

Chip Multiprocessor Based on Dual Instruction Multiple Data Architecture

Masahiro Shimura¹ and Minoru Fujishima^{1,2}

¹School of Engineering, ²School of Frontier Sciences, The University of Tokyo,
5-1-5-703 Kashiwanoha, Kashiwa, Chiba 277-8561, Japan
Phone: +81-4-7136-3849 E-mail: info@axcel.k.u-tokyo.ac.jp

1. Introduction

In recent years, a CMP (chip multiprocessor) has been studied to realize a high-speed parallel operation [1]. However, since the performance of the CMP depends on an algorithm, it is difficult to achieve an operation speed proportional to the number of parallel processors in all cases. On the other hand, an appropriate algorithm realizes a high-speed operation using parallel processors. For example, FFT (fast Fourier transformation) is suitable for parallel processors, and the speed is highly improved by parallel operation [2]. Moreover, a reconfigurable processor is another example realizing a high-speed operation for streaming applications by changing the configuration of the processors dynamically [3]. Although the hardware is not realized yet, a quantum computer is believed to solve NP (nondeterministic polynomial) problems, which require a long computation time by conventional processors, at a high speed [4].

To solve NP problems at a high speed, we are studying a dedicated LSI (large-scale integrated circuit) processor, without quantum computing. In this paper, we propose DIMD (dual instruction multiple data) architecture for on-chip massively parallel processing. In the following sections, after the proposed processor is explained, measurement and future prospects are discussed.

2. DIMD Processor

Optimization of Data Flow Processing

An inverse problem is a problem in which an answer is easy to verify, although obtaining the answer requires a long computation time. Many inverse problems such as factorization, SAT (satisfiability) problem, and TSP (traveling salesperson problem) become NP problems. To solve an inverse problem, the same function is applied to all candidates for an answer and the answer is searched from all the outputs of the function as shown in Fig. 1. As a result, SIMD (single instruction multiple data) architecture is basically suitable because multiple data is calculated by a single instruction. However, since a general algorithm requires a conditional command generated in a given data flow, only some of the PEs (processing elements) are in operation, whereas the other PEs are paused when the SIMD architecture is adopted. To reduce the number of paused PEs, we propose the DIMD architecture, which has two operational codes for true and false instructions at one operation step, and one of the two codes is selected depending on the status of flag registers in the processors as shown in Fig. 2. The block diagram of the PEs used in the DIMD architecture is shown in Fig. 3. All PEs have a local memory and five flags which select either true or false instruction to be operated. They are given different ID numbers to generate different data from the DIMD instructions.

A controller manages the order of operations separately from the PEs using instructions such as branch, arithmetic and logical instructions. The controller and the PEs operate in parallel. As a result, a branch operation by the controller

and a DIMD operation by the PEs are issued at the same time and no additional operation step is required as shown in Fig. 4. The instruction set is summarized in Table 1. Every operation is executed with a single clock, which is similar to a RISC (reduced instruction set computer).

Search

In inverse problems, the answer has to be searched from the multiple results calculated by the processors based on the DIMD architecture. To obtain the final answer, shared registers are utilized among the processors with a binary-tree connection as shown in Fig. 5. The answers of inverse problems are obtained at a high speed by collecting the results calculated by the PE with an ID number of zero. Since one of the registers in the PE with the ID number of zero is shared with the controller, the results calculated by the PE are obtained in the controller. One of the registers in the controller is also shared with all PEs, and it gives an immediate number and results calculated by the controller.

3. Measurement and Discussion

The proposed architecture is implemented in an FPGA (field programmable gate array) with 1.5 million logic gates. Since a PE has no multiplier or divider to minimize the number of logic gates, multiplication and division are realized by executing addition and subtraction operations sequentially. Consequently, 192 PEs with a binary connection and one controller are realized. In the implemented architecture, 7.7 GOPS are achieved at a clock frequency of 40 MHz. A DIMD chip multiprocessor factorizes 64-bit numbers with 2.2 billion steps, while approximately 200 billion steps require by a Pentium4 processor, when a trial division method is adopted. Thus, the DIMD multiprocessor requires only 1/100 of the operation steps of the Pentium4 processor. Nevertheless, as shown in Fig. 6, the operation speed using the proposed processor is slightly worse than that of the Pentium4 processor with a clock frequency of 3.4 GHz since the clock frequency of the DIMD multiprocessor is 1/85 of that of the Pentium4 processor. However, it is noted that 2,048 PEs can be implemented by estimation when the DIMD multiprocessor is fabricated with a custom design using a 90-nm CMOS process. With a clock frequency of 340 MHz, which is 1/10 of that of the Pentium4 processor, 700 GOPS is achieved, which is 80 times faster than that of the newest Pentium4 processor as shown in Fig. 6.

4. Conclusions

We proposed a DIMD multiprocessor operating massive parallel PEs on a chip. The DIMD multiprocessor realized a high-speed parallel operation and a high-speed search operation in a flexible programming environment. As a result of the fabrication using the FPGA, the number of operation steps is reduced up to 1/100 of that of a Pentium4 processor. It is also estimated that custom designing using 90nm CMOS technology will have an operation speed 80 times that of a case using an up-to-date general-purpose processor.

References

- [1] L. Hammond, B.A. Nayfeh, and K. Olukotun, *Computer* **30** (1997) 79.
- [2] L.H. Soo, H. Mori, and H. Aiso, *Trans. of the Institute of Electronics and Communication Eng. of Jpn.* **E68** (1985) 284.
- [3] K. Bondalapati, V.K. Prasanna, *Proceedings of the IEEE* **90** (2002) 1201.
- [4] M. Fujishima, *Fluctuation and Noise Letters* **3** (2003) C9.

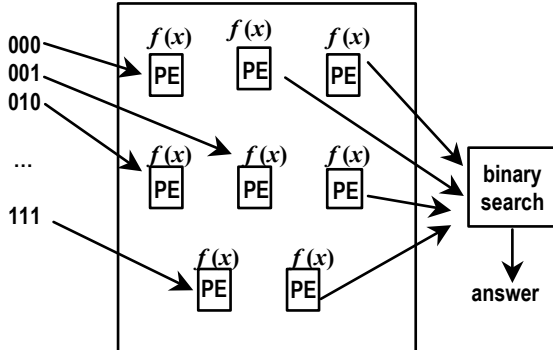


Fig. 1 Schematic data flow for solving an inverse problem where an answer is searched from the calculated results.

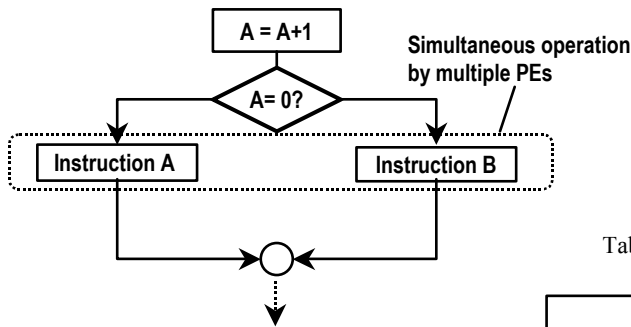


Fig. 2 DIMD operates one of two instructions in all PEs.

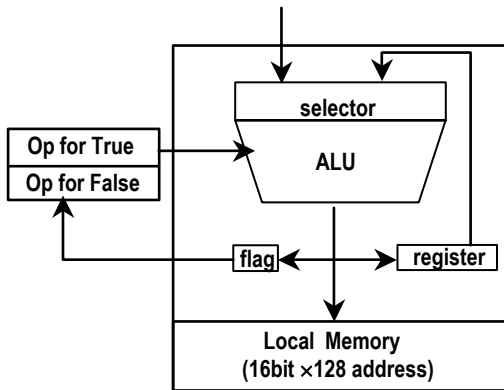


Fig. 3 Schematic block diagram of a PE. The flag determines either "Op for True" or "Op for False".

PE Instruction	Controller Instruction
Step 1	MOV #10, r1
Step 2 L1: ADD [01], r1, r2	NOP
Step 3 SLL r1, r2	NOP
Step 4 ADD r1, r2	SUB r1, #1, r2
Step 5 SLL r2, [02]	BNZ L1

Fig. 4 Branch operation is executed in the background of PE operations. [01] is the data of memory address 1.

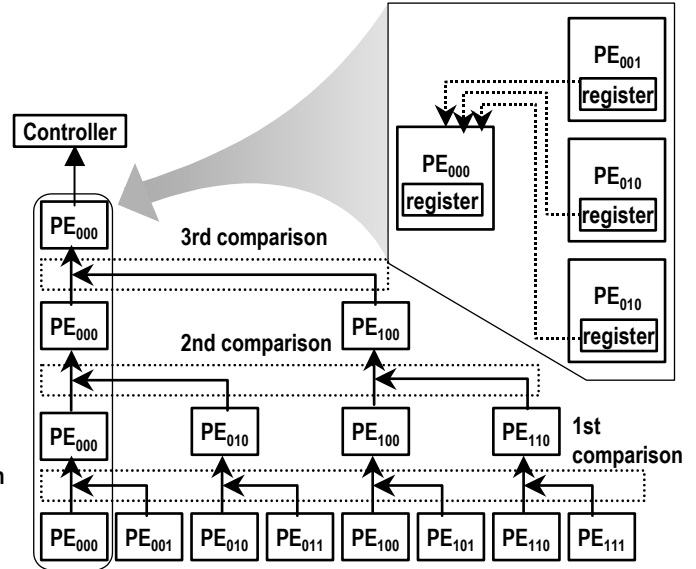


Fig. 5 Some of the PE registers are shared with the binary-tree connection.

Table 1. The instruction set for DIMD multiprocessor, where \$D and \$S are destination and source operands, respectively.

Instructions for a controller			
Instructions for PEs			
ADD \$D, \$S1, \$S2	\$D = \$S1 + \$S2	JMP	Jump to absolute address
ADC \$D, \$S1, \$S2	\$D = \$S1 + \$S2 + C	BR	Jump to relative address
SUB \$D, \$S1, \$S2	\$D = \$S1 - \$S2	PUSH \$S1	Push \$S1 to stack
SBB \$D, \$S1, \$S2	\$D = \$S1 - \$S2 - C	POP \$D	Pop stack to \$D
SBI \$D, \$S1, \$S2	\$D = -\$S1 + \$S2	CALL	Call subroutine
SBIB \$D, \$S1, \$S2	\$D = -\$S1 + \$S2 - C	RET	Return from subroutine
AND \$D, \$S1, \$S2	\$D = \$S1 & \$S2	LD \$D, (M)	\$D = (M)
OR \$D, \$S1, \$S2	\$D = \$S1 \$S2	ST (M), \$S1	(M) = \$S1
XOR \$D, \$S1, \$S2	\$D = \$S1 ^ \$S2		
MOV \$D, \$S1	\$D = \$S1		
SRL \$D, \$S1	\$D = \$S1 >> 1		
NOP	No Operation		

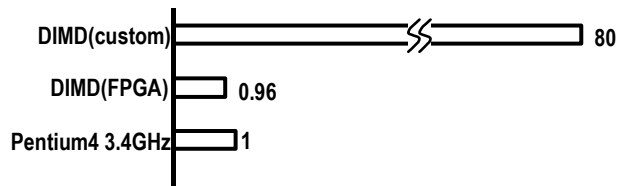


Fig. 6 Performance comparison of the DIMD multiprocessor and the Pentium4 processor with a clock frequency of 3.4 GHz.