

# 量子コンピュータ

江崎 ひろみ <sup>\*1</sup>

## Quantum computer

Hiromi Ezaki <sup>\*1</sup>

Quantum computer is a machine using quantum mechanics principles, which is expected to have extremely high performance compared with a classical computer. In this paper, the author briefly reviews history of classical and quantum computers and show the latest development of quantum computer.

### はじめに

2011年にカナダのベンチャー企業 D-Wave Systems 社が世界初の商用量子コンピュータを販売して以降、近年の量子コンピュータの開発は新しい時代を迎えつつある。量子コンピュータが実現されると、現在インターネットで広く用いられている RSA 公開鍵暗号が容易に破られてしまうことから、その動向は経済界のみに留まらず、政治、外交を含め国政レベルで注目を集めている。本論文では、まず現在用いられている古典コンピュータがどのようなものか触れてから、量子コンピュータでは何が異なるのかを示し、量子コンピュータの開発の現状について紹介したい。最後に量子コンピュータの課題について触れる。

### 古典コンピュータ

古典コンピュータとは、現在広く使用されている通常のコンピュータのことを指す。古典コンピュータをモデル化したものが古典チューリング機械である。古典チューリング機械は図 1 に示すように、テープとプロセッサから成り立っており、テープ上にあるセルには 0 と 1 の数字が書き込まれている。プロセッサにはヘッドが付いていてテープに書いてある数字を読んだり書き換えたりしながら、テープを前後に移動する。

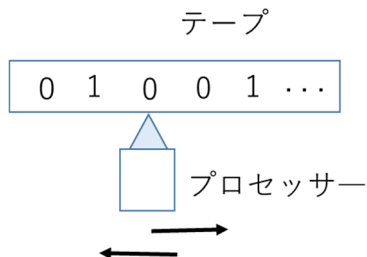


図 1. 古典チューリング機械

テープがコンピュータのメモリに相当し、プロセッサが CPU の役割を果たしている。チューリング機械の動作は状態遷移関数によって表される。状態遷移関数には、「プロセッサの状態が  $p$  でヘッドがあるセルの記号が  $s$  ならば、プロセッサの状態を  $p'$  にして、セルに  $s'$  を書き込み、ヘッドを左に 1 つ動かす」ということが書かれている。

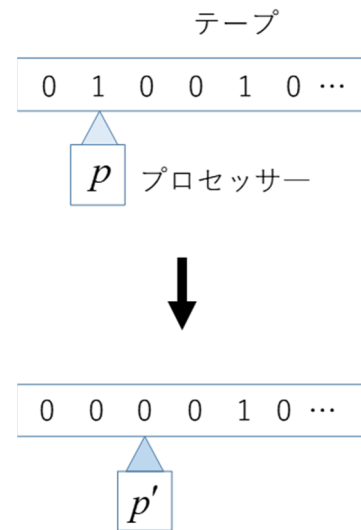


図 2. 古典チューリング機械の動作

たとえば、図 2 ではプロセッサの状態が  $p$  でヘッドがあるセルの記号が 1 のとき、セルの記号を 0 に書き換え、右に一つ移動し、プロセッサの状態を  $p'$  にするという動作を表している。計算のアルゴリズムに対応する状態遷移関数を実行することにより、チューリング機械は現在のコンピュータで実行されているものと同様な計算をすることができる。

<sup>\*1</sup> 東京工芸大学工学部工学科電気電子コース 教授  
2023 年 9 月 21 日 受理

## 量子チューリング機械

古典チューリング機械の量子力学版が量子チューリング機械である。量子チューリング機械は 1985 年の論文でドイチュが初めて定式化を行った<sup>1)</sup>。狭義には、ドイチュのこの論文が量子コンピュータの研究の始まりとされる。

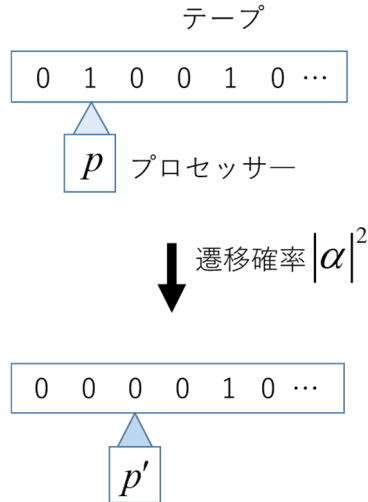


図 3. 量子チューリング機械の動作

図 3 に量子チューリング機械の動作を示した。古典チューリング機械と基本的な動作は変わらないが、状態遷移関数における状態の遷移が決定的ではなく、遷移後の状態は遷移確率  $|c_1|^2$  で状態  $p_1$  へ、遷移確率  $|c_2|^2$  で状態  $p_2$  へなどと確率的に行われる。量子チューリング機械の状態は量子力学的な状態の重ね合わせとなっているのである。このようにして、量子チューリング機械は 1 台の中に無数の状態の重ね合わせを保持することができ、それぞれの状態に対して計算を行うことができる。つまり、量子並列計算が可能となっている。遷移確率 1 で遷移する場合は、量子チューリング機械は古典の場合と同様になるので、古典チューリング機械でできることは量子チューリング機械でも実行可能である。

量子チューリング機械は無数の状態の重ね合わせにより量子並列計算ができるわけであるが、実際に計算の答えを得る際には、状態を観測して重ね合わせにある状態の中の 1 つの状態だけを得ることになる。そのため、正しい解を得るためには、正解となる状態の確率振幅が十分大きくなければならず、解の確率振幅を大きくするための工夫（アルゴリズム）が必要となってくる。量子力学的な状態の重ね合わせにより並列計算を行うアイデアは 1982 年に

ファインマンが提唱していたが、当時は正解を増幅するアルゴリズムがなかったため、多くの計算結果から正解を絞り込むためにまた膨大な計算が必要となってしまう、量子並列計算にはメリットがないと思われていた。

状況が一変したのは、1994 年にショアが素因数分解を量子コンピュータで高速に計算できるアルゴリズムを発表してからである<sup>2)</sup>。ショアのアルゴリズムにおいても、計算結果は確率的であり、正しい答えが得られるまで何回か試行する必要があるが、得られた答えのチェックは容易であるため、試行回数が少なければ十分に機能する。ショアの発表を機に、量子チューリング機械を物理的に実現する量子コンピュータの開発研究が一躍注目を集めるようになった。

## 量子デジタルコンピュータ

古典コンピュータではゲートを組み合わせた回路により計算が行われる。代表的な古典ゲートとしては、AND、OR、NOT がある。これらのゲートのうち、AND と NOT、または OR と NOT の 2 種類のゲートを組み合わせることにより、任意の回路を構成することができる。このことから、これらのゲートは普遍ゲートと呼ばれている。同様に、量子コンピュータにおいても量子普遍ゲートが存在する。量子ゲートを用いた量子コンピュータが量子デジタルコンピュータである。

量子コンピュータにおける計算とは、初期状態  $|\Psi\rangle$  にユニタリ行列  $U$  を演算させ、

$$|\Psi'\rangle = U|\Psi\rangle \quad (1)$$

最終状態  $|\Psi'\rangle$  を得ることであり、このユニタリ行列を量子ゲートと呼ぶ。

1 量子ビットに対するゲートを 1 量子ゲートと呼ぶ。代表的な 1 量子ゲートとしては、重ね合わせ状態を生成できるウォルシュ・アダマールゲート

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

が挙げられる。これらは量子ビットに対して

$$H|b\rangle = \frac{1}{\sqrt{2}} \{ |0\rangle + (-1)^b |1\rangle \} \quad (3)$$

のように作用する。他には、位相シフトゲート

$$T\{c_0|0\rangle + c_1|1\rangle\} = c_0|0\rangle - c_1 e^{\frac{\pi i}{4}}|1\rangle \quad (4)$$

もよく用いられる。

2 量子ビットに対する代表的な演算としては、制御 NOT(CNOT)ゲート

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (5)$$

が挙げられる。これは 2 量子ビットに対して

$$CNOT|0\rangle|0\rangle = |0\rangle|0\rangle, CNOT|0\rangle|1\rangle = |0\rangle|1\rangle \quad (6)$$

$$CNOT|1\rangle|0\rangle = |1\rangle|1\rangle, CNOT|1\rangle|1\rangle = |1\rangle|0\rangle \quad (7)$$

のように、制御ビットが 1 のときのみ、ビットを反転させるものである。任意の量子回路は 1 量子ゲートと CNOT ゲートの組み合わせで実現でき、任意の 1 量子ゲートは H と T の組み合わせで実現できることから、この 3 つが量子普遍ゲートとなる。

量子コンピュータの開発は、量子普遍ゲートを物理系において実現する試みからスタートし、1998 年に核磁気共鳴 (NMR) を使った 2 量子ビットの量子コンピュータの実験がアメリカとイギリスのグループによって初めて成功した。しかし、NMR で量子ビットを増やすのは非常に難しく、7 量子ビット程度が限界であった。現在では、超電導回路やスピンなどの固体中の量子系を用いて研究が進められている。特に、超電導回路の方式については、Intel 社が 49 ビット、Google 社が 72 ビット、IBM 社が 433 ビットなど、数十から数百ビットの規模まで達成されてきている。しかしながら、ショアのアルゴリズムを実行するにはさらなる大規模化が必要であり、汎用の量子デジタルコンピュータの実現にはまだまだ時間がかかりそうである。

日本では、理化学研究所を中心とするグループによって、電子スピンを用いたシリコン量子ドットを用いた研究が進められている。シリコン半導体素子は集積化には有利であるが、電氣的、磁氣的雑音が大きく、スピン操作の高忠実度が求められる。理化学研究所では 99.5% の高忠実度で 2 量子ビット操作を実現している。しかし、量子ビットの規模拡大には高精度の誤り訂正が求められるなど、課題が多く、シリコン素子を用いた大規模量子コンピュータの実現は、少なくとも現状では超電導回路よりも前途多難とい

えよう。

## 量子アナログコンピュータ

量子デジタルコンピュータが量子回路を物理系で実現する試みであるのに対し、量子力学が絡んだ物理現象を用いて問題を解こうとするのが、量子アナログコンピュータである。量子アナログコンピュータは量子コンピュータとは言えないという見方もあるが、動作に量子力学を利用しているという点では“量子コンピュータ”といってもいいだろう。先に述べた D-Wave Systems 社の商用量子コンピュータ (D-Wave マシン) は量子アニーリングを利用して問題を解こうとする量子アナログコンピュータである。量子アニーリングとは量子揺らぎを利用して最適化問題を解く手法で、もともとはスピングラス模型の基底状態を求めるための手法である。したがって、量子アナログコンピュータは最適化問題に特化したマシンであり、現在使われている古典コンピュータのように様々な問題を解くことはできない。最適化問題に特化していることは量子アナログコンピュータの弱点ではあるが、スピングラス模型に帰着可能な問題に関しては非常に高速で解けることが大きな長所となっている。

D-Wave マシンを購入した Google が 2019 年 10 月に発表した“量子超越実験”は、ビットコインが一時的に暴落するなど、物理学界に留まらず経済界にも衝撃をもたらした<sup>3)</sup>。“量子超越”とは古典コンピュータでは実現できないタスクを量子コンピュータが行うことを意味する。Google の論文では、2 次元正方格子状の 53 個の量子ビットに対して、ランダムに選ばれた 1 量子ビット演算と隣接する量子ビットに作用する 2 量子ビット演算を繰り返すというランダム量子回路を実行し、測定をしてビット列をサンプリングするという実験を行った。D-Wave マシンでは 200 秒程度でタスクをこなしたが、同様の実験を現在のスーパーコンピュータで行うと約 1 万年かかると結論付けている。これをもって、“量子超越”を実現したという主張である。しかし、IBM からは一次記憶だけでなく、二次記憶 (ディスクメモリ) も用いれば、現存のスーパーコンピュータを用いても約 2.5 日で実行できるという反論も出ている。

D-Wave マシンが解けるのは最適化問題に限られるため、D-Wave のユーザーは自分が解きたい問題を最適化問題に変換する必要がある。D-Wave マシンを使った応用事例としては、京セラ株式会社が、分子の電子状態やエネルギーの計算に使われるフラグメント分子軌道法に必要なグラフ分割問題に利用した例が報告されている<sup>4)</sup>。ビジネス面での応用事例も多く、例えば、フォルクスワーゲン社は 2017 年に交通量の最適化問題をスピングラス問題に帰着させて、北京の交通渋滞する研究を行った<sup>5)</sup>。他には、オンラインサービスでの最適な商品配置や株や債券などの

資産配分に D-Wave を利用したりする研究が行われている。

このように、現在の量子アナログコンピュータは最適化問題に特化したマシンであるが、解きたい問題をうまく最適化問題に帰着させることができれば、非常に高速に問題を解くことができる。ただし、最適化問題に帰着させられさえすればもとの問題がすべて解けるわけではない。最適化問題に変換するときに効率が大事である。現在の D-Wave マシンは約 5000 量子ビットを使って計算することができるが、それでも大きな整数の素因数分解問題を最適化問題に還元すると、還元したスピングラス問題のサイズがかなり大きくなると予想されている。したがって、現行の D-Wave マシンを使って大きな整数の素因数分解を求めることは難しいようである。

## 量子コンピュータの課題

量子コンピュータは量子ビットの重ね合わせ状態など、量子力学の原理を利用しているコンピュータである。重ね合わせ状態を保つため、量子状態のコヒーレンスを演算の間保たなければならないが、このコヒーレンスの保持と計算のための制御とはトレードオフの関係にある。孤立系にすればコヒーレンスを保てるが、量子ビットを操作するためには量子ビット同士を相互作用させたり、外界から操作したりする必要がある。すると、そのために重ね合わせなどの状態が壊れてしまうわけである。そこで量子デジタルコンピュータの大規模化には誤り訂正が重要となってくる。量子誤り訂正は古典誤り訂正と異なり、ビット符号の反転だけでなく重ね合わせ状態の位相の訂正も必要となる。

量子コンピュータが提案された当初、古典誤り訂正だけでは正確に動作できないことが指摘され、これを理由に量子コンピュータの実現が疑問視されていた時期もあった。その後、量子誤り訂正が提案され、その実装とともに量子コンピュータが実現した。しかし、大規模な量子ビットに対する高精度の誤り訂正は課題が多く、誤り訂正機能を持った大規模の量子デジタルコンピュータの実現はまだまだ先と考えられる。ここでは紙面の関係で誤り訂正には触れられなかったが、今後の量子コンピュータの開発においては、量子ビットの集積化とともに高精度の誤り訂正の実現が鍵となるだろう<sup>9)</sup>。

一方、成功を収めているように見える量子アナログコンピュータだが、D-Wave マシンについてはキメラグラフという特殊な平面グラフ上に量子ビットを配置しているために、任意のスピングラス問題をそのままの形で解くことができないという制約がある。また、断熱性が担保されていないために、基底状態からずれた結果を出力してしまうことが起こりうる。このような点が改善されれば、より広範な問題へ D-Wave マシンの利用が可能となるだろう。

最後に日本の開発状況について触れておこう。量子コンピュータの開発は世界的な投資が拡大する中、日本では投資規模が欧米に比べてはるかに小さく、それが日本の開発の遅れに綱がっているという声もある。しかし、日本においても、先に触れたように理化学研究所や NTT や富士通、東京大学などが量子コンピュータの開発に携わっている。2023 年 3 月には初の国産量子コンピュータが理化学研究所などにより公開された。しかし、まだ 64 量子ビットと少なく、今後の進展が期待される。

## おわりに

本論文では古典チューリング機械から始め、量子チューリング機械の概要、そして量子チューリング機械と等価な量子コンピュータの開発状況を概観した。ここでは、詳細な性能比較については述べなかったが、そもそも量子コンピュータは古典コンピュータよりも性能が高いのか、という素朴な疑問がわいてくる。確かに、最適化問題など一部の問題については量子アナログコンピュータが古典コンピュータを上回るという結果が出ているが、どのような問題に対しても量子コンピュータの方が常に高速に計算を行えるかどうかは結論が出ていない。むしろ、量子コンピュータが現在のスーパーコンピュータよりも上回るのは素因数分解などに限られるのではという予想もある<sup>7)</sup>。

しかし、将来、大規模な量子デジタルコンピュータが実用化されると RSA 公開鍵暗号が破られる恐れが出てくる。RSA 公開鍵暗号は大きな整数の素因数分解の難しさを前提にして設計されているからである。汎用の大規模な量子デジタルコンピュータがいつ頃までに開発されるかは見通せないが、開発されてから対策を議論するのでは遅いので、今から耐量子コンピュータの暗号開発が活発に行われている。アメリカでは既にその候補がいくつかに絞られており、2026 年頃までに耐量子コンピュータ暗号へ移行することを目標にしているそうである<sup>8)</sup>。日本においても、量子コンピュータが脅威となる前に、早めに対策を講じることが必要であろう。

## 参考文献

- 1) D. Deutsch, Proc. R. Soc. Lond., **A400**, 96 (1985).
- 2) P. W. Shor, Proc. 35th Ann. Sym. Found. Comp. Sci., 124 (IEEE, Computer Society Press, New York, 1994).
- 3) F. Arute, et. al., Nature **574**, 505 (2019).
- 4) N. Nishimura, et. al., arXiv:1903.12478 (2019).
- 5) N. Florian et. al., Frontiers in ICT, **4**, 29 (2017).
- 6) Z. Yang et al., IEEE Commun. Surv. & Tut., **25**, 1059 (2023).
- 7) 西野哲郎, 「数理科学」7月号, 61, サイエンス社, 2019.
- 8) 四方順司, 金融研究所ディスカッション・ペーパー2019-J-4, 日本銀行金融研究所, 2019年.