Quantum Information and Computation, Vol. 5, No. 6 (2005) 440–448

A LINEAR-SIZE QUANTUM CIRCUIT FOR ADDITION WITH NO ANCILLARY QUBITS

YASUHIRO TAKAHASHI[a]
NTT Communication Science Laboratories, NTT Corporation
Atsugi, Kanagawa 243-0198, Japan

NOBORU KUNIHIRO[b]
The University of Electro-Communications
Chofu, Tokyo 182-8585, Japan

Received February 10, 2005
Revised June 12, 2005

We construct a quantum circuit for addition of two n-bit binary numbers that uses no ancillary qubits. The circuit is based on the ripple-carry approach. The depth and size of the circuit are $O(n)$. This is an affirmative answer to the question of Kutin [1] as to whether a linear-depth quantum circuit for addition can be constructed without ancillary qubits using the ripple-carry approach. We also construct quantum circuits for addition modulo 2^n, subtraction, and comparison that use no ancillary qubits by modifying the circuit for addition.

Keywords: quantum circuits, addition, ancillary qubits

Communicated by: R Jozsa & C Fuchs

1 Introduction

Shor showed in 1994 that there exists an efficient quantum algorithm for factorization [2]. Shor's algorithm requires elementary arithmetic operations such as addition and modular exponentiation. Using efficient quantum circuits for elementary arithmetic operations, we can construct efficient quantum circuits for Shor's algorithm. For example, using a quantum circuit for addition that uses no ancillary qubits, we can reduce the number of qubits used in the quantum circuit for Shor's algorithm [3]. Therefore, it is of interest to construct efficient quantum circuits for elementary arithmetic operations.

In this paper, we focus on addition of two binary numbers and reducing the number of ancillary qubits since addition is the most basic operation and qubits are very costly resources. There have been many studies of quantum circuits for addition. Vedral et al. and Svore et al. constructed quantum circuits for addition of two n-bit binary numbers that use $O(n)$ ancillary qubits [4, 5]. The circuit constructed by Vedral et al. is based on the ripple-carry approach and the depth and size of the circuit are $O(n)$. The one constructed by Svore et al. is based on the carry-lookahead approach and the depth and size of the circuit are $O(\log n)$ and $O(n)$, respectively. Kutin's quantum circuit for addition uses only one ancillary qubit [1]. The circuit is based on the ripple-carry approach and the depth and size of the circuit are $O(n)$. Draper's uses no ancillary qubits using quantum Fourier transforms [6]. The depth and

[a]Email: takahasi@theory.brl.ntt.co.jp
[b]Email: kunihiro@ice.uec.ac.jp

size of the circuit are $O(n)$ and $O(n^2)$, respectively. As is pointed out in [1], it is not known whether a linear-depth quantum circuit for addition can be constructed without ancillary qubits using the ripple-carry approach.

We construct a quantum circuit for addition of two n-bit binary numbers that uses no ancillary qubits. The circuit is based on the ripple-carry approach and the depth and size of the circuit are $O(n)$. This is an affirmative answer to the above question. To construct the circuit, we use the quantum circuit for the majority of three bits [1]. As we explain in the next section, our use of the circuit for the majority is different from that in [1]. We also construct quantum circuits for addition modulo 2^n, subtraction, and comparison that use no ancillary qubits by modifying the circuit for addition.

2 The Circuit

In the following, we use the standard notation for quantum states and use the standard diagrams for quantum circuits and quantum operations [7]. Note that X represents a NOT gate in circuit diagrams. Our circuit consists of CNOT, Toffoli, and NOT gates. The size of the circuit is defined as the total number of gates. The depth of the circuit is defined as follows. Input qubits are considered to have depth 0. For each gate G, the depth of G is equal to 1 plus the maximal depth of a gate that G depends on. The depth of the circuit is equal to the maximal depth of a gate in the circuit.

Let a and b be two n-bit binary numbers and $a_{n-1}\cdots a_0$ be the binary representation for a, where a_0 is the low-order bit. Similarly, let $b_{n-1}\cdots b_0$ be the binary representation for b and $s_n\cdots s_0$ be the binary representation for $a+b$. Let A_i and B_i denote the memory locations where a_i and b_i are initially located. Let Z be an additional memory location where some value z is initially located. The circuit computes the sum of a and b in place. A_i will contain a_i and B_i will contain s_i and Z will contain $z \oplus s_n$ at the end of the computation, where $\oplus$ denotes addition modulo 2.

In the ripple-carry approach, first, we compute the carry bit c_1 using a_0 and b_0. Then we compute the next carry bit c_2 using a_1 and b_1 and c_1. We continue this procedure and compute the carry bit c_{i+1} $(0 \le i \le n-1)$ in order. More precisely, the carry bit c_{i+1} is computed as follows.

$$c_{i+1} = \mathrm{MAJ}(a_i, b_i, c_i),$$

where $c_0 = 0$ and MAJ is the majority function for three bits. Note that the MAJ can be represented as follows.

$$\mathrm{MAJ}(a_i, b_i, c_i) = a_i b_i \oplus b_i c_i \oplus c_i a_i.$$

After we compute all carry bits, we compute s_i $(0 \le i \le n)$ using the following relationship.

$$s_i = a_i \oplus b_i \oplus c_i,$$

where $a_n = b_n = 0$.

The key ingredient of the circuit is a gate that computes the majority of three bits in place [1]. The gate is depicted in Fig. 1. When we input three bits $z \oplus b_i$, $z \oplus a_i$, $z \oplus c_i$ to the MAJ gate, it is easy to check that the gate outputs three bits $b_i \oplus c_i$, $a_i \oplus c_i$, $z \oplus c_{i+1}$ as shown in Fig. 2. Note that the MAJ gate will be used to compute $z \oplus c_{i+1}$ in our circuit, though the MAJ gate in [1] is used to compute c_{i+1}.

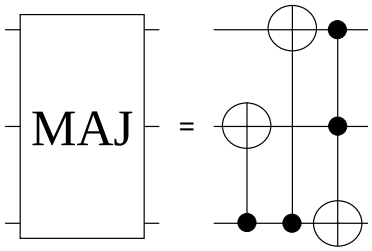

Fig. 1. The MAJ gate that computes the majority of three bits in place. The gate consists of two CNOT gates and one Toffoli gate.

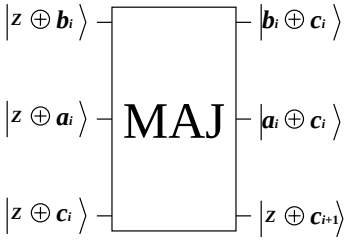

Fig. 2. The MAJ gate for the input $|z \oplus b_i\rangle|z \oplus a_i\rangle|z \oplus c_i\rangle$. The gate outputs $|b_i \oplus c_i\rangle|a_i \oplus c_i\rangle|z \oplus c_{i+1}\rangle$.

Our circuit consists of four stages. In the first stage, the bit value z is added to the memory locations A_i and B_i for $1 \leq i \leq n-1$. To do this, a CNOT gate is applied to a pair of memory locations Z and B_i and then is applied to a pair of memory locations Z and A_i for $1 \leq i \leq n-1$. The CNOT gates write $z \oplus b_i$ into B_i and $z \oplus a_i$ into A_i.

In the second stage, carry bits are computed using MAJ gates and are written into memory locations Z and B_0. To do this, Toffoli and MAJ gates are applied as follows.

1. Apply a Toffoli gate to a tuple of memory locations B_0 and A_0 and Z.
2. Apply a MAJ gate to a tuple of memory locations B_i and A_i and Z and then apply a Toffoli gate to a tuple of memory locations B_i and A_i and B_0 for $1 \leq i \leq n-2$.
3. Apply a MAJ gate to a tuple of memory locations B_{n-1} and A_{n-1} and Z.

The first Toffoli gate writes $z \oplus c_1$ into Z. As in Fig. 2, the first MAJ gate writes $b_1 \oplus c_1$ into B_1 and $a_1 \oplus c_1$ into A_1 and $z \oplus c_2$ into Z. Then, a Toffoli gate writes $b_0 \oplus c_1 \oplus c_2$ into B_0. At the end of the computation in the second stage, MAJ and Toffoli gates write $b_0 \oplus c_1 \oplus c_{n-1}$ into B_0 and $b_i \oplus c_i$ into B_i and $a_i \oplus c_i$ into A_i for $1 \leq i \leq n-1$ and $z \oplus c_n$ into Z. Note that $c_n = s_n$. The first and second stages for $n = 5$ are depicted in Fig. 3. A dashed box represents a MAJ gate. Note that a carry bit written in B_0 is used to erase a carry bit in A_i $(2 \leq i \leq n-1)$ in the third stage.

The third and fourth stages are stages mainly for the reverse computation. In the third stage, a CNOT gate is applied to a pair of memory locations B_0 and A_{n-i} and then a Toffoli gate is applied to a tuple of memory locations B_{n-i-1} and A_{n-i-1} and B_0 for $1 \leq i \leq n-2$. These CNOT and Toffoli gates write b_0 into B_0 and $a_i \oplus b_0 \oplus c_1$ into A_i for $2 \leq i \leq n-1$.

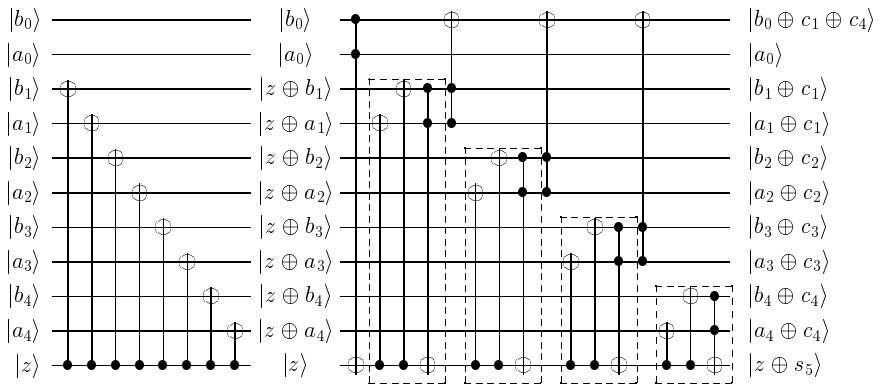

Fig. 3. The first and second stages for $n = 5$.

In the fourth stage, NOT and Toffoli and CNOT gates are applied as follows.

1. Apply a NOT gate to A_0.
2. Apply a Toffoli gate to a tuple of memory locations B_0 and A_0 and A_{n-i} for $1 \leq i \leq n-2$.
3. Apply a NOT gate to A_0.
4. Apply a Toffoli gate to a tuple of memory locations B_0 and A_0 and A_1.
5. Apply a CNOT gate to a pair of memory locations A_i and B_i for $0 \leq i \leq n-1$.

These NOT and Toffoli and CNOT gates write s_i into B_i and a_i into A_i for $0 \leq i \leq n-1$. The third and fourth stages for $n = 5$ are depicted in Fig. 4.

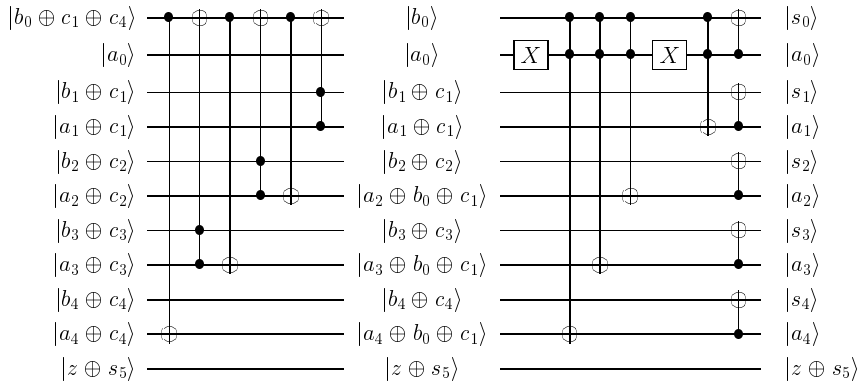

Fig. 4. The third and fourth stages for $n = 5$.

Our circuit for addition is constructed by combining the four stages. The circuit for $n = 5$ is depicted in Fig. 5, where we move some gates to reduce the depth of the circuit. It follows

from the above construction that the depth and size of the circuit are linear in the length of the input and that the circuit uses no ancillary qubits. In the following, we compute the depth and size of the circuit precisely for two n-bit binary numbers, where we assume $n \geq 3$. In the first stage, the number of CNOT gates is $2n-2$ and therefore the depth and size of the stage are $2n-2$. In the second stage, the number of Toffoli gates is $2n-2$ and the number of CNOT gates is $2n-2$. Therefore, the size of the stage is $4n-4$. When a Toffoli gate after a MAJ gate is applied, the Toffoli gate and the CNOT gate in the next MAJ gate can be applied simultaneously. Therefore, the depth of the stage is $3n-2$. In the third stage, the number of CNOT gates is $n-2$ and the number of Toffoli gates is $n-2$. Therefore, the depth and size of the stage are $2n-4$. In the fourth stage, the number of NOT gates is 2, the number of Toffoli gates is $n-1$, and the number of CNOT gates is n. Therefore, the size of the stage is $2n+1$. Note that the first NOT gate can be applied in the second stage and that the last n CNOT gates can be applied simultaneously. Therefore, the depth of the stage is $n+1$. The depth and size of the whole circuit are $8n-7$ and $10n-9$, respectively. The numbers of CNOT, Toffoli, and NOT gates are $6n-6$, $4n-5$, and 2, respectively.

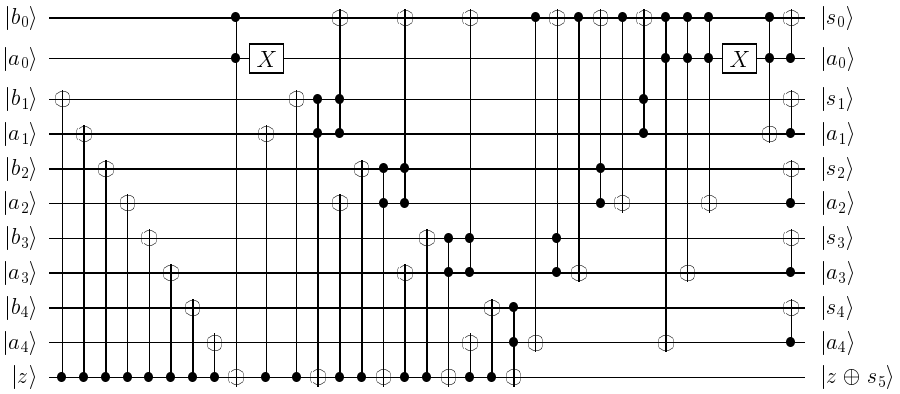

Fig. 5. The circuit for $n = 5$.

3 Addition Modulo 2^n, Subtraction, and Comparison

Quantum circuits for addition modulo 2^n, subtraction, and comparison that use no ancillary qubits can be constructed by modifying the circuit for addition in the previous section. For two n-bit binary numbers a, b, the circuit for addition modulo 2^n outputs a and s_i for $0 \leq i \leq n-1$, where $s_{n-1} \cdots s_0$ is the binary representation for $a+b \bmod 2^n$. That is, we do not compute the high bit of $a+b$. To construct a circuit for addition modulo 2^n, first, we construct a circuit for addition basically using the idea in the previous section except that we regard the memory location Z as A_{n-1}. The circuit for $n = 5$ is depicted in Fig. 6. To obtain the circuit for addition modulo 2^n, we remove two CNOT and one Toffoli gates that are applied to a tuple of memory locations including Z. For $n \geq 3$, the depth and size of the circuit are $8n-10$ and $10n-12$, respectively. The circuit for $n = 5$ is depicted in Fig. 7.

For two n-bit binary numbers a, b and some value z, the circuit for subtraction outputs a and s_i for $0 \leq i \leq n-1$ and $z \oplus s_n$, where $s_{n-1} \cdots s_0$ is the binary representation for $a-b$ and

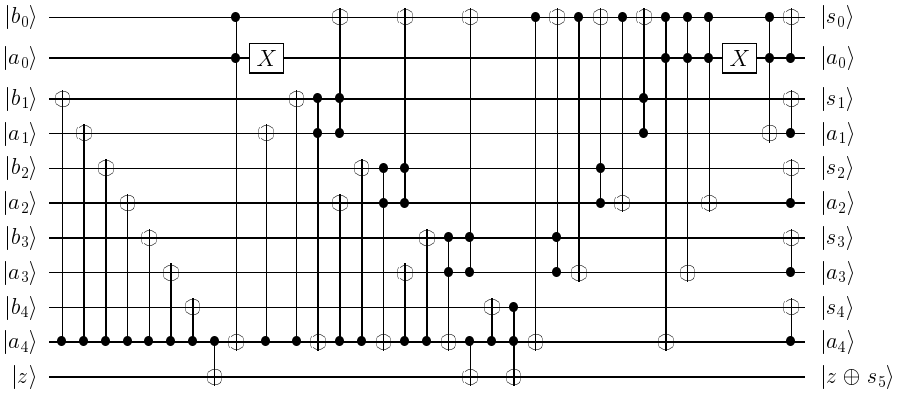

Fig. 6. The circuit for addition constructed by modifying the circuit in Fig. 5.

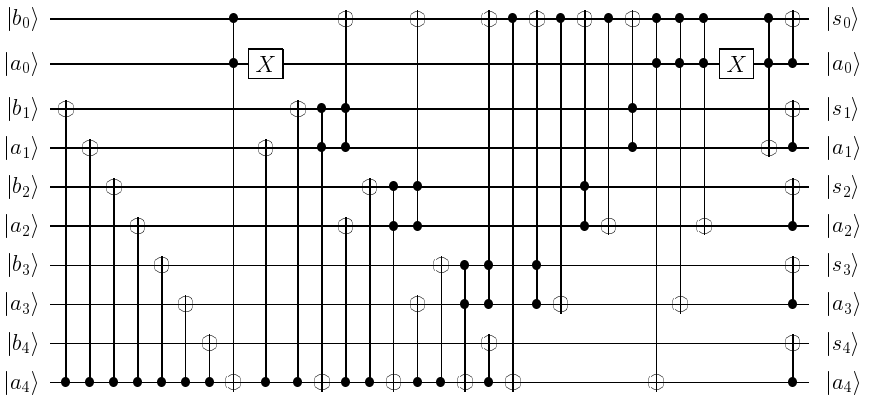

Fig. 7. The circuit for addition modulo 2^n for $n = 5$.

s_n is 1 if $b \leq a$ and 0 otherwise. We use two's complement arithmetic to represent negative numbers. That is, a negative number $-r$ is represented as $r' + 1$, where $r > 0$ and r' is the bitwise complement of r. It is known that

$$a - b = (a' + b)'.$$

Therefore, we construct a circuit for computing $(a' + b)'$. We apply NOT gates to A_i for $0 \leq i \leq n-1$ and then apply the circuit for addition in the previous section. Lastly, we apply NOT gates to all memory locations. For $n \geq 3$, the depth and size of the circuit are $8n - 5$ and $13n - 8$, respectively. The circuit for $n = 5$ is depicted in Fig. 8.

For two n-bit binary numbers a, b and some value z, the circuit for comparison outputs $a, b, z \oplus y$, where y is 1 if $b \leq a$ and 0 otherwise. The circuit is constructed by modifying the circuit for subtraction slightly. That is, we do not need to compute s_i for $0 \leq i \leq n-1$. We add CNOT and Toffoli gates to the circuit for subtraction as follows.

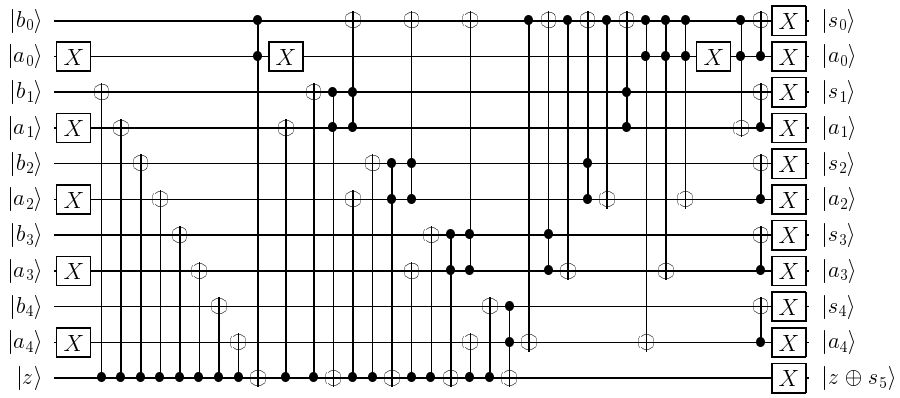

Fig. 8. The circuit for subtraction for $n = 5$.

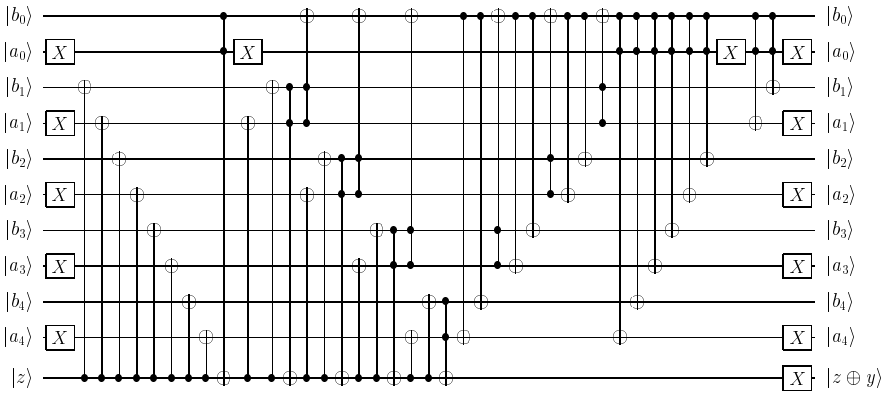

Fig. 9. The circuit for comparison for $n = 5$.

- Apply a CNOT gate to a pair of memory locations B_0, B_{n-i} for $1 \leq i \leq n-2$ in the third stage.
- Apply a Toffoli gate to a tuple of memory locations B_0, A_0, B_{n-i} for $1 \leq i \leq n-1$ in the fourth stage.

Moreover, we remove CNOT and NOT gates from the circuit for subtraction as follows.

- Remove CNOT gates applied to a pair of memory locations A_i and B_i for $0 \leq i \leq n-1$ in the fourth stage.
- Remove NOT gates applied to B_i for $0 \leq i \leq n-1$ in the fourth stage.

For $n \geq 3$, the depth and size of the circuit are $10n-9$ and $13n-11$, respectively. The circuit for $n = 5$ is depicted in Fig. 9.

4 Conclusions and Future Work

We constructed a quantum circuit for addition that uses no ancillary qubits. The circuit is based on the ripple-carry approach and the depth and size of the circuit are linear in the length of the input. We also constructed quantum circuits for addition modulo 2^n, subtraction, and comparison that use no ancillary qubits by modifying the circuit for addition.

Using the circuit for addition, we can reduce the number of qubits that are used in the circuit for Shor's algorithm as in [3]. However, the number of qubits is slightly larger than that in [3] since our circuit for addition cannot reduce the number of qubits when we add a classical bit string to a quantum state. An interesting challenge would be to construct a linear-size quantum circuit for addition that uses no ancillary qubits and can reduce the number of qubits when we add a classical bit string to a quantum state. Such a circuit is useful for reducing the size of the circuit for Shor's algorithm and the number of qubits needed.

Our circuit for addition consists of CNOT, Toffoli, and NOT gates. Toffoli gates can be constructed using 5 controlled rotation gates or using 6 CNOT and 8 single-qubit gates. Moreover, gates that are congruent to Toffoli modulo phase shifts can be constructed efficiently [7, 8, 9]. It is interesting to investigate how our circuit for addition can be compressed when we use these gates in place of Toffoli gates.

Though the depth of our circuit for addition is linear, the depth is larger than that in [1]. The depth of our circuit is $8n - 7$ and the depth of the circuit in [1] is $2n + 4$. It would be interesting to try to construct a quantum circuit for addition that uses no ancillary qubits and where the depth of which is nearly equal to $2n + 4$. Moreover, it would also be interesting to try to construct a logarithmic-depth quantum circuit for addition. Can we construct a logarithmic-depth quantum circuit for addition that uses no ancillary qubits?

Acknowledgements

The authors thank Dr. Yasuhito Kawano, Dr. Seiichiro Tani and Dr. Go Kato for their helpful comments.

References

1. S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton (2005), *A new quantum ripple-carry addition circuit*, The Eighth Workshop on Quantum Information Processing. Also on quant-ph/0410184.
2. P. W. Shor (1994), *Algorithms for quantum computation: discrete logarithms and factoring*, Proc. 35th Annual IEEE Symposium on Foundations of Computer Science, pp. 124–134.
3. S. Beauregard (2003), *Circuit for Shor's algorithm using* $2n + 3$ *qubits*, Quantum Information and Computation, Vol. 3 No. 2, pp. 175–185.
4. V. Vedral, A. Barenco, and A. Ekert (1996), *Quantum networks for elementary arithmetic operations*, Physical Review A, Vol. 54 No. 1, pp. 147–153.
5. T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore (2004), *A logarithmic-depth quantum carry-lookahead adder*, Proc. ERATO Conference on Quantum Information Science, pp. 23–24.
6. T. G. Draper (2000), *Addition on a quantum computer*, quant-ph/0008033.
7. M. A. Nielsen and I. L. Chuang (2000), *Quantum Computation and Quantum Information*, Cambridge University Press.
8. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter (1995), *Elementary gates for quantum computation*, Physical Review A, Vol. 52 No. 5, pp. 3457–3467.

9. D. P. DiVincenzo (1998), *Quantum gates and circuits*, Proc. the Royal Society of London A, Vol. 454 No. 1969, pp. 261–276.