RISC-V MPU - Address Space Isolation for Latency Critical and/or Resource Constrained Systems.

Alexey Khomich[1], Nigel Topham[1] and Paul Stravers[1]*

[1]Synopsys Inc.

Abstract

The Memory Protection Unit (MPU) provides efficient address protection and address translation capabilities for RISC-V cores. It is designed to replace a traditional MMU in resource constrained and/or latency critical systems. The MPU allows multiple stages of operation required for virtualization use cases and compatibility with the RISC-V Hypervisor Extension. Efficient operation is achieved by replacing the memory-based address translation table of an MMU with core local configuration storage allowing flexible definition of memory region boundaries and fixed, low-latency address lookup.

Introduction

Nowadays there are two major memory address isolation mechanisms used by software designs in different scenarios. At the low end there is the physical address protection suitable for simple embedded cases, including real-time systems. Higher end designs traditionally use paged virtual memory and do not include real-time cases. These approaches to address space isolation have significantly different functionality, interfaces, programming models and hardware/software complexity. The difference in hardware and programming models limits the ability to adopt software for different hardware which becomes critical in case of virtualization and/or mixed criticality.

The proposed Memory Protection Unit architecture is targeted to address the issues above and close the distance between different use cases providing efficient configurable hardware that is capable of mixing address protection and translation using the same programming model.

Memory Protection Unit Architecture

Like traditional paged memory, the Memory Protection Unit provides memory address isolation and (optional) translation for S/U-mode and VS/VU-mode memory accesses. S/U-mode address isolation requires a single stage (L1), whereas VS/VU-mode accesses are isolated by two stages, L1 and L2, managed from the virtual environment and the hypervisor respectively. The first L1 stage of address protection provides the same functionality and programming model in normal and virtual privilege mode. The L2 stage allows hypervisor to isolate the virtual machine address spaces.

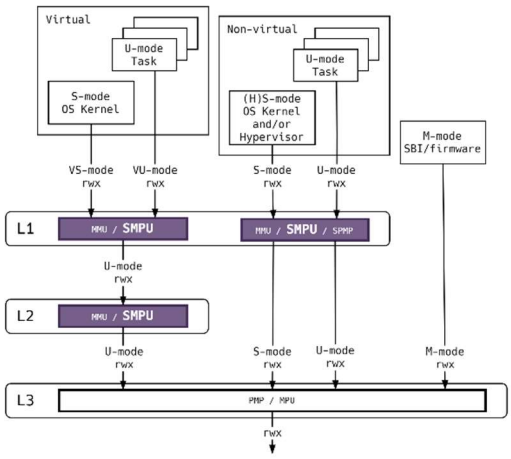

Figure 1. MPU Position at Different Address Isolation Stages

The MPU allows to combine the fine grain address protection and (optional) translation at all levels preserving the predictability of memory operation address matching. This is achieved by limiting the configuration space and keeping it in local storage instead of external memory as in the case of a traditional MMU. However, in designs with less criticality, the MPU may coexist and be fully compatible with a traditional MMU, either at the different address isolation stages or as a run-time switchable module.

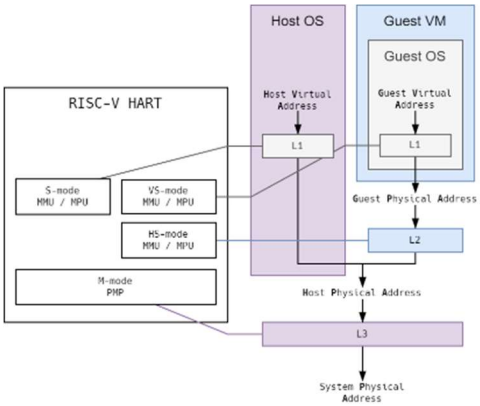

Figure 2. MMU/MPU Coexistence at Different Isolation Stages

The MPU functionality includes mandatory address protection, optional address translation and physical address extension features. All optional features can be independently configured for each isolation stage (L1 and L2). This allows various software architectures, from bare metal S-mode applications with simple address protection to feature rich operating systems with intensive memory allocation.

Except for some differences in the approach to configuring the MPU, software may rely on the same programming model as traditional PMP in simple designs or MMU in more complex cases. This includes, for example, the ability to emulate MMU functionality considering the MPU traps as TLB miss events and implementing the address translation table walker in software.

Memory Protection Unit Configuration Interface

The MPU shares the same configuration interface for protected regions and translated pages, consuming 2 iCSRs for each region/page and allowing either fine grain base/limit or NAPOT virtual/physical page number configuration. Both protected region and translated page share the same set of memory access attributes in a form of URWX flags. The MPU supports configurable translated page sizes, allowing efficient encoding of the configuration space.

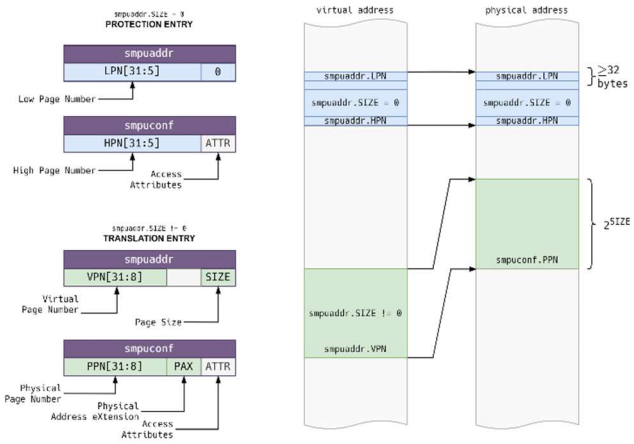

Figure 3. MPU Address Region/Page Configuration (RV32)

As the memory space configuration is part of switchable software context in many scenarios, the MPU design pays special attention to optimize either entire or part of the memory isolation stage configuration. This also includes the atomic enable/disable and safe configuration entries update during run-time.

References

[1] The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA.

[2] The RISC-V Instruction Set Manual, Volume II: Privileged Architecture.