

P1-5

High-Speed Emulation of the Quantum Computing Based on Logic Operations

Kosuke SAITO*, Yasufumi SUZUKI*, Minoru FUJISHIMA* and Koichiro HOH* **

* Graduate School of Frontier Sciences, the University of Tokyo and **CREST

7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Phone: +81-3-5841-6775

Fax: +81-3-5841-8574

E-mail: info@sf.t.u-tokyo.ac.jp

Introduction

It has been about twenty years after the theory of a quantum computer (QC) was proposed. Although it promises to solve such problems as factorization [1] and database retrieving [2] very efficiently, the scale of the realized systems is too small to be practical. In order to seek for new applications working efficiently on the QC, we are studying quantum circuit processors emulating the behavior of the QC on LSI chips, utilizing its massive-parallel computation. In this paper, we propose a new processor, the structure of which is simplified since it is limited to the function required to execute fundamental operations in most of quantum algorithms. The processor, named a logic quantum processor, emulates quantum algorithms at high speed as a useful tool to lead and promote the QC.

Quantum Computer

A certain atom has two spin directions and either direction is observed, depending on probabilities of the observation. One quantum-bit (qubit) QC is realized when these probability amplitudes are operated from the outside of this quantum particle. The n -qubit QC solves problems by amplifying the probabilities of the desirable spin states. Operations in the QC are based on the unitary transformations because the sum of all probabilities is one. A unitary matrix to the probability vector $(|0\rangle, |1\rangle)$ of the target qubit represents a unitary transformation as follows:

$$\begin{pmatrix} |0\rangle \\ |1\rangle \end{pmatrix} = \begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix} \begin{pmatrix} |0\rangle \\ |1\rangle \end{pmatrix}, \text{ where } \begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix} \text{ is a unitary matrix.}$$

One quantum operation changes all probabilities simultaneously and this massive parallelism is the biggest feature and also the biggest advantage of the QC compared with other computers as shown in Fig. 1.

Logic Quantum Processor

Many processing elements (PEs) need to be arrayed to emulate the QC's massive-parallel computation on LSI. Each PE corresponds to each state and holds the state's probability as data. All qubits can be the operation target by connecting all PE pairs with the hamming distance of one. Consequently, PEs comprises a hypercube network as shown in Fig. 2.

Each PE requires two multipliers and one adder in order to calculate all the unitary transformations. It turns out, however, that structure can be simplified if the quantum algorithms are examined in detail. Most of quantum algorithms have four stages as shown in Fig. 3 and only two kinds of probabilities of 0 and p appear until the second stage, where $0 < p \leq 1$. Therefore, probability can

be expressed by one bit when p is regarded as 1. Each one bit is a "flag" which expresses whether the probability of each state exists or not. One-bit logic operations substitute for multiplications and additions, and unitary matrices are summarized in only two kinds of commands as shown in Fig. 4. Thus, the core part of the quantum algorithms is emulated correctly with these flags. Each PE realized by a logic unit is so small that the number of qubits integrated on a chip increases. The processor was named a logic quantum processor (LQP) since it emulates the quantum circuit using logic operations. In the LQP, inquiry operation is used for the substitution of a quantum Fourier transformation and observation in the third and fourth stages in Fig. 3. The inquiry operation *inquires* whether there exist one or more states satisfying a given bit mask with the flag with one. The result of this operation is stored in an answer register. The final solution is found with a binary search within a polynomial steps by generating the next inquiry automatically according to the answer register as shown in Fig. 5. The whole system is shown in Fig. 6.

Implementation of LQP on FPGA

We have coded the architecture of the LQP with Verilog HDL and implemented the LQP on a field programmable gate array (FPGA). Each PE has a local memory and manages multiple states to utilize the hardware resources efficiently. The 16-qubit LQP was realized; where $1024 (= 2^{10})$ PEs with $64 (= 2^6)$ bit local memories are implemented. Pipelines with the depth of 6 are adopted in order to realize high-speed as shown in Fig. 7. The performance is summarized in Table 1 compared with a simulation result using a PC with a 600MHz Pentium III.

Conclusion

A high-speed architecture, LQP, for the emulation of the quantum algorithms was presented. The LQP integrated the restricted functions of the quantum computer with 16 qubits on one chip. The operation time of the LQP is 2000 times faster than a simulation using PC. It is expected that the development of the quantum algorithms progresses by making the best use of LQP's high-speed quantum computing.

Reference

- [1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science (1994), pp. 124-134.
- [2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proc. 28th Annual ACM Symposium on Theory of Computing, pp. 212-219, May 1996.

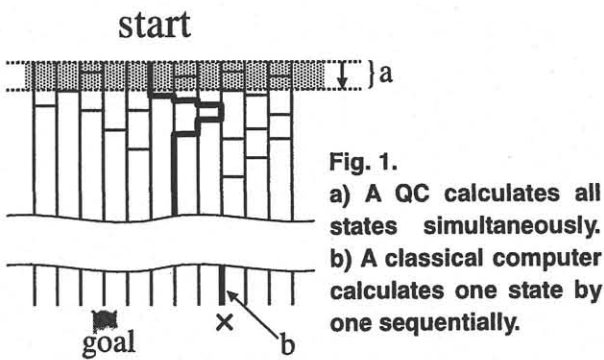


Fig. 1.
a) A QC calculates all states simultaneously.
b) A classical computer calculates one state by one sequentially.

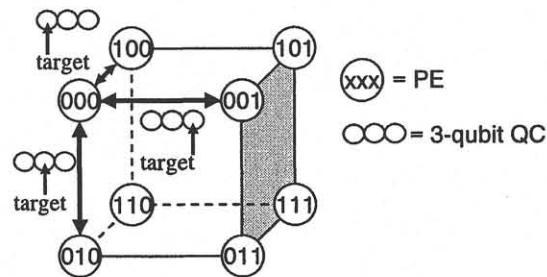


Fig. 2. In the case of three qubits, each PE is connected to three PEs with the hamming distance of one. One of them is chosen according to the target qubit. PEs comprise a hypercube network.

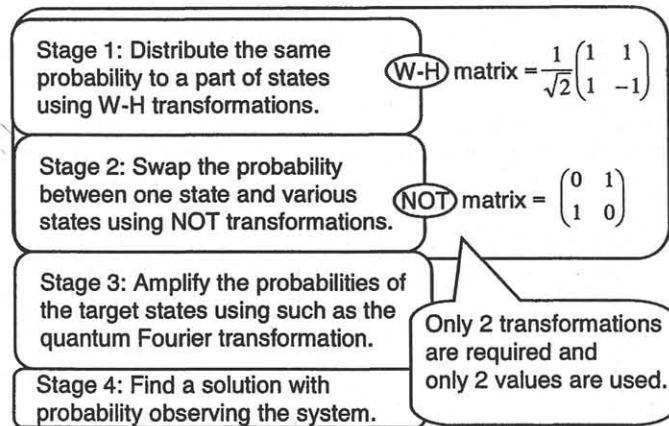


Fig. 3. Quantum algorithms. Until the end of Stage 2, only 2 kinds of probability values and 2 quantum operations appear. "W-H" represents a Walsh Hadamard transformation.

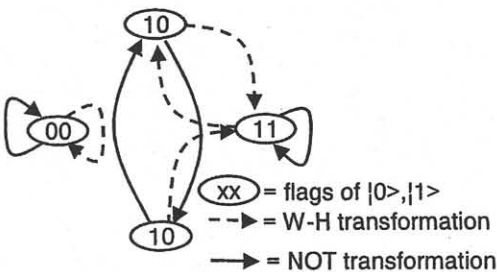


Fig. 4. Only two operations of NOT and W-H are needed to execute the stage 1 and 2 in Fig. 3.

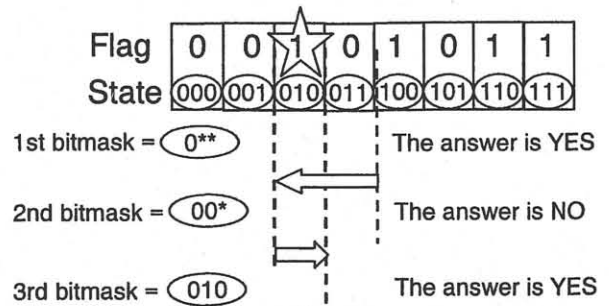


Fig. 5. An example of the inquiry operation. The minimum state with the flag of one is found with a binary search. The answer of this case is 010.

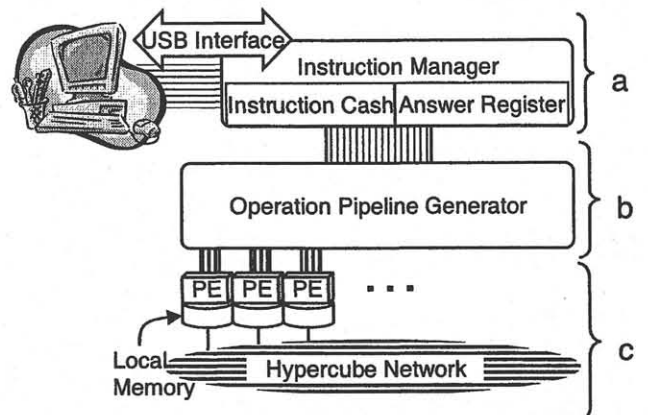


Fig. 6. The LQP a) decodes or regenerates operations according to internal register, b) makes and broadcasts signals to PE pipeline, and c) 1024 PEs with 64 state flags are implemented and connected with the hypercube network.

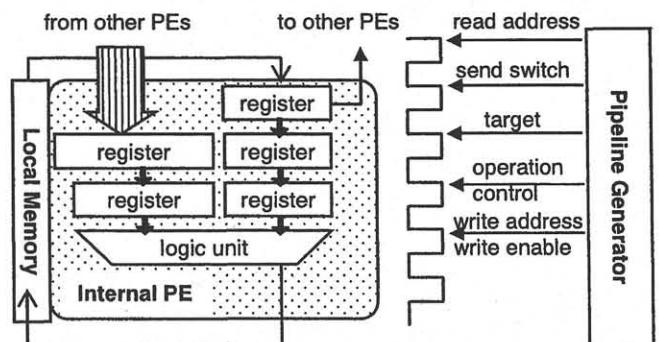


Fig. 7. High-speed operation is available due to the pipeline with the depth of 6.

Table 1. Comparing the performance of the LQP with a simulation result obtained by a software simulation. The LQP operates 2000 times faster than the PC.

	16-qubit LQP	Software simulation
System	APEX 20KE1500 (1.5M Gates)	Mobile Pentium III (600MHz 512kB cash)
Clock frequency	60MHz(1024 parallel)	-----
Performance	1.6M operations/sec	0.8K operations/sec