Xen-Se

A Secure Extended Isolation Hypervisor For Trusted RISC-V Hardware

Authors: Zhixuan Xu, Xinlai Wan and Zhangxi Tan
RISC-V International Open-Source Lab, Tsinghua University

I. Virtualization and RISC-V

With virtualization widely deployed in modern cloud computing platforms for its natural of isolating shared hardware resources and provide security features. How underlying hardware helps virtualization to achieve better performance and security has become one concern of ISAs. VT-x, AMD-V in X86 and Virtualization extension in ARM are all hardware facilities improving virtualization.

Such efforts in RISC-V are of Hypervisor extension, IOMMU and AIA (Advanced Interrupt Architecture), defined specifications for hardware implementations to form a high performance, secure virtualization system.

Hypervisor extension defined virtualization bit in hardware context to indicate the virtualization status. A RISC-V with virtual bit set is in virtual world to access virtual resources with same ABI in normal world.

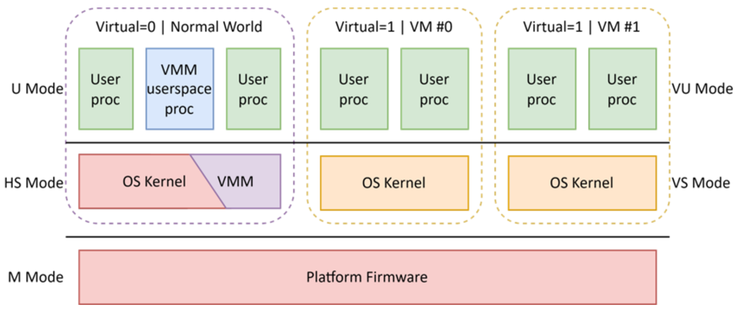

Fig.1 RISC-V Hypervisor extension overview

This virtual resources includes dedicated CSRs and a two-stage translation on memory access. In detail, the origin Supervisor mode in normal world is Hypervisor extended and in virtual world become Virtual Supervisor mode. The two-stage translation added an HS mode managed translation stage from guest physical address to supervisor physical address.

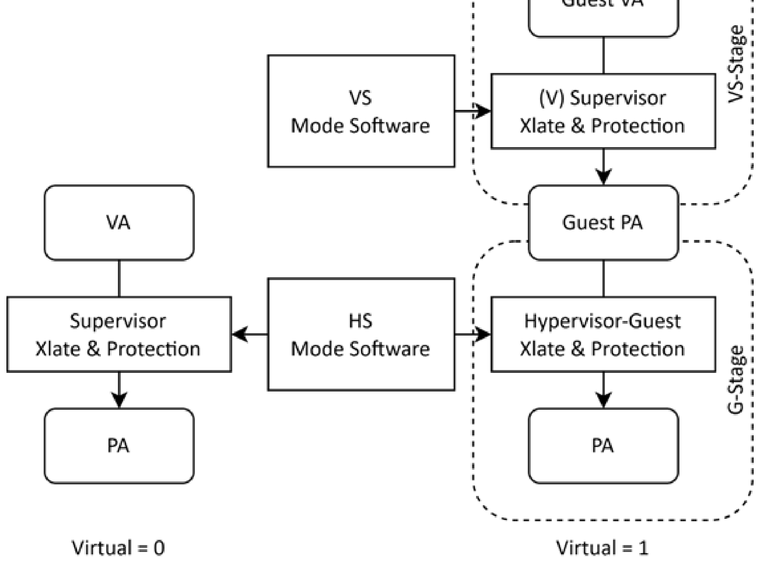

Fig.2 RISC-V 2-stage translation

Virtualization of CPU and memory are the major purpose of H extension and it also provides a natural isolation between virtual machines and the hypervisor. Each VM has its own hardware resources and context of CPU and memory space in the same time.

Another topic of virtualization is I/O virtualization which is defined by the IOMMU and AIA specification in RISC-V.

IOMMU uses the mechanism same as in CPU to translate inbound memory access from I/O devices. Foreach devices with DMA capability, the IOMMU can be configured to translate memory transactions from that device and apply protections, allow the device to access a specific memory region of a VM or a process.

AIA on the other hand handles interrupt delivering. A full featured AIA implementation uses MSI to delivery interrupts from a device to a specify privilege mode in a RISC-V HART.

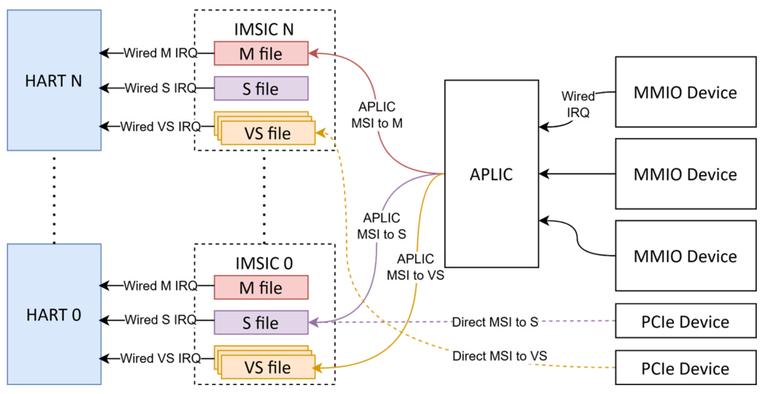

Fig.3 RISC-V full featured AIA system overview

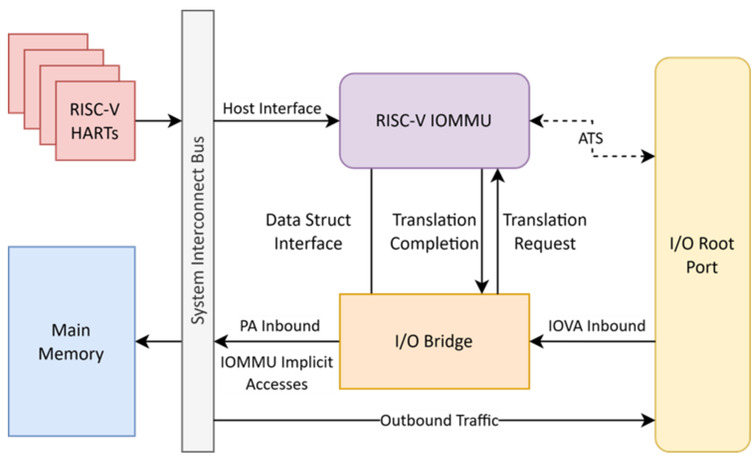

Fig.4 RISC-V IOMMU placement in system

II. Security Considerations & Design

Although hypervisor extension provides natural isolation in RISC-V core and IOMMU provides isolation for system devices, vulnerabilities that lay in software or hardware may be exploited to compromise the isolation. To enhance the security of RISC-V virtualization system, we used same mechanism of enclaves to provide extra protection for virtual machines.

To utilize the PMP in RISC-V, we designed memory tracker and I/O devices tracker to track memory and I/O device assignment to form configuration source of underlying hardware protection mechanisms.

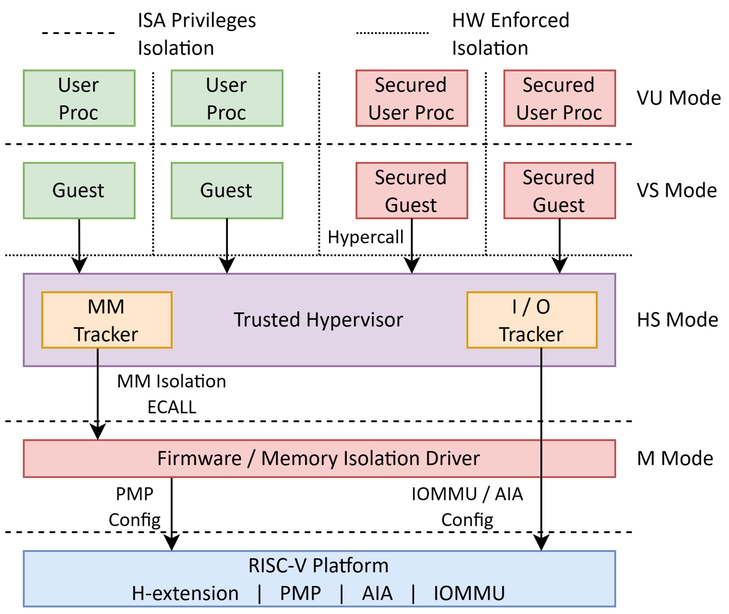

Fig.5 Framework of proposed system

Design of MM tracker and I/O tracker is pretty simple. A memory tracker is attached to each guest that tracks every memory region assigned to it, a memory region that has physical memory bank or device behind it is not allowed to be overlapped while other regions that are not mapped for emulated devices or hypervisor interface can overlap each other. I/O tracker is global unique that tracks each devices in the system, an I/O device could be: 1) simple read only device to bypass all security policies; 2) shared between guests to have only access to emulator region and interrupt to HS mode; 3) private of a secure guest to have access to guest memory and interrupt to VS mode.

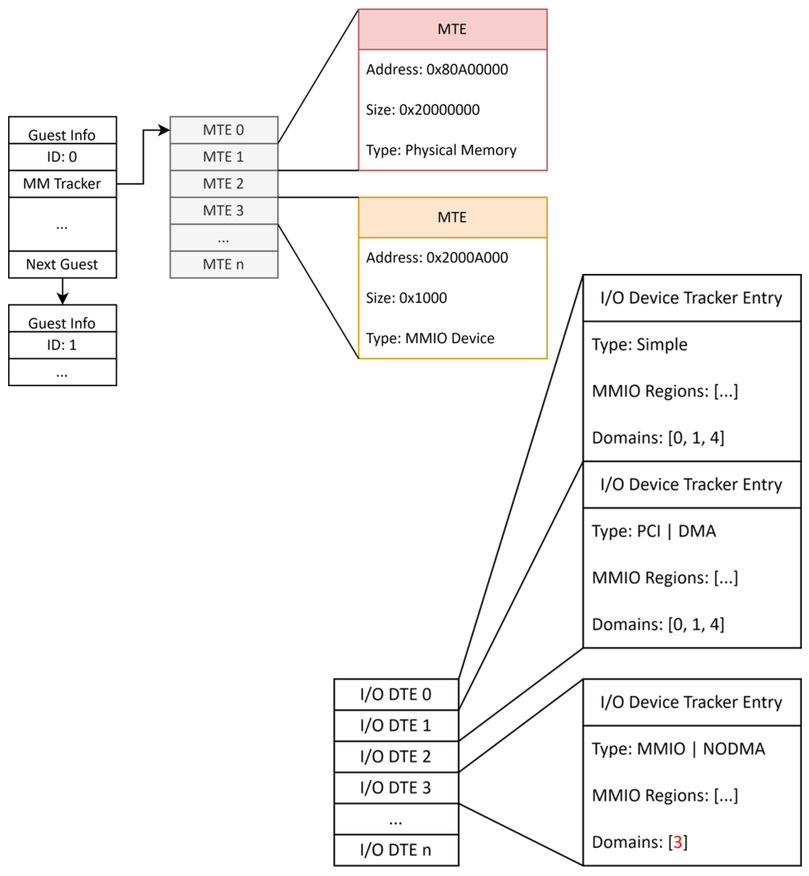

Fig.6 Design of MM and I/O trackers

III. Implementation

Our implementation is based on Xen hypervisor in developing stage, we develop and test the system on a modified QEMU with RISC-V IOMMU models, we also tried to use the developing stage UniKraft as guest in the system to reduce overhead.

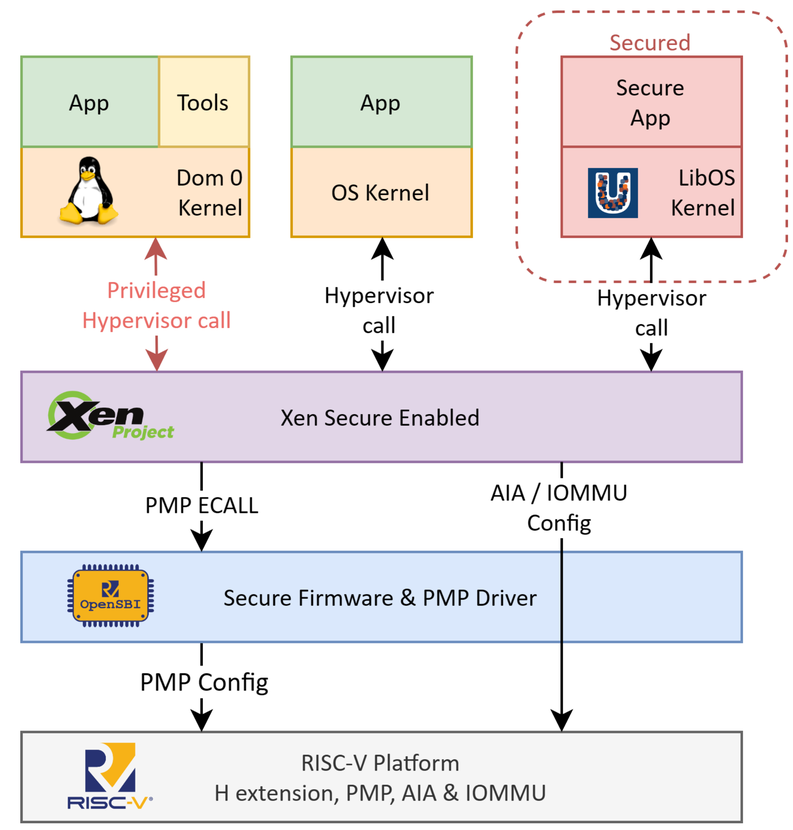

Fig.7 Implementation overview of the designed system.

zhixuan.x@rioslab.org xinlai.w@rioslab.org xtan@rioslab.org