RISC-V Virtualization for a CVA6-based SoC

Bruno Sá, University of Minho
Luca Valente, University of Bologna
José Martins, University of Minho
Davide Rossi, University of Bologna
Luca Benini, ETH Zurich
Sandro Pinto, University of Minho

Abstract

In this work, we describe the implementation of the latest version of the RISC-V Hypervisor extension (v1.0) specification in a RISC-V CVA6-based (64-bit) SoC. We also report the results of performing an extensive evaluation on the current design and we share our experience about the design space exploration for a few microarchitectural optimizations to the memory subsystem. To complete, we have also enhanced the timer infrastructure by implementing the privileged timer Sstc extension. All these efforts we conducted in an attempt to improve performance without compromising area and power.

CVA6 Hypervisor extension

- Compliant with the RV64 Hypervisor extension v1.0
- Nested-MMU support for SV39x4 and VMIDs
- Support for Hyp Instructions and tinst
- Functional validation with **Bao**, Xvisor, and KVM

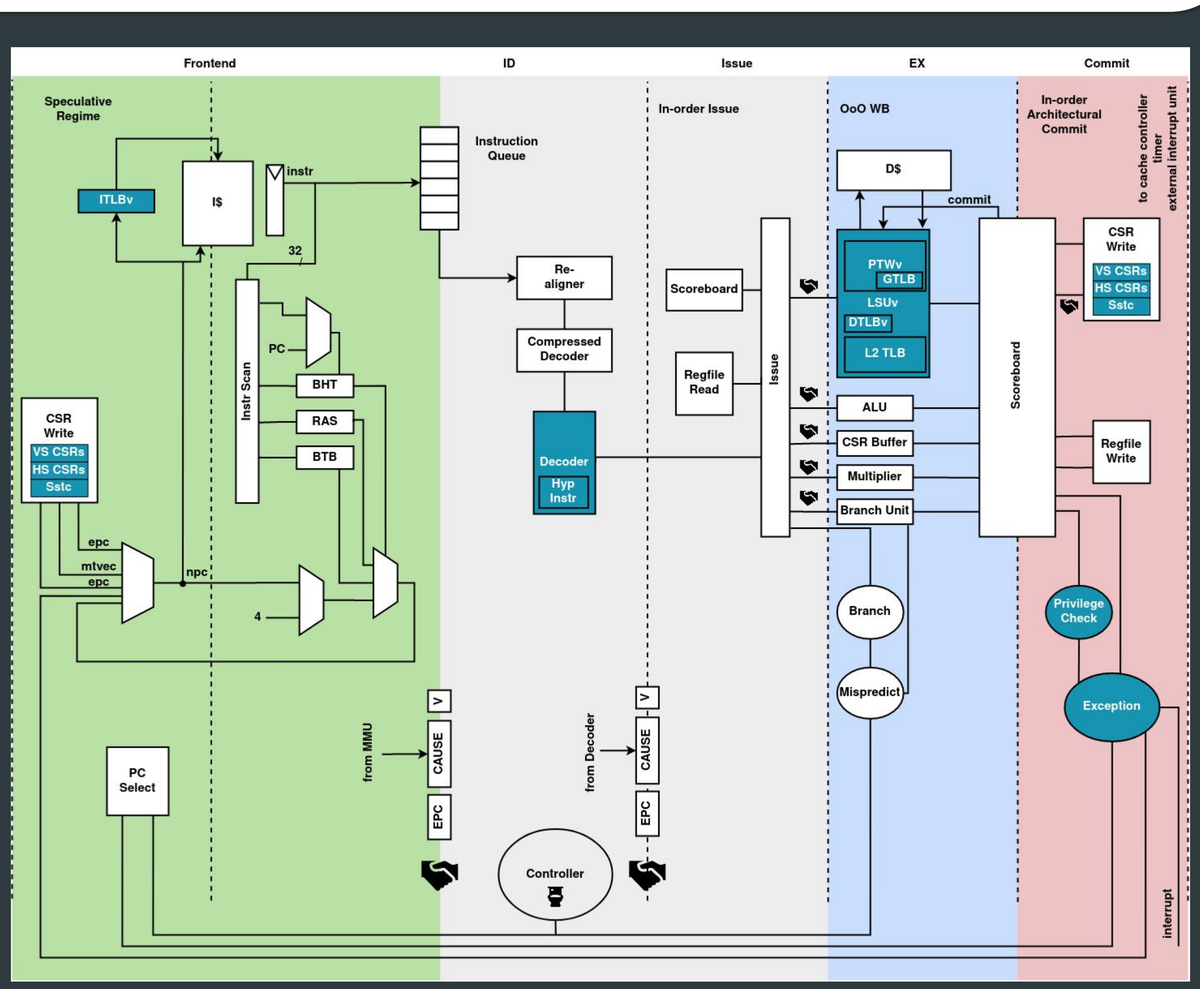
µArch Optimizations

MMU Subsystem

- G-stage TLB for VS-Stage intermediate translations
- Shared L2 I/DTLB with 4K and 2M (optional) page support

Timer Subsystem

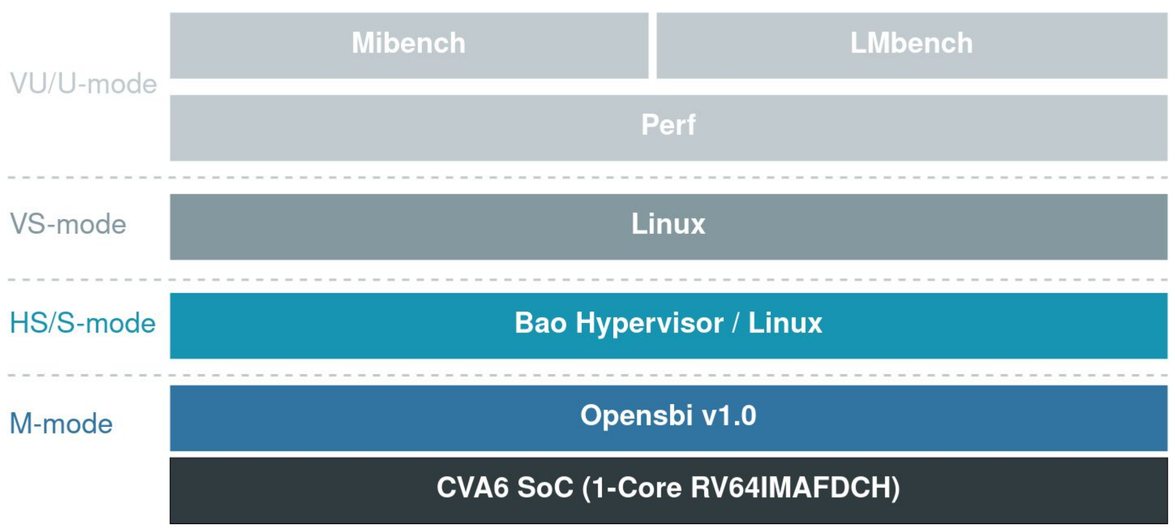
- RISC-V Sstc extension. Timers can be accessed directly from S/HS and VS-modes



Evaluation

- Benchmarking with Mibench and LMbench
- Up to 20 combinations of possible optimizations configurations evaluated:
  - Increase the number of L1 D/I TLB entries
  - GTLB with different number of entries (8 or 16)
  - L2 TLB with different number of entries, page size support (only 4K or 2M or both), and associativity
  - Timer privilege Sstc extension
- Collected microarchitectural events using perf

System Setup



Non-Optimized Core

- Average relative performance overhead of 8% for Mibench
- Worst-case scenario of 14% for the susanc-small
- Hardware resources overhead of ≃ 6% (LUTs and Regs)
- LMbench shows an increase of 640 ns in TLB miss penalty
- Two major sources of performance degradation: i) I/D TLB misses and ii) Exceptions due to timer emulation

Optimized Core

- For the most complete scenario an average relative performance overhead of 2% for Mibench
- Worst-case scenario of 4% for the susane-small
- Hardware resources increase of ≃ 8% (LUTs and Regs) and 24% BRAMs