RISC-V Virtualization: A Case Study on the CVA6

Bruno Sá [1]*, Luca Valente[2], José Martins[1], Davide Rossi[2], Luca Benini[2,3] and Sandro Pinto[1]

[1]Centro ALGORITMI/LASI - University of Minho, DEI
[2]DEI, University of Bologna, Italy
[3]IIS lab, ETH Zurich, Switzerland

Abstract

We report our work on implementing hardware virtualization support in the RISC-V CVA6 core. Specifically, we designed microarchitectural improvements such as a dedicated G-Stage Translation Lookaside Buffer (GTLB) and an L2 TLB to reduce the virtualization performance overhead. Moreover, we conducted a comprehensive design space exploration and post-layout simulations based on 22nm FDX technology to evaluate the trade-offs among performance, power, and area. To evaluate our design, we used the MiBench benchmark on Linux atop Bao hypervisor for a single-core configuration. Our optimized design achieves up to a 16% performance speedup, with an average speedup of 12.5%, compared with a virtualization-aware non-optimized design and a minimal cost of 0.78% in area and 0.33% in power. Our work demonstrates the effectiveness of microarchitectural enhancements in addressing the virtualization performance overhead and provides valuable insights into the PPA trade-offs in designing hardware virtualization support in RISC-V processors.

Introduction

Virtualization is the *de facto* technology used to consolidate and isolate multiple systems into a single hardware platform. Nowadays, virtualization is growing in the embedded and safety-critical systems industry pushed by the market demands to reduce the size, weight, power, and cost (SWaP-C) [1, 2]. This growth has led mainstream instruction set architectures (ISAs) to introduce hardware virtualization technologies, such as RISC-V privileged ISA hypervisor extension.

In this work, we report the implementation of the support for hardware virtualization in the RISC-V open-source core CVA6 [3]. Overall we make the following contributions. First, our implementation is fully compliant with the Hypervisor extension privileged specification version 1.0 and with the RISC-V timer "*stimecmp/vstimecmp*" extension (Sstc) [1]. Second, we designed a set of virtualization-oriented enhancements to the CVA6 nested memory management unit (nested-MMU) aiming to alleviate the virtualization overheads: (i) a TLB coupled to page table walker (PTW) to store second-stage translations (i.e., GTLB in our terminology), and (ii) an second level TLB (i.e., L2 TLB). In addition, we also carried out a design space evaluation (DSE) with trade-offs on parameters from three different microarchitectural modules (i.e., L1 TLB, GTLB, L2 TLB) and their respective impact on the functional performance. Last, we selected 6 designs from the DSE and conducted post-layout simulations of implementations in 22nm FDX technology to perform the PPA analysis. Results show that the optimal design point achieves a maximum of 16% and a minimum of 8% relative functional performance speedup (approx. 12.5% on average), with a penalty of 0.78% in the area and 0.33% in power.

CVA6 Virtualization Support

CVA6 Privileged Specification Extensions. We have implemented two RISC-V extensions on the CVA6 core: (i) the ratified privileged Hypervisor extension version 1.0; and (ii) the Sstc extension to allow timers to be directly manage in S-mode and VS-mode without the firmware intervention.

CVA6 Microarchitecure otimizations. We designed the following enhancements to the MMU subsystem to improve the virtualization performance. First, a GTLB located in the nested-PTW to store guest-physical addresses (GPA) to host-physical addresses (HPA) and accelerate VS-Stage translations (i.e., the first stage of translation). Second, a second level TLB (i.e., L2 TLB) to increase the TLB reach. The L2 TLB is highly parameterizable, i.e., designers can easily configure the TLB size, associativity, and page sizes support (4KiB and 2MiB).

*Acknowledgments: This work has been supported by Technology Innovation Institute (TII), the European PILOT (EuroHPC JU, g.a. 101034126), the EPI SGA2 (EuroHPC JU, g.a. 101036168), and FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope UIDB/00319/2020 and Scholarships Project Scope SFRH/BD/138660/2018 and SFRH/BD/07707/2021. Corresponding author: id10037@alunos.uminho.pt

[1] https://github.com/riscv/riscv-time-compare/

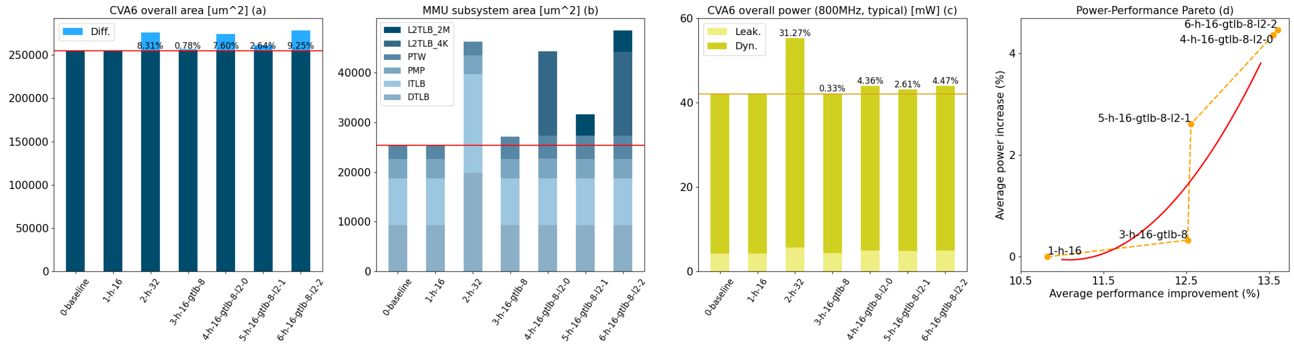

Figure 1: *Area and Power results*

Table 1: *7 selected configurations for PPA analysis in GF22nm*

	SSTC support	I/DTLB #entries	8-entries GTLB	4k L2 TLB	2MB L2 TLB
0-vanilla	x	16	x	x	x
1-h-16	✓	16	x	x	x
2-h-32	✓	32	x	x	x
3-h-gtlb-8	✓	16	✓	x	x
4-h-gtlb-8-l2-0	✓	16	✓	✓	x
5-h-gtlb-8-l2-1	✓	16	✓	x	✓
6-h-gtlb-8-l2-2	✓	16	✓	✓	✓

Results

For the evaluation, we performed two main assessments: (i) a DSE evaluation focus on the functional performance speedup; and (ii) physical implementation with a detailed PPA analysis.

Design Space Exploration: Evaluation

For the DSE evaluation, we focus on assessing the functional performance speedup of each specific module (L1 TLB, GTLB, L2 TLB, and *Sstc* extension) with different configurations. We defined a set of parameters we wanted to try out and their respective configurations for each module. For example: for the GTLB we select the number of entries as the design parameter and 8, and 16 as possible configurations. It is possible to obtain a maximum of 288 distinct combinations by accounting for all the modules, parameters, and design configurations. However, we selected and assessed only 23 design configurations. Our DSE evaluation focuses on functional performance, and for that, we used Mibench Benchmark Suite (automotive subset). Moreover, we have also collected the post-synthesis hardware utilization targeting the Genesys2 FPGA at 100MHz. Lastly, we elected 6 design configurations with the best functional performance speedup results for the PPA analysis (see Table 1).

Power, Performance, Area Analysis

For the PPA analysis, we estimate the frequency, power, and area for each configuration listed in Table 1, targeting implementation in 22nm FDX technology from Global Foundries. Figure 1 summarizes the results for the area, power, and power-performance measurements results. We observed that all design configurations reached the defined target frequency of 800MHz in the worst corner. Then, we carefully compare the area and power consumption by running a dense 16x16 FP matrix multiplication at 800MHz with warmed-up caches. Overall, we expect a small margin error less than 10% with the pre-silicon measurements. In addition, we extracted energy efficiency using the power measurements and the functional performance speedup from the DSE (not depicted here due to lack of space). In brief, we concluded that the SSTC extension has a negligible impact on the area and that the GTLB with 8 entries and Sstc support (*3-h-16-gtlb-8*) is optimal design configuration with an average functional performance speedup of 12.5% and an impact of 0.78% in the area and 0.33% in power.

Conclusion

This work presents a comprehensive study and evaluation of hardware virtualization support in the RISC-V CVA6 core. This study provides valuable insights into the design and optimization of hardware support for virtualization in RISC-V.

References

[1] José Martins et al. "Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems". In: *Workshop on NG-RES*. Vol. 77. 2020. ISBN: 978-3-95977-136-8. DOI: 10.4230/OASIcs.NG-RES.2020.3.

[2] Bruno Sa, Jose Martins, and Sandro Pinto. "A First Look at RISC-V Virtualization from an Embedded Systems Perspective". In: *IEEE Transactions on Computers* (2021), pp. 1–1. DOI: 10.1109/TC.2021.3124320.

[3] Florian Zaruba and Luca Benini. "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". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.11 (2019), pp. 2629–2640. DOI: 10.1109/TVLSI.2019.2926114.