Evaluating IOMMU-Based Shared Virtual Addressing for RISC-V Embedded Heterogeneous SoCs

Cyril Koenig[1], Enrico Zelioli[1], Luca Benini[1,2]
[1]Integrated Systems Laboratory, ETH Zurich
[2]Department of Electrical, Electronic, and Information Engineering, University of Bologna

1. Motivation

Shared Virtual Addressing allows for **zero-copy offloading** and simplifies programming heterogeneous platforms. However, IO page walking can cause significant overhead. [1] [2]

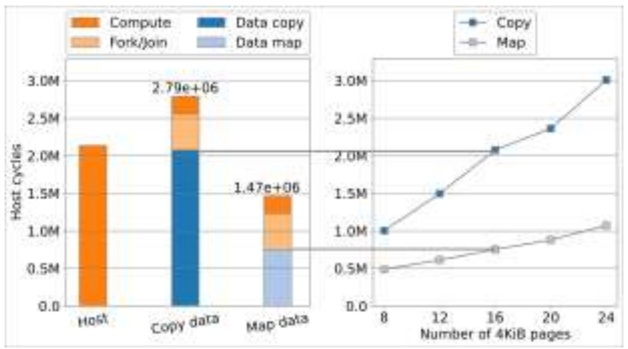
- ***We propose an open-source platform to evaluate IOMMU overhead on heterogeneous benchmarks using SVA.***
- ***We show that IOMMU overheads fall below 5% for proposed kernels when integrating a last-level cache (LLC).***

2. Zero-copy offloading

Copying data to physically contiguous device memory prevents heterogeneous accelerations of memory-bound kernels.

Creating page table entries for IOMMU is significantly faster.

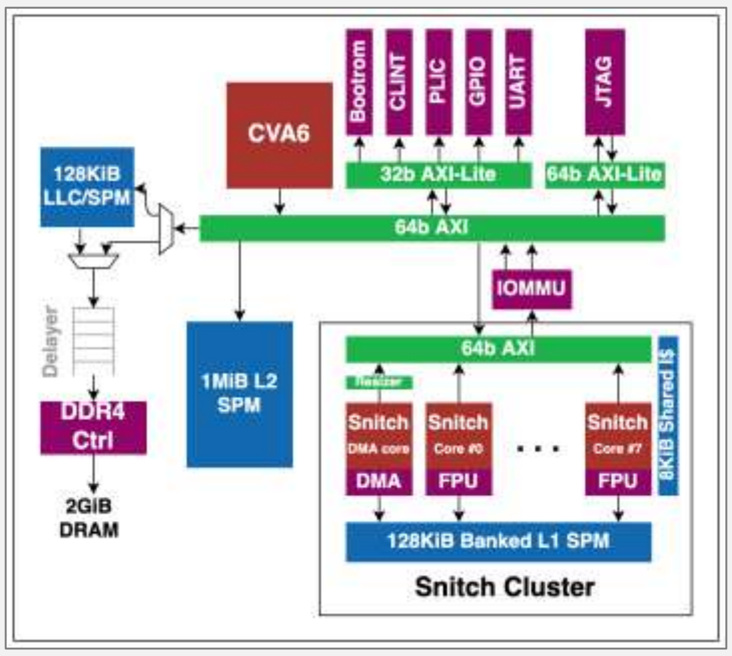
Heterogeneous AXPY with and without zero-copy



3. Platform Architecture

The proposed platform contains:

- **Linux** capable CVA6 **core**
- **Programmable Many-Core Accelerator** (PMCA)
- **RISC-V IOMMU** [3] with four IO-TLB entries



source code and FPGA flow

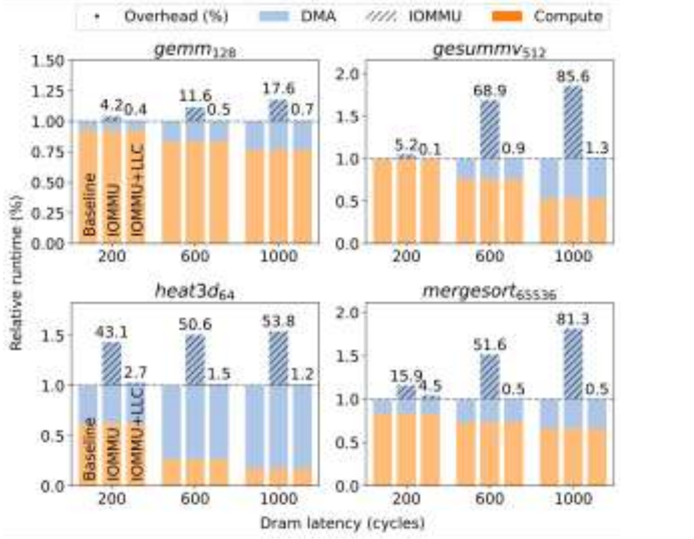
paper

A last-level cache with bypass unit allow the device DMA to fully utilize the DDR bandwidth (with appropriate SW coherency) .

A reconfigurable delayer emulates different memory latencies.

4 IOMMU Overhead Evaluation

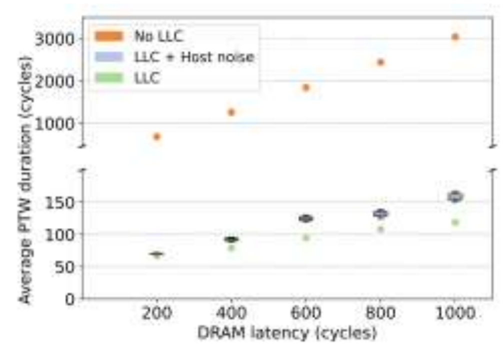
Accelerator runtime (using double-buffering) with IOMMU overhead at different DRAM latencies



Even on compute bound kernels (GeMM), IO page walking can increase accelerator's runtime by 17.6% under high DRAM latency. Memory bound kernels (GeMV) can face even much larger overheads: 85.6%

Previous work [1] [2] propose architectural changes in host and device MMUs to face this issue. We show that adding a shared last-level cache suffices to reduce this overhead below 5%.

Even with interfering host traffic. We show that thanks to the LLC, average page table walking time is greatly reduced: from 3000 cycles to 150 cycles in high latency memory systems.



4 Conclusion

- The **platform RTL** is **available on Github** for further research on shared virtual addressing and page table walking overhead.
- Our study shows that **last-level caches are a key enabler to** heterogeneous acceleration with **SVA**, reducing IOMMU overhead below 5% of the accelerator's runtime.
- Scratchpad-based accelerator that typically exploit DRAM with **DMA engines** can rely on SW coherency and **LLC bypass**.

References

[1] Y. Hao, et Al. "Supporting Address Translation for Accelerator-Centric Architectures"

[2] Fu et Al. "Active Forwarding: Eliminate IOMMU Address Translation for Accelerator-rich Architectures"

[3] M. Rodríguez, et Al. "Open-source RISC-V Input/Output Memory Management Unit (IOMMU) IP"

pulp-platform.org pulp-platform @pulp_platform cykoenig@iis.ee.ethz.ch