Unified Data/Instruction Cache with Bank-Based Multi-Port Architecture

Koh Johguchi1, Zhaomin Zhu1, Hans Jürgen Mattausch1, Tetsushi Koide1 and Tetsuo Hironaka2

1Hiroshima Univ., Research Center for Nanodevices and Systems, 1-4-2 Kagamiyama, Higashi-Hiroshima, Japan 739-8527, Japan
2Hiroshima City Univ., Faculty of Computer Sciences, 3-4-1 OzuakaHigashi, AsaMinami-Ku, Hiroshima, 731-3194, Japan
Phone: +81-82-424-6265, FAX: +81-82-424-3499, E-mail: {jouguchi, zzm, koide, hjm}@sxsys.hiroshima-u.ac.jp

1. Introduction
Modern processors simultaneously fetch, decode and execute many instructions. This results in the demand for a large access bandwidth of the processor’s memory components and has already led to register files with many ports. However, for the cache memory the conventional solution of 1-port-data and instruction caches is still in use. Since the demand for parallelism tends to increase at a high rate, the cache system will become the bottleneck of processor performance and has to be innovated.

We propose to improve the access bandwidth of the cache with a 1-port-based multi-level Data/Instruction Cache (DHT), which can simultaneously realize small area and high performance. The two-dimensional bank decoder reduces the overhead of bank selection and allows easy matrix arrangement of the banks [1].

Fig. 1 shows the structure-example of a direct mapped cache which uses HMA [2]. The cache index, consisting of line number (LN) and line offset (LO), is divided into two portions, a bank internal address (BI) and a bank number (BN). BI is used for selecting a cache word or tag within memory banks, and BN is used for selecting the respective banks within data/instruction or tag memory. BN uses the lower rank bits in order to make sure that consecutive lines and words within lines are located in different banks, so that they can be accessed in parallel without access conflict.

2. Multi-port Cache with HMA Structure
HMA is a 1-port-bank based multi-port memory architecture which further improves area consumption and performance of the conventional crossbar architecture. The crossbar’s switching network is distributed into the bank structure, which decreases global wiring and transistor number. A two-dimensional bank decoder reduces the overhead for bank selection and allows easy matrix arrangement of the banks [1].

Fig. 1 shows the structure-example of a direct mapped cache which uses HMA [2]. The cache index, consisting of line number (LN) and line offset (LO), is divided into two portions, a bank internal address (BI) and a bank number (BN). BI is used for selecting a cache word or tag within memory banks, and BN is used for selecting the respective banks within data/instruction or tag memory. BN uses the lower rank bits in order to make sure that consecutive lines and words within lines are located in different banks, so that they can be accessed in parallel without access conflict.

3. Combination of Instruction and Data Cache

3.1. Advantages of a Unified Cache
Using the proposed HMA cache, instruction and data cache can be unified without loss in access bandwidth, but with the advantage of a lower miss rate at the same storage capacity. On the other hand, using a bank-based multi-port cache, access to one bank is restricted to 1 port, and access-conflict rate may increase.

The miss-rate advantage of the unified cache and the required number of banks for sufficiently small access-conflict rate are examined for the example of a 4-way superscalar processor. The simulation is carried out with a modified version of SimpleScalar [3]. Dhrystone and SPEC95 (gcc, ipexg, etc.) are used as benchmarks.

The results for split unified and direct-mapped cache are shown in Figs. 2 and 3. The storage capacity values in the figures show the total capacity of instruction and data cache, being the same for the unified cache. According to Fig. 2, in the case that sufficient capacity is not prepared, the unified cache has higher miss rate than the split cache because data rewriting takes place frequently. However, if the total miss rate becomes lower than 10%, as required in real processors, the miss rate of the unified cache is clearly lower than that of the split cache. Moreover, it turns out that the miss rate of the unified cache is approximately equal to that of a split cache at 25% reduced storage capacity. We conclude from the result of Fig. 3, that the access-conflict rate becomes sufficiently low when more than 16 banks are provided.

3.2. Optimum Combination of Instruction and Data Cache with HMA Structure

Nomally the accesses to the instruction cache are consecutive. For a 4-way superscalar processor, it is therefore expected, that one instruction port with 4-time larger word length will deliver sufficient instruction-fetch performance. The optimum number of data-access ports is estimated to be 2 or 3.

Above considerations suggest that an optimized unified data/instruction cache should have different word length for data and instruction ports. Fig. 4 shows our HMA proposal of a unified write-through cache with 2 data ports and 1 instruction port, with 4 times larger word length, for 4-way superscalar processors. Although, it uses internally only a 1- to 3-port converter with a relatively small area-overhead, the externally available access bandwidth corresponds to 6 ports, due to the 4 times increased word length of the instruction port.

4. Test-Chip Design for an HMA Cache

For the test chip of an HMA cache memory, a configuration with 4 ports was chosen and the design was carried out in a 0.18µm CMOS technology. The chip-layout shown in Fig. 5 contains all needed new functional units. The design data are summarized in Table I. Small area and short delay are achieved with a dynamic CMOS circuit technology and effective floor planning. The area-overhead of the 1:4-port converter for the 1Kbyte bank of Fig. 6 is less than 25%.

We also applied a new access method which overlaps bank-conflict management and bank decoding with the precharging phase of the banks. As a result, bank-access time, complete cache-access time and power dissipation are 1.9ns, 3.8ns and 247mW at 250MHz, respectively, as determined with layout-based simulation.

5. Conclusions

In this paper, a bank-based unified data/instruction cache with multiple ports has been proposed and the advantages have been verified by simulation. Especially important is our method of providing a different word length for data and instruction ports, which takes advantage of the internal bank structure. To minimize bank conflicts, we use an addressing method, which insures that the words in one cache-line and also consecutive cache-lines are located in different banks. A test-chip design of a 4-port bank-based cache in 0.18µm CMOS technology showed, that the area-overhead for the 4 ports is about 25%. A minimum clock cycle time of 3.8 ns could be achieved with a dynamic CMOS circuit technology and by overlapping the external bank access with the bank-internal precharge.

The proposed bank-based multi-port cache is also very attractive for low power dissipation, because the number of activated banks, determining power dissipation, corresponds to the port number and is independent of the total number of banks in the cache.
Acknowledgements
This research is sponsored by Semiconductor Technology Academic Research Center (STARC).
The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Hitachi Ltd., Dai Nippon Printing Corporation and Cadence Design Systems, Inc. This work was supported in part by the 21st Century COE program and a JSPS research fellowship, No.1605246, Japanese Government.

References
Unified Data/Instruction Cache with Bank-Based Multi-Port Architecture

Koh Johguchi1, Zhaomin Zhu1, Hans Jürgen Mattausch1, Tetsushi Koide1, Tetsuo Hironaka2
1 Research Center for Nanodevices and Systems (RCNS), Hiroshima University
2 Department of Computer Engineering, Hiroshima City University

Memory Hierarchy of Processors

Multi-Port Cache Applications

Special Issues of Conventional Multi-Port Cache & Proposed Solution

Motivation

By using a multi-port cache, an instruction and data cache can be unified.

Bank Row Decoder (N Ports)

1:N Port Converter is integrated into the banks to achieve modularity.

Bank Number

High-speed SRAM cells are connected to banks with 1-port SRAM cells.

Multi-Bank Type

Solution for Multi-Port Caches

(1) Low area efficiency
- Application of 1-port SRAM bank based multi-port memory (HMA).
- Dynamic CMOS technology can be used for port converters and conflict manager.

(2) Increase of delay time for multi-port memory
- Synchronous 2-stage access mode for holding precharge phases.

(3) Difference of the access pattern between data and instruction caches (in order to unify of two caches)
- Development of a new multi-port cache with the port dependent word-length and interleaved cache-line words

Motivation

Tag memory and data memory both use HMA.

Bank number is determined by the lower bits of the index, so that conflicts will not take place when addresses are consecutive.

Unified Data/Instruction Cache with HMA

New Access Method for Multi-Port Memory

Simulation

Test Chip Design

4-Port Unified Data/Instruction Cache Test Chip

4-Port Unified Data/Instruction Cache Test Chip

Test Chip Specification

Using HMA and dynamic CMOS technology, we have achieved high speed and area-eficiency, simultaneously.

Layout Design of Bank (1st Level)

Area increase is proportional to square of port number.

High access conflict rate, because access-conflict penalty is only 1 clock.

When access conflict rate and miss rate are the same, multi-port capability is more viable than single-port.

Area-overhead for multi-port capability is about 30%.(Considering only 1st level)

Conclusions

- Test-chip design of a unified data/instruction HMA cache with 4 ports, 16KByte and 64 data-banks. Cycle time is 3.9ns and area overhead for multi-port capability is less than 25%.

- Unified data/instruction cache advantages.
  - Reduced miss rate
  - High area efficiency

- Concept proposal for additional unification of trace cache with data/instruction cache.