

Bank-Type Multiport Register File for Highly-Parallel Processors

Tetsuya Sueyoshi¹, Hiroshi Uchida¹, Yosuke Mitani², Ken Hiramatsu²,

Hans Jürgen Mattausch¹, Tetsushi Koide¹, and Tetsuo Hironaka²

¹Research Center for Nanodevices and Systems, Hiroshima University, 1-4-2 Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

²Graduate School of Information Sciences, Hiroshima City University, 3-4-1 Ozuka-Higashi, Asaminami-ku, 731-3194, Japan

Phone: +81-824-24-6265 Fax: +81-824-22-7185 email: {sueyoshi,hjm,koide}@sxsys.hiroshima-u.ac.jp

1. Introduction

Common techniques for improving processor performance exploit possible parallelism on the instruction level, as in superscalar processors or on the level of program threads as in SMT (Simultaneous Multi-Threading) processors. It is expected that the number of simultaneously issued instructions as well as the number of thread run simultaneously will continue to increase. To support this development, it is necessary to supply sufficiently increased access ports as well as entry numbers of the register file. For example, a processor design that supports an 8-issue/4-thread SMT processor has to be equipped with a 24-port register file having 512 entries [1]. In such a multiport register file with many ports and large capacity, if the conventional multiport-cell architecture is used, the problems of substantially increased area, access time, and power consumption arises. In order to solve these problems, we propose to apply a multiport register file with multi-bank structure, each bank having just 1 port.

2. Architecture Concept for the Bank-Type Multiport Register File

2.1 Hierarchical Multiport-memory Architecture (HMA)

We have selected the Hierarchical Multiport-memory architecture (HMA) [2] for the bank-type register file. HMA reduces the multiport-memory's area by using 1-port-cell banks, like the classical bank-type memory, using a crossbar. The difference between HMA and the conventional crossbar architecture is that 1:N-port convertor circuits are attached to each bank to realize a distributed crossbar functionality. In this way the necessary number of transistors and global wirings can be reduced without degrading the functionality [3]. In order to avoid access conflicts to the banks, an access conflict management circuit is included on the 2nd level of HMA. Access conflicts to the banks for each port are detected, and permission/prohibition of these accesses are controlled.

The block diagram of the proposed HMA register file is shown in Fig. 1. At the beginning of an access cycle the access conflict management circuit and row/column bank selector determine, which port can access which bank. For a read access, an address is supplied from the selected read-port and the data from the accessed register in the bank is transferred to the read-port. At the time of writing, address and data are supplied from the selected write-port and the data is written into the accessed register of the bank.

2.2 Access Method with Hidden Precharge

The conventional access method of a synchronous memory, divides the clock cycle into two phases: the memory access phase and the precharge phase, the latter to prepare the memory for the access in the following clock cycle (Fig. 2 (a)). Therefore, the cycle time is the sum of the actual memory-access phase and the precharge phase. However, it is desirable to have a clock cycle time, which is equal to and not longer than the actual memory-access time. Consequently, we propose an access method for a synchronous HMA register file, which hides the precharge phase by carrying out conflict

management, port selection, decoding etc. simultaneously with the precharge phase of the banks (Fig. 2 (b)). In Fig. 2, the register file access up to the bank's wordline driver is operated simultaneously with bank precharging when clock is "0". When the clock becomes "1", the remaining access part, starting with wordline-driver activation, is carried out. In this way equal length of clock-cycle time and actual register file access time are realized.

3. Performance Evaluation of a Superscalar Processor with Bank-Type Register file

The main problem of a bank-type register file is due to possible access conflicts. If there is access from two or more ports simultaneously to one bank, only one port can access the bank and the others cannot. In consequence the processing time of the related instructions is delayed and the processor-performance is degraded. To handle this problem we proposed to apply a register-access scheduling, which includes a register renaming under consideration of the bank structure, a register access queue, etc. [4]. In order to examine the validity of the scheduling approach, we compared the performance of our bank-register file with an ideal multiport memory by a trace-driven simulator. The register file is assumed to have 12 ports (8 read-ports, 4 write-ports) and 128 registers, which is typical for a processor executing four instructions in parallel. The result of the simulation experiment is shown in Fig. 3. It turns out that access conflict number can be kept low by using the register-access scheduling, and that an instruction processing performance nearly equivalent to an ideal 12-port register file is obtained when 4 or more banks are used.

4. CMOS Design Study of the Proposed HMA Register File

We have designed 3 types of 12-port register files with 128 registers in a 0.35μm CMOS technology with 3-metal layers, and performed an analog circuit simulation (HSPICE) at the layout level to compare their performance. The first design implements the conventional multiport register file with a 12-port SRAM cell and the other two designs implement the proposed HMA register file, one using the access method with hidden precharge and the other using the conventional access method. Layout comparison of 12-port cell register file and the HMA register file with hidden precharge and 4 banks is shown in Fig. 4. We also propose an L-type floorplan for the banks in the HMA register file (Fig. 5). The advantage of this floorplan is that the global wirings for outputs and inputs can be realized on top of the banks. Thus length and area consumption of the wiring can be minimized. The summary of the register file design comparison in Table I shows that the HMA register file with the hidden-precharge access can achieve enormous improvements, reducing the area by 70%, the access cycle by 49% and the power consumption by 81%, when compared with the conventional multiport cell based register file. 25% of the access-cycle time reduction and 20% of power-consumption reduction are a result of the access technique with hidden precharge.

5. Conclusion

We proposed a hierarchical bank-type register file construction for highly parallel processors in order to solve the problems of increasing area, access time and power consumption, which result from the required large port number. A detailed analysis showed that enormous reductions in area (70%), access cycle (49%), power consumption (81%) are possible, when compared with the conventional multiport-cell-based register file. The bank-type register file was found to be applicable in superscalar processors without loss in processor performance, when applying the new access scheduling method.

Acknowledgments

This research is supported by Semiconductor Technology Academic Research Center (STARC).

The VLSI chip in this study has been fabricated in the chip

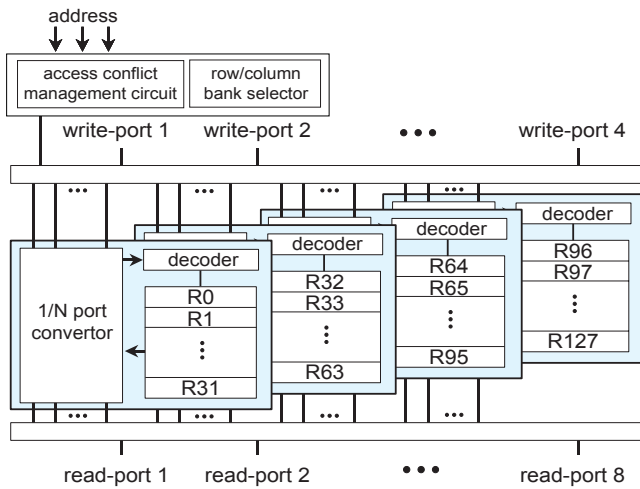


Fig.1: Proposed hierarchical bank-type register file architecture.

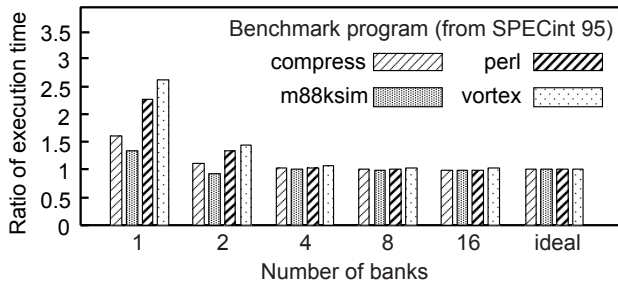


Fig.3: Evaluation of register access scheduling efficiency for a bank-type register file. An execution time of 1 corresponds to an ideal register file.

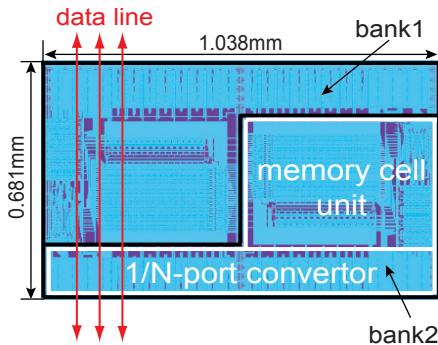


Fig.5: Floorplan of HMA-register-file bank with L-shape structure.

fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation and Cadence Design Systems, Inc.

References

- [1] R. P. Preston, et al., "Design of an 8-wide superscalar RISC microprocessor with simultaneous multithreading," Dig. of Tech. Papers, ISSCC2002, pp. 334-335, 2002.
- [2] H. J. Mattausch, et al., "Area-efficient multi-port SRAMs for on-chip data-storage with high random-access bandwidth and large storage capacity," IEICE Trans. Electron., Vol. E84-C, No. 3, pp. 410-417, 2001.
- [3] S. Fukae, et al., "Optimized bank-based multi-port memories through a hierarchical multi-bank structure," Proc. of SASIMI2003, pp. 323-330, 2003.
- [4] Y. Mitani, et al., "Access conflict resolution methods for superscalar processors with multi-bank register file," Tech. Rep. of IEICE, ARC-2002-150-9, pp. 41-46, 2002 (in Japanese).

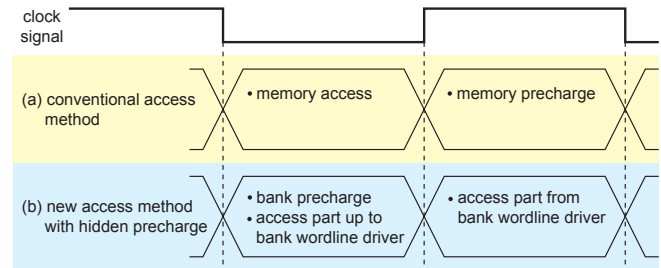
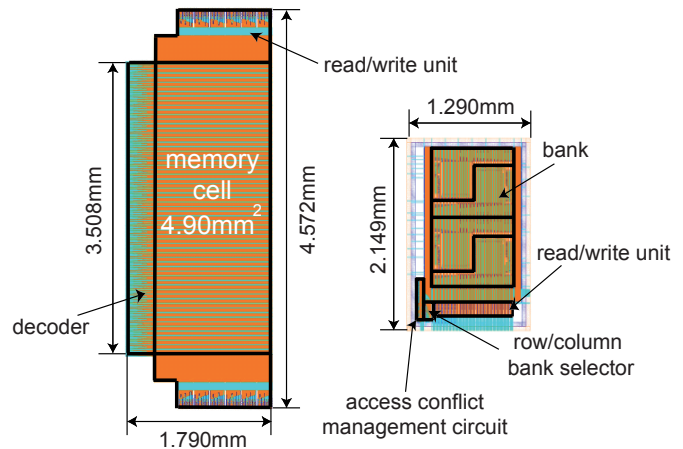


Fig.2: New access method with hidden precharge for a synchronous bank-type multiport memory.



(a) Multiport SRAM cell based register file (conventional). (b) Hierarchical bank-type register file.

Fig.4: Layout comparison in 0.35μm CMOS technology (12-ports, 128-register).

Table I: Design-comparison results between conventional multi-port-cell-based and hierarchical bank-type register files (RFs).

	Multiport SRAM cell based RF (conventional)	HMA RF with conventional access method	HMA RF with hidden precharge access method
Area [mm ²] (ratio)	7.84 (1)	2.23 (0.28)	2.36 (0.30)
Cycle time [ns] (ratio)	14 (1)	10.7 (0.76)	7.2 (0.51)
Power consumption (@50MHz) [mV] (ratio)	543 (1)	212 (0.39)	103 (0.19)