Practical running time of factoring by quantum circuits

Noboru Kunihiro[1] *

[1] *The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585, Japan.*

Abstract. We evaluate the exact number of gates for circuits of Shor's factoring algorithm. We estimate the running time for factoring a large composite such as 530 bit numbers by appropriately setting unit times. For example, we show that on the condition that each Toffoli gate is operated with 70μsec, the running time for factoring 530 bit number is 1 month even if the most efficient circuit is adopted. Consequently, we find that if we adopt the long unit-time devices or qubit-saving circuits, factorization will not be completed within feasible time and we point out that long unit time may become a new problem preventing a realization of quantum computers.

Keywords: Shor's factoring algorithm, quantum circuit, practical running time

1 Introduction

The security of the RSA cryptosystems is based on the difficulty of factoring a large composite integer. The present world record of the factorized largest composite is RSA-160, which is a 530-bit number. This number was factorized within 1 months by using 109 PCs.

Shor proposed a quantum algorithm which factorize a composite number in polynomial time [1]. Our goal is to estimate the actual time for factoring by using Shor's algorithm. In order to attain it, we need to evaluate the number of elementary gates, such as Toffoli or rotation gates. Although the number of qubits and the order of the number of gates have so far been studied, we need to evaluate the exact number of gates for factoring circuits in order to attain the above purpose. In this paper, we evaluate the exact number of gates (not only its order) for three previously proposed circuits of modular exponentiation. The unit time is different from various devices. Hence, by setting the unit time appropriately, we estimate the running time. We show that if we adopt some device with long unit time or if we adopt qubit-saving circuits, factoring a large composite may not be completed within feasible time. Our results lead to the conclusion that the long unit time may become a new problem for realization of quantum computers.

2 Circuits for Modular Exponentiation

Let N be a n-bit composite number to be factored. Shor's factoring algorithm is composed of two parts: a modular exponentiation and an inverse of quantum Fourier transform. The aim of the modular exponentiation is to construct a state

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |a^x \bmod N\rangle \tag{1}$$

from the initial state $1/\sqrt{2^m}\sum_{x=0}^{2^m-1}|x\rangle|1\rangle$, where a is a randomly chosen integer less than N and $m = 2n$. The aim of the inverse of quantum Fourier transform is to obtain a period of the function: $a^x \bmod N$ from Eq.(1). The former is *difficult* than the latter.

The modular exponentiation Mod-$EXP(a) : |x\rangle|0\rangle \rightarrow |x\rangle|a^x \bmod N\rangle$ is composed of m controlled modular multiplications [2]. The modular multiplication: Mod-$MUL(d) : |y\rangle \rightarrow |dy \bmod N\rangle$ is composed of two modular product-sum operations and one $SWAP$ operation. The modular product-sum operation: Mod-PS : $|y\rangle|t\rangle \rightarrow |y\rangle|t + dy \bmod N\rangle$ is composed of n controlled modular additions. The modular addition: Mod-$ADD(d) : |y\rangle \rightarrow |y+d \bmod N\rangle$ is composed of some additions: $ADD(d) : |y\rangle \rightarrow |y + d\rangle$.

We show how to construct Mod-ADD from ADD. Two controlled qubits x_i, y_j are included in the Mod-ADD operation since we use each one controlled qubit in Mod-EXP and Mod-PS, respectively. We describe how to construct $C(x_i, y_j)$-Mod-$ADD(d)$ from ADD. We have two strategy for constructing the above operation. In ADD operation, we use three registers: R_1, R_2 and R_3. These consist of 1 qubit, n qubits, and 1 qubit, respectively. The ADD operator is applied to the register connected R_1 and R_2. We only represent Type2.

Step1 $C(x_i, y_j)$-$ADD(d)$
Step2 $ADD(2^n - N)$
Step3 $NOT(R_1), C(R_1)$-$NOT(R_3)$ and $NOT(R_1)$
Step4 $C(R_3)$-$ADD(N)$
Step5 $C(x_i, y_j)$-$NOT(R_1)$
Step6 $C(x_i, y_j)$-$ADD(2^n - d)$
Step7 $C(R_1)$-$NOT(R_3)$
Step8 $C(x_i, y_j)$-$ADD(d)$
Step9 $NOT(R_1)$

The Type1 ADD is obtained by adding two controlled qubits $C(x_i, y_j)$- and combining Step1 and Step2, which become $C(x_i, y_j)$-$ADD(d + 2^n - N)$. The Type1 consists of one C^3-ADD, three C^2-ADD, two C^3-NOT and four C^2-NOT. The Type2 consists of three C^2-ADD, one C-ADD, one ADD, one C^2-NOT, two C-NOT and three NOT. Note that $C(x_i, y_j)$- is unnecessary for operators in Steps 2, 3, 4, 7 and 9. Which type is effective depends on how to construct ADD.

3 Construction of ADD circuits

Three constructions have been known for ADD. These are classical ADD, ADD using generalized Toffoli, and Quantum Addition. We describe how to construct these ADDs and evaluate the exact number of gates.

3.1 Classical ADD (C-ADD)

The classical ADD [2] is based on classical addition circuits. This circuit needs $n - 1$ clean ancilla qubits as

*kunihiro@ice.uec.ac.jp

carries. The average number of gates for C-ADD is given by $(2n-3, 2n-\frac{3}{2}, \frac{3}{2}n-2)$. The first element is the number of Toffoli gates, and the second and the third elements are the number of C-NOT and NOT gates, respectively. The number of gates for operating one C^2-Mod-ADD is given by $(6n-9, 8n-\frac{15}{2}, \frac{17}{2}n-\frac{19}{2}, \frac{7}{2}n-\frac{3}{2}, \frac{3}{2}n+1)$ in the case of Type2. We omit the case of Type1 since the Type2 is more efficient. Since Mod-EXP consists of $2nm$ C^2-Mod-ADD and m C-$SWAP$, the total number of gates for Mod-EXP is given by $m(12n^2-18n, 16n^2-15n, 17n^2-18n, 7n^2-n, 3n^2+2n)$. Next, we decompose C^4-, C^3-NOT into Toffoli gates. The following are known about the decomposition of C^k-NOT into Toffoli gates [3]. *If there are $k-2$ clean (or unclean) ancilla qubits, C^k-NOT can be decomposed into $2k-3$ (or $4k-8$) Toffoli gates.* If we use C-ADD, we can apply the first condition in decomposing almost every C^k-NOT since we can use unused carry bit as clean ancilla qubits. Then, C^k-NOT, where $k=3,4$ can be decomposed into 3 and 5 Toffoli gates, respectively. Hence, the number of gates for constructing Mod-EXP are given by Eq. (2). The number of qubits are given by $m+3n+1$.

$$\text{Type2: } m(125n^2-153n, 7n^2-n, 3n^2+2n) \quad (2)$$

3.2 ADD using generalized Toffoli (GT-ADD)

First, we describe the circuit for adding 2^i into $|b_{n-1}\dots b_0\rangle$. By the sequence: $C(b_i,\dots,b_{n-1})$-$NOT(b_n)$, $C(b_i,\dots,b_{n-2})$-$NOT(b_{n-1}),\dots,C(b_i)$-$NOT(b_{i+1}), NOT(b_i)$, the above is realized. If we add $a=a_{n-1}\dots a_0$ into $|b_{n-1}\dots b_0\rangle$, we run the above sequence for i such that $a_i=1$. This circuit needs no ancilla qubits [2]. The average number of gates for GT-ADD are given by $(1/2, 1, 3/2, ..., n, n)$. The i-th element is the number of C^{n+1-i}-NOT gates and the last one is the number of NOT gates.

The total number of gates for Mod-EXP is given as follows. $\#C^i$-$NOT = n(4n-4i+13)$, where $4 \le i \le n+3$, $\#C^3$-$NOT = 4n^2+4n$, $\#C^2$-$NOT = 3n^2+9n$ and $\#C$-$NOT = 2n$. In this case, we omit the Type2 since Type1 is more effective.

The C^k-NOT gates, except for C^{n+3}-NOT, can be decomposed into $4(k-2)$ Toffoli gates since we can use more than $k-2$ qubits as *unclean* ancilla qubits. By adding one ancilla qubit, we can decompose C^{n+3}-NOT. Hence, the total number of Toffoli gates is given by Eq. (3). The number of qubits are given by $m+2n+3$.

$$m\left(\frac{8}{3}n^4+10n^3+\frac{43}{3}n^2+24n, 2n, 0\right). \quad (3)$$

3.3 Quantum ADD (Q-ADD)

By applying quantum ADD (q-$ADD(a)$) [4] to the input state $|\phi(y)\rangle$, we obtain $|\phi(y+a)\rangle$, where $\phi(y)$ is the quantum Fourier Transform (QFT) of y. Hence, ADD can be realized as follows. First, we apply QFT to $|y\rangle$ to obtain $|\phi(y)\rangle$. Second, we apply q-$ADD(a)$ to obtain $|\phi(y+a)\rangle$. Finally, we apply QFT^{-1} to obtain $|y+a\rangle$. This circuit also needs no ancilla qubits.

Next, we evaluate the number of gates. In this case, we omit the Type1 since the Type2 is more effective. Let a rotation gate:$R_k=(1,0;0,\exp(2\pi i/2^k))$. QFT is composed of $n+1$ Hadamard gate: H and $n+2-i$ controlled rotation gates: C-R_i, where $2 \le i \le n+1$. The q-ADD operation is composed of $(n+2-i)/2$ times R_i gates on average. Hence, the total number of gates for Mod-EXP are given as follows. $\#C^2$-$R_i = 3mn(n+2-i), (1 \le i \le n+1)$, $\#C$-$R_i = mn(n+2-i), (1 \le i \le n+1)$, $\#R_i = m(9n+2)(n+2-i), (2 \le i \le n+1)$, $\#R_1 = mn(n+1)$, $\#H = m(8n+2)(n+1)$, $\#C^2$-$NOT, \#C$-$NOT, \#NOT = mn, m(6n+4), m(4n+4)$, respectively. The C^2-R_i (or C-R_i)gate can be decomposed into six (two) C-NOT and eight (four) single qubit operation [3]. After the decomposion, the number of C-NOT gates becomes $m(10n(n+1)(n+2)+6n+4)$ and the number of single qubit operators becomes $m(n+1)(n+2)(37n+2)/2$. It is known that if i is large enough, R_i can be approximated as identity [4]. Then, the total number of gates can be reduced to $O(mn^2\log n)$. The number of qubits are given by $m+2n+2$.

4 Evaluation of the running time

Table1 shows the number of gates for factoring 530 and 1024 bit numbers. We set $m=1$ and $m=2n$ in the evaluation of the number of qubits and gates, respectively. Next, we estimate the running time for 530 bit numbers by setting the four various unit time, which is a time for operating elementary gate (Toffoli, R_i gate). Table 2 shows the running time for factoring 530 bit number.

Table 1: # of qubits and gates for 530 and 1024 bits

	World Record ($n=530$)		Recommended ($n=1024$)	
	qubits	# of gates	qubits	# of gates
C-ADD	1592	3.71×10^{10}	3074	3.80×10^{11}
GT-ADD	1064	2.25×10^{14}	2052	6.03×10^{15}
Q-ADD	1063	6.10×10^{12}	2051	8.48×10^{13}
		1.57×10^{11}		1.22×10^{12}

(In Q-ADD, the below shows with approximation.)

Table 2: Running Time for 530bit composite

unit time	1msec	0.1msec	1μsec	1nsec
C-ADD	1.18Y	43D	10H	37S
GT-ADD	7134Y	710Y	7.1Y	2.6D
Q-ADD	190Y/5.0Y	19Y/181D	70D/1.8D	1.7H/2.6M

(Y: years, D: days, H: hours, M: minutes, S: seconds)

The 530 bit number was factorized within 1 month even if classical computers were used. Consider the condition for quantum computers to exceed classical computers. If we adopt C-ADD, the Toffoli gate should be operated within 70μsec. If we adopt Q-ADD, the elementary gates should be operated within 16μsec. In order to exceed classical computers, either a quantum computer with 1592-qubits and 70μsec unit time or with 1064-qubits and 16μsec unit time must be realized.

References

[1] P. W. Shor, in Proc. of the 35th FOCS, pp. 124-134, 1994.

[2] V. Vedral, A. Barenco, and A. Ekert, Physical Review A, vol. 54, no. 1, pp. 147–153, 1996.

[3] A. Barenco, C. H. Bennett, et.al, Physical Review A, vi. 52, pp. 3457–3467, 1995.

[4] T. G. Draper, `quant-ph/0008033v1` , 2000.