A Unified AI Accelerator Interface for Scalable RISC-V Architectures

Fucong Qiu[1,2], Mingjin Gao[1,2], Dan Tang[1,3], Yungang Bao[1,2] and Tao Xie[3,4]

[1]State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Beijing Open Source Chip Research Institute
[4]School of Computer Science, Peking University

Introduction

- As an open and scalable instruction set architecture, RISC-V has rapidly gained widespread acceptance across computing tasks. To enhance processor versatility, general-purpose interfaces like the Rocket Custom Coprocessor Interface (ROCC) [1] and Core-V eXtension Interface (CV-X-IF) [2] were introduced.
- While effective for general applications, these interfaces struggle to meet AI workloads' demands—specifically massive parallelism and high-throughput data processing. Specialized interface solutions have emerged to address AI-specific challenges. The Open Vector Interface (OVI), proposed by Semidynamic, is a notable example that extends vector accelerator capabilities [3].
- However, such specialized interfaces typically cater to specific accelerator classes, thereby limiting their adaptability and expandability. In response, this work proposes a unified interface that serves as a comprehensive solution for diverse AI accelerator.

Features

- **Unified AI Accelerator Interface.** Besides supporting vector acceleration, the interface also incorporates matrix and tensor modes, enabling broader applicability across a range of AI tasks.
- **Diverse Data Access Paths.** Building on existing data access mechanisms, the interface also integrates multiple data paths, including L1 cache access, bus access, coprocessor extension access, and inter-coprocessor data movement, to support various AI dataflow patterns.
- **Virtual Memory Interface.** The interface allows accelerators to acquire physical addresses from the processor by supporting both private TLBs and direct MMU access, ensuring flexible and efficient memory address translation.
- **Decoupled Microarchitecture Design.** In contrast to traditional interfaces, which impose rigid microarchitectural constraints, our proposed design supports multiple decoupled architectures, including flexible instruction decoding schemes and versatile accelerator CSR management strategies, improving compatibility with diverse accelerator designs and system configurations.

Components

- **Instruction Issuance and Response**
  - **Issue** coordinates instruction delivery and status feedback. Supports multiple decode schemes and omits scalar-operand addresses if the first-stage decoder resides on the processor.
  - **Scalar** transmits scalar operands and incorporates a flexible CSR management mechanism, relaying CSR data when centralized in the processor.
  - **Commit** handles instruction cancellation during mispredicted speculative execution.
  - **Completed** returns execution results to the processor and shares CSR status under centralized management.
- **Virtual Memory**
  - **Translation** provides PTEs or physical addresses to accelerator, supporting private TLB or direct MMU access, and reports memory access exceptions to the processor.
- **Data Access and Movement**
  - **Mem_fast** and **Mem_far** respectively enable rapid access to L1 and remote L2/L3 caches or memory, both leveraging ID signals to support out-of-order memory accesses.
  - **Extension** facilitates partial access to register data within the accelerator by conveying data and index information.
  - **Move** handles data movement between coprocessors by transmitting addresses, indices and synchronous signals.

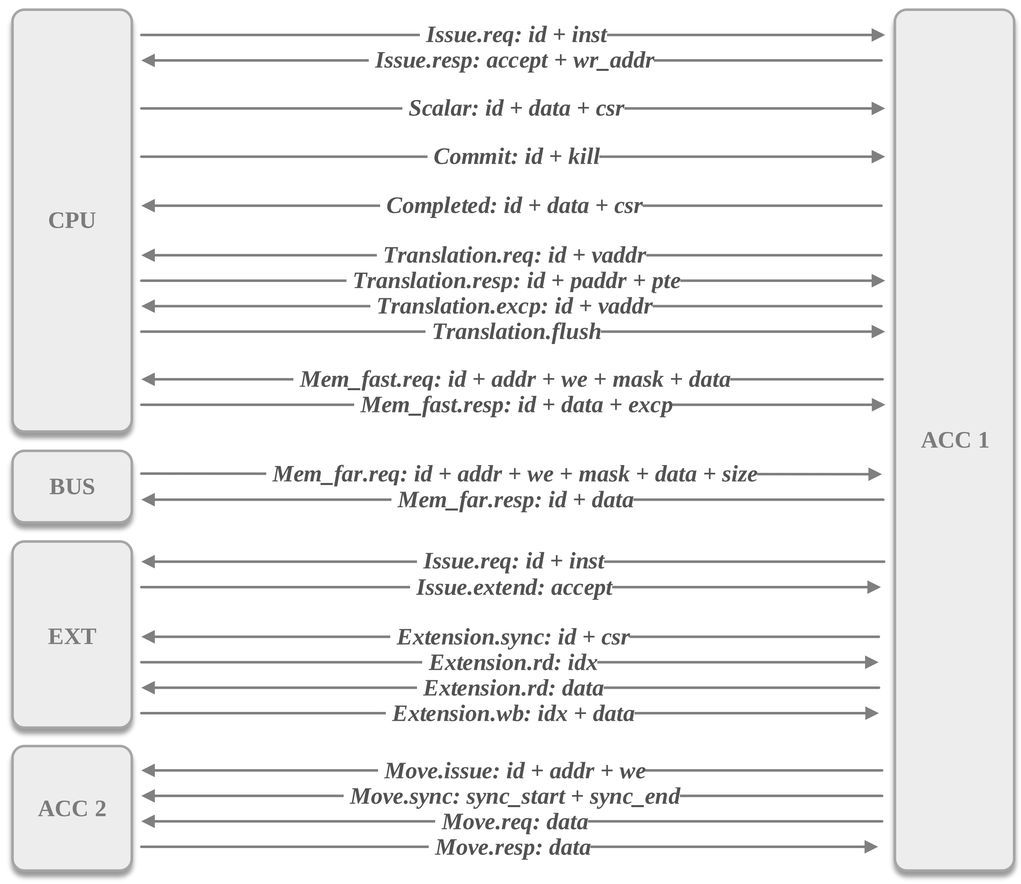

Figure 1: Overview of signals interfacing with accelerator 1.

Conclusion

- In this paper, we have proposed a flexible, open AI accelerator interface that supports a variety of RISC-V extensions, diverse data access, virtual memory mechanisms, and decoupled microarchitectural designs via flexible decoding and CSR management.
- While the interface exhibits strong theoretical scalability, adaptability, and performance, practical validation is still required. Future efforts plan to prioritize design refinements and real-world testing to assess its efficacy.

References

[1] K. Asanovic, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio et al. "The Rocket Chip Generator". In: EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17 4 (Apr. 2016), pp. 6–2.

[2] OpenHW Group. OpenHWGroup Specification: Core-V eXtension Interface (CV-X-IF). OpenHW Group, 2024. [Online]. Available: https://docs.openhwgroup.org/projects/openhw-group-core-v-xif/en/v1.0.0/

[3] Semidynamics. AVISPADO - VPU Interface. GitHub, 2021. [Online]. Available: https://github.com/semidynamics/OpenVectorInterface