RIMT - ACPI table for RISC-V IOMMU

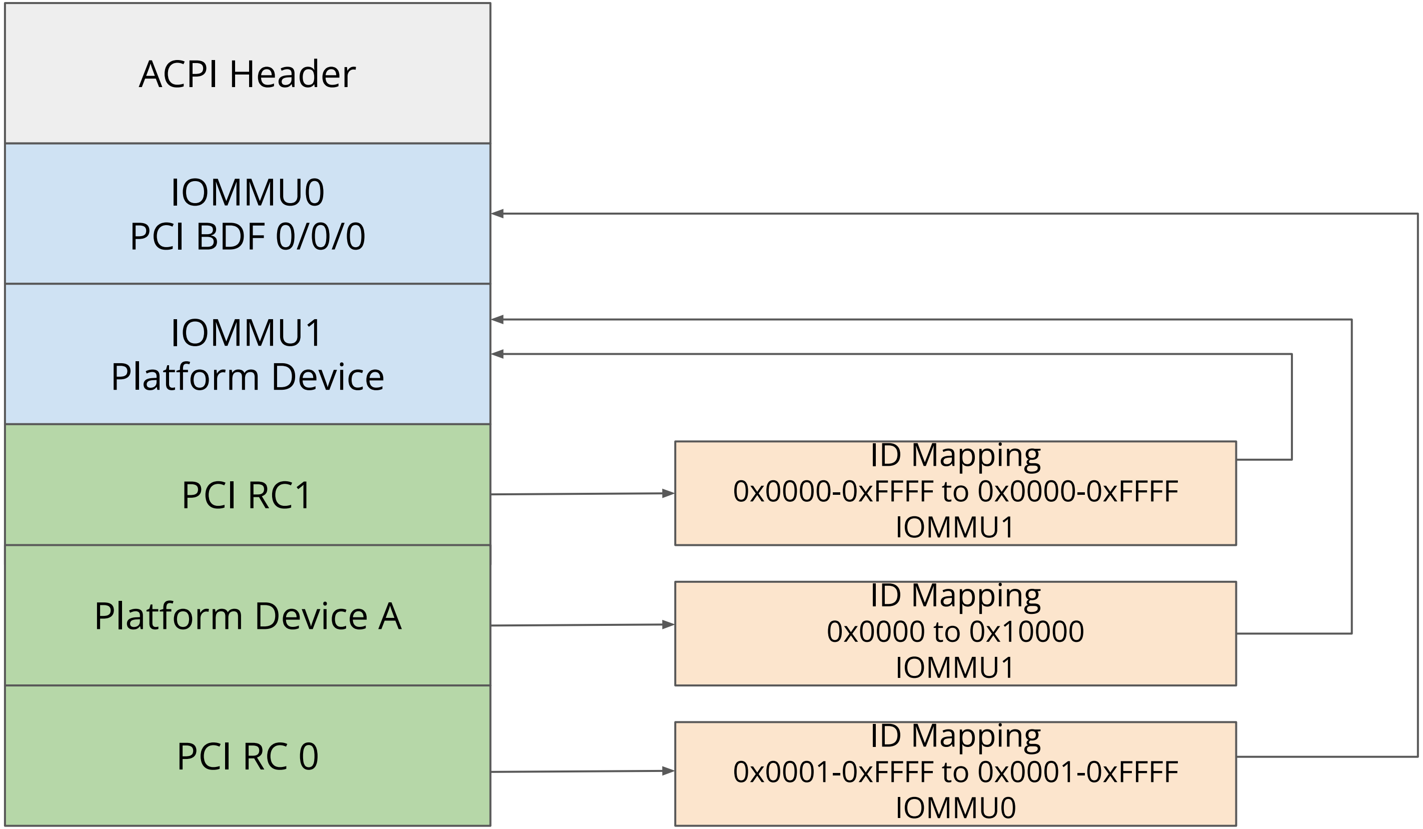
Sunil V L <sunilvl@ventanamicro.com>

09/19/2024

Background

- **RISC-V IO Mapping Table (RIMT)**
  - ACPI table to communicate IOMMU information to the OS.
  - Every CPU vendor has such table maintained by the respective organizations (outside the ACPI specification).
    - Intel - DMAR
    - AMD - IVRS
    - ARM - IORT
  - Similarly, RVI is defining the RIMT specification for RISC-V.

RIMT - Overview



ACPI Header

IOMMU0
PCI BDF 0/0/0

IOMMU1
Platform Device

PCI RC1

Platform Device A

PCI RC 0

ID Mapping
0x0000-0xFFFF to 0x0000-0xFFFF
IOMMU1

ID Mapping
0x0000 to 0x10000
IOMMU1

ID Mapping
0x0001-0xFFFF to 0x0001-0xFFFF
IOMMU0

IOMMU probe

- In ARM, IORT is parsed early during boot and platform devices are created for the SMMU. Later SMMU driver will claim those devices and initialize them.
- In RISC-V, Platform IOMMU device must exist in the namespace as well. This is required to properly order the probing of its wired IRQs.
  - No need to create the platform device explicitly by RIMT parsing code.
  - The namespace IOMMU device will have a fwnode automatically.
  - Need to link this with the RIMT node to create the devices to IOMMU mapping.

IOMMU probe (contd)

- RISC-V IOMMU can be implemented either as a PCI device or a platform device.
  - IOMMU needs a fwnode. But PCI devices will not have one in ACPI. So, fwnode needs to be created explicitly.
  - NOTE: PCI IOMMU should be implemented such that it is scanned before other devices in the same Root Complex behind the IOMMU.
- Hence, the fwnode should be registered with RIMT data structures as part of the IOMMU driver probe.
- The device drivers behind the IOMMU should support deferred probe. Should we detect and add dependencies to ensure probe order in case of platform IOMMU?

ACPI common path changes

- acpi_iommu_configure_id() calls iort_iommu_configure_id() directly from common path.
    - Proposal: call arch_iommu_configure_id() a weak function that can be implemented by different architectures.

References

- **Specification (Under internal review)**
  - https://github.com/riscv-non-isa/riscv-acpi-rimt/releases/download/v1.0.0-rc1/rimt-spec.pdf
- **Proof of Concept**
  - https://github.com/vlsunil/qemu/tree/acpi_rimt_poc_v1
  - https://github.com/vlsunil/linux/tree/acpi_rimt_poc_v1