7 \\
1 & \omega^4 & 1 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega & \omega^7 & \omega^5 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & \omega^2 & \omega^4 & \omega^6 & -\omega & -\omega^3 & -\omega^5 & -\omega^7 \\
1 & \omega^4 & 1 & \omega^4 & -\omega^2 & -\omega^6 & -\omega^2 & -\omega^6 \\
1 & \omega^6 & \omega^4 & \omega^2 & -\omega^3 & -\omega & -\omega^7 & -\omega^5
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$

$$1 = -\omega^4$$
$$\omega^1 = -\omega^5$$
$$\omega^2 = -\omega^6$$
$$\omega^3 = -\omega^7$$

$$\omega^2 = i$$
$$\omega^4 = -1$$
$$\omega^6 = -i$$



$$\omega^{\left(j+\frac{N}{2}\right)2k} = \omega^{2jk+Nk} = \omega^{2jk}\omega^{Nk} = \omega^{2jk}$$

$$\omega^{N} = 1$$
$$\omega^{\frac{N}{2}} = -1$$

$$\omega^{\left(j+\frac{N}{2}\right)(2k+1)} = \omega^{2jk+j+Nk+\frac{N}{2}}$$
$$= \omega^{2jk}\omega^{j}\omega^{Nk}\omega^{\frac{N}{2}} = \omega^{j}\omega^{2jk}\,\omega^{\frac{N}{2}} = -\omega^{j}\omega^{2jk}$$

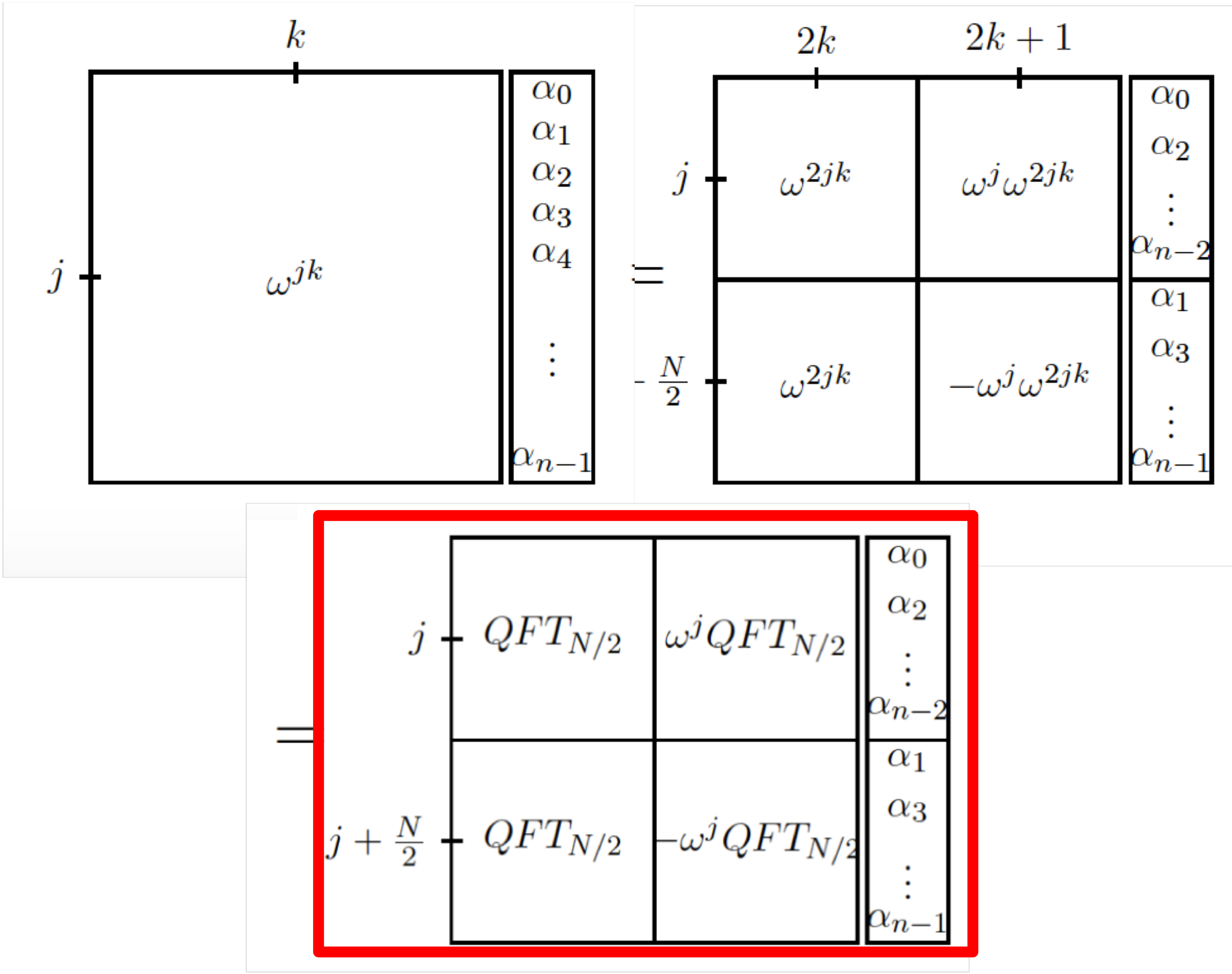
DFT_8は、次のようにDFT_4に分解できる

$$
\mathrm{D}FT_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & -i \end{pmatrix}
$$

$$
\left(\begin{array}{cccc|c}
1 & 1 & 1 & 1 & \\
1 & i & -1 & -i & \omega^j \mathrm{D}FT_4 \\
1 & -1 & 1 & -1 & \\
1 & -i & -1 & -i & \\
\hline
1 & 1 & 1 & 1 & \\
1 & i & -1 & -i & -\omega^j \mathrm{D}FT_4 \\
1 & -1 & 1 & -1 & \\
1 & -i & -1 & -i &
\end{array}\right)
\begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ a_1 \\ a_3 \\ a_5 \\ a_7 \end{pmatrix}
$$

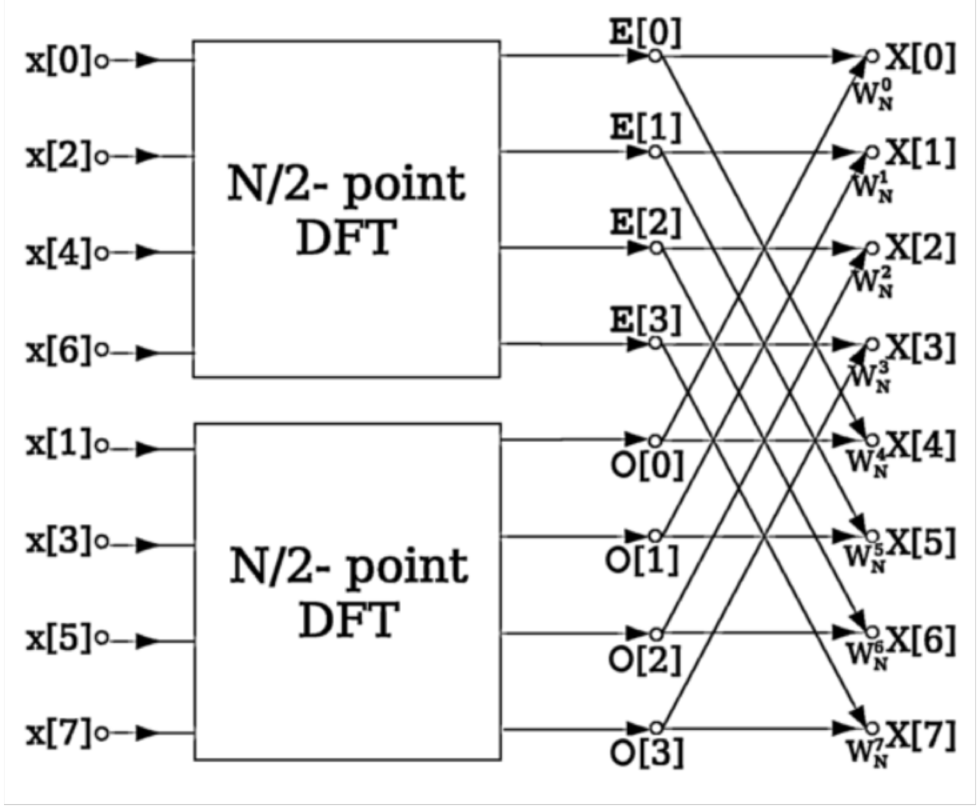
$$
\begin{aligned}
1 &= -\omega^4 \\
\omega^1 &= -\omega^5 \\
\omega^2 &= -\omega^6 \\
\omega^3 &= -\omega^7
\end{aligned}
$$

$$
\boxed{\begin{aligned}
\omega^2 &= i \\
\omega^4 &= -1 \\
\omega^6 &= -i
\end{aligned}}
$$


k
j
ω^{jk}
α_0 α_1 α_2 α_3 α_4 $\vdots$ α_{n-1}
$=$
$2k$ $2k+1$
j
$\frac{N}{2}$
ω^{2jk} $\omega^j\omega^{2jk}$
ω^{2jk} $-\omega^j\omega^{2jk}$
α_0 α_2 $\vdots$ α_{n-2}
α_1 α_3 $\vdots$ α_{n-1}
$=$
j
$j+\frac{N}{2}$
$QFT_{N/2}$ $\omega^j QFT_{N/2}$
$QFT_{N/2}$ $-\omega^j QFT_{N/2}$
α_0 α_2 $\vdots$ α_{n-2}
α_1 α_3 $\vdots$ α_{n-1}

Cooley–Tukey FFT algorithm

$$
\begin{aligned}
X_k &= E_k + e^{-\frac{2\pi i}{N}k} O_k \\
X_{k+\frac{N}{2}} &= E_k - e^{-\frac{2\pi i}{N}k} O_k
\end{aligned}
$$

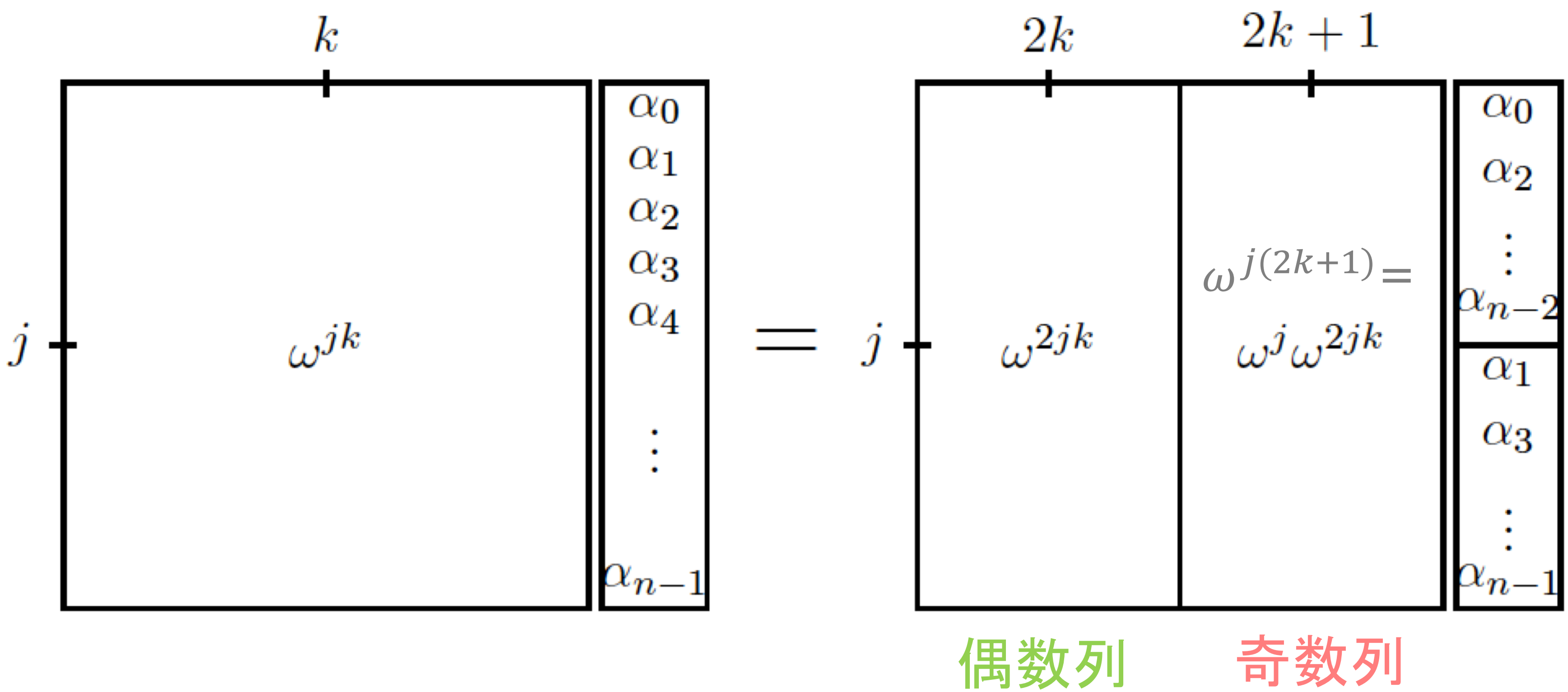


Quantum Fourier Transform

FFTからQFTへ

FFTからQFTの回路を考えるには、この関係がヒントになる

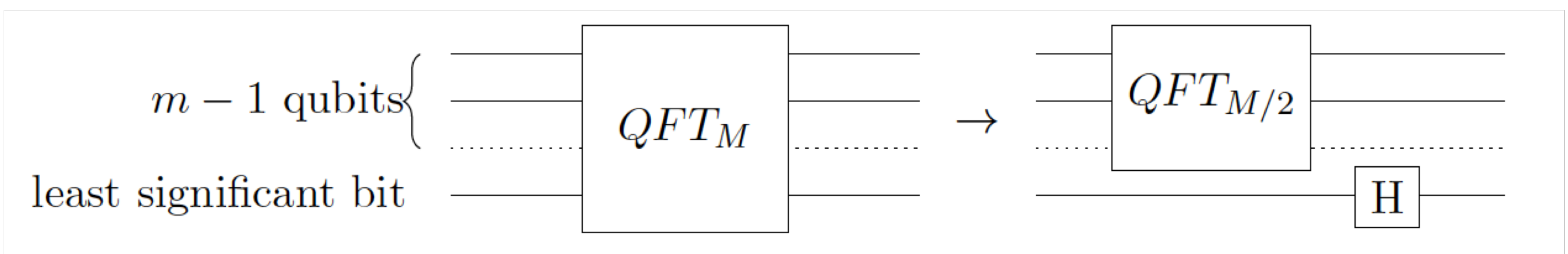
最後の「バタフライ」のところまで行かない



$$\hat{f}(j) = \left(F_{M/2}\overrightarrow{f_{\text{even}}}\right)(j) + \omega^j \left(F_{M/2}\overrightarrow{f_{\text{odd}}}\right)(j)$$

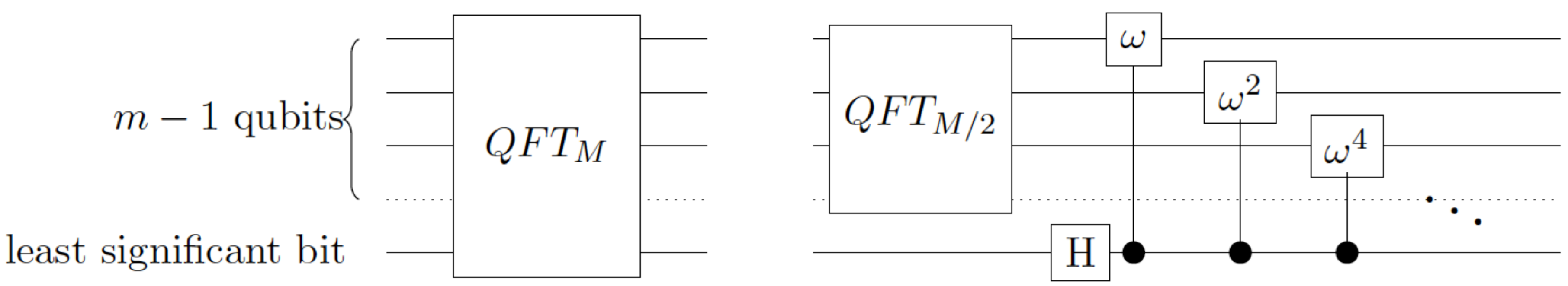
QFTをQuantum Gateで実装する

- QFTは、FFTに動機付けられているので、同じようなステップで進むのだが、いくつかの点で実装が異なっている。
- FFTでは、偶数部分と奇数部分に分け、奇数項にフェーズω^jを掛けた。QFTでは、このステップは、かなり簡単になる。偶数項も奇数項も重ね合わせの状態にある。奇数項は最小bitが1の項で、奇数項は最小bitが0である。それゆえ、奇数項にも偶数項にも、$QFT_{M/2}$を適用できる。
- こうやればいい。単に、m-1 msb(most significant bit)に$QFT_{M/2}$を適用して、lsb（least significant bit）にアダマールを適用して、再結合すればいい。

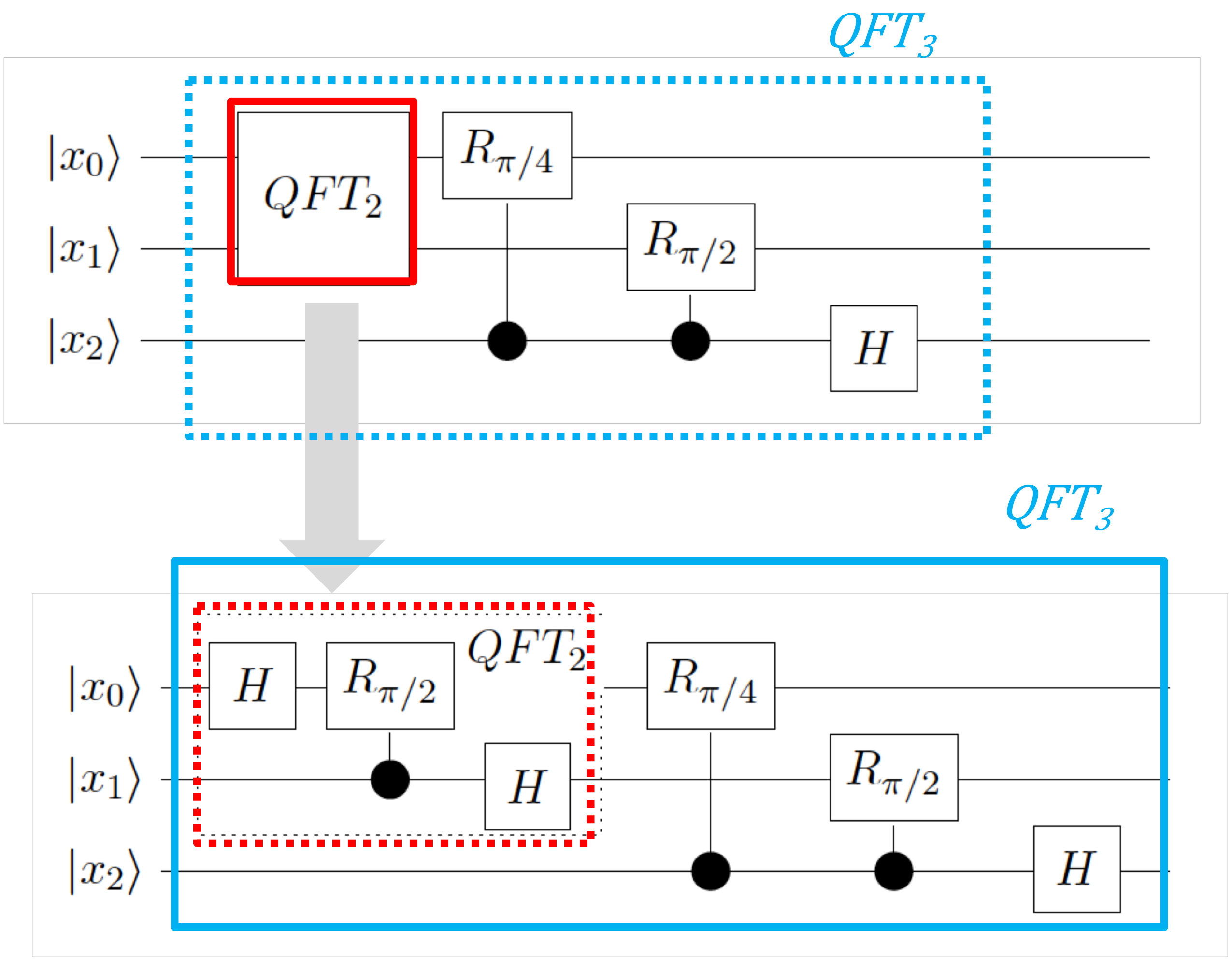


Controll Phase Shift

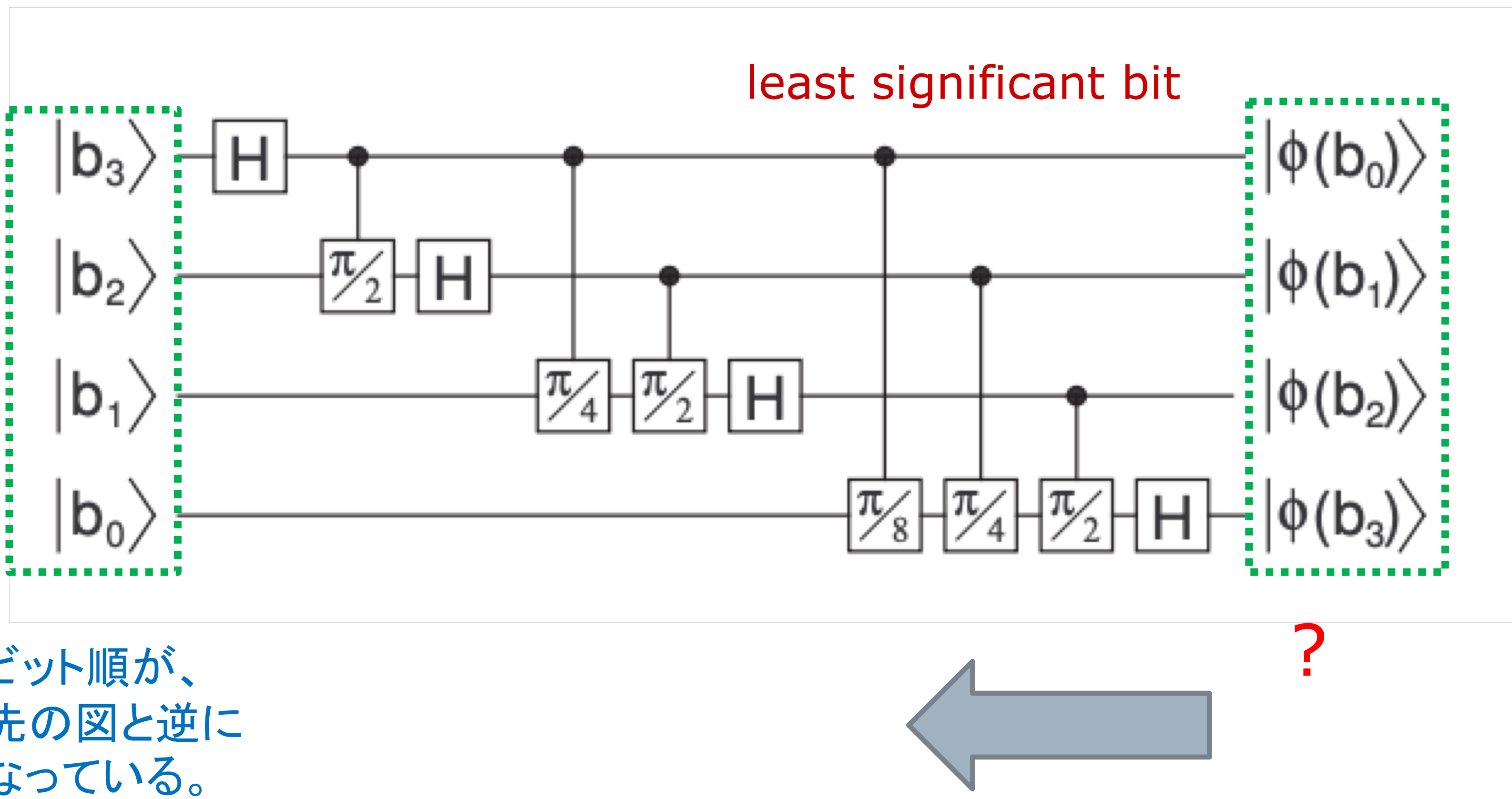
- フェーズをかける作用だが、それぞれの奇数項 j に、ω^j を掛ける。奇数項は最小bitが1の項で、奇数項は最小bitが0である。lsbが 1の時のみ、偶数項にには何もしないように、コントロール-Phase Shiftを使う。
- コントロール-Phase Shiftで利用されるフェーズω^jは、k番目のbitについて言えば、$j = 2^k$ である。



QFT$_3$ の回路の例

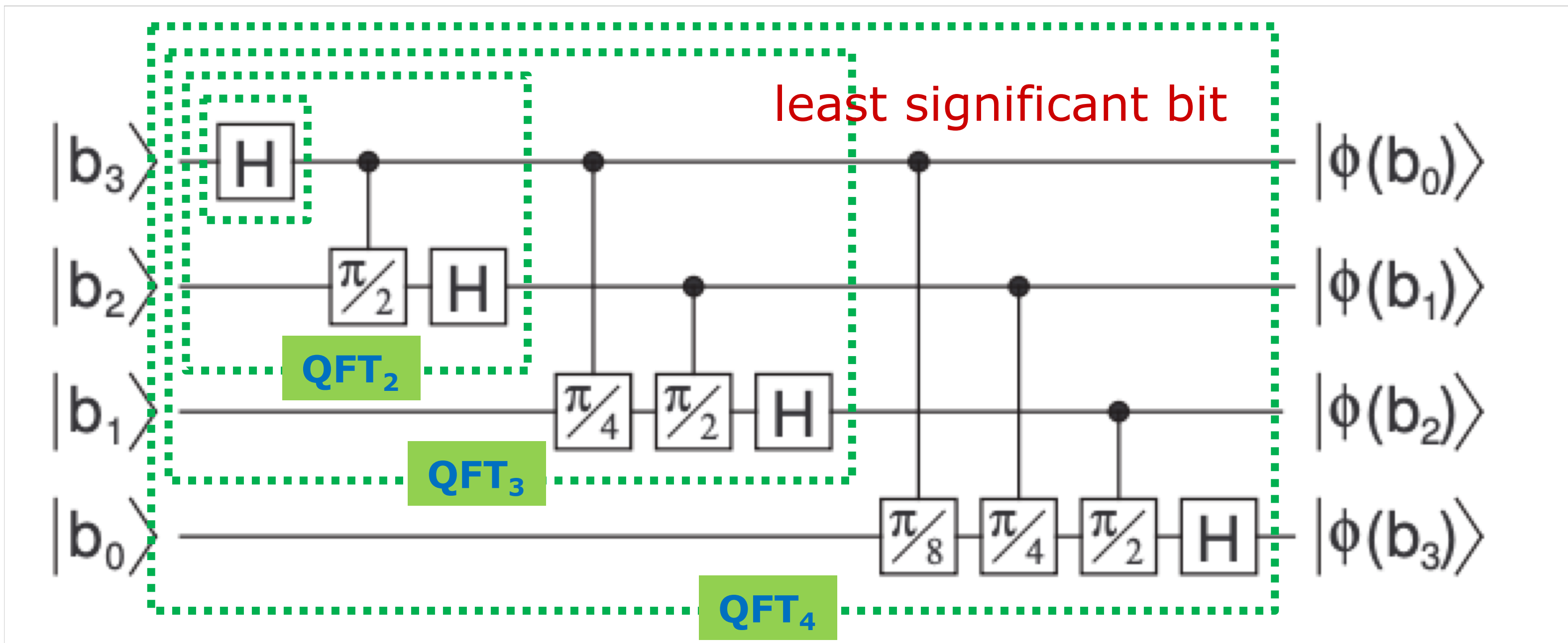


$QFT_4^\dagger$ の回路の例



ビット順が、
先の図と逆に
なっている。

$QFT_4^\dagger$ の回路の例



ビット順が、
先の図と逆に
なっている。

QFTの定義から、回路を導出する

Discrete Fourier Transformと Quantum Fourier Transform

- Discrete Fourier Transform

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i jk/N} .$$

同じ形をしている。

- Quantum Fourier Transform

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

$$\sum_{j=0}^{N-1} x_j |j\rangle \longrightarrow \sum_{k=0}^{N-1} y_k |k\rangle ,$$

ただし、|j>, |k>は、
|1011..01>のような
形をしている。
n個の基底のテンソル積。
組み合わせとしては、
2^n個ある。

次のページで見るように
これを二進数としてみる。

Quantum Fourier Transform

- $N=2^n$として、n qubitの量子コンピュータの計算基底を $|0\rangle, |1\rangle, \ldots, |2^n-1\rangle$ とする。
  この時、$j=j_1j_2j_3\ldots j_n$で、$j = j_1 2^{n-1} + j_2 2^{n-2} + \ldots + j_n 2^0$ を表す。
  同様に、$0.j_l j_{l+1} \ldots j_m$で、$j_l/2 + j_{l+1}/4 + \ldots + j_m/2^{m-l+1}$を表す。

  この時、QFTは、次のように表現できる。

$$|j_1, \ldots, j_n\rangle \rightarrow$$

$$\boxed{\frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)}{2^{n/2}}}$$

以下、それを説明しよう。　　QFTの基本公式

少し予行練習
Phase Shift ゲート R_kをシリアルに適用する

$$Phase\ Shift\ Gate\ \ R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

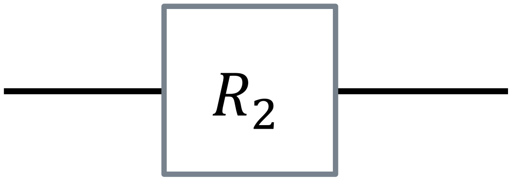
$$R_1 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

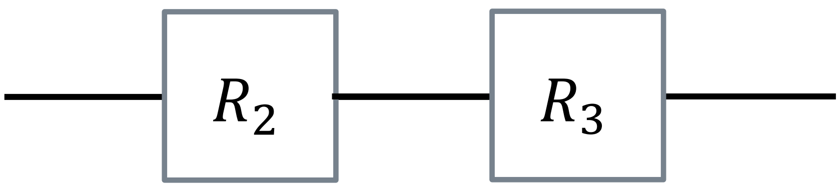
$$R_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} i \end{pmatrix}$$

$$R_4 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^4} \end{pmatrix} = \cdots$$

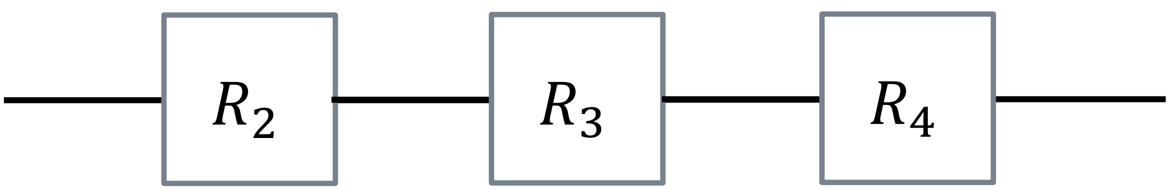
少し予行練習
Phase Shift ゲート R_kをシリアルに適用する



$$\underset{R_2}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}} |\psi>$$

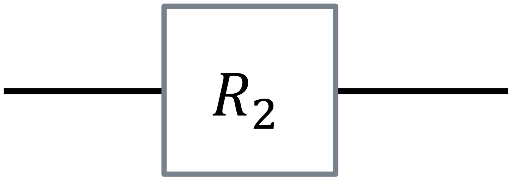


$$\underset{R_3}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix}} \underset{R_2}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}} |\psi>$$



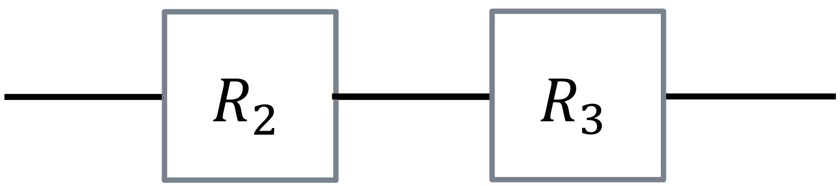
$$\underset{R_4}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/4} \end{pmatrix}} \underset{R_3}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^3} \end{pmatrix}} \underset{R_2}{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix}} |\psi>$$

少し予行練習
Phase Shift ゲート R_kをシリアルに適用する

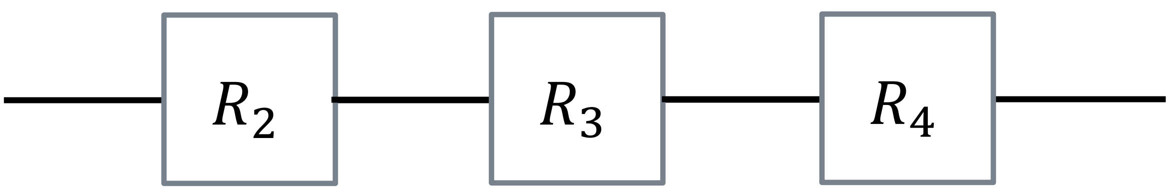


$$\underset{R_2}{\begin{pmatrix}1 & 0\\0 & e^{2\pi i/2^2}\end{pmatrix}}|\psi>$$

$$\begin{pmatrix}1 & 0\\0 & e^{2\pi i\left(\frac{1}{2^1}+\frac{1}{2^2}\right)}\end{pmatrix}$$



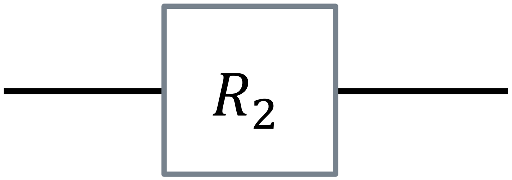
$$\underset{R_3}{\begin{pmatrix}1 & 0\\0 & e^{2\pi i/2^3}\end{pmatrix}}\underset{R_2}{\begin{pmatrix}1 & 0\\0 & e^{2\pi i/2^2}\end{pmatrix}}|\psi>$$



$$\begin{pmatrix}1 & 0\\0 & e^{2\pi i\left(\frac{1}{2^1}+\frac{1}{2^2}+\frac{1}{2^3}\right)}\end{pmatrix}$$

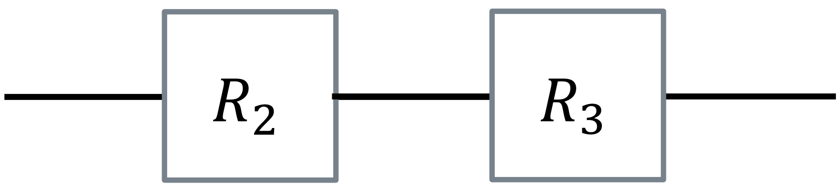
$$\underset{R_4}{\begin{pmatrix}1 & 0\\0 & e^{2\pi i/2^4}\end{pmatrix}}\underset{R_3}{\begin{pmatrix}1 & 0\\0 & e^{2\pi i/2^3}\end{pmatrix}}\underset{R_2}{\begin{pmatrix}1 & 0\\0 & e^{2\pi i/2^2}\end{pmatrix}}|\psi>$$

少し予行練習
Phase Shift ゲート R_kをシリアルに適用する

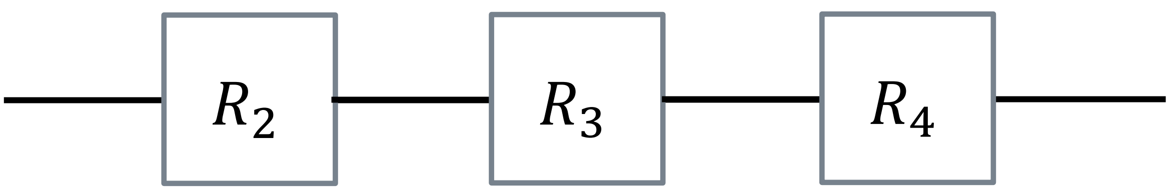


$$\underset{R_2}{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^2}\end{pmatrix}}|\psi>$$

$$\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i\ 0.11}\end{pmatrix}$$



$$\underset{R_3}{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^3}\end{pmatrix}}\underset{R_2}{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^2}\end{pmatrix}}|\psi>$$



$$\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i\ 0.111}\end{pmatrix}$$

$$\underset{R_4}{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^4}\end{pmatrix}}\underset{R_3}{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^3}\end{pmatrix}}\underset{R_2}{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^2}\end{pmatrix}}|\psi>$$

$$|j\rangle \rightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j \boxed{k/2^n}} |k\rangle$$

QFTの定義から

$$k = k_1 k_2 \ldots k_n = k_1 2^{n-1} + k_2 2^{n-2} \ldots + k_n 2^0$$

$$k/2^n = \frac{k_1}{2} + \frac{k_2}{2^2} + \cdots + \frac{k_n}{2^n}$$

$$= \frac{1}{2^{n/2}} \underbrace{\overset{0/1}{\boxed{\sum_{k_1=0}^{1}}} \cdots \overset{0/1}{\boxed{\sum_{k_n=0}^{1}}}}_{n\text{個}} e^{2\pi i j \left(\boxed{\sum_{l=1}^{n} k_l 2^{-l}}\right)} |k_1 \ldots k_n\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \boxed{\bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle}$$

$$= \frac{1}{2^{n/2}} \underbrace{\sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1}}_{n\text{個}} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle$$

$$= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle\right]$$

$k_l = 0$ の時と

$k_l = 1$ の時

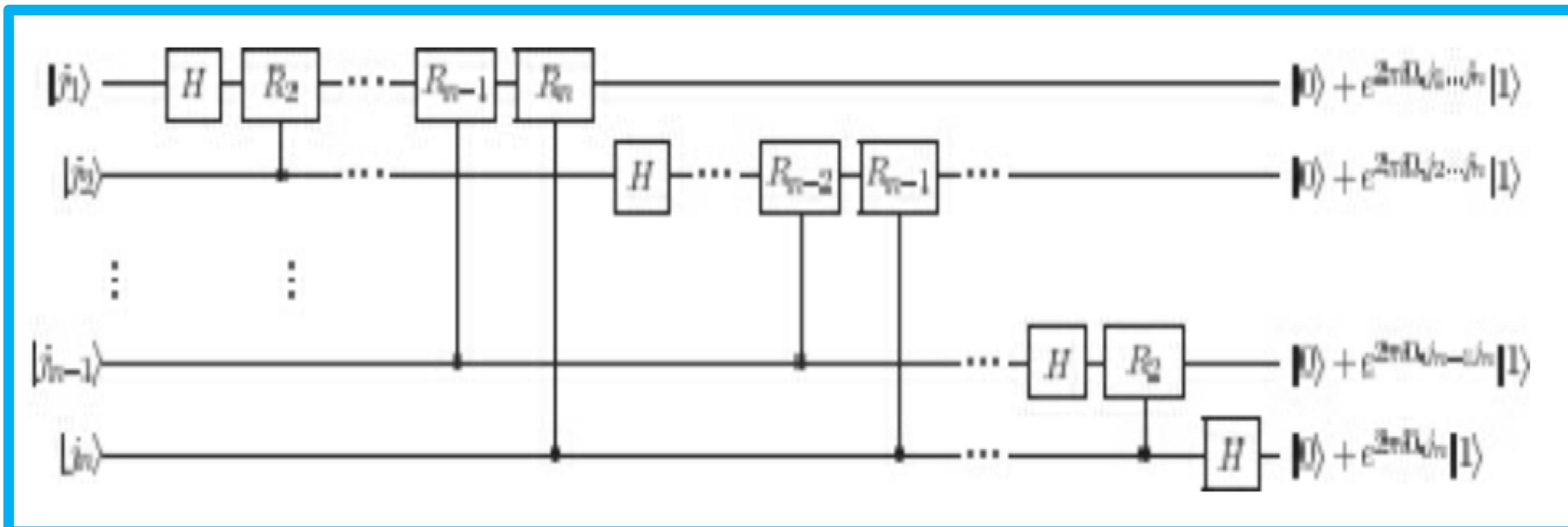
$j=j_1j_2\ldots j_n=j_12^{n-1}+j_22^{n-2}\ldots+j_n2^0$

$$= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle\right]$$

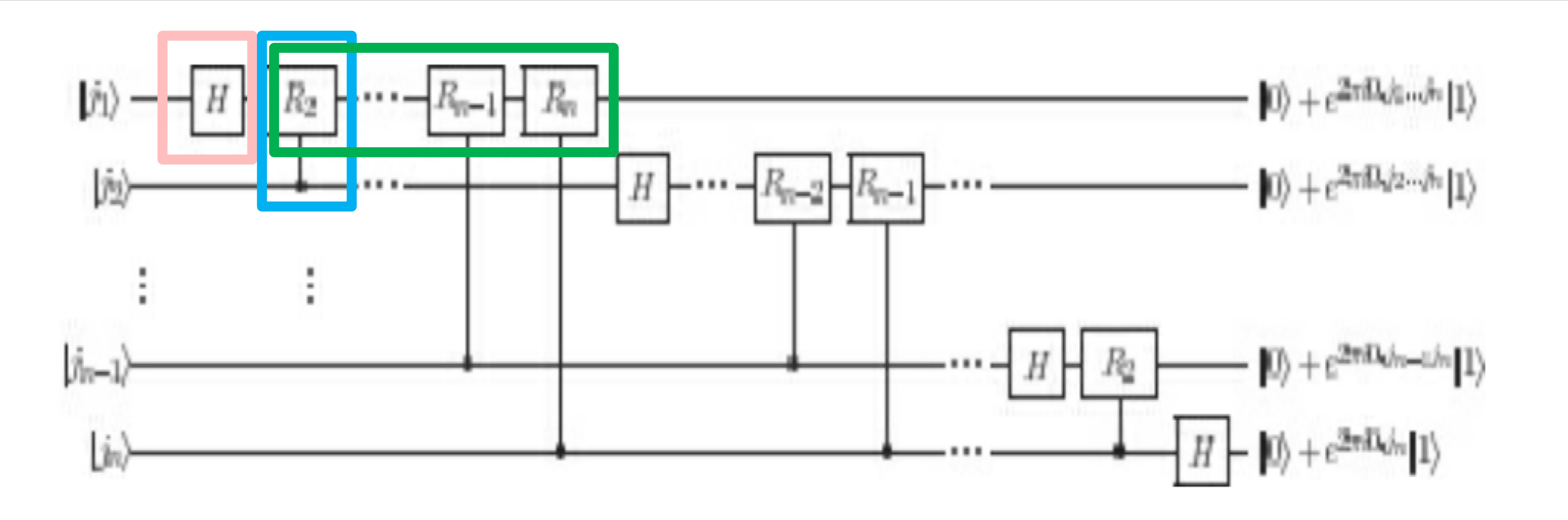
$$= \boxed{\frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)}{2^{n/2}}}.$$

QFTの基本公式

QFTの基本回路



ただし、 $R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$



アダマールを |$j_1 j_2 \ldots j_n$> の最初のビットj_1に適用すると、次の状態になる。

$$\frac{1}{2^{1/2}}\left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right)|j_2 \ldots j_n\rangle$$

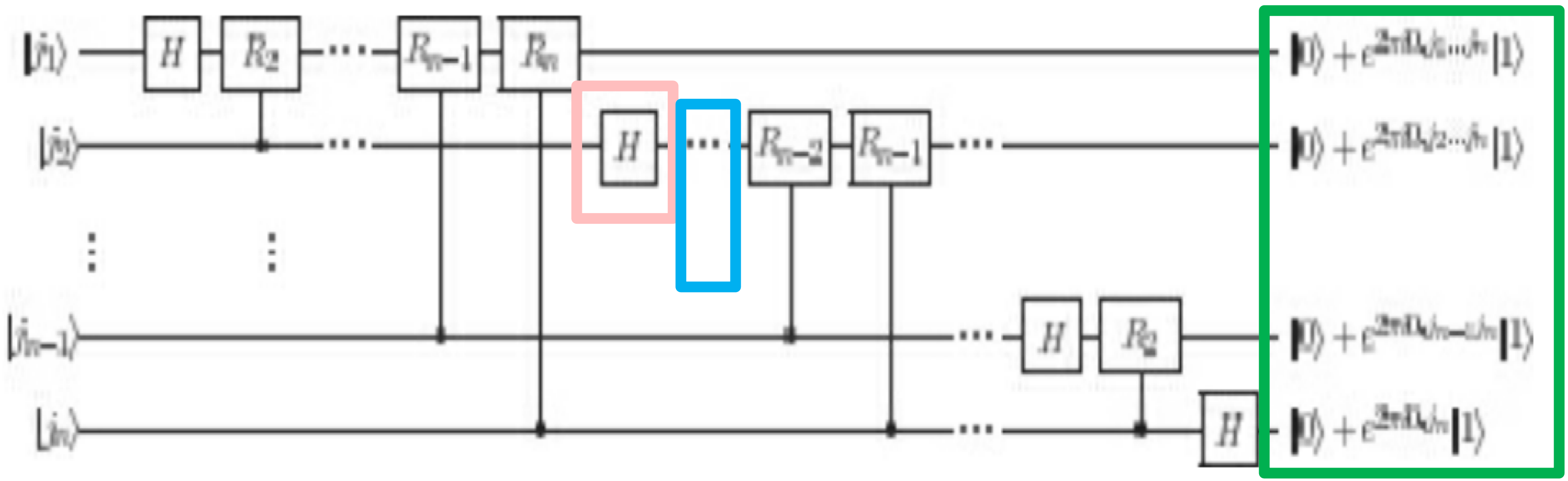
j_1=1 の時 $e^{2\pi i 0.j_1} = -1$、それ以外は、+1

コントロール-R_2ゲートを適用すると、状態は、次のようになる。

$$\frac{1}{2^{1/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right)|j_2 \ldots j_n\rangle$$

コントロール-R3, R4,...Rnゲートを順次適用すると、|1>の係数が変化する。

$$\frac{1}{2^{1/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}|1\rangle\right)|j_2 \ldots j_n\rangle$$



アダマールを |$j_1 j_2 \ldots j_n$> の二番目のビットj_2に適用すると、次の状態になる。

$$\frac{1}{2^{2/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2}|1\rangle\right)|j_3 \ldots j_n\rangle$$

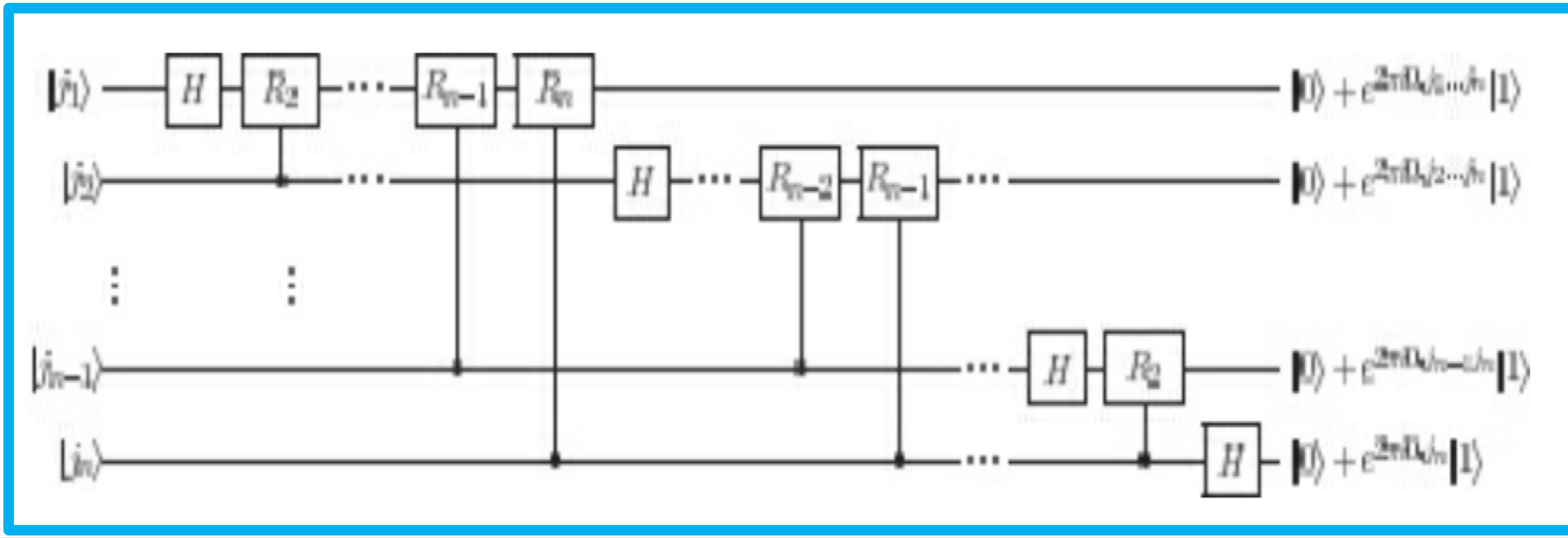
コントロール-R_2ゲートを適用すると、状態は、次のようになる。

$$\frac{1}{2^{2/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle\right)|j_3 \ldots j_n\rangle$$

これらの操作を繰り返すと、最終的に次の状態を得る。

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)$$

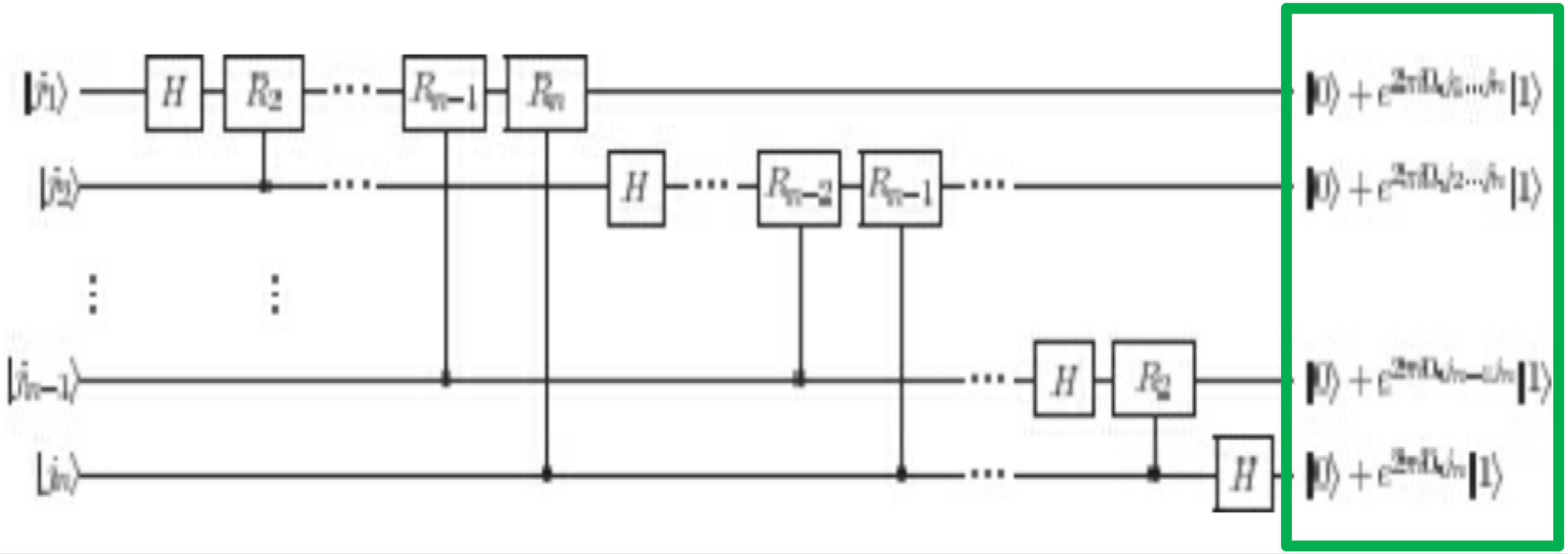
QFTの基本回路



QFTの基本公式

$$
= \frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)}{2^{n/2}}.
$$

一つ問題が



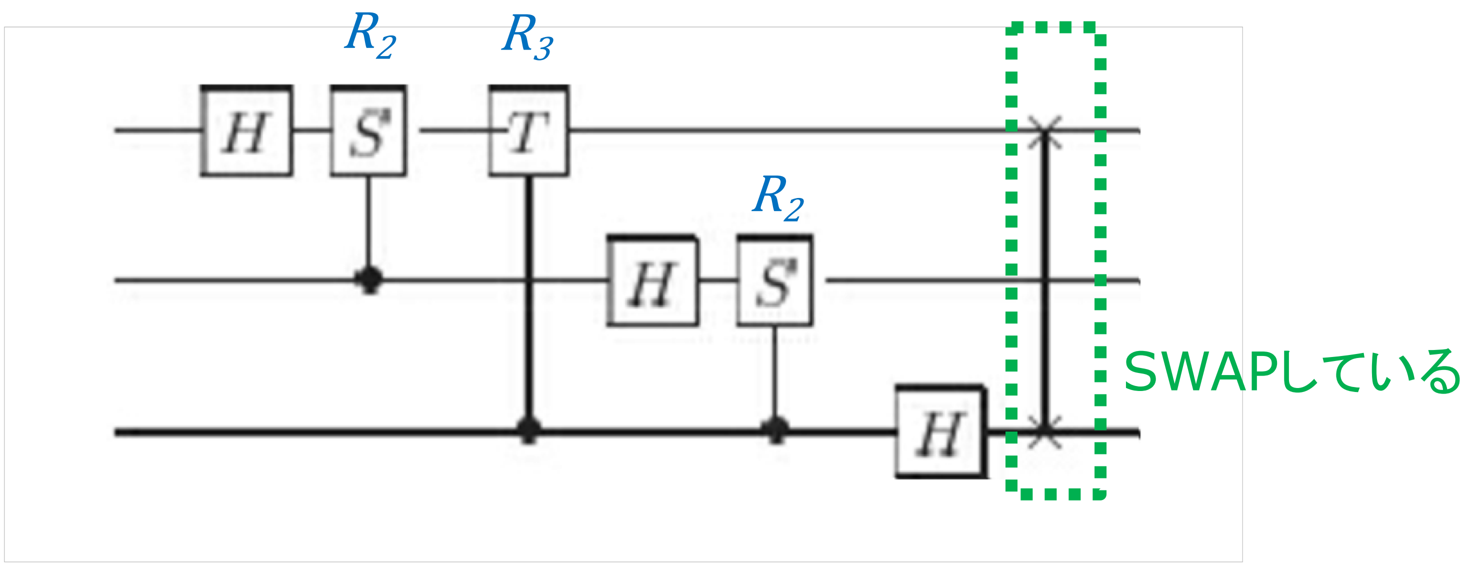
テンソル積の順序が真逆

QFT$_n$($|j_1 j_2 j_3 \cdots j_n>$)

$$= \frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)}{2^{n/2}}.$$

回路の方の最終出力を、逆順にする

3-qubitのQFT回路（再び）



Phase Shift

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix}$$

π/8

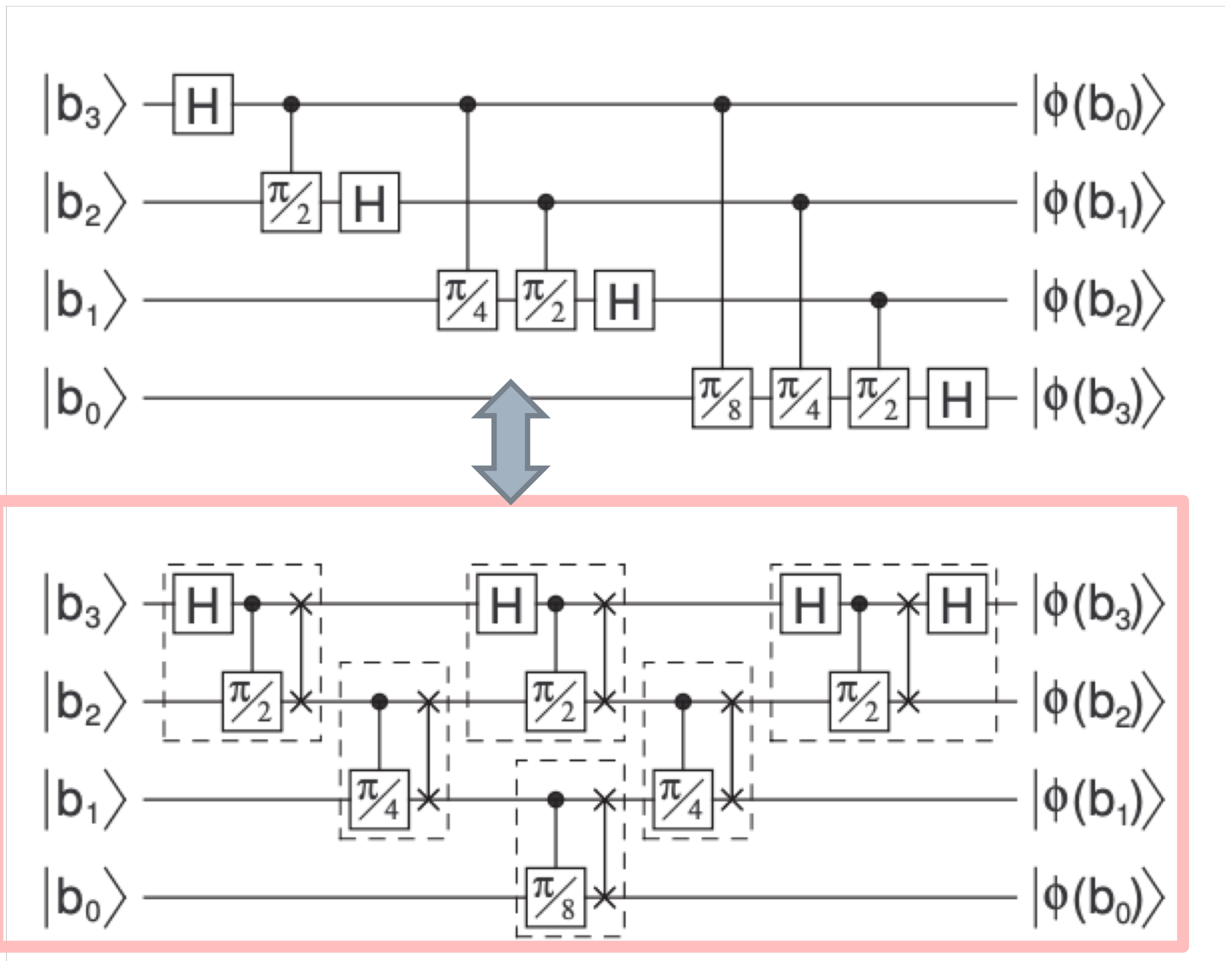
$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

一般のPhase shift $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$

3-qubitのQFTと等価なDFT

$$
\frac{1}{\sqrt{8}}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{bmatrix}
$$

QFT_4 の回路の例（SWAPつき）



Implementation of Shor's Algorithm on a Linear Nearest Neighbour Qubit Array
https://arxiv.org/pdf/quant-ph/0402196.pdf

QFT$_4$ の回路の例（SWAPつき）

```
Circuit:
(0, 0): ─H───@^0.5───×───H────────────@^0.5───────×───H────────────@^0.5───×───H
             │       │                │           │                │       │
(0, 1): ─────@───────×───@^0.25───×───@───────────×───@^0.25───×───@───────×────
                         │        │   │           │   │        │
(1, 0): ─────────────────┼────────┼───@^0.125─────×───┼────────┼────────────────
                         │        │   │           │   │        │
(1, 1): ─────────────────@────────×───@───────────×───@────────×────────────────
```

[Cirq](https://github.com/quantumlib/Cirq)/[examples](https://github.com/quantumlib/Cirq/tree/master/examples)/quantum_fourier_transform.py
https://goo.gl/jwjSV3

単純だけど大事なこと

n-qubitのQFTの入力に、n個の|0>、$|0>^{\otimes n}$を与えると、出力はどうなるか？

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_1 j_2\cdots j_n}|1\rangle\right)$$

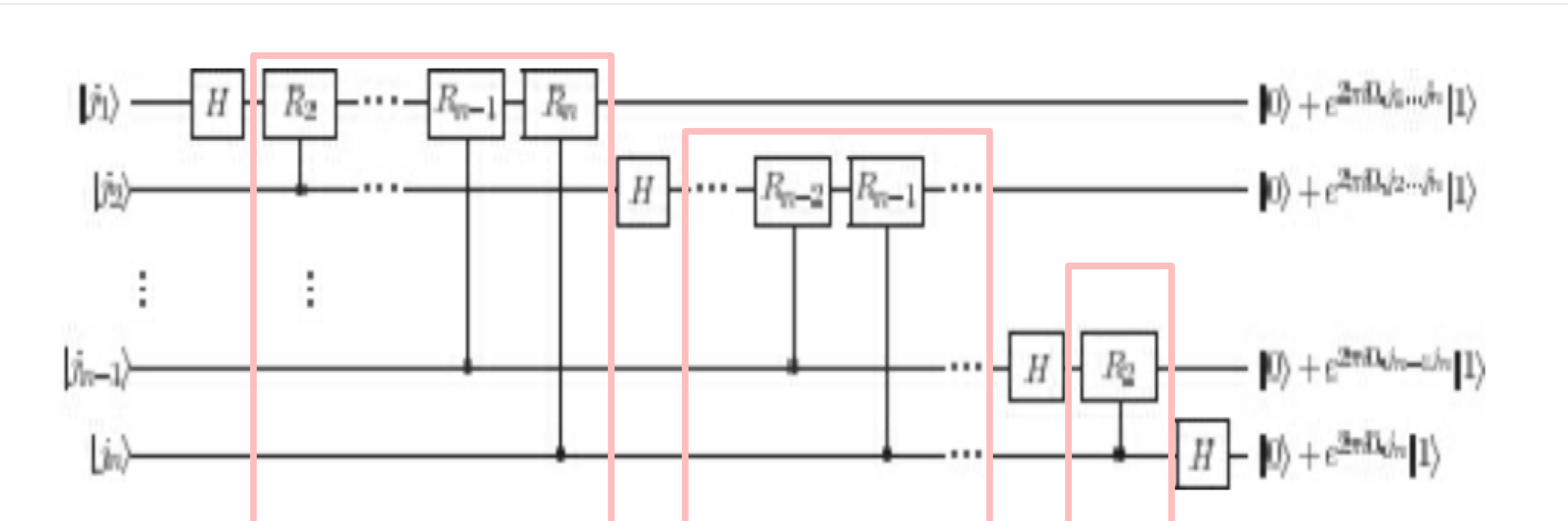
$j_1,j_2,\ldots j_n$は、すべて0だから、|1>にかかっているPhaseは、すべて1になる。
よって、出力は、
$1/\sqrt{N}$ (|0>+|1>)⊗ (|0>+|1>)⊗ … ⊗ (|0>+|1>)になる。

QFTによって、|00…00>+|00…01>+ …|11…11> という、
可能な全ての2^n個の基底の重ね合わせの状態が生まれるということ。

これは、先に「Quantum Parallelism」のところで見た、n個のアダマール
回路の働きに等しい。

そのことは、先の回路図を見てもわかる。

線で囲ったphase-shiftのコントロールR_nが、$j_1, j_2, \ldots j_n$ がすべて0だと働かないことになる。



結局、残ったのは、n個のアダマールである。

Appendix

Period Finding

Period Finding

- 周期的な関数 f を計算するブラックボックスが与えられているとしよう。周期的な関数とは、次のような関数のことだ。
  f(x)=f(y) となるのは、$x \equiv y \pmod r$ の時のみ
  例えば、関数 sin x も cos x も、mod 2π で周期関数だ。
- 「周期探し」の目標は、f（そのブラックボックス）が与えられた時、周期 r を見つけることである。
- 古典的には、この f に何度も値を代入して繰り返しを見つければいいのだが、f が、n ビットの入力を受け付けるのなら、最悪 2^n 回の問い合わせを、このブラックボックスに対して行うことになる。他にもやり方はあるかもしれないが、古典的には、指数関数的な時間が必要になる。

Quantum Period Finding

- 量子コンピュータでは、n個のqubit で、$N=2^n$個の入力の組み合わせの重ね合わせの状態が表現できる。ここで、フーリエ変換の周期・周波数と線形シフトの性質を利用する。
- まず、最初に、線形シフトを含む重ね合わせの状態にアクセスする。そして、フーリエ変換で、線形シフトを除去する。
- こうしたアプローチは、Shorのアルゴリズムの発見以前に、Simonのアルゴリズムとして知られていたものである。

周期的な重ね合わせを準備する

- 最初に次の状態をQFTで準備する。

$$|0\rangle\,|0\rangle \xrightarrow{QFT_N} \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\,|0\rangle$$

- U_fを f を計算するブラックボックスだとする。

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\,|0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\,|f(x)\rangle$$

- ここで、|f(x)>を測定する。
  |f(x)>は、ある値 f(x0)として観測されるのだが、同時に、|x>も、f(x0)の元のイメージに崩壊する。fは、周期的だから、その値は、{x0, x0+r, x0+2r, ... , x0+(N/r -1)r }の重ね合わせになる。

- |f>の観測は、次のような変化を引き起こしたことになる。

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\otimes|f(x)\rangle \xrightarrow{\text{measure}|f\rangle} \sqrt{\frac{r}{N}}\sum_{i=0}^{N/r-1}|ir+x_0\rangle\,|f(x_0)\rangle$$

- 第一レジスターには、関数fの周期 r と同じ周期の重ね合わせの情報が含まれている。
- ただ、このアルゴリズムを繰り返し、第一レジスターを測定しても、異なるx0についての情報が得られるだけである。

フーリエ・サンプル y=kN/r を取得する

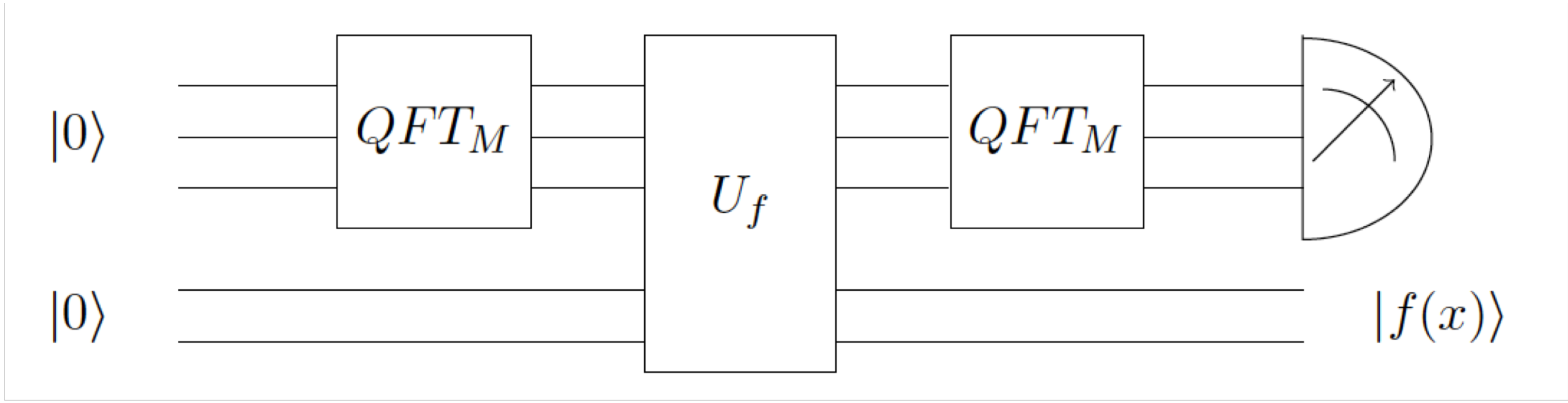
- 次のようなフーリエ変換の性質を思い出そう。
  もし、f が N/r=k なる周期r を持つなら、フーリエ変換された$\hat{f}$は、周期 kを持つことを。さらに、線形シフトの影響は、$\hat{f}$のフェーズ部分に現れるだけである。
- だから、第一レジスターのフーリエ変換を取ると、N/r の整数倍の状態が残るだけである。

$$\sqrt{\frac{r}{N}}\sum_{i=0}^{N/r-1}|ir+x_0\rangle \xrightarrow{QFT_N} \frac{1}{\sqrt{r}}\sum_{i=0}^{r-1}\left|i\frac{N}{r}\right\rangle \phi_i$$

  ここに、ϕ_i
  は、$x0$の線形シフトの影響で現れたフェーズの部分である。

r を推定する

- このアルゴリズムを繰り返して、いくつかの異なる N/r の整数倍の値を得ることができる。最悪でも、iog Nの繰り返しで、これらの値の最小公倍数として、N/rを求めることができる。



f(x)=x (mod 2) N=8の例

$$|0\rangle\,|0\rangle \xrightarrow{QFT_8} \frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x\rangle\,|0\rangle$$

$$\frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x\rangle\,|0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x\rangle\,|x \bmod 2\rangle$$

|f(x)>=|1>の時

$$\frac{1}{\sqrt{8}}\sum_{x=0}^{7}|x\rangle \otimes |f(x)\rangle \xrightarrow{measure|f\rangle} \frac{1}{2}(|1\rangle+|3\rangle+|5\rangle+|7\rangle)\otimes|1\rangle$$

$$\frac{1}{2}(|1\rangle+|3\rangle+|5\rangle+|7\rangle) \xrightarrow{QFT_8} \frac{1}{\sqrt{2}}(|0\rangle-|4\rangle)$$

N/r=4, N=8 から、r=2

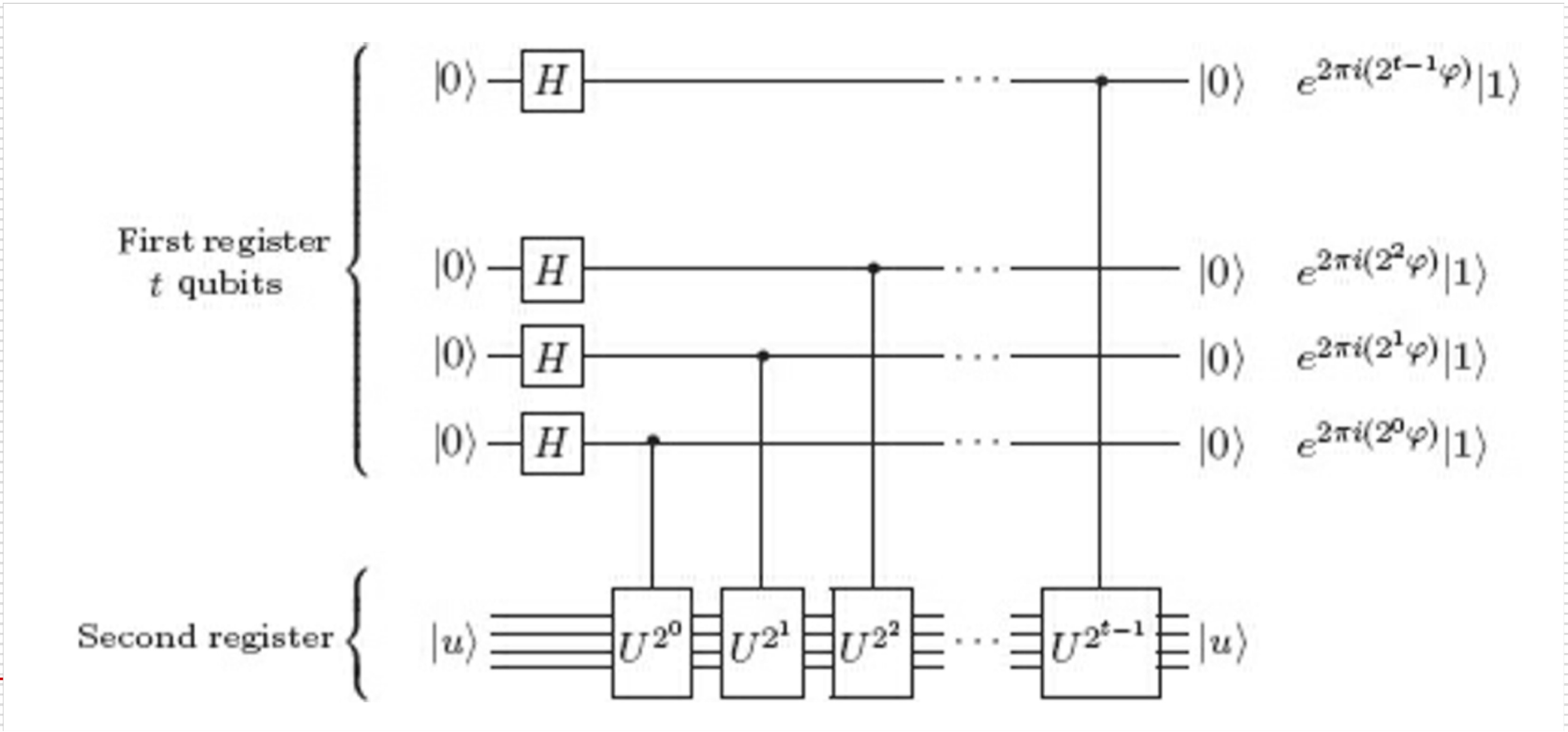
Phase Estimation

Phase Estimation

- 量子フーリエ変換は、Phase Estimationと呼ばれる一般的な手続きの重要なキーとなる。Phase Estimationは、数多くの量子アルゴリズムで重要な役割を果たす。
- 固有ベクトル|u>, 固有値$e^{2\pi i\varphi}$を持つユニタリ演算子Uを考えよう。ただし、φの値は未知とする。Phase Estimation アルゴリズムの目標は、φを評価することである。
- 評価を実行するために、状態|u>を準備し、コントロール-U^{2^j}(jは非負の整数)演算を実行できる、ブラックボックスが利用できるものとする。

- Quantum Phase Estimation 手続きは、二つのレジスターを使う。
- 第一のレジスターは、初期状態が|0>のt個のqubitを含んでいる。tをどのように選べばいいかは、どの程度の精度で φを評価するか、また、どの程度の確率でこの手続きが成功するかに依存する。
- 第二のレジスターは、|u>の状態で始まり、|u>を保存するに、必要なだけのqubitからなる。

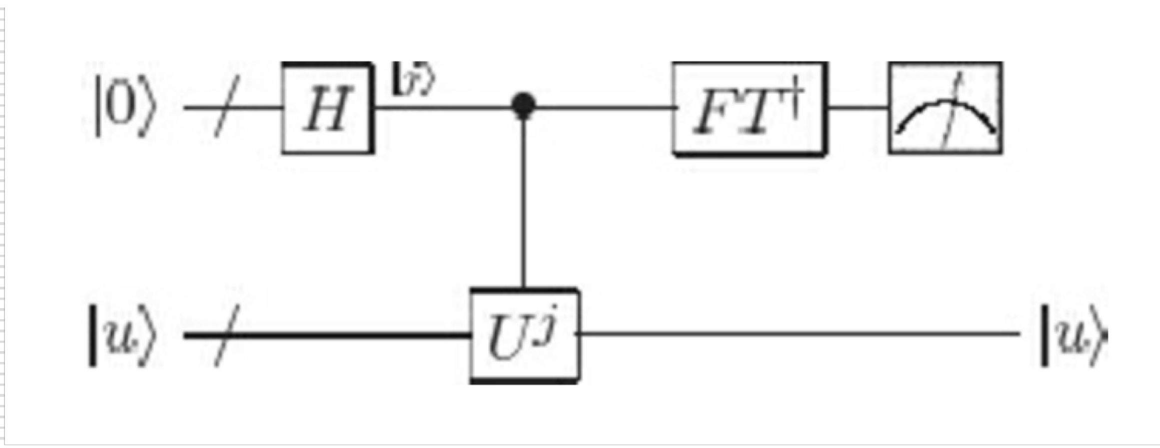
- Phase Estimationは、次の二つの段階で構成される。
- 第一段階では、次のような回路を適用する。



- この回路は、まず、第一レジスターに、アダマール変換を適用する。続いて、第二レジスターに、コントロールUを実行する。このコントロールUは、2のべき乗で指数が増えていく。
- 第一レジスターの最終状態は、次のようになる。

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right) = \frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{2\pi i\varphi k}|k\rangle .$$

- Phase Estimationの第二段階では、第一レジスターに、逆フーリエ変換を適用する。
- Phase Estimationの第三段階では、第一レジスターを、計算基底で測定する。



逆フーリエ変換

$$\frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1} e^{2\pi i\varphi j}|j\rangle|u\rangle \rightarrow |\tilde{\varphi}\rangle|u\rangle$$

1.	$\vert 0\rangle\vert u\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\vert j\rangle\vert u\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\vert j\rangle U^j\vert u\rangle$	apply black box
	$= \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}e^{2\pi i j\varphi_u}\vert j\rangle\vert u\rangle$	result of black box
4.	$\rightarrow \vert\widetilde{\varphi_u}\rangle\vert u\rangle$	apply inverse Fourier transform
5.	$\rightarrow \widetilde{\varphi_u}$	measure first register