

8-QUBIT QUANTUM-CIRCUIT PROCESSOR

Shin-ichi O'uchi^{*}, *Minoru Fujishima*^{**}, and *Koichiro Hoh*^{**,***}

^{*}School of Engineering, the University of Tokyo,
^{**}School of Frontier Sciences, the University of Tokyo,
^{***}CREST, Japan Science and Technology Corporation

ABSTRACT

Quantum-Circuit Processor (QCP) is a parallel processor that executes quantum algorithms at a processing speed comparable to a quantum computer. The QCP emulates the quantum superposition, utilizing massive silicon devices in the fractal hardware-architecture. Its design cost is relaxed due to that recursive structure although a huge amount of devices are required for it.

In this paper, 8-qubit QCP was fabricated within a programmable logic device. This is the first realization of the 8-qubit-hardware quantum computing as far as we know. A factoring algorithm is demonstrated by using it in this paper.

1. INTRODUCTION

We have proposed a method for tackling some problems intractable in the conventional computer, by putting the quantum computing into LSI, and fabricated the one chip parallel processor based on it [1]. Using parallel operation of an amount of devices available in the present LSI technology, our system executes quantum algorithms in the same processing steps as the quantum computer. Although the quantum computer is expected to solve those focusing problems efficiently, its realization is very hard due to its weakness against the disturbance from the environment in reality. The proposed system may give a solution to this circumstance. Optimizing the hardware for the quantum computing, we obtained the fractal hardware-architecture for the system. This structure also helps us to expand the scale of hardware in massive-parallel-computing system.

In our previous study, we fabricated a chip which operates 32 processing-elements (PEs) in parallel. This chip has the function equivalent to a 5-quantum-bit (qubit) quantum computer [1]. We optimized the circuit and successfully fabricated an 8-qubit chip in this study, although the NMR quantum computer has realized the 5-qubit algorithms in a quantum-mechanical substance [2] and the algorithm in the larger scale has been executed in the form of simulation on the workstation.

In the following section, we will describe the first demonstration of an 8-qubit quantum-algorithm with our hardware.

2. FUNDAMENTALS OF QUANTUM-CIRCUIT

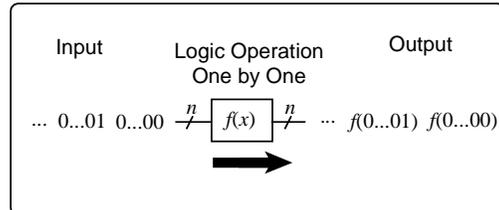
In our system, quantum computing is emulated according to the quantum circuit model. In this section, the overview of the quantum circuit is mentioned.

The quantum circuit consists of the qubits and quantum gates. The qubit is able to hold both logic 0 and 1 with an arbitrary ratio in a quantum-superposition,

$$|\psi\rangle = \omega_0|0\rangle + \omega_1|1\rangle. \quad (1)$$

Here, the probability amplitude ω_0 and ω_1 are complex numbers whose squared absolute values represent the probabilities of observing logic 0 or 1 in the qubit. Thus, they satisfy $|\omega_0|^2 + |\omega_1|^2 = 1$. The quantum gates on the

Conventional Logic Circuit



Quantum Logic Circuit

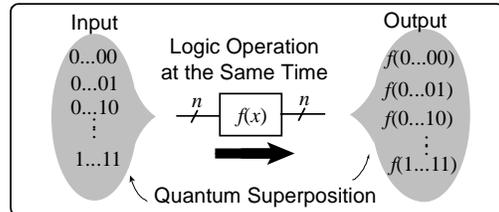


Fig. 1. Comparison of the conventional computer and the quantum computer. The quantum computer processes 2^n data at the same time while the conventional computer processes in serial. Taking this parallelism effectively, the quantum computer enables the efficient algorithm for the intractable problems in the conventional computers.

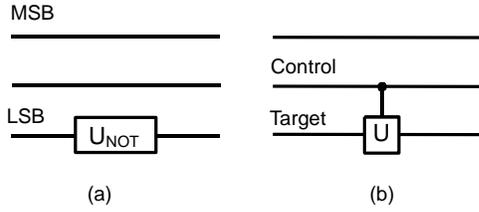


Fig. 2. Symbols of (a) the quantum NOT gate operated on a single qubit and (b) the controlled unitary gate operated on two qubits.

qubits are physical operation represented by the unitary transformations on the qubits. In the n -qubit quantum circuit, a gate operation on a single qubit processes the 2^n logic bases; all the possible bases as shown in Fig. 1. By contrast, the conventional logic circuit processes them serially. This is called quantum parallelism. For example, the operation on the least significant qubit in a three-qubit quantum circuit shown in Fig. 2(a) processes $8(=2^3)$ data in parallel. Namely, the unitary matrix

$$U_{\text{NOT}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2)$$

is operated on the state pairs $\omega_{000}-\omega_{001}$, $\omega_{010}-\omega_{011}$, $\omega_{100}-\omega_{101}$, $\omega_{110}-\omega_{111}$, as

$$\begin{bmatrix} \omega'_{000} \\ \omega'_{001} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_{000} \\ \omega_{001} \end{bmatrix} = \begin{bmatrix} \omega_{001} \\ \omega_{000} \end{bmatrix}. \quad (3)$$

In general, the arbitrary one-bit operation U processes the probabilistic amplitudes of every logic-bases pairs whose logic difference exists only in the operated qubit. The two or more-qubit logic gate, which is called controlled gate, enables the universal quantum gates, used together with the one-qubit logic gates mentioned above [3,4]. Figure 2(b) shows the symbol of the controlled gate in a tree-qubit quantum circuit for instance. This gate executes the transformation U on the target qubit (the least significant qubit in Fig. 2(b)) only when the control qubit (the second qubit in the figure) is in the state $|1\rangle$. Therefore, only two pairs $\omega_{010}-\omega_{011}$ and $\omega_{110}-\omega_{111}$ are transformed by the gate while the pair $\omega_{000}-\omega_{001}$ and $\omega_{100}-\omega_{101}$ are left as they are. After those gate operations, the output of the quantum circuit is observed in the respective qubits. As the result of the n -step observations, the most probable n -bit solution is obtained.

In the quantum-mechanical system, only n substances are needed for the 2^n parallelism. However, 2^n devices are needed to realize same 2^n parallelism as the quantum computer in the LSI because it is one of a classical system.

3. QUANTUM-CIRCUIT PROCESSOR

In the presenting system, the computation in the n -qubit quantum circuit is realized by 2^n PEs in the Single-Instruction-stream-Multiple-Data-stream (SIMD) parallel

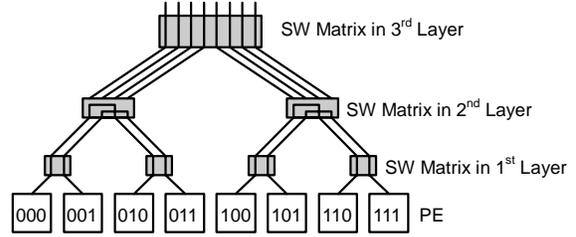


Fig. 3. Fractal-tree network for the communication between PEs. The schematic represents the connection of switch matrices in the case of the gate operation on the second qubit. The switch matrices in the second layer exchange the PE data while the other layers pass them

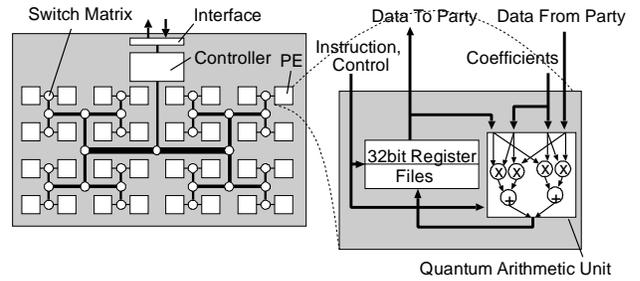


Fig. 4. Fractal structure in a 5-qubit chip for example. PEs and switch matrices are placed in the H-tree configuration. In each PE, 32-bit register-files and an arithmetic unit are implemented. The instructions, controlling signals and transformation coefficients are broadcasted from the controller to each PE.

processor, where the computations superposed in the quantum mechanical system are expanded into the massive hardware in the parallel processor. We named it the Quantum-Circuit Processor (QCP). In this section, main features of the hardware and the instructions in the QCP are described.

3.1. Fractal Hardware Structure

The quantum gate operations are emulated by the numerical operation of matrix calculations as shown in (3). This operation is executed by the pairs of PEs corresponding to the state pairs in (3). Each PE has its own address corresponding to one n -bit logic-basis, and holds the probability amplitude of that state in its own dedicated register. 2^n PEs and n -dimensional hypercube network are necessary in the QCP for the n -qubit universal computation.

We utilize a binary fractal-tree structure to realize the hypercube network in two-dimensional LSI. Figure 3 is the schematic of the fractal-tree network. A switch matrix is placed at each node of the tree, and the PEs are placed at the terminal of the tree.

The PEs and switch matrices are placed in the H-tree type fractal structure as shown in Fig. 4. Each PE consists

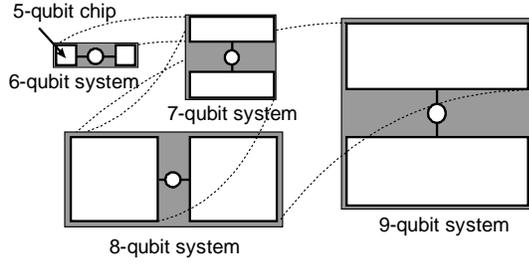


Fig. 5. Scaling-up method of the processor system. $(n+1)$ -qubit system is composed with two n -qubit systems and one switch-matrix recursively.

of 32-bit registers and the arithmetic unit. Single instruction, switch-controlling signals, and the matrix coefficients are broadcasted by controlling unit. Then, the probabilistic amplitude in each PE is calculated using the local register, the party's register and the broadcasted coefficients. The result is written back to the local register.

This system is suitable for extending into larger scale owing to the fractal structure. For example, $(n+1)$ -qubit system is recursively realized by combining two n -qubit systems and one extra switch matrix as shown in Fig. 5. This is helpful to construct massive parallel computation system effectively with the LSI.

3.2. Dedicated Instructions

The universal quantum gates in the QCP consist of six kinds of quantum-gate instructions, Rotation for the specific qubit (ROT)

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, \quad (8)$$

Phase Shift for the state $|0\rangle$ of the specific qubit (PHAS0)

$$\mathbf{P}_0(\phi_0) = \begin{bmatrix} e^{i\phi_0} & 0 \\ 0 & 1 \end{bmatrix}, \quad (9)$$

and Phase Shift also for the state $|1\rangle$ of the specific qubit (PHAS1)

$$\mathbf{P}_1(\phi_1) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi_1} \end{bmatrix}, \quad (10)$$

and their extension which have arbitrary number of controlling qubits, Controlled Rotation (CROT) and Controlled Phase Shift (CPHAS0, 1).

Observation of a single qubit is calculated in three steps for 3-qubit computation by using dedicated instructions, the Probability CALculation (PCAL) and Probability SUMmation (PSUM).

The complete instruction set including system operations, initializing and halt operations are summarized in Table I.

TABLE I
SUMMARY OF INSTRUCTIONS AND THEIR FUNCTIONS

Mnemonic	Function
PHAS0(q_t, ϕ_0)	Performs phase shift on state 0 of qubit q_t as shown in (9).
PHAS1(q_t, ϕ_1)	Performs phase shift on state 0 of q_t as shown in (10).
CPHAS0(q_c, q_t, ϕ_0)	Performs phase shift on state 0 of q_t controlled by qubit q_c .
CPHAS1(q_c, q_t, ϕ_1)	Performs phase shift on state 1 of q_t controlled by q_c .
ROT(q_t, θ)	Performs rotation on q_t as shown in (8).
CROT(q_c, q_t, θ)	Performs rotation on q_t controlled by q_c .
PCAL	Computes observation probability of respective basis.
PSUM(q_t)	Computes sum of probability-register values only q_t -th digit is different.
REDUCE(q_t, mode)	Reduces states in four modes based on the probability.
INIT	Initializes state to 000..0
HALT	Halts computation.

TABLE II
SPECIFICATION OF THE FABRICATED CHIP

Type of PLD	Altera EP20K1500EFC33-3
Qubits	8
Gates	1.3M
Power supply	3.3V
Clock speed	30MHz

4. EXPERIMENT

By downsizing the circuitry of the PE in the previous 5-qubit chip [1], we succeeded in the implementation of an 8-qubit chip in a single programmable logic device (PLD). Its specification is summarized in Table II. With this 8-qubit chip, we demonstrated Shor's algorithm for factoring the integer [5]. Shor's algorithm completes a factorization in polynomial steps of the inputted bit-length while it is thought to be hard on the conventional computers. In this demonstration, we took the integer 15 for example. To execute the factorization, the qubits are partitioned into two set, 4 qubits as register 1 and another 4 qubits as register 2. On these "quantum" registers, we perform the quantum operation as the following, which is realized by the quantum circuit as shown in Fig. 6. (i) We make a superposition of all integers a in the range of 0 to 15 in the

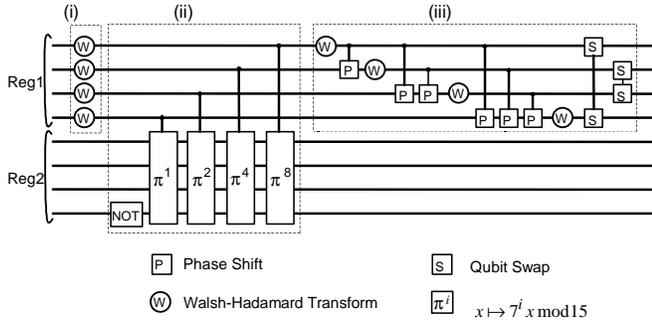


Fig. 6. Quantum circuit for factoring the integer 15 [2,6]. The upper 4 qubits represent the register 1 and the lower represent the register 2. (i) The superposition of all integers is made by the operation of Walsh-Hadamard transform, which is expressed by the matrix $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, on every qubit in the register 1. (ii) Successive π controlled by the binary of the register 1 executes the function of $(a,1) \mapsto (a, 7^a \text{ mod } 15)$. Combined with the NOT operation on the first qubit, those successive π make the superposition of $7^a \text{ mod } 15$ from the state 0. (iii) Quantum Fourier transform [6] finds the period of a in the register 1. “Qubit swap” is the counterpart of the bit shuffling in the FFT, which is realized by CPHAS and CROT instructions.

register 1. (ii) Next, we compute $7^a \text{ mod } 15$ in the register 2 utilizing unitary-transformation π^i controlled by qubits in the register 1. As a result of this stage, we get the superposition of states $(a, 7^a \text{ mod } 15)$. (iii) After we perform the quantum Fourier transformation on the register 1, finally, we find a multiple of $16/r$ in the register 1 by the observation, where r is the order of $7^a \text{ mod } 15$. In this demonstration, $r=4$. The factors of 15, 3 or 5, can be obtained from $\text{gcd}(7^{r/2}-1, n)$ and $\text{gcd}(7^{r/2}+1, n)$ successfully only when 4 or 12 is observed in the register 1. We should repeat the above routine (i) — (iii) until we observe either of them. The statistics of the number of repetition to find 3 or 5 in the experiment is shown in Fig. 7. As a result, we got the expected times of the repetitions 1.9177 in our fabricated system. The single execution of this algorithm takes 131 instructions including observation operation.

5. CONCLUDING REMARKS

We fabricated the 8-qubit QCP in a single chip and demonstrated the Shor’s algorithm with it. Its fractal hardware structure is suitable for the low-cost design of the massive parallel computing system. This processor shows a good performance in the simulation of the quantum computer utilizing its parallel-hardware operation. Further, extending this processor so as to execute not only unitary transformations but also some classical operations, we may utilize it for the study of novel algorithms which complement both classical and quantum features.

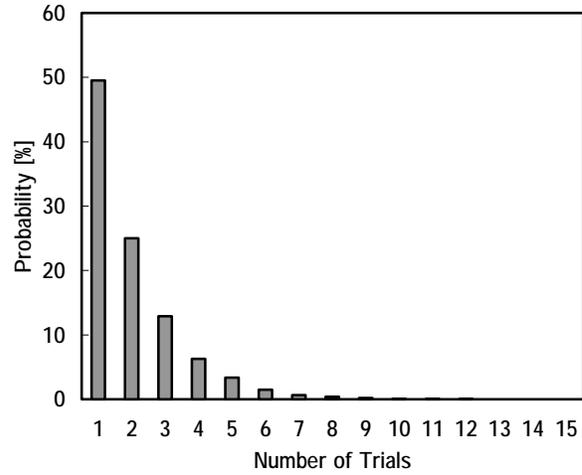


Fig. 7. Statistics of the number of trial for obtaining the solution 3 and 5. The candidates of the observed solution in the register 1 are 0, 4, 8 and 12. The solution is given when 4 or 12 is observed among them. The horizontal axis represents the number of trials of the procedure shown in Fig. 6 till observing 4 or 12. This graph was obtained by 10000 times experiments.

REFERENCES

- [1] S. O’uchi, M. Fujishima, and K. Hoh, “A programmable SIMD processor for universal quantum-circuit simulator”, in *the 2001 International Conference on Solid State Devices and Materials*, Tokyo, 2001, pp. 402-403.
- [2] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, R. Cleve, and I.L. Chuang, “Experimental realization of an order-finding algorithm with an NMR quantum computer”, *Phys. Rev. Lett.*, vol. 85, pp. 5422-5455.
- [3] D. Deutch, A. Barenco, and A. Ekert, “Universality in quantum computation”, *Proc. R. Soc. London Ser. A*, vol. 449, pp. 669-677, 1995.
- [4] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, “Elementary gates for quantum computation”, *Phys. Rev. A*, vol. 52, pp. 3457-3467, 1995.
- [5] P.W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, *SIAM J. Comput.*, vol. 26, pp. 1484-1509, 1997.