Open-source SPMP-based CVA6 Virtualization

Manuel Rodríguez José Martins Bruno Sá Sandro Pinto

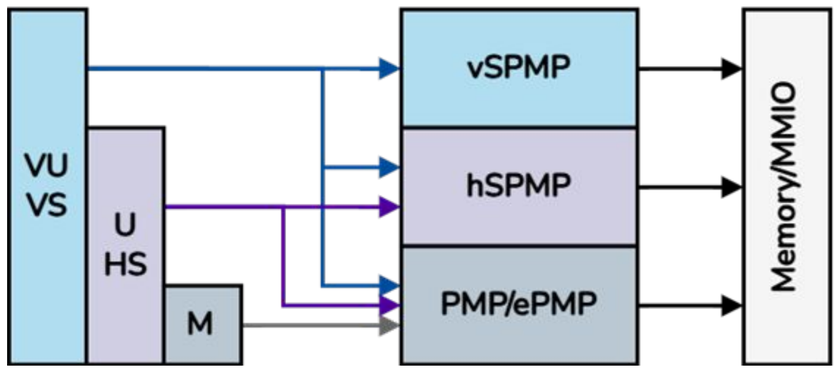
Centro ALGORITMI/LASI - UMinho, OSYX Technologies

Abstract

This work presents the first open-source design and implementation of the RISC-V SPMP for hypervisor. We integrated the dual-stage SPMP into an MMU-less 64-bit CVA6 core with the Hypervisor extension. The modified core was functionally validated using a software stack built around the Bao hypervisor. Preliminary results regarding FPGA resource utilization are also provided. Future work includes benchmarking the SPMP-based platform and comparing its performance with conventional MMU-based virtualization. Additionally, we plan to contribute to the RISC-V community by developing a QEMU model of the dual-stage SPMP.

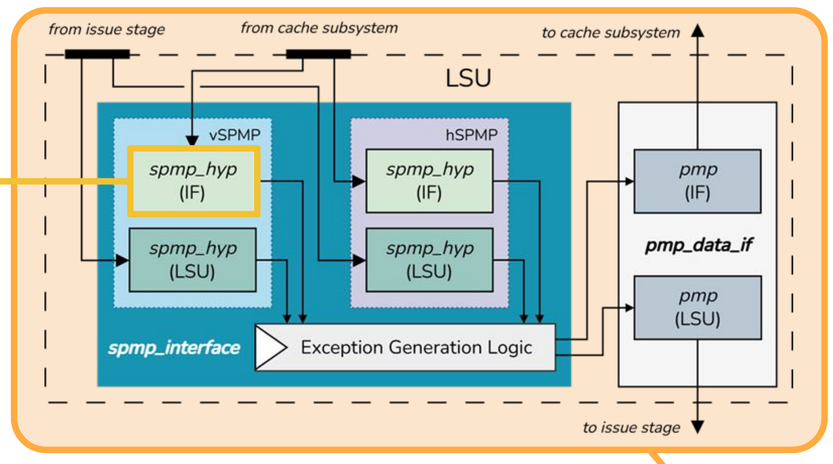
RISC-V Supervisor Physical Memory Protection for Hypervisor

- **RISC-V SPMP** → Provides memory isolation for MMU-less processors
- **Vanilla SPMP** → Isolates S-mode OSes (e.g., RTOS) and U-mode apps
  - Similar to M-mode PMP, but with S-mode and U-mode rules
- **SPMP for Hypervisor** -> Dual-stage SPMP
  - Single unified SPMP controlled by HS-mode: **hSPMP**
    → Isolates hypervisor (HS-mode) accesses from VM accesses (VS/VU)
  - Virtual SPMP controlled by VS-mode: **vSPMP**
    → Isolates guest OSes (VS-mode) from VU-mode apps



SPMP-based CVA6 Virtualization: Hardware and Software

- Hardware Baseline → CVA6 release 5.2.0 w/ Hyp extension
  - RV64IMAFDCH
  - No MMU (*CVA6ConfigMmuPresent* = 0)
- SPMP for Hypervisor integrated within the LSU
  - Dedicated dual-stage SPMPs for IF and LS requests
  - Parameterizable SPMP instantiation and number of entries
  - Fully combinatorial address matching logic
- Software → Bao hypervisor
  - First hypervisor running on a virtualization-ready, SPMP-based RISC-V processor
  - Two-layer MPU/PMP Design: VMPU and PMPU

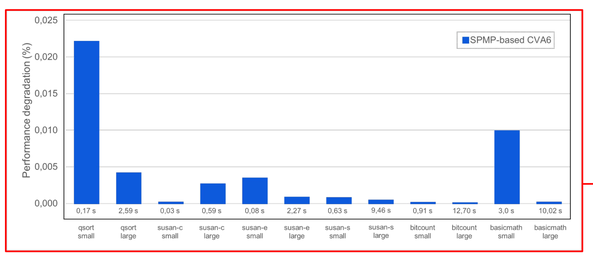


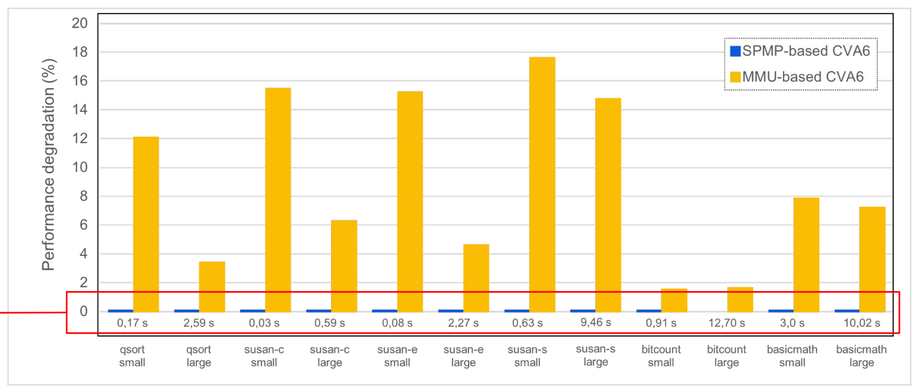
Preliminary Evaluation

- Slightly higher utilization of LUTs in (#3) LSU compared to (#4) due to SPMP matching logic
- Negligible increase in FF utilization
  - FF utilization is the same in (#2) and (#3)
  - Number of SPMP entries only affects matching logic (comb)

	CVA6 Configuration	LSU Utilization	
		Lookup Tables	Flip Flops
(#1)	Bare CVA6	**3633 (100%)**	**1951 (100%)**
(#2)	SPMP-based CVA6 w/ 32 entries	8671 (+139%)	2085 (+6,9%)
(#3)	SPMP-based CVA6 w/ 64 entries	12324 (+239)	2085 (+6,9%)
(#4)	MMU-based CVA6	11370 (+213)	7646 (+292%)

- MiBench benchmark suite performed in three configs:
  - Baremetal (*reference*), Bao+VM (SPMP), Bao+VM (MMU)
- SPMP-based CVA6 ≈ solo execution
  - Negligible overhead
  - Not similar to MMU patterns -> Noise





Universidade do Minho

osyx technologies

CROSS-ON