HPDcache: Open-source high-performance L1 data cache for RISC-V cores

César Fuguet

HPDcache: Open-Source High-Performance L1 Data Cache for RISC-V Cores

César Fuguet
cesar.fuguettortolero@cea.fr
Univ. Grenoble Alpes, CEA, List
F-38000 Grenoble, France

CCS CONCEPTS

• **Computer systems organization → Processors and memory architectures**.

KEYWORDS

cache memory, data cache, architecture, processor, RISC-V, open-source hardware

ACM Reference Format:
César Fuguet. 2023. HPDcache: Open-Source High-Performance L1 Data Cache for RISC-V Cores. In *20th ACM International Conference on Computing Frontiers (CF '23), May 9–11, 2023, Bologna, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3587135.3591413

1 INTRODUCTION

For many compute applications the performance bottleneck is the memory bandwidth and latency. This is particularly true in the domain of High-Performance Computing (e.g. scientific applications). Cache memories are an essential component of modern processors, which further pushes the "Memory Wall". Caches in the domain of HPC must enable both high memory throughput and energy efficiency.

This work introduces a High-Performance, Out-of-Order (OoO), L1 Data Cache (HPDcache) compatible with RISC-V processor cores [3]. This HPDcache was successfully integrated with the CVA6 core [4] (replacing its original L1 data cache). It is meant to be compatible with other cores with some tailoring in the load/store unit of the target core.

The HPDcache SystemVerilog Register-Transfer Level (RTL) sources are contributed as open-source with a permissive solderpad v2.1 license. It is available through the OpenHW Github [1].

2 RELATED WORKS

There are other remarkable open-source L1 data caches. The repository of the CVA6 core provides two different flavors of L1 Dcaches: write-back, and write-through. These caches, however, have some limitations: only one miss is supported per requester; there is no prevention mechanism for head-of-the-line blocking; they lack of configurability on both, requesters and memory interfaces.

The L1 Dcache of the Rocket and BOOM RISC-V cores [5] supports multiple outstanding read misses through a multi-entry Miss Status Holding Register (MSHR). However, in this implementation, the MSHR must be implemented in flip-flops (FFs), which limits it to a small number of entries (and thus a small number of outstanding misses) because of the area cost. Furthermore, this cache is implemented in Chisel, which requires additional effort for integration in non-Chisel-based design flows.

3 FEATURES

The HPDcache is a highly-configurable, high-performance L1 Data cache (Dcache) for RISC-V cores. It implements the following features:

- Set-associative cache with configurable (at compilation) number of ways and sets;
- Multiple ports for requests with support for wide data;
- Write-Through Policy with write-buffer supporting write-coalescing and multiple outstanding write requests;
- Pipelined micro-architecture for high clock frequencies;
- Single-cycle latency for read hit and write requests;
- Non-blocking pipeline with read under multiple misses;
- OoO execution of requests to prevent head-of-line blocking;
- Support for atomic memory operations (AMOs);
- Support for Cache Management Operations (CMOs) [2];
- Dedicated Configuration-and-Status Registers (CSR) address space to access performance counters and runtime configuration registers;
- Stride-based, programmable, hardware memory prefetcher;
- Native FIFO-like (ready-valid) memory interfaces and adapter for AMBA AXI5 interface;
- Different bus widths are supported on the memory interface, from 64 to 512 bits.

4 ARCHITECTURE

Figure 1 gives an overview of the micro-architecture of the HPDcache. The HPDcache implements a 3-stage pipeline in order to balance the logic complexity and, in particular, to reduce the number of SRAM-to-SRAM timing paths, which are challenging because of their associated access and setup times. The following subsections explain some of the features in the HPDcache.

4.1 Access latency

The access latency of requests on the cache is variable and depends on different factors. In brief, both cacheable read requests that "hit" in the cache and cacheable write requests may be processed and acknowledged in a single cycle. The latency in other cases depends mostly on the latency of next levels in the memory hierarchy.

CF '23, May 9–11, 2023, Bologna, Italy

ACM ISBN 979-8-4007-0140-5/23/05.
https://doi.org/10.1145/3587135.3591413

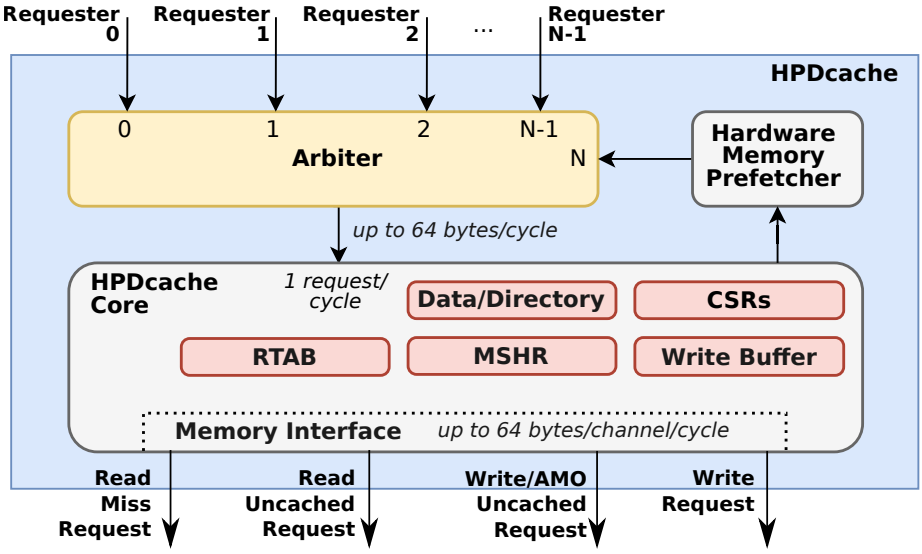

Figure 1: HPDcache Micro-architecture Overview

4.2 Multiple request ports

The HPDcache implements a configurable number of request ports. Each port can carry any of the supported types of operations: reads, writes, AMOs and CMOs. These multiple ports can be used by different load/store units implemented within the core. For example, two ports can be associated to the normal load/stores from the core, and two other ports can be associated to an accelerator/coprocessor implemented within that core. The HPDcache implements an additional port for the hardware memory prefetcher.

The data width of each port is also configurable. The maximum data width depends on the implemented RAM organization for the cache data. The constraint is that the cache must be able to access the requested data in a single cycle. The maximum data width is 512 bits, which is the maximum supported cache-line size.

4.3 Read under multiple miss

In order to support multiple miss requests to the memory, the HPDcache implements a deep MSHR (up to 128 entries). The MSHR keeps track of all outstanding miss requests.

To reduce the area footprint of this structure, that can be significant when the number of entries is high, the MSHR can be implemented in single-port SRAM (higher bits/μm^2 ratio than FFs). However, when there is a new request, the cache needs to check the MSHR to detect possible collisions. A set-associative organization is implemented in the MSHR to reduce the area cost of this collision detection mechanism.

4.4 Out-of-Order execution of requests

The HPDcache processes the requests in the order it consumes them from requesters. However, there are some blocking conditions where the pipeline may put a request on-hold in a multi-entry buffer called the Replay Table (RTAB). Some examples of these blocking conditions are: cacheable read on a pending miss (hit on the MSHR); or read miss on an address with a pending write in the write buffer.

The idea is to improve the throughput of the cache by reducing the number of cases where there is a head of the line blocking at the interface of any of the requesters. New requests with no address collision with on-hold requests in the RTAB are executed first, in an out-of-order fashion.

5 FUTURE WORK

A silicon prototype of a RISC-V based accelerator implementing this HPDcache is currently being fabricated in GF22FDX technology in the context of the *European Processor Initiative (EPI)* project.

In addition, new features, and industrial-grade verification is being done in the context of the European KDT Tristan project. New features include: hybrid write-through/write-back write policy and scratchpad mode.

ACKNOWLEDGMENTS

This work has been performed in the context of the EPI and TRISTAN projects. EPI has received funding from the European Union's Horizon research and innovation program under Grant Agreement EPI-SGA1: 826647. TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under Grant Agreement nr. 101095947. We also thank the OpenHW Group organization for hosting this project.

REFERENCES

[1] CV-HPDcache 2023. *HPDcache Github Repository.* Retrieved February 21, 2023 from https://github.com/openhwgroup/cv-hpdcache
[2] RISC-V Cache Management Operation Task Group. 2022. *RISC-V Base Cache Management Operation ISA Extensions.* Technical Report.
[3] Andrew Waterman and Krsten Asanovic. 2019. *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA.* Technical Report. SiFive Inc. and EECS Department, University of California, Berkeley.
[4] Florian Zaruba and Luca Benini. 2019. The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27, 11 (2019), 2629 – 2640. https://doi.org/10.1109/TVLSI.2019.2926114
[5] Jerry Zhao, Ben Korpan, Abraham Gonzalez, and Krste Asanovic. 2020. SonicBOOM: The 3rd Generation Berkeley Out-of-Order Machine. (May 2020).