Discovery

Efficient quantum computation with a reversible logic circuit of Excess-3 Adder

Jeong Ryeol Choi

For a classical computer, the elements of computing circuits release heat according to Landauer's principle whenever they loses an information during a computing process. However, quantum computers do not emit such heat because there is no information loss in quantum computing due to the reversibility of its processes. Owing to this, the research for the reversible quantum computation has attracted considerable interest up until now. We suggest an improved reversible circuit of decimal Adder in quantum computation, which operates with Excess-3 code. It is shown that this circuit works well, while it has the fewest garbage operation lines among the circuits ever designed with the same purpose. Our circuit is composed of 14 operation lines; among them, four operation lines are garbage lines. We can carry out efficient arithmetic operations of addition with decimal numbers using this circuit.

INTRODUCTION

In recent days, the technique for quantum computation has been developed step by step in parallel to the advance of the quantum information theory. Elementary quantum computers have been realized by multinational corporations that invest in this field, such as Google, Intel, IBM, and Microsoft. The ability of quantum computers is much superior then that of the classical computers in particular domains of information process and/or arithmetic operation. While existing computers operate according to classical mechanics that has been mainly developed by Newton in early days, the basic theory underlying the operation of quantum computers is quantum mechanics which is important in modern physics.

If the size of electronic circuits' elements becomes comparable to an atomic size, significant quantum effects take place [1]. In such situations, classical mechanics cannot be applicable in executing and analyzing the computation processes. The core properties of quantum mechanics, which are crucial in manipulating quantum information, are superposition and entanglement.

The computing and information processes in classical computers are irreversible because the output informations are typically less than the input ones. Landauer's principle says that such information loss in a computing instrument accompanies by the release of a certain amount of heat. Hence, the classical computers emit heat whenever they lose information [2]. If a bit of information erases during a computing or data process, the amount of emitting heat is $kT\ln 2$ where k is the Boltzmann constant and T is the absolute temperature at which the computing operation takes place. If a computation process of a circuit for classical computing is complicated, the circuit may release more heat. The

circuits in quantum computation however dissipate no such heat because their operating processes are totally reversible, leading no information loss from the system. Thanks to this, quantum computers consume low electric power during they operate. This is a merit of using quantum computing algorithm in computers.

The operation of computing can be carried out using a specific code. Numerical values in a computer are represented by means of such codes. For the availability of data process in a computer, the input decimal numbers which we use ordinarily should be converted to certain codewords that the given computer adopts. Binary code is the most basic one among diverse codes because its representation is very simple. However, it is known that there is no one-to-one correspondence between decimal and binary numbers. Due to this, errors take place during the conversion of the decimal numbers to the binary ones in a computer or vice versa [3,4]. Hence, for the error-free computing processes, computers require a kind of decimal codes rather than the binary one. A well-known decimal code in computing systems is binary-coded decimal (BCD). In the BCD code, decimal numbers are encoded to binary ones so that they can be available to the computing equipments. Usually, each decimal digit of BCD code is expressed by four bits. For more details of BCD code and its relation with other codes, refer to Ref. [5]. Excess-3 code is another useful decimal one. If we add 0011 in a specific BCD code, it becomes the corresponding codeword of Excess-3 code. Notice that such code conversion or the conversion between other classes of codewords can be easily carried out by specific code-converters [6,7]. First application of Excess-3 code in reversible quantum computation has been shown in Ref. [8] as far as we know.

In this work, we will suggest a reversible circuit of decimal Adder that can fulfill quantum computation with Excess-3 code and will demonstrate that it works well. Through this circuit design, we have tried the simplification of computational circuit in order to make its operation simple. If we think that decoherence can be taken place during

Professor, Department of Physics, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon, Gyeonggi-do 16227, Republic of Korea
Corresponding author: Department of Physics, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon, Gyeonggi-do 16227, Republic of Korea, Republic of Korea, e-mail: choiardor@hanmail.net
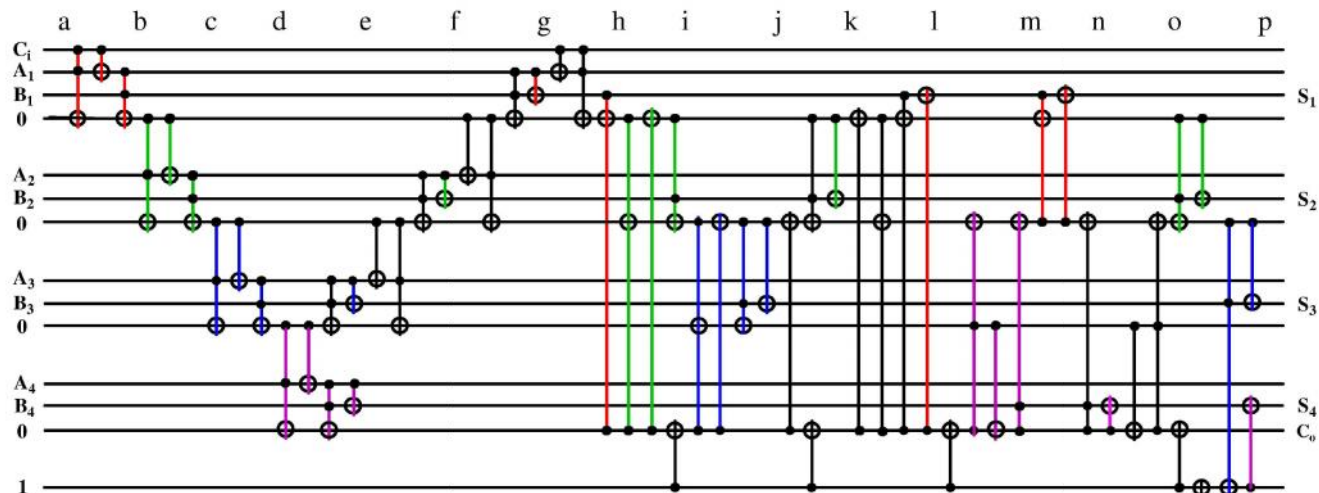
Figure 1 Reversible decimal Adder that operates with Excess-3 code

Table 1 Detailed evolution of the state of each qubit line in Fig. 1 during the computational process for the case introduced in Example 1

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	
C_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B_1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	S_1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	S_2
0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	
A_3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_3	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	S_3
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
A_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	S_4
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	C_o
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	

Table 2 The same as table 1, but for the case introduced in Example 2

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	
C_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B_1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	S_1
0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	
A_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B_2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	S_2
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	
A_3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_3	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	S_3
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
A_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	S_4
0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	0	1	C_o
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	

discovery

a quantum computation especially when the computing process is complicated [9], such simplification is important. Hence, the devise of simplified computing circuits without degradation of their function is necessary for the advancement of quantum computers. The realization of high-ability quantum computers is expected through such progress, which are competitive or surpass to classical ones.

LOGIC CIRCUIT FOR REVERSIBLE EXCESS-3 ADDER

If a quantum computing circuit is complicated, it is apt to occur decoherence during the execution of computation by the elements of a logic circuit. The system then reduces to a classical-like one and, as a consequence, the quantum computer can no longer process the computation algorithm. Hence, we should bear in mind, when we design quantum computing circuits, that computing operation and their related circuits should be as simple as possible. For this reason, the simplification of quantum logic circuits by minimizing the number of garbage outputs as well as reducing quantum costs is important in the development of quantum computers [10]. In this work, we design a reversible Adder in the domain of quantum computation using Excess-3 code. Because input qubits become output qubits as a whole at a later time in a reversible computation, there is no information loss during the progress of quantum computation. This is the reason why the quantum computing circuits based on reversibility are favorable compared to classical ones.

We propose a circuit of decimal Adder that fulfills quantum computing with Excess-3 code as shown in Fig. 1. This circuit is designed in the spirit of reducing garbage operation lines in order to make the circuit simple. The flowchart for adding process with Excess-3 code, which is helpful for understanding the data process in this circuit, is given in Ref. [5]. As you can see from Fig. 1, our circuit is composed of 14 operation lines. The first line among them is for the carry that flows into this circuit from the previous computational step, whereas the 13th line is used for output carry. The computation process is given by $(A_4A_3A_2A_1) + (B_4B_3B_2B_1) +$ input carry$[C_i] = (S_4S_3S_2S_1) +$ output carry$[C_o]$.

In this circuit, we have eliminated a garbage operation line from the previously designed Excess-3 Adder that was reported in Ref. [11]. From this removal, the computing circuit became much simple. By comparing Fig. 1 given above with Fig. 1 in Ref. [11], one can easily confirm that the fourth garbage operation line in the circuit of Ref. [11] has been eliminated when we design the circuit in this work. The computing procedures that are allocated to that removed line have been undertaken by another operation line in the present circuit, which is the last line in Fig. 1. The number of garbage lines in Fig. 1 is four, which is smallest among those of the circuits reported with the same purpose so far; as you can see, the first three garbage lines have the initial value 0, whereas the initial value of the last garbage line is 1. In the next section, we will check whether computing processes in this circuit produce exact results.

ACCURACY TESTS OF THE COMPUTING RESULTS

We will examine correctness of arithmetic results of the circuit given in Fig, 1. For this purpose, we will check the computation process for two examples of addition.

● **Example 1**: We first consider a simple arithmetic operation which is 3+5=8 (in Excess-3 code: 0110 + 1000 = 1011). This procedure do not produces output carry.

● **Example 2**: For the purpose of clarity, it may be necessary to check the computing process more generally, by considering the case that the computational result produces an output carry. For this purpose, we regard the case 9+7=16 (1100 + 1010 = 0100 1001).

Let us suppose that there is no input carry in the circuit for the above both examples. This means that $C_i = 0$.

The detailed evolutions of each qubit value for Examples 1 and 2 are given in tables 1 and 2, respectively. By inspecting the tables directly, we can confirm that the computing result of the Adder for Example 1 is $(S_4S_3S_2S_1) = (1011)$, whereas that for Example 2 is $(S_4S_3S_2S_1) = (1001)$ with an output carry. By comparing these results with the exact ones represented above, we see that the newly designed decimal Adder gives correct results.

CONCLUSION

Arithmetic processes in the logic circuits of a quantum computer are reversible in general, while those of a classical computer are non-reversible. A reversible logic circuit for a decimal Adder which operates based on Excess-3 code has been designed in this work. By taking advantage of the reversibility of our designed circuit, it may be possible to establish low power consuming computational systems [12].

The quantum resources in quantum computers are considerably vulnerable from the influence of their environment. The quantum coherence should be maintained throughout the computing process. However, it can be easily destroyed by the effects of various external interventions. As a consequence, during a computational operation through the principle of quantum mechanics, the qubit informations easily collapse to the classical ones. This phenomenon suppresses interferences between the system and its surroundings, leading to decoherence of the system [9]. Decoherence takes place through the interaction of a quantum system with diverse perturbations such as external noise and thermal radiation. Notice that such decoherence phenomena are main obstacles in realizing highly developed quantum computers which can be commercially used.

It may be impossible to remove overall decoherence under practical circumstances due to the inevitable coupling of qubits to the external environment. However, the efforts for protecting quantum computing systems from decoherence are continuously carried out by researchers in this field [13]. The appearance of such decoherence can be reduced by minimizing the running time of the computation process through the simplification of the circuits [14]. This is the reason why the simplified logic circuits that operate with a simple computing process are important in the quantum computation. By comparing our circuit represented in Fig. 1 with the previously designed ones, we can confirm that our circuit in this work is the most simple among the circuits ever designed with the same purpose.

The most recently devised previous arithmetic logic circuit for decimal Adder operating with Excess-3 code is given in Ref. [11]. This circuit has 15 operation lines. Among them, five lines are garbage lines. One of garage lines in the circuit proposed in Ref. [11] has been removed in our present work. Hence, our circuit has 14 operation lines. Among them, four lines are garbage lines.

We have proved that this newly designed circuit works well by examining its arithmetic operation with the two examples. If we use this circuit in the quantum computation, the computing process can be much simple, leading to having a positive influence on preserving quantum informations from their destruction via decoherence.

REFERENCES

1. J. R. Choi, Squeezing effects applied in nonclassical superposition states for quantum nanoelectronic circuits. *Nano Convergence*, vol. 4, 17 (2017).
2. R. Landauer, Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.*, vol. 5, no. 3, 183-191 (1961).
3. H. Thapliyal, H. R. Arabnia, R. Bajpai, K. K. Sharma, Partial reversible gates(PRG) for reversible BCD arithmetic. Proceedings of the 2007 International Conference on Computer Design(CDES'07), Las Vegas, USA, June 2007, pp. 90-91 (CSREA Press).
4. H. Thapliyal, S. Kotiyal, M. B. Srinivas, Novel BCD adders and their reversible logic implementation for IEEE 754r format, Proceedings of the 19th IEEE/ACM International Conference on VLSI Design (VLSI Design 2006), Hyderabad, India, 2006, pp. 387-392. IEEE Computer Society Press.
5. J. R. Choi, Optimal logic circuit design for reversible quantum computation based on Excess-3 code. *Discovery*, vol. 52, no. 246, 1483-1493 (2016).
6. P. K. Rao, P Raveendra, K. V. Rao, Implementation of effective code converters using reversible logic gates. *Int. J. Engineer. Res. Appl.*, vol. 6, no. 5, 17-21 (2016).
7. A. Rahman, Md. A. Habib, A. N. Bahar, Z. Rahman, Novel design of BCD to Excess-3 code converter in quantum dots cellular automata (QCA). Global J. Res. Eng. F: Electr. Electron. Eng., vol. 14, no. 4, 7-11 (2014).
8. K. H. Yeon, J. R. Choi, D. Kim, M.-S. Kim, M. Maamache, Reversible quantum computation using Excess-3 code. *AIP Conf. Proc.*, vol. 1444, 310-313 (2012).
9. A. Fedorov, L. Fedichkin, V. Privman, Evaluation of decoherence for quantum control and computing. *J. Comp. Theor. Nanosci.,* vol. 1, no. 2, 132-143 (2004).
10. M. Mohammadi, M. Eshghi, M. Haghparast, A. Bahrololoom, Design and optimization of reversible BCD Adder/Subtractor circuit for quantum and nanotechnology based systems. *World Appl. Sci. J.*, vol. 4, no. 6, 787-792 (2008).
11. J. R. Choi, J. N. Song, Design of a simplified reversible decimal adder in quantum computing based on excess-3 code. *Discovery*, vol. 54, no. 270, 231-234 (2018).
12. J. Ikonen, J. Salmilehto, M. Möttönen, Energy-efficient quantum computing. *NPJ Quantum Inf.*, vol. 3, 17 (2017).
13. L.-M. Duan, G.-C. Guo, Reducing decoherence in quantum-computer memory with all quantum bits coupling to the same environment. *Phys. Rev. A*, vol. 57, no. 2, 737-741 (1998).
14. D. Rotta, F. Sebastiano, E. Charbon, E. Prati, Quantum information density scaling and qubit operation time constraints of CMOS silicon-based quantum computer architectures. *NPJ Quantum Inf.*, vol. 3, 26 (2017).