

# プログラマの為の量子コンピュータ入門

## Part 0:

### イントロダクション(プロローグ)

---



宮地直人 (miyachi@langedge.jp)

Ver1.0 2019年6月27日

属性:  @le\_miyachi

技術: 古典PKIプログラム

仕事: ぼっち有限公司 (電子署名系)

量子: 独自に勉強 (書籍・勉強会)

趣味: 勉強会の開催、OSS開発

活動: OsSAL.org (オッサル) 他  
オープンソース署名 & 認証ラボ

## Richard Feynman (リチャード ファイマン)

“If you think you understand quantum mechanics, you don't understand quantum mechanics.”

「もしあなたが量子力学を理解できたと思うならば、それは量子力学を理解できていないということだ。」

※ ということできっと私も理解できていないので間違い等ご指摘を！

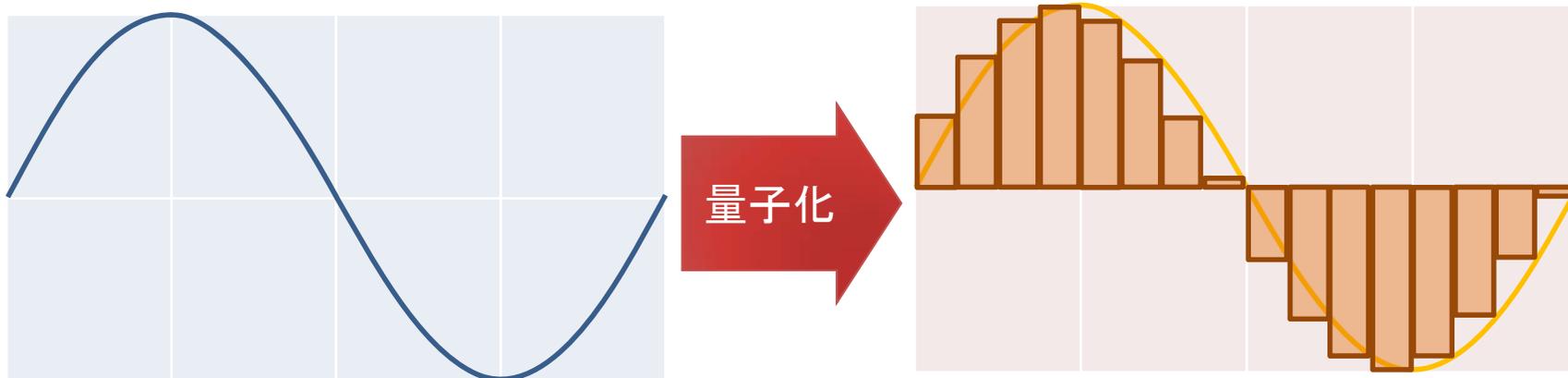
# 量子とは？

一般的な定義：

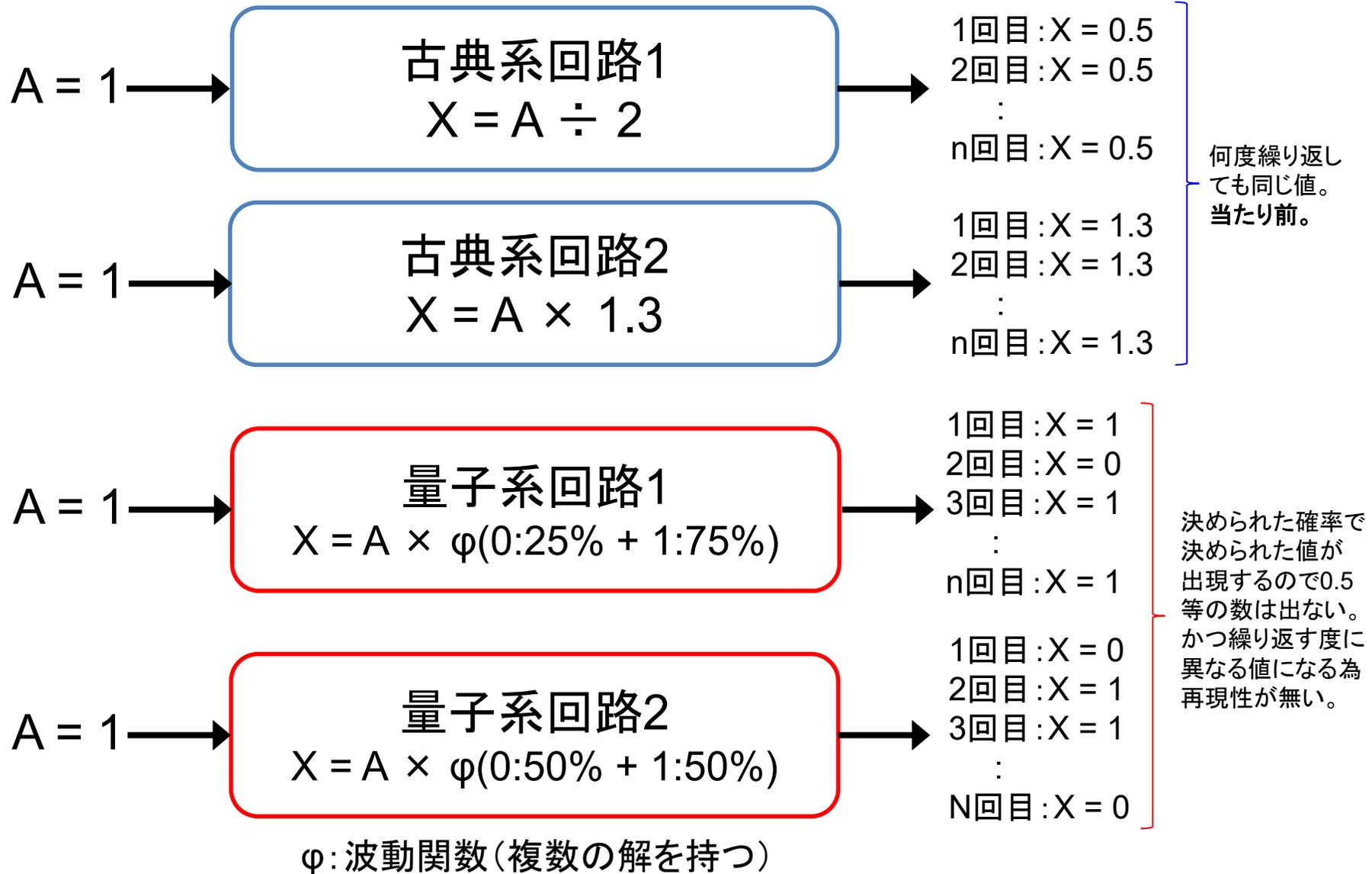
物理量を実数では無く単位量の整数倍で表す場合にその単位量を「量子」と呼ぶ。

※ 量子論/力学以外でも使われる場合がある。

例：アナログ波形のデジタル化を量子化と呼ぶ。

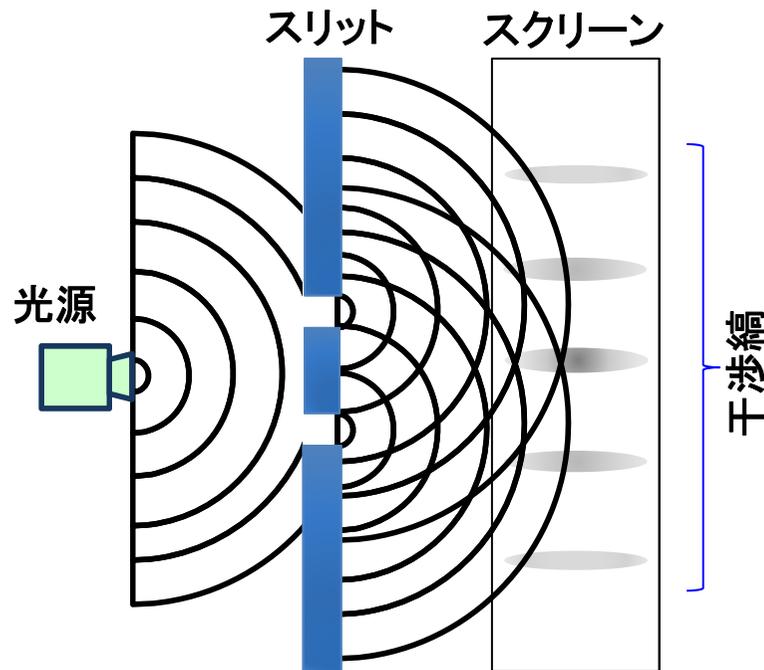


# 古典系と量子系の測定

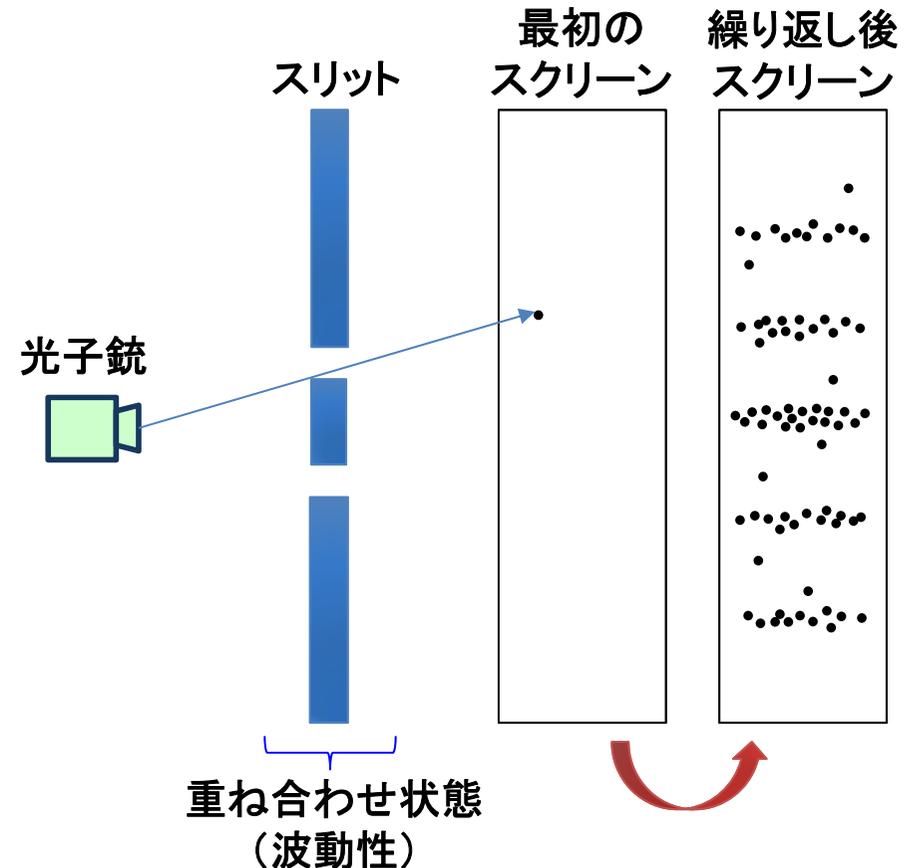


# 二重スリット実験（光の波動性）

二重スリット通過後に干渉縞が表示される。言うことは光に波動性があることを示す。干渉縞は同じ位相を持つ光源が2つあり、距離により位相が強めあう場所と弱めあう場所がある為に生じる現象である。

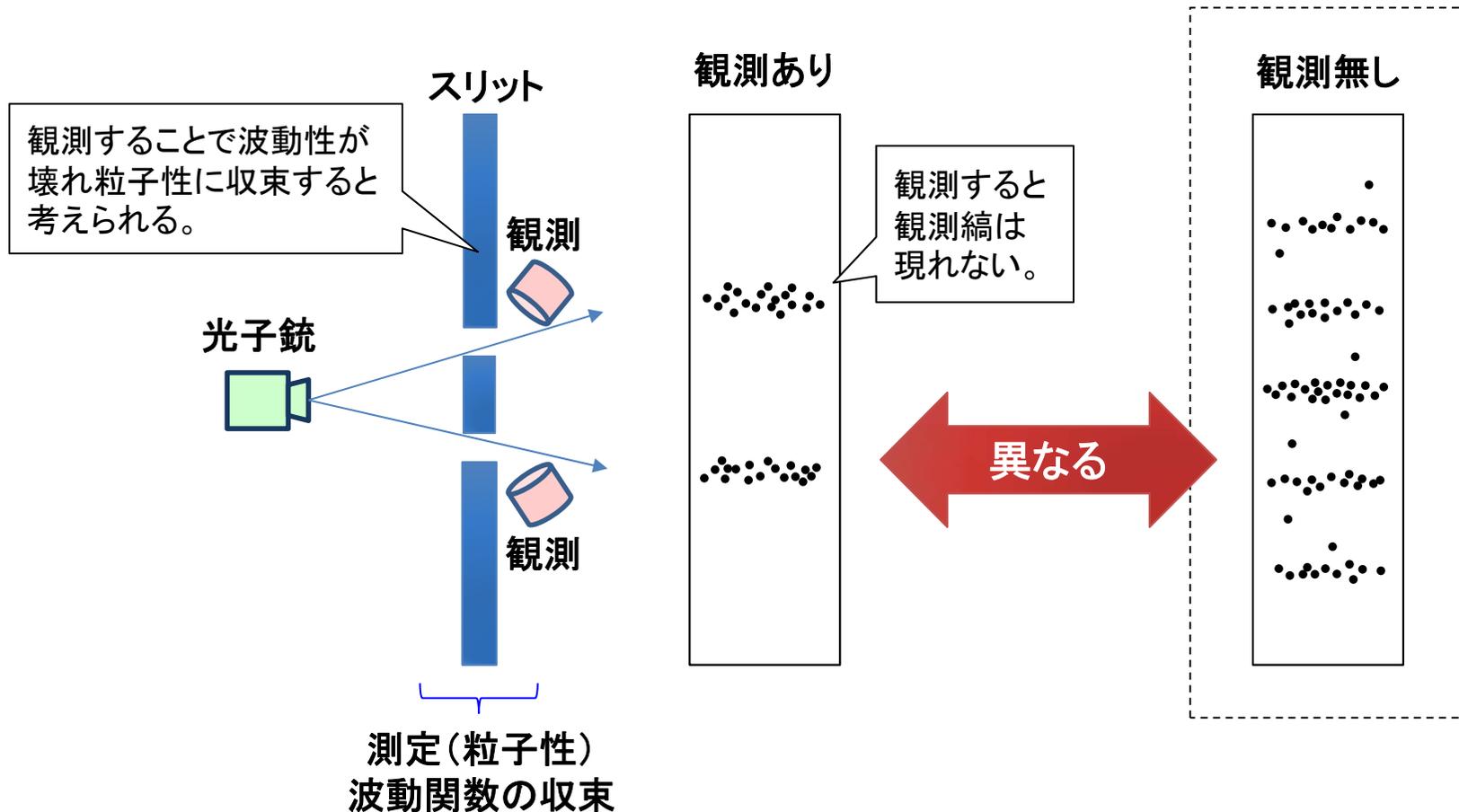


光子1個ずつ発射可能な光子銃と光子1個を認識できるスクリーンを使う。光子1個ずつを打ち出しても繰り返すうちに干渉縞が出てくる。つまり光子1個でも波動性があることになる。

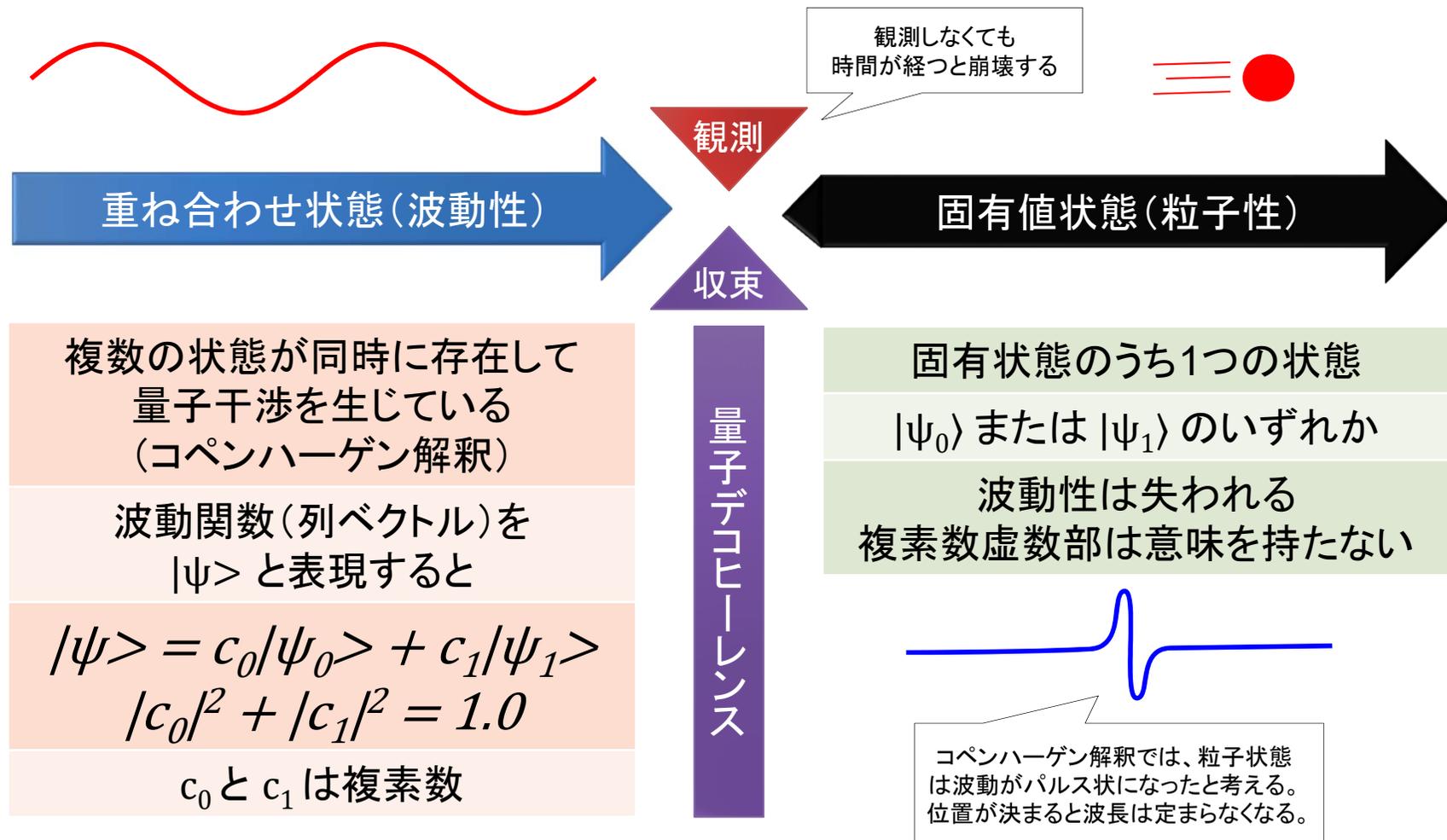


# 二重スリット実験の観測問題

観測問題: どちらのスリットを通過したかを観測することにより干渉縞が出なくなる。  
コペンハーゲン解釈では観測することで量子干渉が壊れて粒子として収束していると考えられる。  
しかし他にも解釈は存在しており(平行宇宙論等)、正確なことは分かっていない。



# 波動と粒子の二重性 (量子重ね合わせ)



※ コヒーレンス時間 (状態の量子干渉が失われるまでの時間) は短い。

## 量子ビット（重ね合わせの実現）

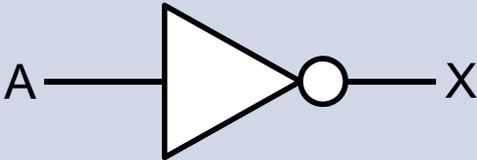
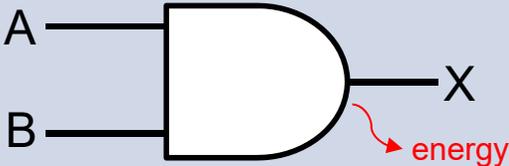
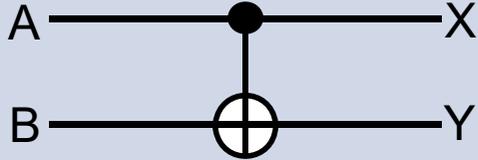
- 0と1の量子重ね合わせの状態が可能な単位。
- 観測により2つの固有値（0か1）に収束する。
- 物理的に実現するには幾つかの方法がある。

方式	概要	開発
超伝導 量子ビット	現在主流となっている超伝導状態のシリコン回路で量子ビットを実現する方式（極低温）	IBM, Google, D-Wave
イオントラップ	捕獲したイオンをレーザーで冷却して利用（室温） 理論的には量子ビット間の全結合が可能	IonQ
量子ドット	原子10～50個で構成した微小半導体を利用（極低温？）	Intel
トポロジカル	超電導体とトポロジカル絶縁体による量子ビット（極低温？）	Microsoft
NVセンター ダイヤモンド窒素-空孔中心	ダイヤモンドの炭素を窒素に置き換えて生じる欠損部に電子を捕獲して量子ビットに利用（室温）	（研究レベル）
QNN 量子ニューラルネットワーク	光パルスを量子ビットとして利用（室温） 全結合によるアニーリング型の計算が可能（らしい）	NTT/NII/東大 （ImPACT）

# 可逆コンピュータ

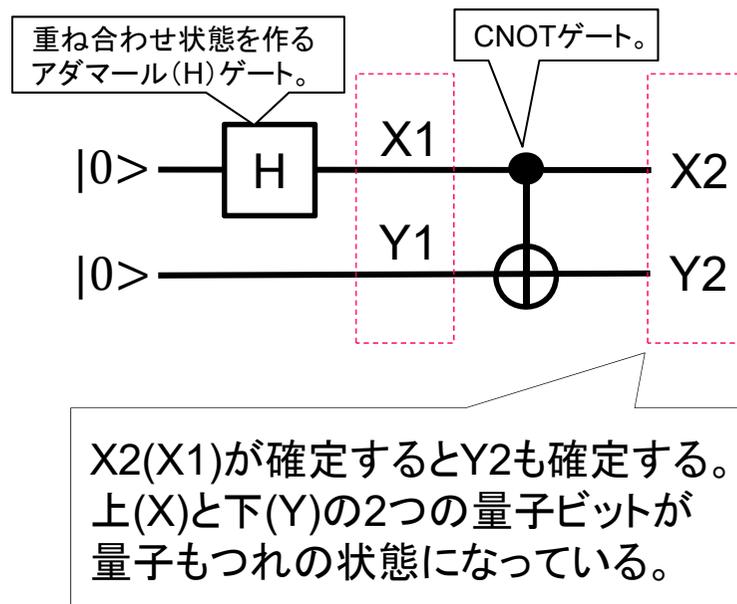
可逆コンピュータは非量子な場合には低電力になっても低性能になる為にほぼ実用化されていない。

情報が失われる時にエネルギー(熱)が消費される。  
 可逆コンピュータが実現すると低電力化が可能となる。  
 IBMから1960年代に出た理論で今も研究が続いている。  
 ※ 量子コンピュータは可逆コンピュータの一種である。

NOTゲート	ANDゲート	CNOTゲート																																									
																																											
<table border="1" data-bbox="421 1075 674 1270"> <thead> <tr> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </tbody> </table> <p data-bbox="383 1289 674 1362">出力から入力が再現できるので可逆</p>	A	X	1	0	0	1	<table border="1" data-bbox="891 1054 1205 1378"> <thead> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p data-bbox="1223 1193 1379 1362">出力から入力が再現できない</p>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1" data-bbox="1503 1054 1912 1378"> <thead> <tr> <th>A</th> <th>B</th> <th>X</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> <p data-bbox="1711 1394 1973 1442">Yだけ見るとXOR</p>	A	B	X	Y	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1	0
A	X																																										
1	0																																										
0	1																																										
A	B	X																																									
0	0	0																																									
0	1	0																																									
1	0	0																																									
1	1	1																																									
A	B	X	Y																																								
0	0	0	0																																								
0	1	0	1																																								
1	0	1	1																																								
1	1	1	0																																								
可逆	非可逆 (量子計算では使えない)	可逆																																									

# 量子もつれ(量子エンタングルメント)

- 量子もつれは、2つの粒子(量子ビット)が互いに影響をおよぼし合い、一方を測定すると、もう一方の値が確定する現象である。
- 複数の量子ビット間を、量子もつれにより関連付けることで量子回路を構築する。以下HとCNOTによる例。



$$X1 = |0\rangle : 50\% + |1\rangle : 50\%$$

$$Y1 = |0\rangle$$

$$X2 = X1 = |0\rangle : 50\% + |1\rangle : 50\%$$

X1が  $|0\rangle$  なら Y2も  $|0\rangle$  (Y1そのまま)

X1が  $|1\rangle$  なら Y2も  $|1\rangle$  (Y1を反転する)

$$X2Y2 = |00\rangle : 50\% + |11\rangle : 50\%$$

※  $|01\rangle$  や  $|10\rangle$  は 0%

この辺り詳しくは次回。

# 量子コンピュータの概要

1. 0と1の2つの基底を持つ複数の量子ビットを利用。
2. **重ね合わせた状態**のまま量子並列演算を行う。
3. 量子ビット同士を**絡み合わせて量子回路**を構築。
4. 重ね合わせ状態の出力を観測して固有値に収縮。
5. **繰り返し実行**し確率的な固有値の**出力分布**を得る。  
→ 二重スリット実験時に光子の分布により干渉縞を確認するようなもの。



# 量子コンピュータの種類

## 量子ゲート型 (狭義の量子コンピュータ)

方式: 量子ゲートの量子回路による量子計算

対象問題: 汎用 (ただし量子アルゴリズムの範囲内)

開発企業: IBM/Google/Intel/Alibaba/Microsoft等

## 量子アニーリング型 (正確には量子シミュレータ)

方式: イジングモデルを使った量子シミュレーション

対象問題: 最適化問題特化 (深層学習等への応用)

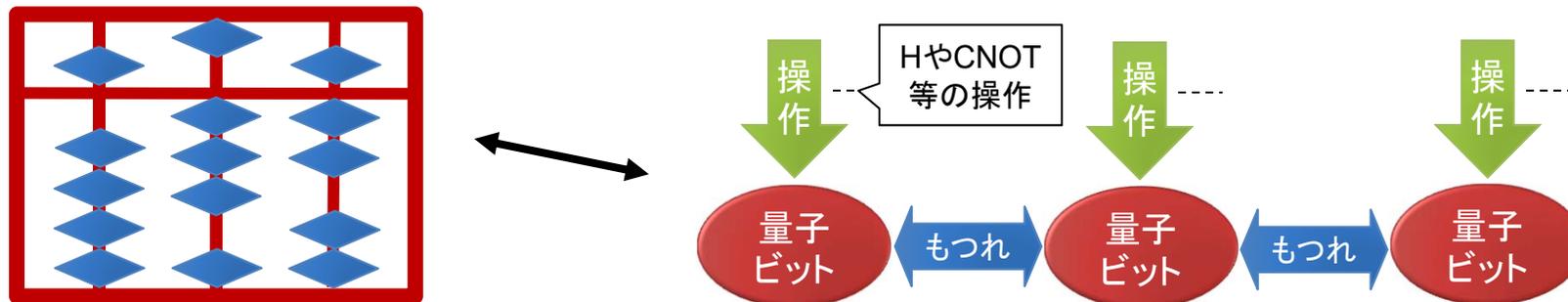
開発企業: D-Wave (非量子型では富士通と日立)

※ 非量子: 富士通「デジタルアニーラ」、日立「CMOSアニーリングマシン」。

※ 他に光を使ったCIM(コヒーレントイジングマシン)もあるがここでは省略。

# 量子ゲート型とソロバン

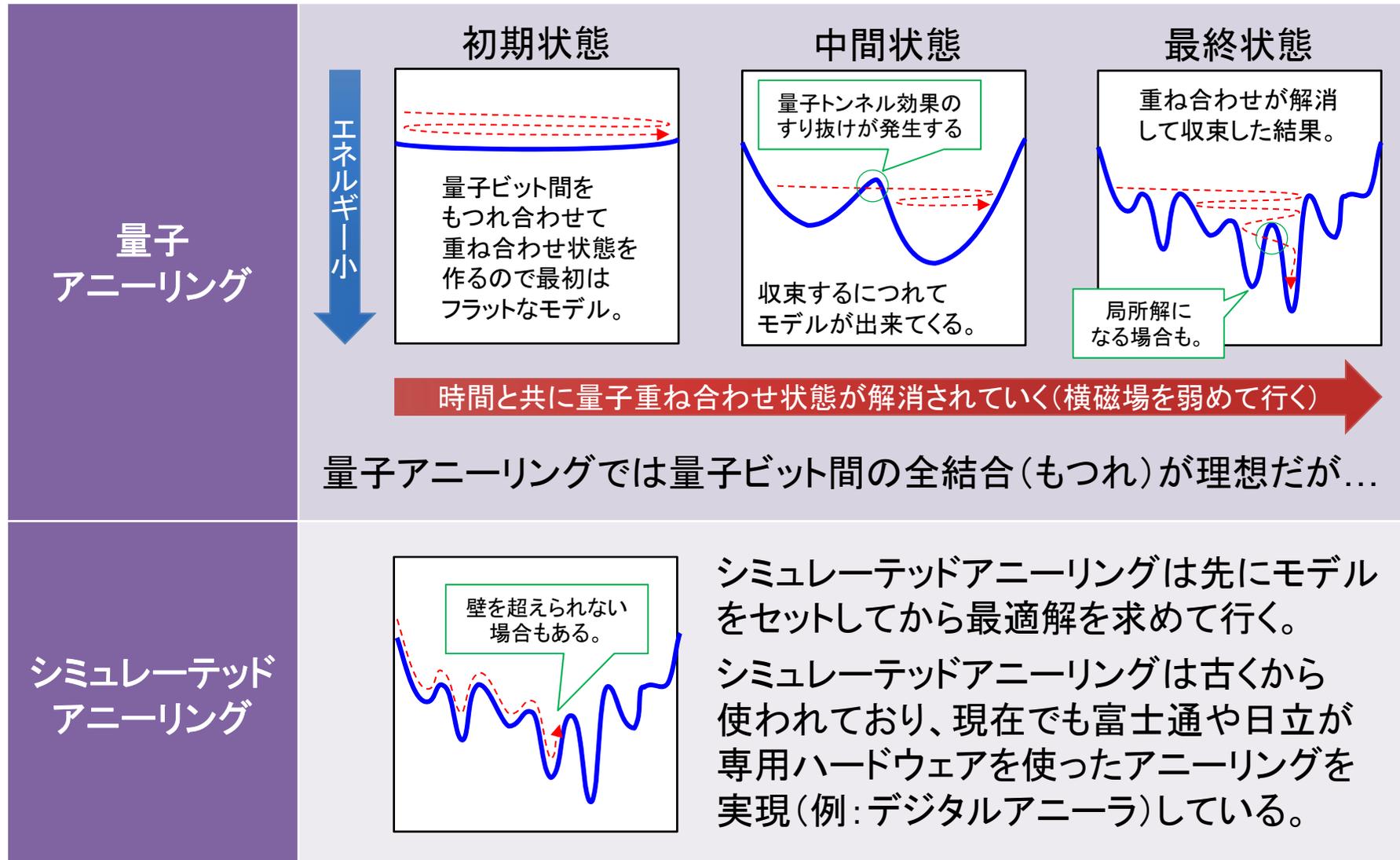
ソロバンは珠(たま)を配置したハードウェアを、指で弾いて行くことで計算を進めて行く。各珠の間には桁上がり等の関連性がある。



量子ゲート型は量子ビットを配置したハードウェアを、レーザーや電界で弾いて行くことで計算を進めて行く。各量子ビット間には量子もつれによる関連性がある。

ソロバンは可逆回路でもある。またソロバンは同じ操作をすれば毎回同じ値になるが量子では異なる。

# 量子アニーリング型の計算



# NISQの時代(今後5~10年程度)

## Q2B: QUANTUM FOR BUSINESS 2017

物理学者 John Preskill による基調講演の論文

「*Quantum Computing in the NISQ era and beyond*」

<https://arxiv.org/abs/1801.00862>

## NISQ: Noisy Intermediate-Scale Quantum

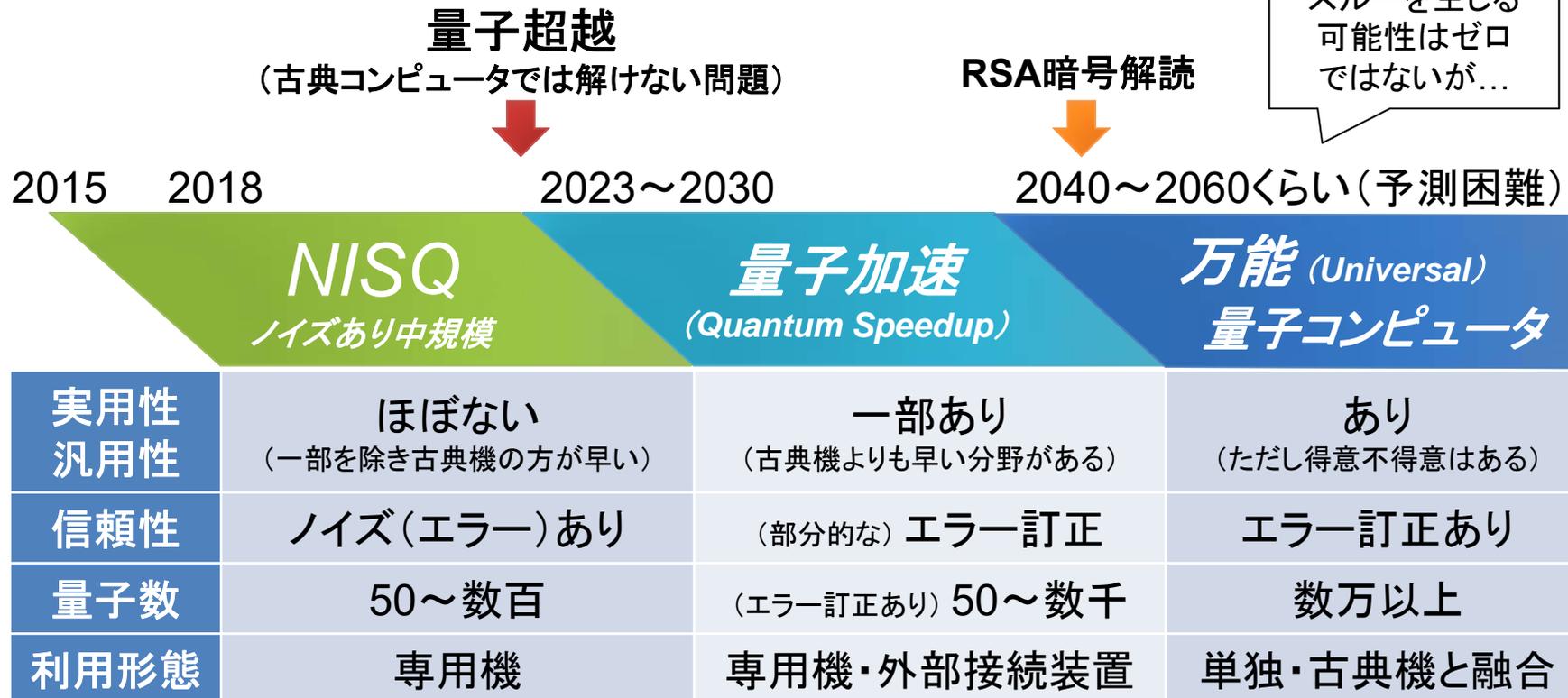
ノイズエラーがあり中規模量子ビット数の時代

50~数百量子ビット程度

私見:

現在は標準ハードウェアと標準ソフトウェア(アルゴリズム)を確立する時期で、特にソフトウェアは量子シミュレータを使って勉強しておくことで量子コンピュータ時代に備える。

# 量子コンピュータの未来予想



## 参考:(ノイマン型)古典コンピュータの歴史



# 量子計算フレームワーク（量子ゲート型）

IBMやGoogleは自社量子コンピュータを使う為の量子計算フレームワークを公開している。実機だけではなくシミュレーション機能を持っているので、量子プログラミングの勉強用として最適だが小規模の量子回路のみとなる。

項目	IBM Qiskit	Google Cirq
ロゴ	 Quantum Information Science Kit	
構成	Terra: 量子計算の基盤部 (Python) Aqua: 量子アルゴリズムのライブラリ OpenQASM: 量子低レベル中間言語	Cirq: 量子計算基盤 Python ライブラリ OpenFermion: 量子化学ライブラリ
提供	オープンソース (GitHub)	オープンソース (GitHub)
取得	<a href="https://qiskit.org/">https://qiskit.org/</a> <a href="https://github.com/Qiskit">https://github.com/Qiskit</a>	<a href="https://github.com/quantumlib/Cirq">https://github.com/quantumlib/Cirq</a>
情報	<a href="https://qiskit.org/documentation/ja/">https://qiskit.org/documentation/ja/</a>	<a href="https://cirq.readthedocs.io/en/latest/">https://cirq.readthedocs.io/en/latest/</a>
その他	IBM Q Experience: GUI 利用 ※ GUI から OpenQASM に展開し実行 <a href="https://quantumexperience.ng.bluemix.net/">https://quantumexperience.ng.bluemix.net/</a>	2018年夏に正式公開されたライブラリ 量子コンピュータ実機はまだ使えない

## Q-LEAP (光・量子飛躍フラッグシッププログラム)

<https://www.jst.go.jp/stpp/q-leap/>

文部科学省の日本国産プロジェクト。

今後10年で国内産学共同で100量子ビット  
(量子超越)の量子コンピュータやインフラ  
(開発環境)を開発する...予定だそうです。

人材育成も目的とのことですので、今後  
日本でもますます量子計算ができる人が  
求められる時代が来るでしょう。

## 量子ゲート型の状況は...

レベルから言えばアセンブラ言語の利用に近い。

アセンブラ言語はハードウェアに密接に関連しているのでハードウェアを理解しないと使えない。

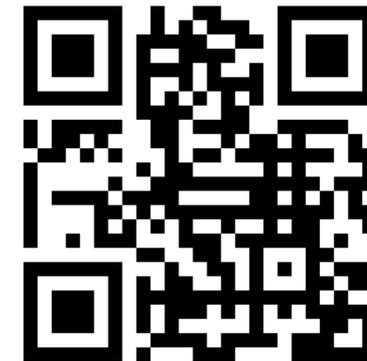
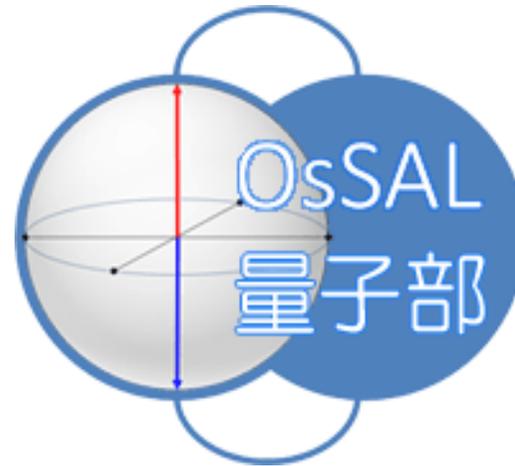
同様に量子回路もIBMはQASM言語に変換して実行している。つまりアセンブラ言語だ。IBMのQiskitはQASMのPython用ラッパーに過ぎない。

今後、量子アルゴリズムAPIを備えたSDKが必要。

プログラマが勉強すべきは量子アルゴリズムAPIの使い方になるだろう。現在SDKを提供している各社はそのレベルを目指しているように見える。

# サル量子部

<https://www.ossal.org/qc/>



**2019夏** に勉強会(サル量子#1)を開催予定

## 古典プログラマの為の 量子プログラミング入門

第1部: 関連数学と1量子ビット計算

第2部: 量子ゲート型のプログラミング

第3部: 量子アニーリング型のプログラミング

ショアの  
アルゴリズムも  
やります。

# サル量子 #1の内容

Part 0: イントロダクション(プロローグ)

## Part 1: 関連数学と1量子ビット操作

1-1: 線形代数学の基本知識

1-2: ブラケット記法と量子計算

1-3: ブロッホ球と1量子ビット操作

1-4: IBM Q

## Part 2: 量子ゲート型のプログラミング

2-1: 複数量子ビット操作

2-2: 量子アルゴリズムの基本

2-3: ドイチェ アルゴリズム

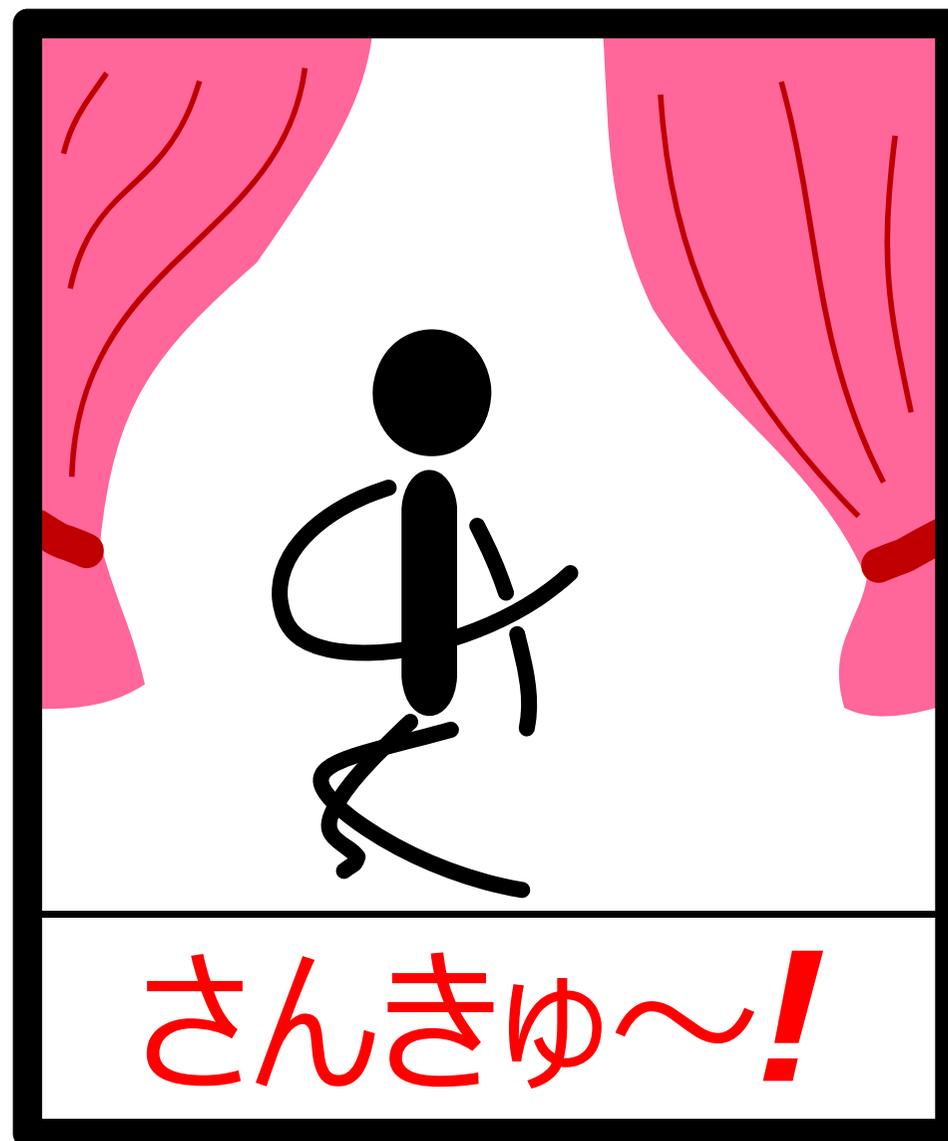
2-3: グローバー検索(量子検索)

2-4: 量子フーリエ変換

2-5: ショアのアルゴリズム

2-6: エラー訂正問題

2-7: Cirq (Google)・Blueqat (MDR)



<http://scienceinoh.jp/schrodinger/>