Quantum-inspired on-chip parallel processors

Minoru Fujishima*
Dept. of Frontier Informatics, Univ. of Tokyo, 5-1-5-703 Kashiwanoha, Kashiwa JAPAN 277-8561

ABSTRACT

A quantum computer has been the focus of considerable attention since it realizes a much higher operating speed than conventional computers. However, there are many important issues regarding algorithms as well as hardware in the realization of a quantum computer. To realize the calculation speed of a quantum computer, several quantum-computing emulators utilizing parallel operation using integrated circuits were fabricated. Consequently, an emulator of 75 quantum bits was realized, and the feasibility of a high-speed quantum-computing emulator using integrated circuits has been proved.

Keywords: quantum computing, LSI, emulator, parallel processing, quantum algorithm, FPGA

1. INTRODUCTION

1.1 RESEARCH BACKGROUND

A quantum computer is a dream computer in which large-scale parallel operation is performed in an instant using quantum mechanics, particularly quantum superposition. It has attracted attention because it compromises the safety of secret codes since Shor's algorithm[1], which solves factorization at high speed, was proposed. It is believed that a quantum computer solves difficult problems, besides factorization, which cannot be solved by conventional computers at high speed. Nevertheless, it is far from realization in practice because various aspects of quantum computers remain to be studied. On the other hand, the performance of microprocessors using large-scale integrated circuits (LSIs) has improved rapidly with improvements in integration. As a result, LSIs have the potential to realize a quantum-computing emulator with a calculation capability equal to that of a quantum computer. The emulator is not only easy to realize compared with a real quantum computer, but may raise the operation capability of a microprocessor remarkably. This project was started based on this concept.

1.2 RESEARCH TARGET

The research target of the quantum-computing emulator, which is called a quantum processor hereafter, is as follows.

- The quantum processor aims at solving so-called "difficult" problems, namely, NP (nondeterministic polynomial) problems such as factorization at high speed.
- Within the operation scale dealt with by the quantum processor, the quantum processor has a calculation speed higher than that of a real quantum computer. Namely, its calculation speed does not degrade exponentially similar to that observed in software emulation with increasing operation scale.
- The quantum processor solves various problems as well as factorization. This feature is quite different from that of a real quantum computer which solves only factorization in polynomial time.

When these conditions are fulfilled, a high-speed LSI quantum processor using no quantum mechanics will realize a new processor paradigm.

* fuji@k.u-tokyo.ac.jp; phone 81 4 7136-3846; fax 81 4 7136-3847; www.axcel.k.u-tokyo.ac.jp

Noise and Information in Nanoelectronics, Sensors, and Standards II, edited by Janusz M. Smulko, Yaroslav Blanter, Mark I. Dykman, Laszlo B. Kish, Proc. of SPIE Vol. 5472 (SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.545364

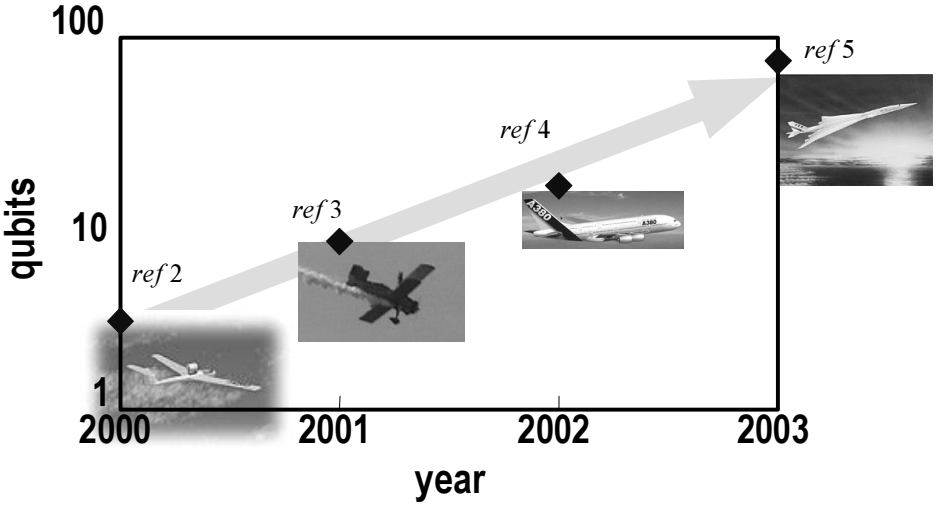


Fig. 1. The relation of the fabrication year of the quantum processor and the achieved number of the qubits.

1.3 RESEARCH PLAN

Since emulating precision and calculation speed are generally tradeoffs, it is important to review all functions used in quantum algorithms for a quantum computer to realize high-speed operation. As a result, the command unrelated to execute the algorithm should be removed, considering the minimum required functions for high-speed operation in a quantum computer. In our project, four kinds of emulators[2-5] were fabricated. The progress is shown in Fig. 1. Although the first processor fabricated in 1999[2] emulated only three qubits, the latest processor fabricated in 2003[5] emulated a quantum algorithm of 75 qubits. It was realized in accordance with research strategy not to emulate unnecessary operations to improve the calculation speed of the processor. In this paper, the difference between the first processor emulating the physical model and the latest quantum processor specialized in the minimum required functions is described. The execution result of the program by the fabricated processor is also described, considering the minimum required functions for the quantum processor.

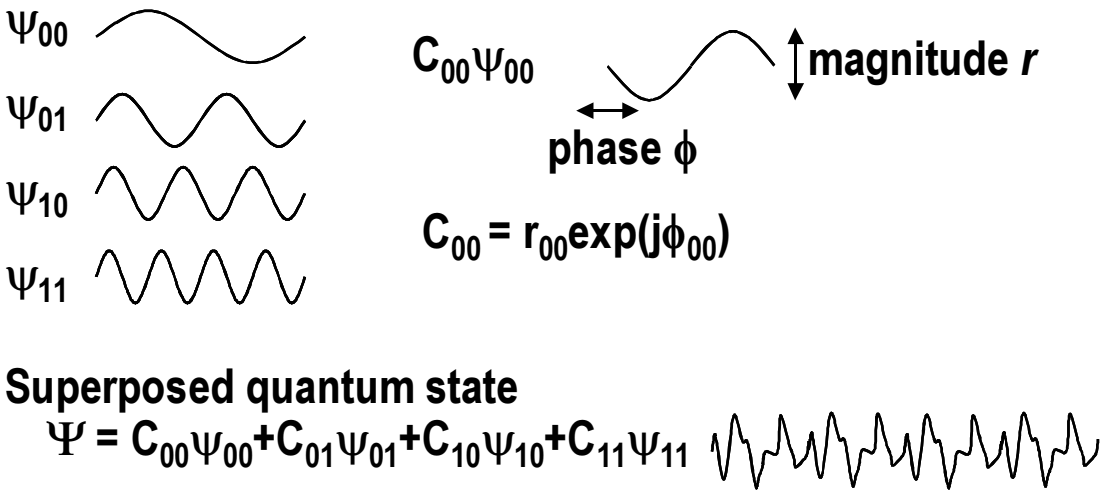


Fig. 2. The quantum state emulated by the superposition of sinusoidal waves.

2. PHYSICAL ACCURATE EMULATION - SPECTRUM COMPUTING PROCESSOR –

In conventional computers, only one number is storable in one register. On the other hand, in a quantum computer, multiple numbers are simultaneously storable in the quantum register called a qubit. Generally, when n qubits are

interacting, the quantum computer holds 2^n states simultaneously. These states are described by the linear combination of the probability amplitudes with complex numbers. The squares of their absolute values become probabilities. The probability amplitudes with complex numbers are decomposed by amplitudes and phases in polar coordinates. On the other hand, a wave is also described by amplitude and phase. Thus, the expression of the quantum states by superposing sinusoidal signals with different frequencies is proposed to emulate the quantum computer, as shown in Fig. 2.

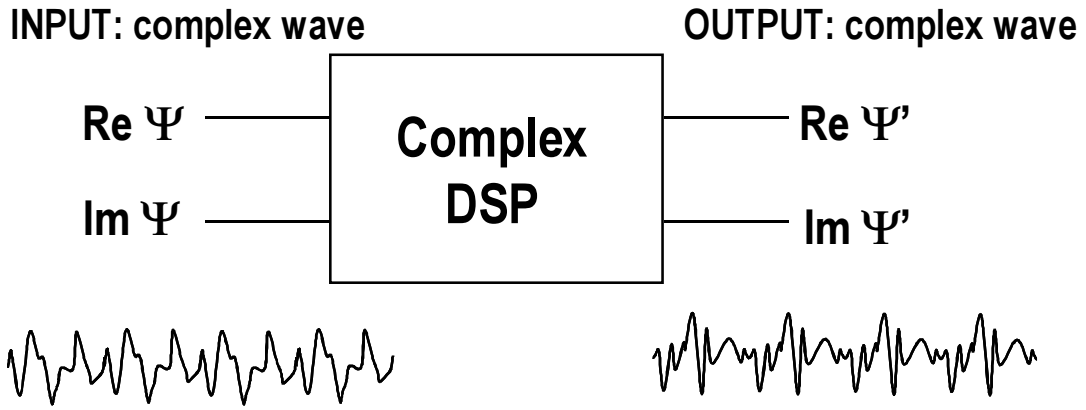
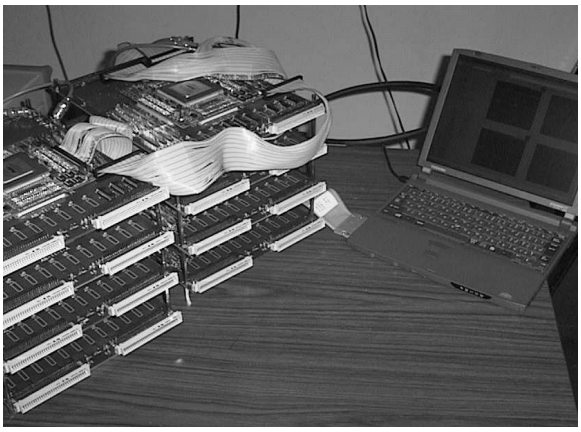


Fig. 3. Block diagram of the spectrum computer.

The sinusoidal signal of each frequency corresponds to a quantum base on a one-to-one basis. Since the quantum state made by n qubits is described by the linear combination of 2^n bases, it is also described by the superposition of sinusoidal waves with 2^n frequencies. The quantum states changing with quantum operations are emulated by changing the superposed waveforms. The emulating processor using this method is named a spectrum computer. Since probability amplitude is a complex number, the spectrum computer uses two signal lines corresponding to a real number and an imaginary number. A block diagram is shown in Fig. 3. After the two signals expressing the quantum state are given to a spectrum computer, digital signal processing of a complex number is applied, and a complex number is newly obtained at the output.



Main Specifications

PLD type	EPF10K250A-3 Altera
Number of PLD	8
Number of Qubits	3
Number of Logic Gates	1.0×10^6
Clock Frequency	5MHz

Fig. 4. Photograph of a spectrum computer system and a summary of the specifications.

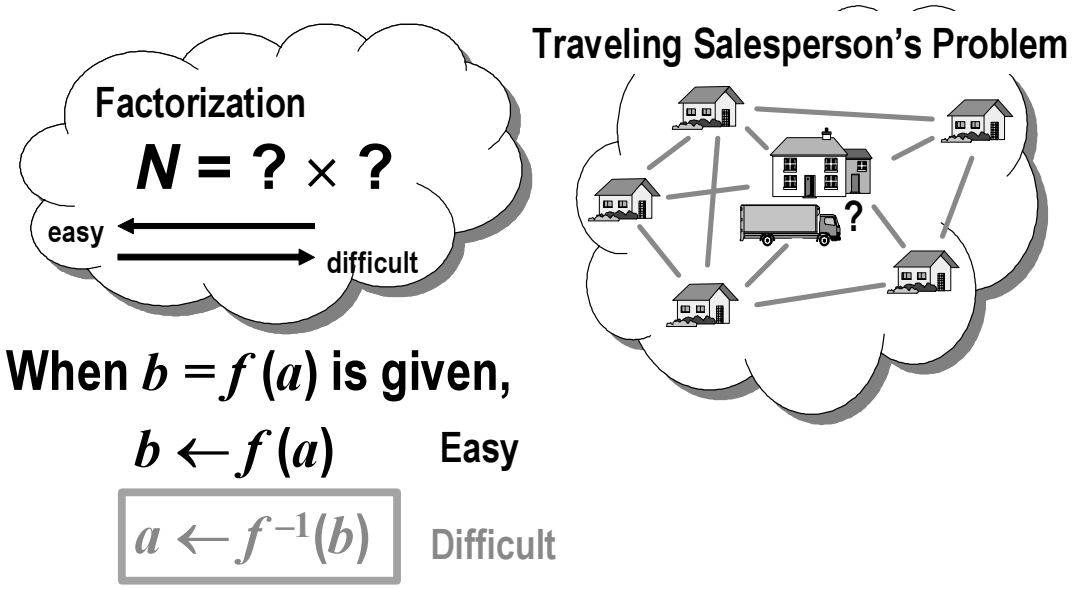


Fig. 5. NP problems.

The photograph and specifications of a spectrum computer are shown in Fig. 4. By expressing the superposition of the quantum state using sinusoidal superposition, the hardware emulation of a quantum computer was successfully realized for the first time in the world. However, digital signal processing requires a large amount of hardware with a clock frequency of only 5 MHz, which is insufficient for practical use, and only small-scale algorithms can be executed by the complex hardware since the available qubits for emulation are three. Using the quantum processor, improvements in operating speed and the problem scale to be solved are required for solving a problem like factorization.

3. HIGH-SPEED EXECUTION OF THE QUANTUM ALGORITHM – QUANTUM INDEX PROCESSOR –

3.1 GENERALITIES OF QUANTUM ALGORITHM

The spectrum computer emulates the physics of a quantum computer with an LSI. However, it is not necessary to emulate the physics of a quantum computer accurately to perform quantum algorithms. Thus, quantum algorithms were examined to realize larger-scale emulation at high speed.

General problems to be solved by a quantum computer at high speed are called NP problems. An example of these problems is shown in Fig. 5. Generally, when function $b = f(a)$ is given, it is easy to obtain output b from input a. However, it is difficult to obtain input a from output b in many cases. This type of difficult problem is called NP problem which cannot be solved in polynomial time using a conventional computer. The quantum computer is expected to solve NP problems at high speed.

In case of solving NP problem, the algorithm used with a quantum computer is as follows[6]. A quantum computer calculates the outputs corresponding to the candidate of multiple inputs in parallel simultaneously. When the conventional computer finishes $b = f(a)$ in polynomial time, a quantum computer obtains multiple b's from multiple a's in polynomial time. Next, the target output from all outputs b's are searched so that the corresponding input a may be an answer. When this search is also finished in polynomial time, NP problem is solved in polynomial time as a whole. Here, an important point is that each of the parallel operations and searches must be completed in polynomial time. As a result, a method for performing parallel operation and searches using a quantum processor is considered.

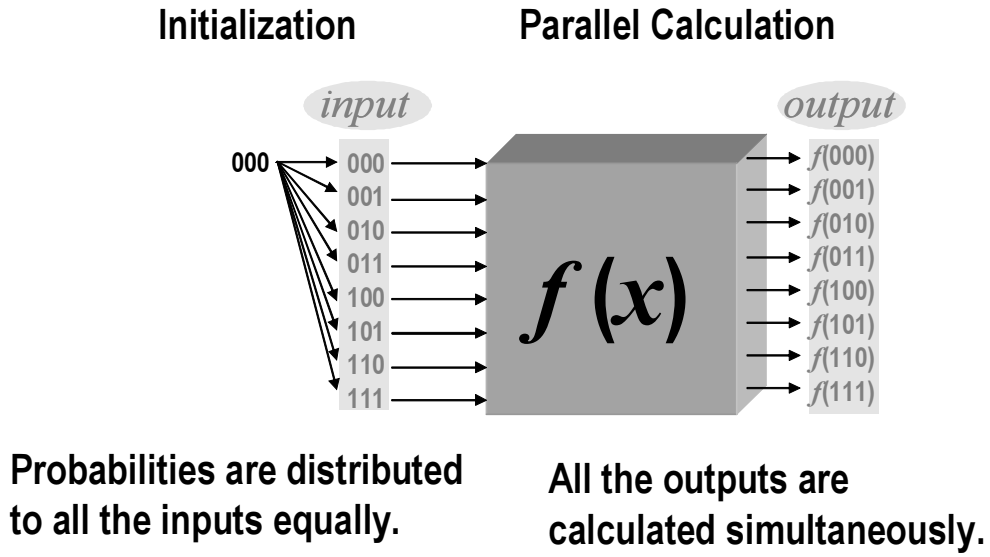


Fig. 6. Parallel calculation in a quantum computer.

3.2 PARALLEL PROCESSING

The conceptual figure of a parallel operation using a quantum computer is shown in Fig. 6. The superposed state of all inputs is generated in a quantum computer by equally distributing probabilities to all the inputs as candidates of a solution. The function $f(x)$ is calculated from the superposed inputs, and all outputs are obtained simultaneously. Namely, the inputs of a quantum computer need to be superposed to calculate multiple states simultaneously. However, there is only one output for a certain input in the function $y = f(x)$ since the input and the output correspond to each other on a one-to-one basis. Consequently, only one output is necessary for a certain input, and superposed outputs are not required. This situation is shown in Fig. 7.

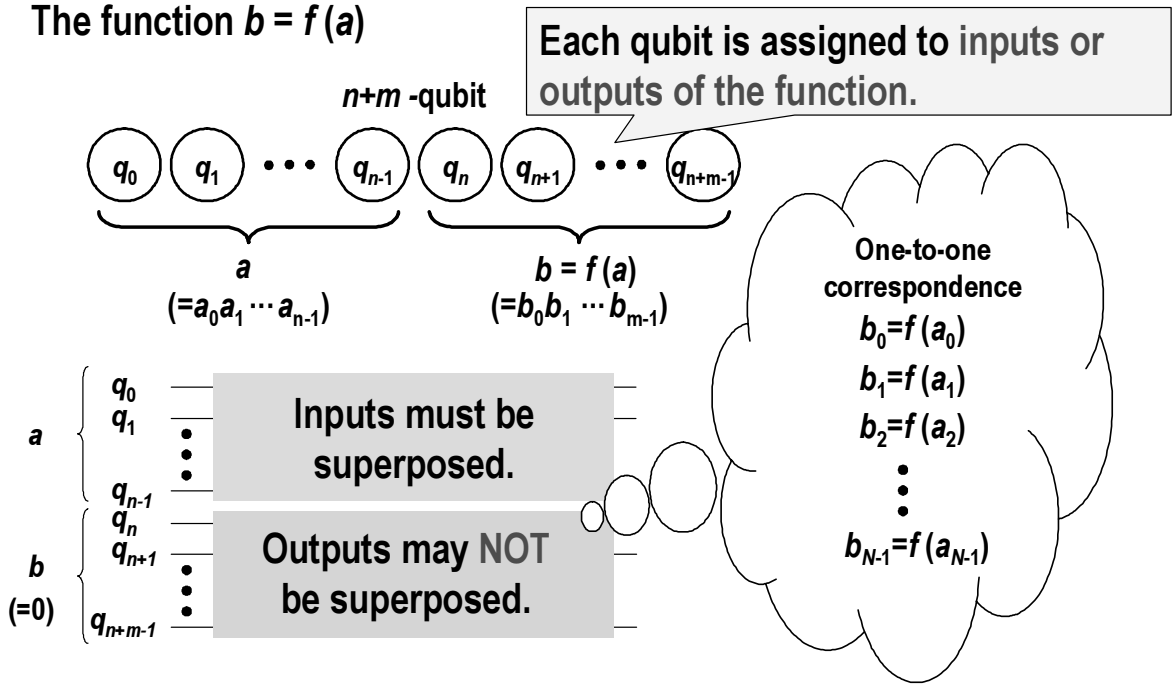


Fig. 7. A difference of the roles of the input and the output qubits in a quantum computer.

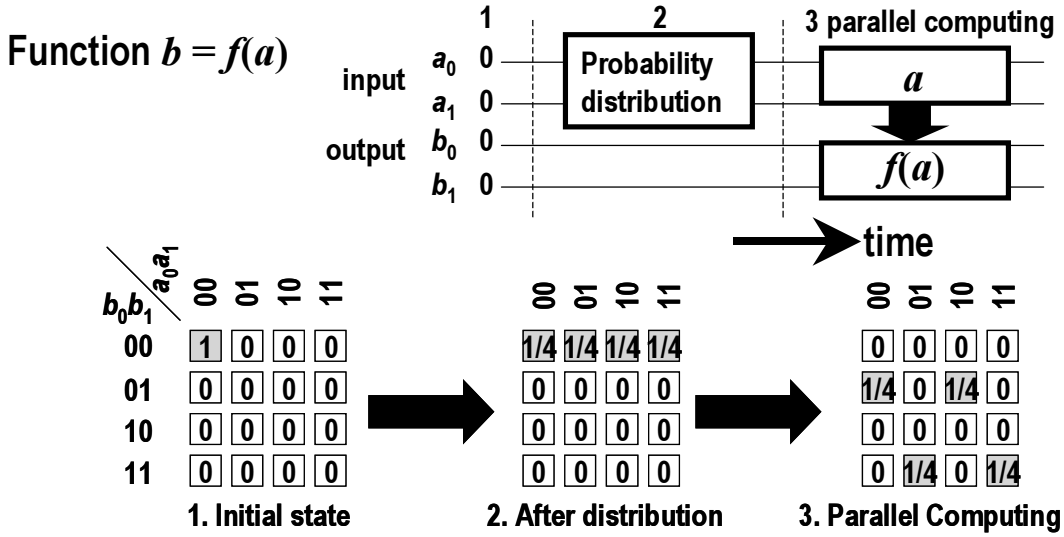


Fig. 8. Probability transitions of quantum states.

Probabilities other than one output become zero for one input since outputs are not superposed. The transitions of the probabilities of quantum states in parallel operation are shown in Fig. 8, where the values of the probabilities are unchanged but the position of the nonzero probability changes after the probabilities are equally distributed to each input in initialization. Furthermore, the one-to-one correspondence of inputs and outputs also shows that the probabilities of most quantum states are zero. From these two facts, it is found that the quantum states can be described by the positions of the quantum states where nonzero probabilities are stored, even when all the values of probabilities or probability amplitudes are not stored. In this case, the amount of memory is significantly reduced and calculation time is also improved. This processor which saves and calculates the position of the nonzero quantum state, namely the quantum index, was named the quantum index processor (QIP).

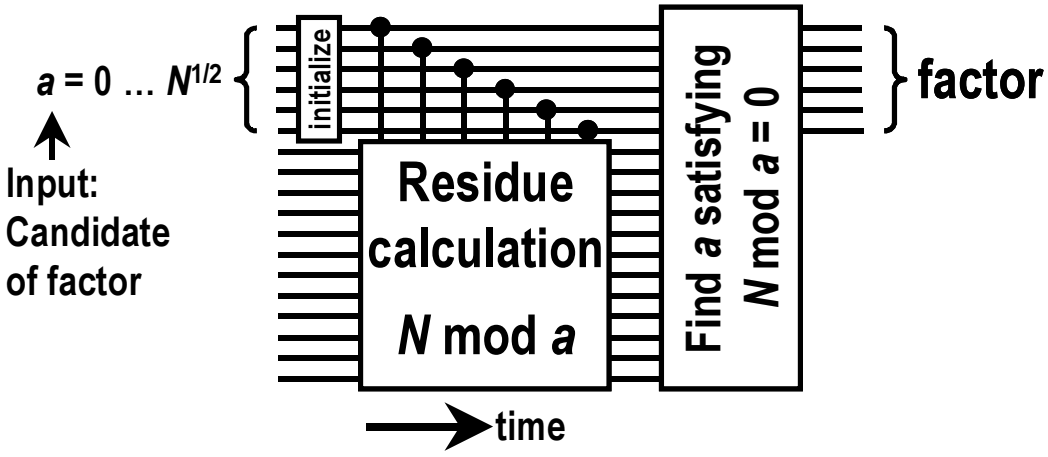


Fig. 9. Simple algorithm for factorization.

3.3 SEARCH

In a quantum computer, search is performed after parallel operation to solve NP problems. Figure 9 shows a simple application to factorization. In Fig. 9, probabilities are equally distributed among the candidates of the factors in the factorization of N. Then, residues are calculated when N is divided by all the candidates of the factors. Finally, the answer is derived when the input corresponding to the output of zero is found. On the other hand, the quantum computer finishes initialization and residue calculation in polynomial time using parallel operation. However, the quantum computer does not obtain the input corresponding to a specific output in polynomial time. In factorization, the

specific output is the residue of zero. Therefore, a quantum computer cannot finish the algorithm shown in Fig. 9 in polynomial time. Shor's algorithm shown in Fig. 10 is used to finish factorization in polynomial time using a quantum computer.

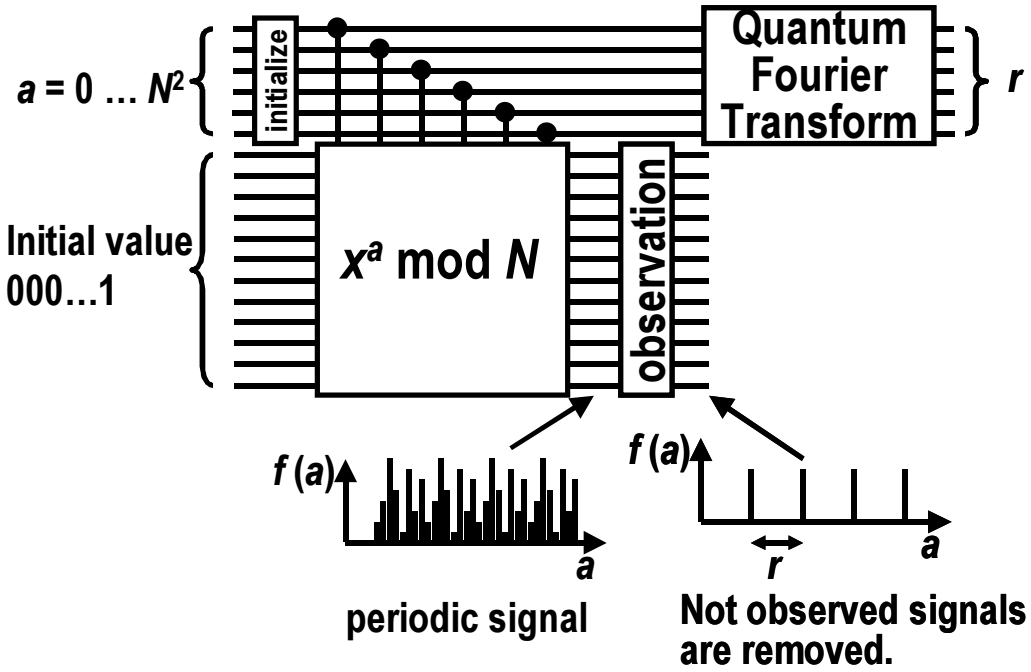


Fig. 10. Shor's algorithm.

Shor's algorithm does not calculate residues but calculates $x^a \bmod N$ to factorize, where x is an arbitrary number and a is the vector of seriate integers which is superposed. It is known that $x^a \bmod N$ is periodic, and that the factor is calculated from a cycle. Fourier transform is performed to obtain the cycle of $x^a \bmod N$ after parallel computing and observation of the output to eliminate unnecessary harmonics. The quantum computer finishes the quantum Fourier transform in polynomial time. Consequently, the quantum computer finishes factorization in polynomial time by not calculating residues directly but using a periodic function. Since the quantum Fourier transform to obtain a cycle is the only efficient search algorithm for a quantum computer, residue calculation has changed into the calculation of the periodic function. As a result, the periodic function has to be found to finish NP problem in polynomial time with a quantum computer, as used with Shor's algorithm. This will impose severe restrictions on algorithms.

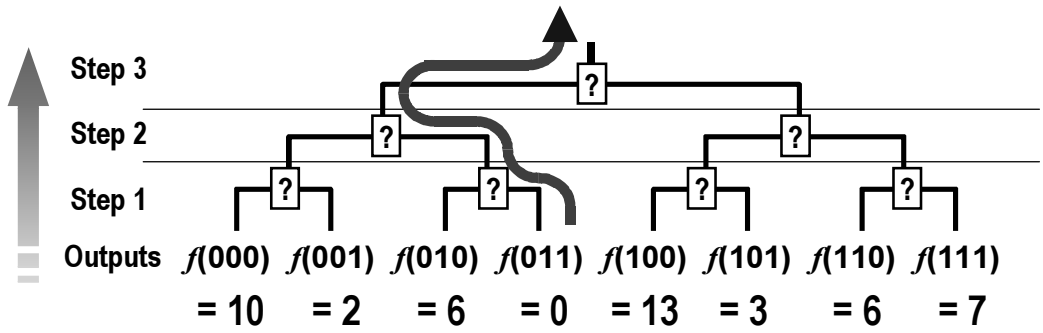


Fig. 11. Binary search.

On the other hand, in the case of LSI emulation, search can be performed efficiently even when Fourier transform is not applied. For example, the input corresponding to a specific output can be determined in polynomial time using a binary search method. The outline of binary search by LSI is summarized in Fig. 11, in the case where parallel operation has

finished. In such a search, we determine in which of two groups the solution exists. When a solution exists, the value of the input corresponding to the solution is returned. When a solution does not exist, the nonexistence flag is returned. Then, out of two groups of calculation results, the existence of a solution is investigated similarly and an output is defined. The input corresponding to the solution can be found by repeating this method in polynomial time. This operation is equivalent to "absolute observation" in a quantum computer, which observes 0 or 1 intentionally. When quantum mechanics is used, 0 or 1 is only obtained stochastically, and the specific value is not always observed regardless of the method used. However, LSI, which enables absolute observation, finishes search in polynomial time in this way.

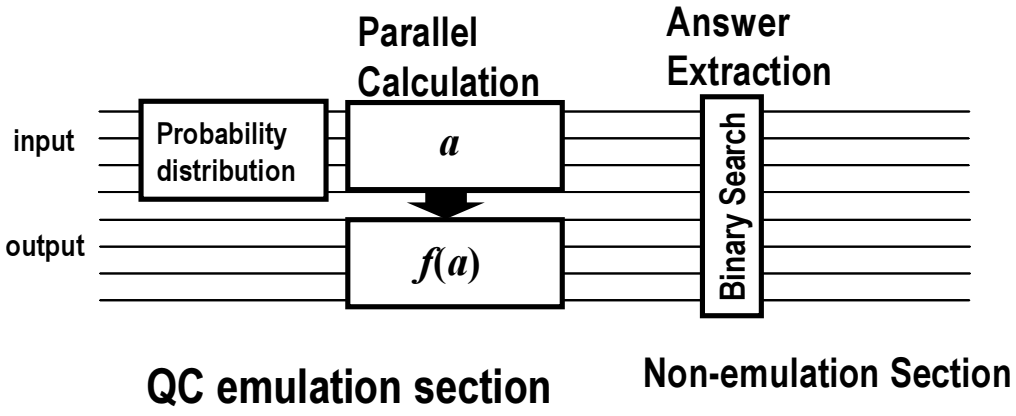


Fig. 12. Generic algorithm to solve NP problems using the quantum processor.

3.4 QUANTUM INDEX PROCESSOR

The general algorithm using the QIP is shown in Fig. 12. The QIP emulates parallel operation of a quantum computer since there are no restrictions on algorithms. However, the QIP does not emulate the search algorithm of a quantum computer and uses binary search since a quantum computer uses only periodic functions to apply Fourier transforms. The outline of the QIP is shown in Fig. 13. In the QIP, the position of the nonzero quantum state, namely quantum index, is used as an input. The quantum index of an output is obtained by performing a logic operation. This operation is executed for all the inputs in parallel. A solution is searched using binary search after parallel operations. A photograph of the fabricated QIP and a screen shot of the programming environment are shown in Fig. 14. Two FPGAs are implemented in a single board, where one stores a program to control the other. The latest processor realizes the 2,048 processing elements (PEs) in parallel, and performs parallel operations and searches. Performance comparison between the QIP and the previous version, which is third generation[4], is shown in Fig. 15. Operation time is measured by performing Shor's algorithm. It is noted that operation time shown as a dotted line is estimated since large-scale operation cannot be executed using the previous version. The QIP improves calculation speed by a factor of 10^{18} compared with the previous version by optimizing the preservation of the nonzero quantum state.

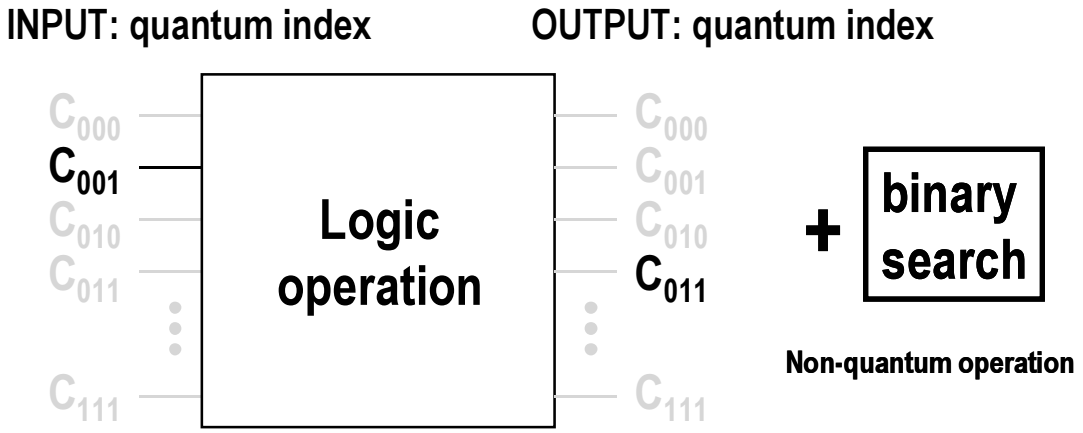


Fig. 13. Conceptual block diagram of the quantum index processor.

The QIP can calculate Shor's algorithm at high speed by calculating a periodic solution. As described before, the binary search is adopted in the QIP to overcome the restrictions of a quantum computer. Thus, even when Shor's algorithm is not used, the QIP finishes factorization at high speed. The speed of factorization not using Shor's algorithm is measured and the comparison results are shown in Fig. 16. Compared with the case where Shor's algorithm is used, the factorization speed improves by a factor of 10^4. The residue calculation can calculate at a higher speed than Shor's algorithm because its algorithm is much simpler. In this case, the quantum processor performs flexible algorithms compared with the quantum computer.

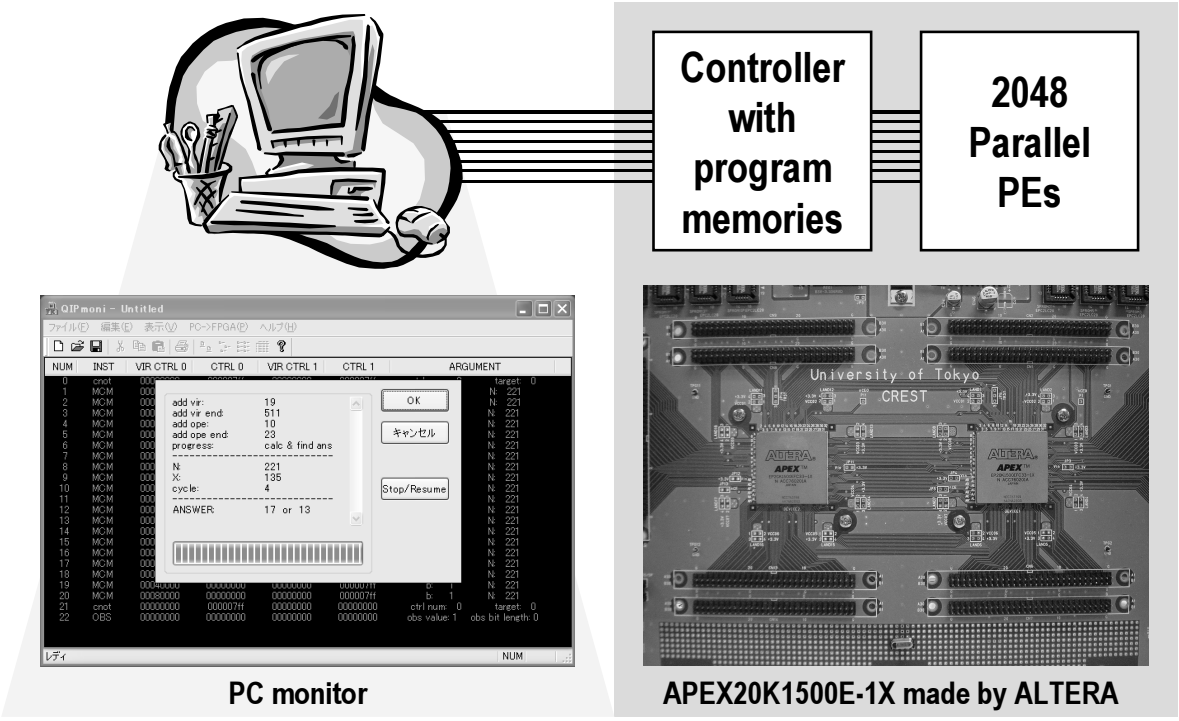


Fig. 14. Screen shot of program environment and print board implementing two FPGAs for the quantum index processor.

	Old version	QIP
Number of qubit	16	75
Frequency	40 MHz	80 MHz
Factorization of 20 bits	10^{18}sec (40 billion years)	5.19sec

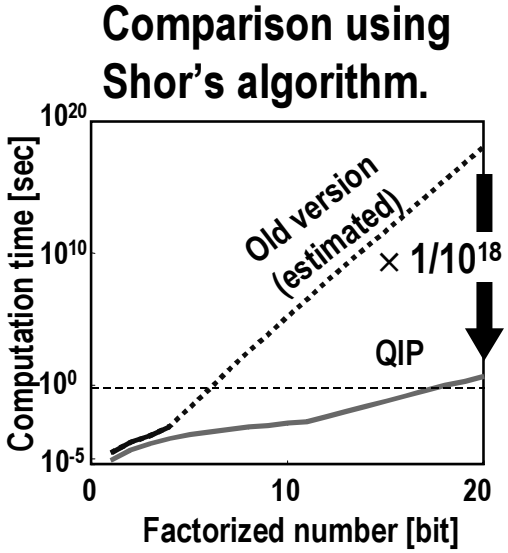


Fig. 15. Performance comparison between the QIP and third generation in the quantum processors.

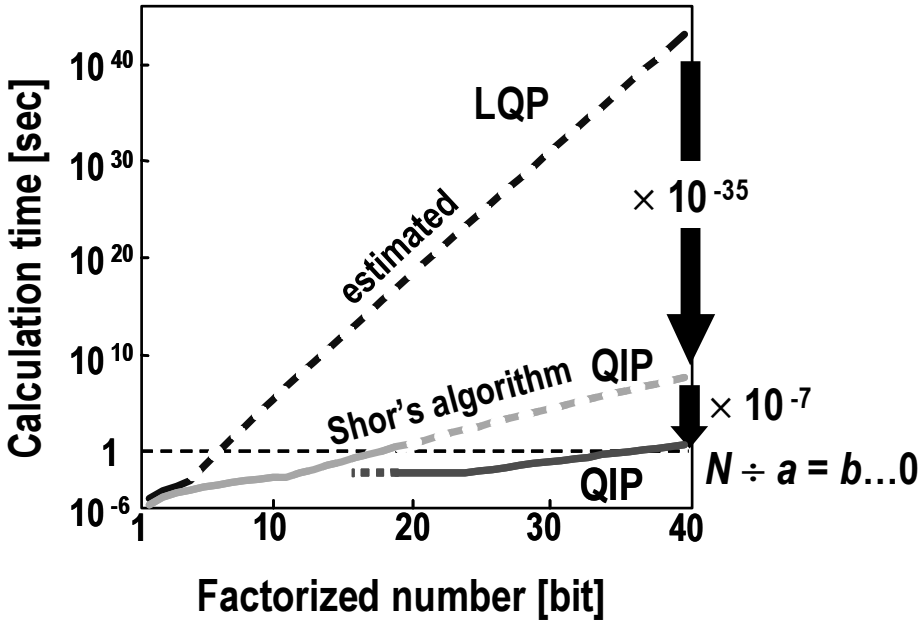


Fig. 16. Comparison of the speed of factorization when using and not using Shor's algorithm.

4. CONCLUSION AND FUTURE PROSPECTS

A quantum processor utilizing the capability of LSIs, which has a computing performance more than a quantum computer, has been studied. Consequently, a much larger scale calculation at high speed has been realized by only emulating the minimum required functions to solve NP problem, compared with the case of emulating the quantum state directly. A comparison between the progress of the number of the qubits realized by a conventional quantum processor and the progress of the number of qubits of a quantum computer is shown in Fig. 17.

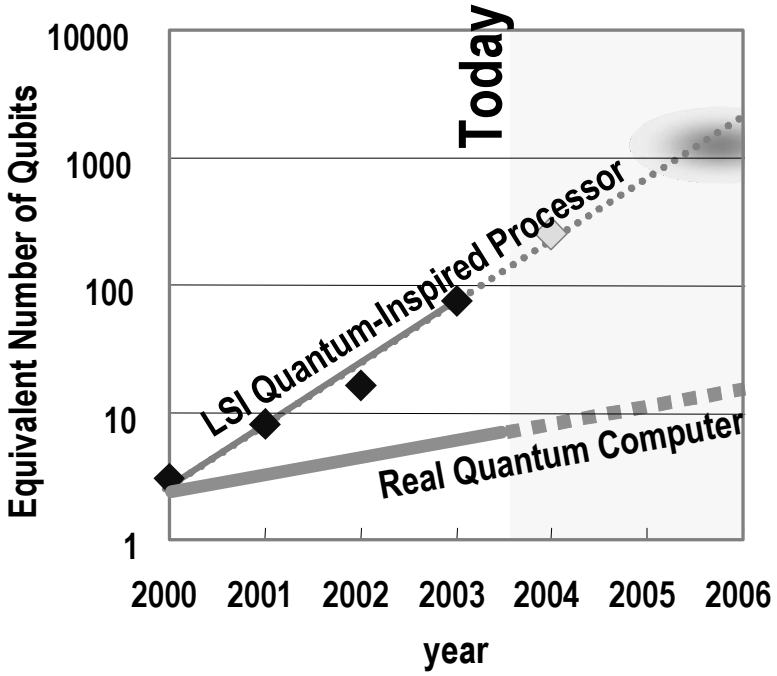


Fig. 17. Increasing qubits of the quantum processor and the quantum computer.

The number of qubits in the quantum processor is expanding by approximately three times per year. Currently, the number of qubits in a quantum processor is ten or more times larger than that of a quantum computer since the quantum computer realizes only seven qubits using nuclear magnetic resonance. It is noted that only one qubit increases every one and half years when each transistor is assigned to each quantum state in a quantum computer, since the degree of integration of LSI increases twice every one and half years. Thus, the increase in the number of the qubits in the quantum processor is mainly achieved by the evolution of algorithms. In the future, NP problems will be solved on a practical scale when the 1,000-qubit operation is attained. It will be realizable three years from now with the continuing expansion of the operation scale. Many issues have been resolved by algorithms although issues to be solved still remain. Furthermore, the quantum processor is versatile in the case of algorithms for general optimization problems since fewer search restrictions apply to the quantum processor than the quantum computer. Expansion of operation scale and flexibility will disseminate the quantum processor.

REFERENCES

1. P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Proc. 35th Annual Symposium on Foundations of Computer Science,* 124-134, 1994.
2. S. O'uchi, M. Fujishima and K. Hoh, "Emulation of quantum computing by finite impulse reponses," *Extended Abstracts of 1999 International Conference on Solid State Devices and Materials*, 96-97, 1999.
3. S. O'uchi, M. Fujishima and K. Hoh, "Fractally-structured CMOS processor for quantum-circuit emulator," *Jpn. J. Appl. Phys.* **41**, 2329-2334, 2002.
4. M. Fujishima, K. Saito and K. Hoh, "16-qubit quantum computing emulation based on high-speed hardware architecture," *Jpn. J. Appl. Phys.*, **42**, 2182-2184, 2003.
5. M. Fujishima, K. Inai, T. Kitasho and K. Hoh, "75-qubit quantum computing emulator," *Extended Abstracts of the 2003 International Conference on Solid State Devices and Materials*, 406-407, 2003.
6. M. Fujishima, "LSI-based efficient emulation overcoming algorithmic restrictions inherent in quantum computers," *Fluctuation and Noise Letters*, **3**, C9-C17, 2003.