Vol.2022-QS-5 No.10
2022/3/24

Shor のアルゴリズム量子回路の簡略化と量子コンピュータにおける実装実験

坪井あさと[1]　永田真[1]　三木拓司[1]

概要：Shor のアルゴリズムは，特定の数 N に対して N よりも小さく互いに素である数 a を選択し，$a^r \bmod N = 1$ となる最小の整数 r を発見することで，N を多項式時間で効率的に素因数分解するアルゴリズムである．Shor のアルゴリズム量子回路の実装において，同様の数 N を対象とする場合においても，使用する量子ゲートの数や配置順によって数種類の量子回路を考えることが可能である．その中で，より量子ゲート操作回数の少ない量子回路を選択して実行することで，測定結果の正確さを向上させることができる。

本稿では $N = 15, 35$ を対象とする Shor のアルゴリズム量子回路を基本的なルールに従って実装した後，Toffoli ゲートや一連のゲートの重複使用に着目した簡略化を施した量子回路を実装する．そして，実装したそれぞれの量子回路の測定結果について比較を行い，正確さの向上を確認する．量子回路を実行する量子コンピュータはイオントラップ型の量子コンピュータ IonQ を用いる．本稿では，$N = 15, a = 7$ においては 7.0%，$N = 35, a = 4$ においては 13.5%の正確さの向上を確認したので報告する．

キーワード：Shor のアルゴリズム，量子コンピュータ，IonQ

Simplifying Shor's factoring algorithm and implementation experiment by using quantum computer

ASATO TSUBOI[†1]　MAKOTO NAGATA[†1]
TAKUJI MIKI[†1]

Abstract: For implementing Shor's factoring algorithm, we choose positive integer $a < N$ co-prime to N, and find the order r which satisfies $a^r \bmod N = 1$. The algorithm results in factoring integer N in polynomial time. When implementing the quantum circuit of the algorithm, we can think of several quantum circuits which have different number of quantum gates and alignment sequence. By choosing the quantum circuit which have less gates, we can improve the accuracy of measurement data.

We implement the quantum circuit of Shor's factoring algorithm for $N = 15$, $a = 7$ and $N = 35$, $a = 4$ following basic rules, and simplify them by focusing on Toffoli gates and repetitive sequences. Before and after simplifying, we compare the measurement data, and confirm the results of improving accuracy. We use IonQ to implement these quantum circuits. In this paper, we report improving accuracy by 7.0% in the case of $N = 15$, $a = 7$, 13.5% in the case of $N = 35$, $a = 4$.

Keywords: Shor's factoring algorithm, quantum computer, IonQ

1. はじめに

1.1 RSA 暗号の危殆化可能性

公開鍵暗号の標準である RSA 暗号は，素因数分解問題の難しさを安全性の根拠としており，情報セキュリティ分野における重要な研究課題である．RSA 暗号では，互いに異なる 2 つの素数 p, q の積 N を用いて暗号化を行うが，RSA 暗号の法 N が素因数分解されれば，暗号文は解読されてしまう[1]．しかし，多項式時間で効率的に素因数分解を行う古典的アルゴリズムは存在しないことが知られている．

NISQ（Noisy Intermediate-Scale Quantum device）は量子ビット数が数百個程度の小～中規模の量子コンピュータであり，今後数年以内に実用化される可能性が高い．NISQ は大量の量子ビットを必要とする「量子誤り訂正」を前提としておらず，量子ゲート操作等によるエラーが重畳される．NISQ は特定の計算において古典コンピュータを凌駕する性能を発揮すると期待されるが，即座に RSA 暗号を解読する程の性能は有していない．しかし，将来的に 200 万量子ビットを持つ量子コンピュータが実現すれば，RSA 暗号は 8 時間で解読可能と推察されている[2]．

1.2 現状の Shor のアルゴリズム実装動向

Shor のアルゴリズムは，多項式時間かつ高確率に合成数 $N = pq$ を素因数分解するアルゴリズムである[3]．

現在、複数の研究グループによって Shor のアルゴリズム量子回路の実行結果が報告されている．しかし、その多くが $N = 15, 21$ に対するものである[4][5]．また、特定の N, a

1 神戸大学大学院科学技術イノベーション研究科
（〒657-8501 神戸市灘区六高台町 1-1）
Graduate School of Science, Technology and Innovation, Kobe University
坪井あさと: asato.tsuboi@cs26.scitec.kobe-u.ac.jp
永田真: nagata@cs.kobe-u.ac.jp
三木拓司: miki@cs26.scitec.kobe-u.ac.jp

でのみ有効な簡略化手法（log を用いた量子レジスタの圧縮手法など）を用いており，RSA 暗号で扱うような巨大素数の素因数分解に応用できない手法も含まれる[6][7].

Shor のアルゴリズムでは位数 r と呼ばれる数を求め，それを利用して素因数分解を行う．$N = 15, 21$ において位数 $r = 2$ や $r = 4$ を求めることに成功したという報告があるが，$N = 35$ において位数 $r = 6$ を求める実験は成功しなかったという報告がある[8].

本稿では，量子ビット数が少なく，量子ゲート操作のエラー率が高い現状の制約下において，Shor のアルゴリズムを実行する量子回路を簡略化することで，測定結果の正確さを向上させる．まず，基本的なルールに従って Shor のアルゴリズム量子回路を構築した後，Toffoli ゲートおよび関連する量子ゲートの重複使用に着目して簡略化を施す．

これにより，現有する量子コンピュータにより，どの程度の N まで素因数分解することが可能か，あるいは，RSA 暗号で用いられるような巨大な数を素因数分解するために量子コンピュータに求められる性能（量子ビット数や量子ゲートエラー率など）の水準はどれほどかを評価する．

本稿では試行錯誤の結果，$N = 35, a = 4, r = 6$ に対する簡略化手法を見出した．結果，測定結果の正確さが向上し，実験は成功したので，4 章及び 5 章で詳細に報告する．

2. 量子ビットの結合方式

2.1 隣接結合型

隣接する量子ビットのみが相互に接続された方式を隣接結合型と定義する．この方式の代表的な量子コンピュータとして IBMQ がある．IBMQ は超伝導型クラウド量子コンピュータであり，1 つの量子ビットに対して 1～3 個の量子ビットが隣接している．IBMQ は数十種類の量子コンピュータを展開しているが，量子ビット数や全体の形状はそれぞれ異なる．その一例として ibmq_lima を挙げる．この量子コンピュータは 5 量子ビットで構成され， 図 1 に示すような T 字のトポロジーとなっている．

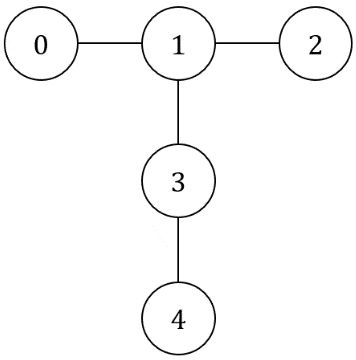

図 1　ibmq_lima のトポロジー
Figure 1　The topology of ibmq_lima.

T 字の量子コンピュータの特徴として，1 番で示された量子ビットが 3 つの量子ビットと接続されていることが挙げられる．量子ゲート操作には CNOT ゲートのような 2 量子ビット以上に関係する操作がある．他の複数の量子ビットと接続する回数が最も多い量子ビットについて，これを 1 番の量子ビットと割り付けることで，測定結果の正確さを最大化できる．これは，IBMQ において未接続の量子ビット同士で CNOT ゲート論理などを作用する場合に，SWAP ゲートを用いた疑似的に接続で対応することに起因する. SWAP ゲート自体は CNOT ゲート 3 つから構成され，CNOT ゲート 3 つ分のエラーが重畳される．特に，Shor のアルゴリズムでは CNOT ゲートや Toffoli ゲート（CCNOT ゲート）が多用されることから，SWAP ゲートの多用は測定結果の正確さに大きく影響する．

ところで超伝導型の量子コンピュータは 1999 年にはコヒーレンス時間（重ね合わせ状態が崩れてしまうまでの時間）がナノ秒にも満たなかったが，現在ではミリ秒に達している．しかし，まだ実用的なレベルに達しているとは言えない[9].

超伝導型の量子コンピュータの量子ビット数は大幅に増加を続けており, IBM Quantum Summit 2021 においては 127 量子ビットの量子プロセッサ「Eagle」が公開された[10]. 今後も飛躍的に量子ビット数が増加することが期待される.

2.2 全結合型

全ての量子ビット同士が接続した方式を全結合型と定義する. この方式の代表的な量子コンピュータとして IonQ がある. IonQ はイオントラップ型クラウド量子コンピュータである．IonQ は 11 量子ビットから構成され，全ての量子ビットが相互に結合されている[11]. 図 2 に IonQ のトポロジーを示す.

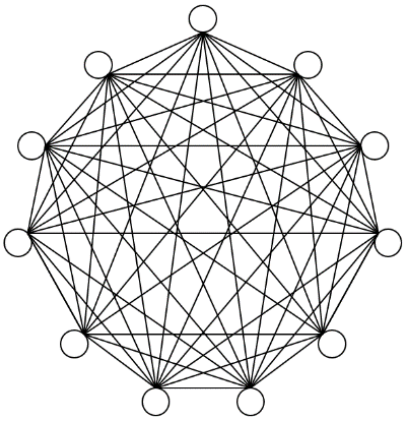

図 2　IonQ のトポロジー
Figure 2　The topology of IonQ.

全ての量子ビットが接続されていることから，2 量子ビット以上に対するゲート操作において，IBMQ のように SWAP ゲートを挿入する必要がない.

ここで IonQ はコヒーレンス時間が 0.2 秒で，IBMQ と比較して長いというメリットがあるが，超伝導型量子コンピュータと比較して量子ビット数が 11 個と少ない．2021 年には IonQ で誤り訂正を成功したという報告が挙がっているが[12]，規模が大きい量子回路を誤り訂正も含めて実行するためには，量子ビット数のさらなる増加が求められる．

3. Shor のアルゴリズム量子回路実装

3.1 位数発見問題

ある自然数 N と，それよりも小さく互いに素な整数 a を選択する．その後，$a^r \bmod N = 1$ となる最小の整数 r を発見する．これは，$f(x) = a^x \bmod N$の周期を求めることと同義であり，これを位数発見問題と呼ぶ．この r を位数と呼び，これを多項式時間で効率的に導く古典的アルゴリズムは存在しない．

3.2 Shor のアルゴリズム

素数 p, q の合成数 $N = pq$ を素因数分解することを考える．ランダム数 a を選択後，Shor のアルゴリズムによって位数発見問題を多項式時間で解く．これにより，位数 r を求める．最後に，$\gcd\left(a^{r/2} \pm 1,\ N\right)$を計算することで，素数 p, q が求まる．

3.3 Shor のアルゴリズム量子回路実装

ここでは，簡単のために $N = 15$, $a = 7$ で説明を行う．この場合，$f(x) = 7^x \bmod 15$を計算すると$f(x)$ =1, 7, 4, 13, 1 …となり，位数 $r = 4$ であることが分かる．

表 1 に$f(x)$の値を 2 進数で表現した真理値表を示す。ただし，$c = (c_0, c_1, c_2)$, $q = f(c) = (q_0, q_1, q_2, q_3)$とする．

表 1　$N = 15$, $a = 7$ における 3 ビット制御レジスタと標的レジスタの真理値表

Table 1　Truth table of 3-bit control register and target register for $N = 15$, $a = 7$

c_0	c_1	c_2	q_0	q_1	q_2	q_3
0	0	0	0	0	0	1
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	1	1	0	1
1	0	0	0	0	0	1
1	0	1	0	1	1	1
1	1	0	0	1	0	0
1	1	1	1	1	0	1

今後，c を制御レジスタ，q を標的レジスタと呼ぶことにする．まず，制御レジスタには H ゲート（アダマールゲート）を作用させる．H ゲートは量子状態を重ね合わせ状態に遷移させることができる．これにより，制御レジスタの初期化を行う．また、今回は制御レジスタを 3 ビットとする．制御レジスタのビット数が多いほど，位数決定の信頼性が高くなる．

次に，標的レジスタの量子ビットを真理値表に表された量子状態のみを取るように量子ゲート操作を行う．この量子回路の実装方法は 1 種類とは限らず、いくつか考えることができる．よって、実装方法によって測定結果の正確さが大きく変化するため，効率的に量子回路を組むことが求められる．

最後に，制御レジスタに対して逆量子フーリエ変換を施し，測定を行う．測定結果を基に，連分数展開などの手法によって位数 r を一定確率で求めることができる．

3.4 量子回路の基本的な実装方法

論理積に相当する量子ゲート操作を図 3 に示す．また，その真理値表を表 2 に示す．

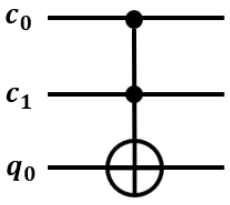

図 3　論理積に相当する量子ゲート操作

Figure 3　Quantum gates equivalent to logical AND.

表 2　論理積の真理値表

Table 2　Truth table of logical AND.

c_0	c_1	q_0
0	0	0
0	1	0
1	0	0
1	1	1

論理和に相当する量子ゲート操作を図 4 に示す．また，その真理値表を表 3 に示す．

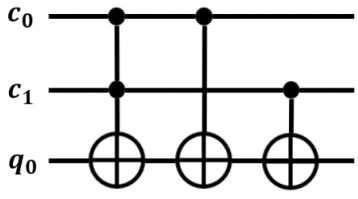

図 4　論理和に相当する量子ゲート操作

Figure 4　Quantum gates equivalent to logical OR.

表 3　論理和の真理値表

Table 3　Truth table of logical OR.

c_0	c_1	q_0
0	0	0
0	1	1
1	0	1
1	1	1

また，否定は X ゲートで表現する．基本的にはこれらのルールに従って，量子回路を構成していくことにする．

4. Shor のアルゴリズムの実装実験

4.1 実験環境

本稿では測定結果の正確さに焦点を当てている．2.1 節で述べたように IBMQ では余分な SWAP ゲートの挿入により，Shor のアルゴリズムとは本質的に関係のないエラーの影響が大きい．また，IBMQ では使用できる量子ビット数に制限があり，Shor のアルゴリズムの簡略化評価を行うに十分な量子ビットが確保できない．よって，本稿の実験では 2.2 節で述べた全結合型の IonQ を用いる．

4.2 $N = 15, a = 7$ の量子回路

図 5 に $N = 15$, $a = 7$ における量子回路図を示す．位数 $r = 4$ である．[13]の量子回路を参考にしているが，説明の都合上，量子ゲートの配置順を変更している（結果に影響はない）．

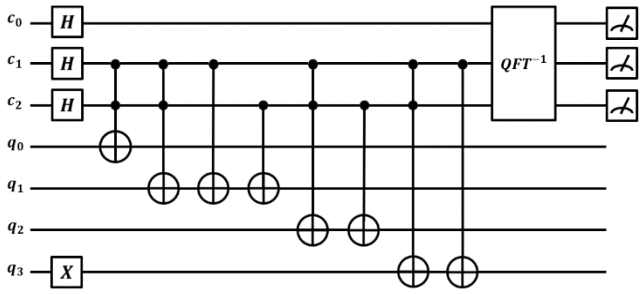

図 5　$N = 15$, $a = 7$ の Shor のアルゴリズム量子回路
Figure 5　Quantum circuit of Shor' factoring algorithm for $N = 15$, $a = 7$.

まず，q_0 について考える．表 1 より，

$$q_0 = c_1 \cdot c_2 \tag{1}$$

が成り立つ．よって，図 3 と同様にして q_0 を実装できる．

次に，q_1 について考える．表 1 より，

$$q_1 = c_1 + c_2 \tag{2}$$

が成り立つ．よって，図 4 と同様にして q_1 を実装できる．

次に，q_2 について考える．表 1 より，

$$q_2 = \overline{c_1} \cdot c_2 \tag{3}$$

が成り立つ．この際，図 3 の操作を行った後に c_2 を制御ビットとする CNOT ゲート操作を行う．これにより，q_2 を実装できる．

最後に，q_3 について考える．表 1 より，

$$q_3 = \overline{c_1 \cdot \overline{c_2}} \tag{4}$$

が成り立つ．まずは q_3 に X ゲートを作用させることで全体の否定を表す．その後，図 3 の操作を行い，c_1 を制御ビットとする CNOT ゲート操作を行う．これにより，q_3 を実装できる．

以上のように量子ゲート操作を行った後，制御レジスタの量子ビットに逆量子フーリエ変換を施し，測定を行う．

この実装を行った場合，CNOT ゲートは 4 回，Toffoli ゲートは 4 回使用される．Toffoli ゲートは図 6 に示すように 6 個の CNOT ゲートで構成されるため，量子回路全体では CNOT ゲートが 28 回使用されたことになる．なお，この回数は mod 計算部分のゲート数のみをカウントしており，逆フーリエ変換や測定時の調整用に挿入する SWAP ゲートは含めていないことに注意されたい．

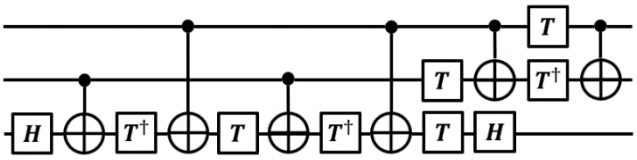

図 6　Toffoli ゲート
Figure 6　Toffoli gate.

4.3 簡略化の指針

簡略化の方針を述べる．3.4 節に示した基本ルールに忠実に量子ゲートの配置が終わった後，Toffoli ゲートに着目する．全く同様の 2 量子ビットを制御ビットとする Toffoli ゲートが複数存在する場合は，最初の 1 つだけを残し，後はその Toffoli ゲートの標的ビットを制御ビットとする CNOT ゲートで代用する．

Toffoli ゲートに限らず，複数の連続する量子ゲート操作が再度行われている部分があれば、同じように簡略化を施していく．これらの処理をすることで総量子ゲート数が減少し，測定結果の正確さが向上することを見出した．

4.4 $N = 15, a = 7$ の量子回路（簡略化後）

図 5 の量子回路を簡略化した量子回路図を図 7 に示す．

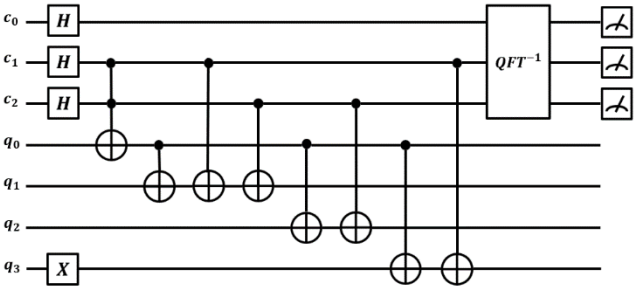

図 7　$N = 15$, $a = 7$ の Shor のアルゴリズム量子回路（簡略化後）
Figure 7　Simplified quantum circuit of Shor' factoring algorithm for $N = 15$, $a = 7$.

簡略化により，CNOT ゲートの制御ビットとして制御レジスタ c の量子ビットだけでなく，標的レジスタ q の量子ビットも採用された．

図 5 において c_1 と c_2 を制御ビットとする Toffoli ゲート

が 4 度使用されている．また，q_0 は 1 つ目の Toffoli ゲート操作以降，量子状態は変化しない．よって，それ以降の 3 つの Toffoli ゲートは，q_0 を制御ビットとする CNOT ゲートで代用し，簡略化することが可能である．

この事実から，図 7 では図 5 の 3 個の Toffoli ゲートを，q_0 を制御ビットとする CNOT ゲートに置き換えることで，$q_1 \sim q_3$ を実装した．

この実装を行った場合，CNOT ゲートは 7 回，Toffoli ゲートは 1 回使用される．よって，量子回路全体では CNOT ゲートは 13 回使用されたことになる．これは図 5 の場合と比較して約半減している．

以上のように標的レジスタの量子ビットを上手く活用することで、CNOT ゲート数の大幅な削減が可能であることが分かる．

4.5 $N = 35, a = 4$ の量子回路

$N = 35, a = 4$ の場合，$f(x) = 4^x \bmod 35$を計算すると$f(x)$ =1, 4, 16, 29, 11, 9, 1 …となり，位数 $r = 6$ であることが分かる．表 4 に$f(x)$の値を 2 進数で表現した真理値表を示す. また，図 8 に $N = 35, a = 4$ における量子回路図を示す.

表 4　$N = 35, a = 4$ における 3 ビット制御レジスタと標的レジスタの真理値表

Table 4　Truth table of 3-bit control register and target register for $N = 35, a = 4$

c_0	c_1	c_2	q_0	q_1	q_2	q_3	q_4
0	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1
1	0	1	0	1	0	0	1
1	1	0	0	0	0	0	1
1	1	1	0	0	1	0	0

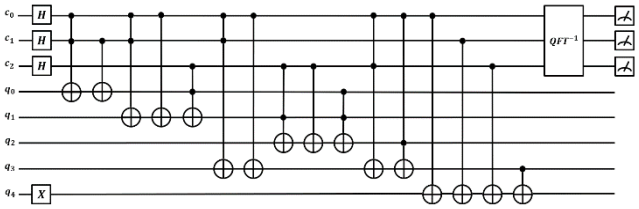

図 8　$N = 35, a = 4$ の Shor のアルゴリズム量子回路

Figure 8　Quantum circuit of Shor' factoring algorithm for $N = 35, a = 4$.

今回は，既に標的レジスタ中の量子ビットを制御ビットとして用いている．これは，$q_1 \sim q_4$ のように状態が 1 となる回数が奇数回の場合，規則性のある制御レジスタの情報だけでは表現することができないからである．

図 8 の量子回路において，$q_0 \sim q_2$ は以下のように表される．

$$q_0 = \bar{c_0} \cdot c_1 \tag{5}$$

$$q_1 = c_0 \cdot \bar{c_1} + c_2 \cdot q_0 \tag{6}$$

$$q_2 = c_2 \cdot \bar{q_1} + q_0 \cdot q_1 \tag{7}$$

q_3, q_4 はやや変則的である．q_3 は制御レジスタ $c = (100)_2$ の時のみ 1 である．また，q_4 は制御レジスタ中の量子ビットが偶数個 1 の場合に加え，q_3 が 1 の場合に 1 となる．

この実装を行った場合，CNOT ゲートは 8 回，Toffoli ゲート 8 回使用される．よって，量子回路全体では CNOT ゲートは 56 回使用されたことになる．

4.6 $N = 35, a = 4$ の量子回路（簡略化後）

図 8 の量子回路を簡略化した量子回路図を図 9 に示す.

まず，q_0 については CNOT ゲートの配置が 1 つ後ろに移動しているが、操作自体に変更はない.

次に，q_1 についてである．c_0 と c_1 を制御ビットとする Toffoli ゲートが再度使用されていることから，q_0 を標的ビットとする Toffoli ゲートの次に q_1 を標的ビットとする CNOT ゲートを加えて簡略化を施した．この際，q_0 に対する CNOT ゲート操作の影響を受けないようにする必要があるため，q_0 で CNOT ゲートの配置を変更した．また，c_2 と

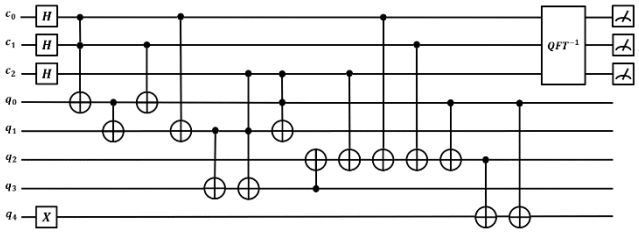

図 9　$N = 35, a = 4$ の Shor のアルゴリズム量子回路（簡略化後）

Figure 9　Simplified quantum circuit of Shor' factoring algorithm for $N = 35, a = 4$.

q_0 を制御ビットとする Toffoli ゲートは，2 つ後ろに移動しているが操作に変更はない.

次に，q_3 についてである．図 8 において c_0 と c_1 を制御ビットとする Toffoli ゲートと c_0 を制御ビットとする CNOT ゲートが連続しているが，これは図 9 の 3 つ目の CNOT ゲートを施した後の q_1 を制御ビットとした CNOT ゲートで簡略化が可能である．また，c_0 と c_2 を制御ビットとする Toffoli ゲートと c_0 と q_2 を制御ビットとする Toffoli ゲートの 2 つは，c_2 と q_1 を制御ビットとする Toffoli ゲートで簡略化が可能である.

次に，q_2 についてである．c_2 と q_1 を制御ビットとする Toffoli ゲートは，q_3 を制御ビットとする CNOT ゲートで代用することができる．また，q_0 と q_1 を制御ビットとする Toffoli ゲートは，c_0 と c_1 と q_0 をそれぞれ制御ビットとする 3 つの CNOT ゲートで簡略化可能である.

最後に，q_4についてである．q_0 と q_2 の状態に着目すれば，CNOT ゲート 2 つで簡略化が可能である.

この実装を行った場合，CNOT ゲートは 11 回，Toffoli ゲ

ートは 3 回使用される．よって，量子回路全体では CNOT ゲートは 29 回使用されたことになる．これは図 8 の場合と比較して約半減している．

5. 測定結果

5.1 測定環境

以降の測定結果は最終アップデートが 2021 年 10 月 25 日，21 時 11 分（UTC）の IonQ におけるものである．1 量子ビットゲート操作に対する平均忠実度は 0.9973，2 量子ビットゲート操作に対する平均忠実度は 0.97838 となっている．また，Amazon Braket simulator による理想的な結果と，IonQ による実測値を比較することで正確さの向上を視覚的に理解しやすいものとしている．

5.2 $N = 15, a = 7$ の量子回路測定結果

図 10 に $N = 15, a = 7$ における Shor のアルゴリズム量子回路（図 5）の測定結果を示す．青色で示した棒グラフ（Amazon braket simulator）は理想的な結果であり，制御レジスタの量子状態として，|000⟩, |010⟩, |100⟩, |110⟩ にピークが観測されれば実験は成功と言える．

図 10 の黄色で示した棒グラフは IonQ による実測値を示している．これによると，|000⟩ のピークに関しては理想値と誤差はほぼ見られないが，|010⟩, |100⟩, |110⟩ のピークに関しては理想値との誤差が大きい．また，|001⟩, |011⟩, |101⟩, |111⟩ という本来観測が想定されない量子状態もそれぞれ 5%以下の確率であるが観測されている．この結果に関しては，誤差はあるもののピークを識別する分には問題ないと言える．

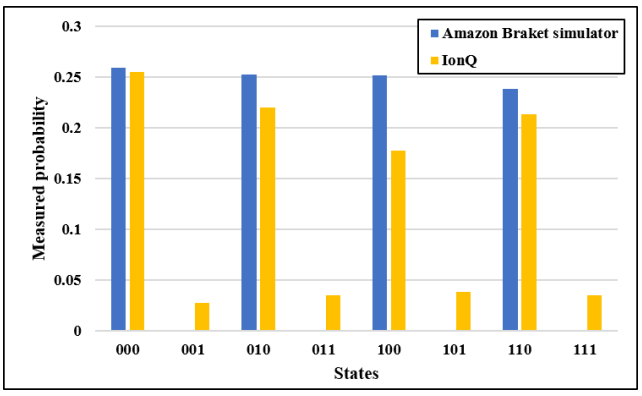

図 10　$N = 15, a = 7$ の Shor のアルゴリズム量子回路（図 5）の測定結果

Figure 10　Measurement data of quantum circuit of Shor' factoring algorithm for $N = 15$, $a = 7$ (Figure 5).

次に簡略化後の $N = 15, a = 7$ における Shor のアルゴリズム量子回路（図 7）の測定結果を図 11 に示す．

図 11 によると，|110⟩ のピークは理想値と外れているが，|010⟩, |100⟩ のピークに関しては改善された．また，理想値以外の量子状態の観測確率も低減されている．

図 10 と図 11 の結果を基に，各量子ビットの理想値と実測の相対誤差を算出した．そして，（1 - 相対誤差平均）を計算すると，図 10 では 0.864，図 11 では 0.934 となり，正確さが 7.0%向上したことを確認した．

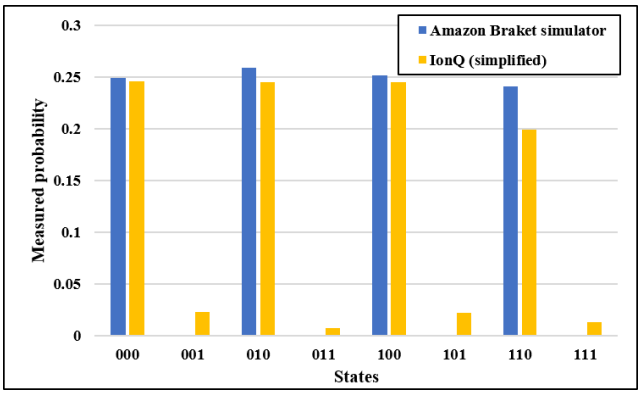

図 11　簡略化後の $N = 15, a = 7$ の Shor のアルゴリズム量子回路（図 7）の測定結果

Figure 11　Measurement data of simplified quantum circuit of Shor' factoring algorithm for $N = 15$, $a = 7$ (Figure 7).

5.3 $N = 35, a = 4$ の量子回路測定結果

図 12 に $N = 35, a = 4$ における Shor のアルゴリズム量子回路（図 8）の測定結果を示す．これによると，制御レジスタの量子ビットの量子状態として，|000⟩, |100⟩ にピークが観測されれば，実験は成功と言える．

しかし，IonQ の実測値では 0.1 ~ 0.15 程度の均一な量子状態が観測され，実験は不成功であった．

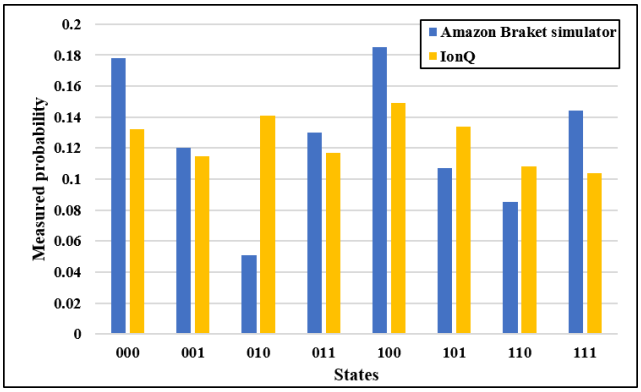

図 12　$N = 35, a = 4$ の Shor のアルゴリズム量子回路（図 8）の測定結果

Figure 12　Measurement data of quantum circuit of Shor' factoring algorithm for $N = 35$, $a = 4$ (Figure 8).

次に，簡略化後の $N = 35, a = 4$ における Shor のアルゴリズム量子回路（図 9）の測定結果を図 13 に示す．図 13 によると，実測値においても |000⟩, |100⟩ のピークが観測されており，その他の量子状態全体に関しても理想値の傾向に追従している．よって，実験は成功と言える．

（1 - 相対誤差平均）を計算すると，図 12 では 0.605，図 13 では 0.740 となり，正確さが約 13.5%向上したことを

確認した．ただし，これらのピーク情報では位数 $r = 6$ を正確に求めることはできない．よって，より位数推定の信頼性を高めるためには制御レジスタの量子ビット数を 3 ビットから 4 ビットに増加することが必要である．IonQ によって，その場合の量子回路を実装し測定を行ったが，量子ゲート数が多くなりエラーが測定結果に大きく影響した結果，実験は成功しなかった．

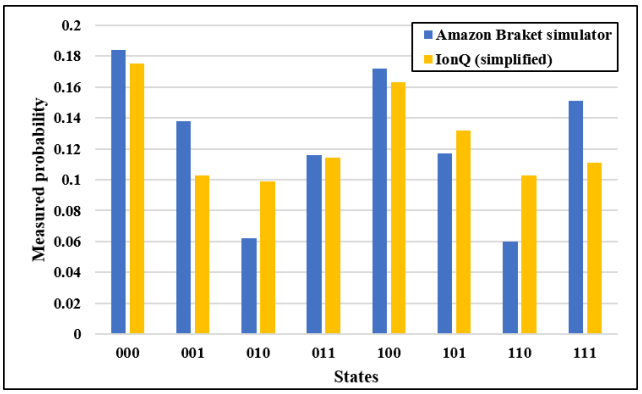

図 13　簡略化後の $N = 35$, $a = 4$ の Shor のアルゴリズム量子回路（図 9）の測定結果

Figure 13　Simplified quantum circuit of Shor' factoring algorithm for $N = 35$, $a = 4$ (Figure 9).

5.4 測定結果のまとめ

測定結果の正確さについて表 5 に結果をまとめる．また CNOT ゲート総数の比較について表 6 に結果をまとめる．

表 5　簡略化による正確さの向上結果

Table 5　The results of improving accuracy by simplifying.

	$N = 15, a = 7$	$N = 35, a = 4$
Before	0.864	0.605
After	0.934	0.740

表 6　CNOT ゲート総数の比較

Table 6　The comparison of total CNOT gates.

	$N = 15, a = 7$	$N = 35, a = 4$
Before	28	56
After	13	29

6. おわりに

本稿では，Shor のアルゴリズム量子回路の簡略化により，位数 $r = 6$ である $N = 35$, a = 4 を対象とした実験に成功した．この結果から，量子ビットやゲート操作によるエラーという根本的なエラーの改善だけではなく，Shor のアルゴリズム量子回路の実装方法によってもエラーを低減させることができることが確認された．

今後の展望として，特定の N を対象とする Shor のアルゴリズム量子回路について，測定結果の正確さを最大化する回路を自動的に生成するシステムを構築することで，RSA 暗号などの解読に必要な最小ゲート数を把握し，量子ビットに求められるエラー率の水準を明確化していくことを目指す．

謝辞　本研究は JSPS 科研費 JP21K14198 の助成を受けたものです．

参考文献

[1] 下山 武司, 伊豆 哲也, 小暮 淳, 素因数分解と RSA 暗号の安全性, J-STAGE 応用数理 19 巻 1 号 p. 28-32, 2009.
[2] 株式会社日本総合研究所 先端技術ラボ, 量子コンピュータの概説と動向 ～量子コンピューティング時代を見据えて～, 2020.
[3] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26, 1484-1509 (1997).
[4] 高安 敦. Shor のアルゴリズム実装動向調査. CRYPTREC EX-3005-2020 (2021).
[5] Monz, T. et al. Realization of a scalable shor algorithm. Science 351, 1068-1070 (2016).
[6] Skosana, U & Tame, M. Demonstration of Shor's factoring algorithm for N = 21 on IBM quantum processors. Nature. Scientific Reports 11, 16599 (2021).
[7] Lanyon, B. P. et al. Experimental demonstration of a compiled version of Shor's algorithm with quantum entanglement. Phys. Rev. Lett 99, 250505 (2007).
[8] Amico, M., Saleem, Z. H. & Kumph, M. An experimental study of Shor's factoring algorithm on IBM Q. Phys. Rev. A 100, 012305 (2019).
[9] 向井 寛人, 朝永 顕成, 蔡 兆申, 超伝導量子コンピュータの基礎と最先端, J-STAGE 低温工学 53 巻 5 号 p. 278-286, 2018.
[10] IBM THINK Blog Japan, IBM Quantum Eagle、100 量子ビットの壁を破る, https://www.ibm.com/blogs/think/jp-ja/ibm-quantum-eagle-breaks-the-100-qubit-barrier/
[11] Wright, K. et al. Benchmarking an 11-qubit quantum computer. Nature Communications 10, 5464 (2019).
[12] Egan, L. et al. Fault-tolerant control of an error-corrected qubit. Science 598, 281-286 (2021).
[13] 中山 茂. Python 量子プログラミング入門 (2018).