Security Level:

IOMMU: A Detailed view

Anurup M.
Sanil Kumar D.

Nov, 2014

www.huawei.com

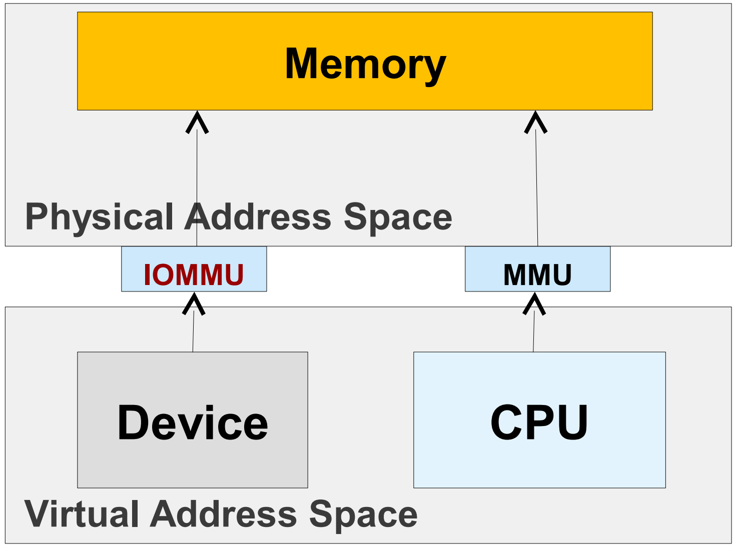
Contents

- **IOMMU Introduction**
- IOMMU for ARM
- Use cases
- Software Architecture
- Summary

Input/Output

→ Memory Management Unit

IOMMU

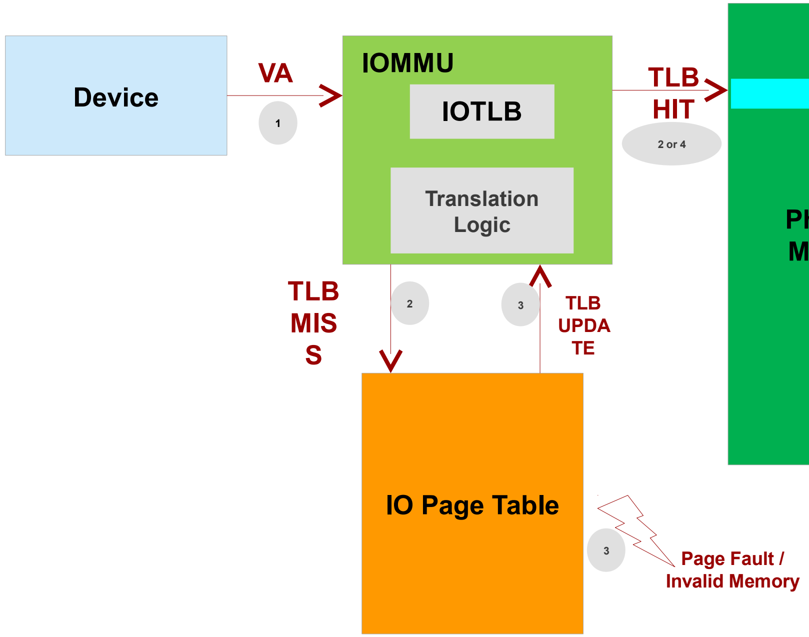


Virtual Addresses /Device Addresses → Physical Addresses

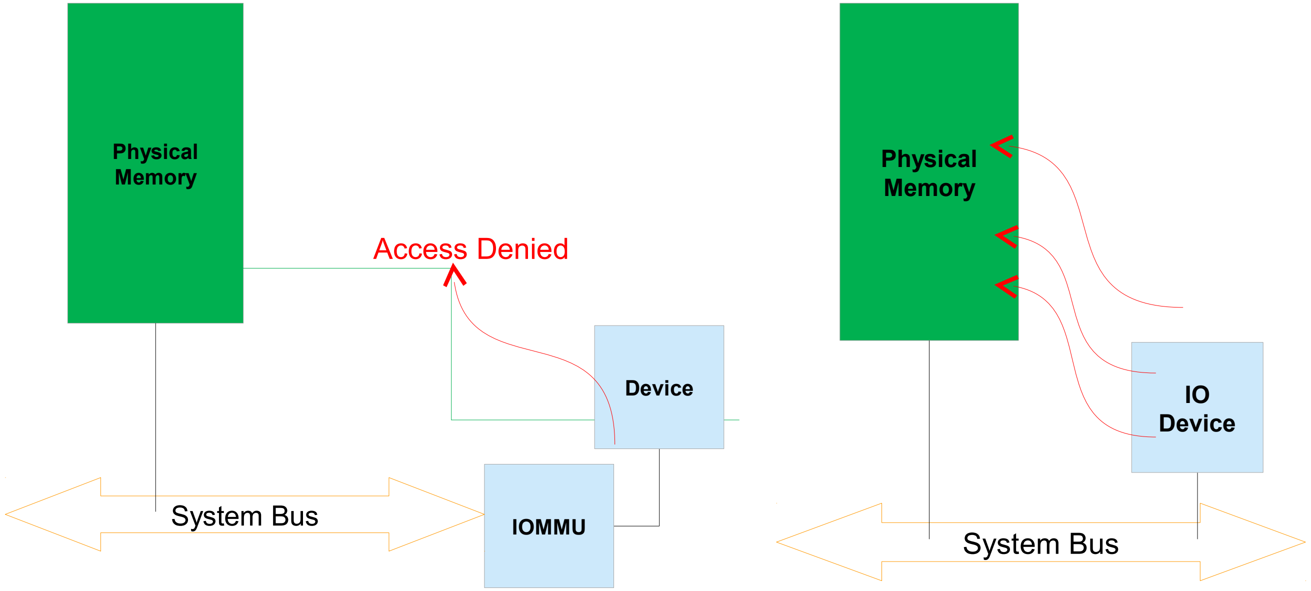
Why IOMMU ?

- Address Translation
- Isolation

Without IOMMU many devices are unable to address the complete address range supported by the host processor



IODevices can corrupt the memory without memory isolation



IOMMU provides the unique address translation for the device address to address more than its actual capability.
(This translation is independent of MMU)

IOMMU provides memory protection and enables secure memory access

IOMMU: Pros and Cons

Pros

- Large Memory Allocation; No need to be physically contiguous

- Devices can access physical memory addresses higher than 4GB (non-DAC devices as well)

- Can be programmed to make the memory region appear to be contiguous to the device on the bus .

- Device Isolation (avoid DMA attacks)

- Peripheral memory paging can be supported by an IOMMU.
  (PCIe Address Translation Services (ATS) Page Request Interface (PRI) extension can detect and signal the need for memory manager services.)

- Interrupt remapping.

Cons

- Degradation of performance from translation and management overhead (can be mitigated by a TLB)

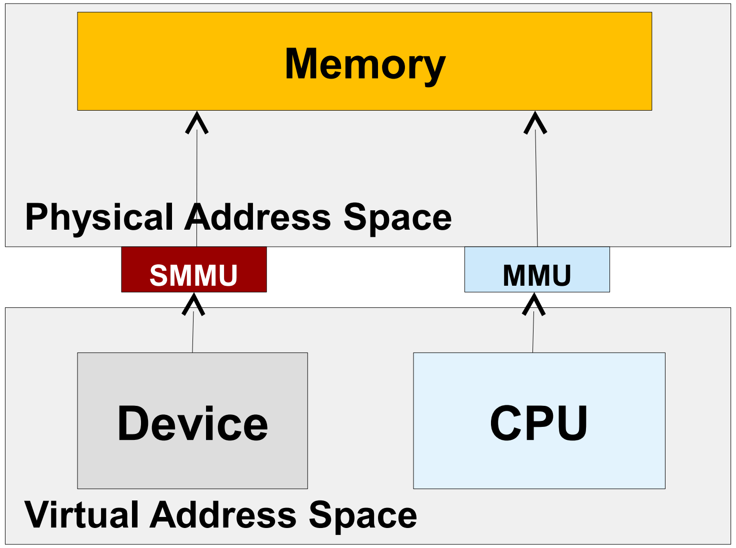
- Consumption of physical memory for the added I/O page (translation) tables. This can be mitigated if the tables can be shared with the processor.

Contents

- n IOMMU Introduction
- n **IOMMU for ARM**
- n Use cases
- n Software Architecture
- n Summary

IOMMU in ARM

- IOMMU in ARM is named as **System Memory Management Unit (SMMU)**
- Supports Address translation and isolation
- Two stage transaltion to support Virtualization
  - Stage 1, from VA (Virtual address) to IPA (Intermediate Physical Address)
  - Stage 2, from IPA to PA (Physical Address) – hypervisor will define translation tables to perform this.
- ARM releases SMMU specifications to support the implementations
  - *http://infocenter.arm.com*
  - *Currently SMMUv2 is the latest official release*
  - *The mainline Linux kernel has the SMMUv2 driver implemented*
  - *This driver currently supports (drivers/iommu/arm_smmu.c)*
    - *SMMUv1 and v2 implementations*
    - *Stream-matching and stream-indexing*
    - *v7/v8 long-descriptor format*
    - *Non-secure access to the SMMU*
    - *4k and 64k pages, with contiguous pte hints*
    - *Up to 42-bit addressing (dependent on VA_BITS)*
    - *Context fault reporting*

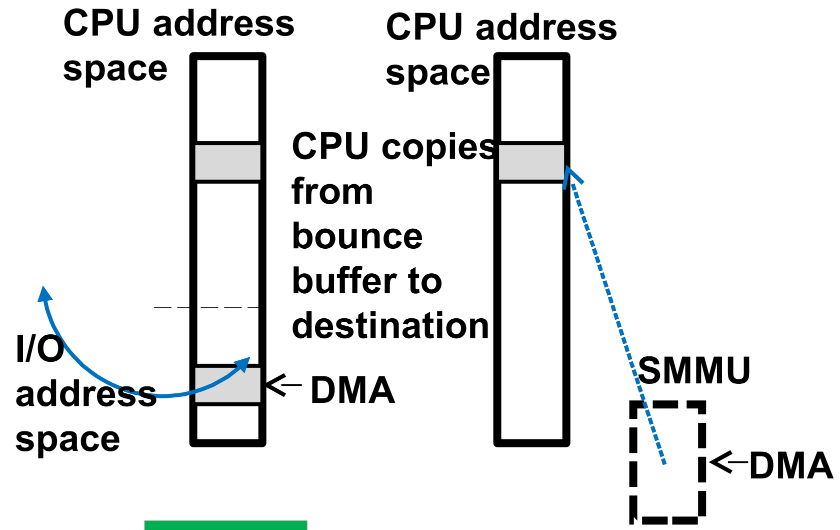


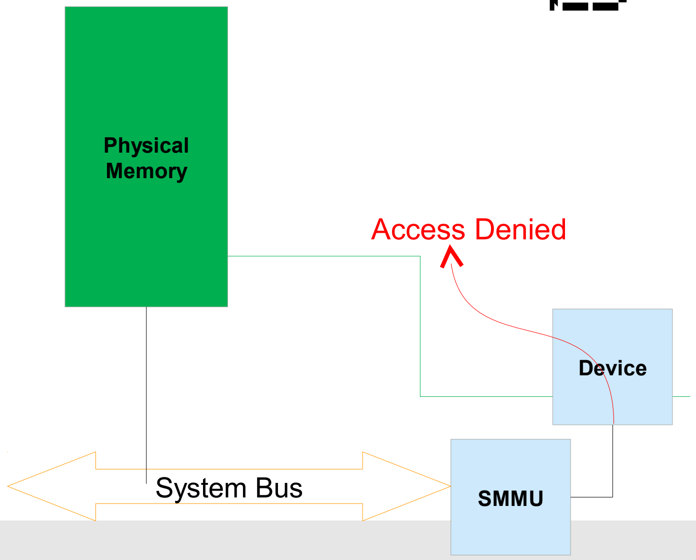
Contents

- IOMMU Introduction
- IOMMU for ARM
- **Use cases**
- Software Architecture
- Summary

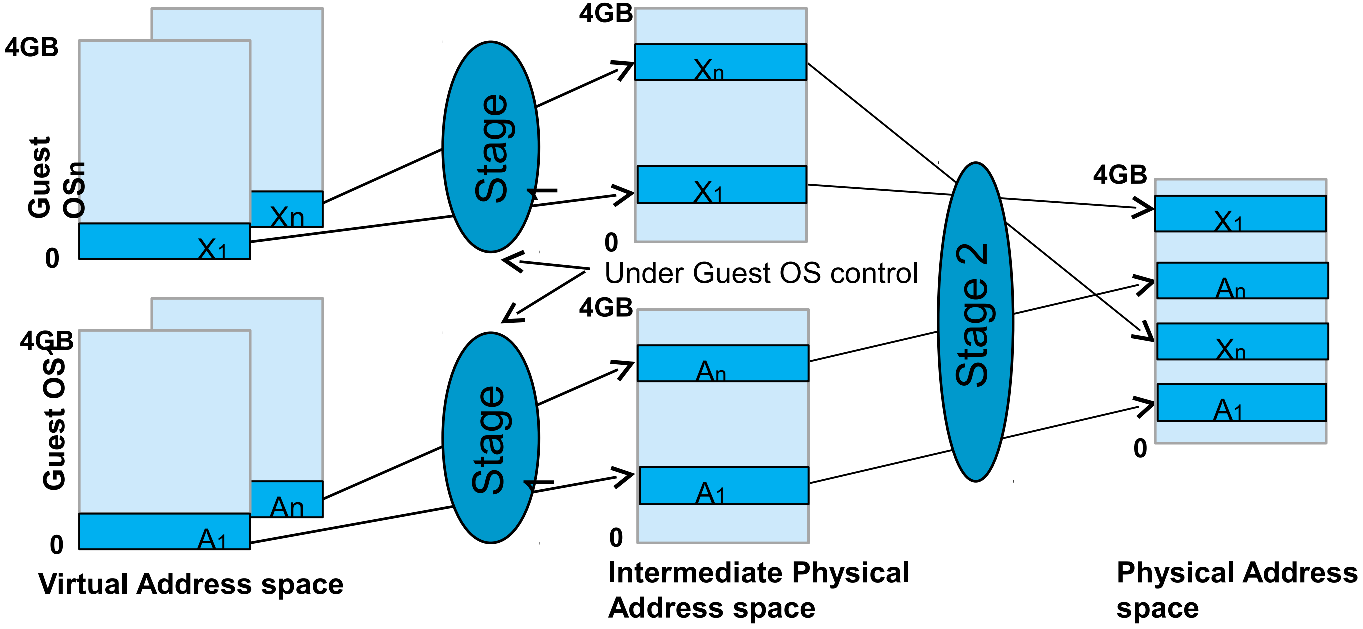
Use cases

DMA





Virtualization

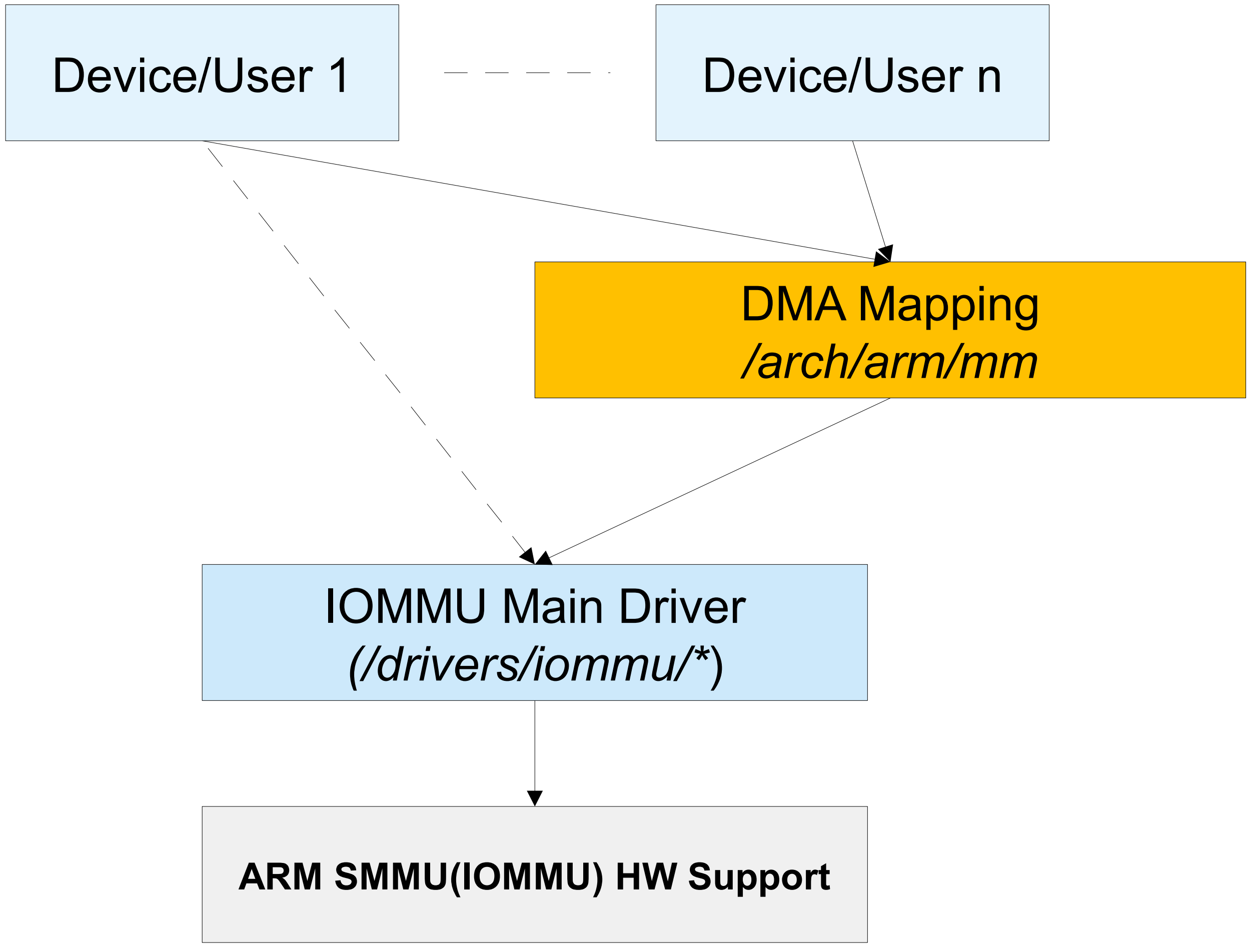


- Enables devices to address more than its addressing capability without DAC or bounce buffers
- Two stage address translation helps to manage virtual devices along with isolation
- Memory protection helps in DMA and Virtualization use cases
- Scatter gather DMA capabilities
- High performance user space drivers

Contents

n IOMMU Introduction
n IOMMU for ARM
n Use cases
n **Software Architecture**
n Summary

Software Architecture



Device/User 1

Device/User n

Clients that uses the iommu

DMA Mapping

/arch/arm/mm

Attaches clients to use iommu awared dma ops *(dma-mappings.c)*

IOMMU Main Driver

(/drivers/iommu/)*

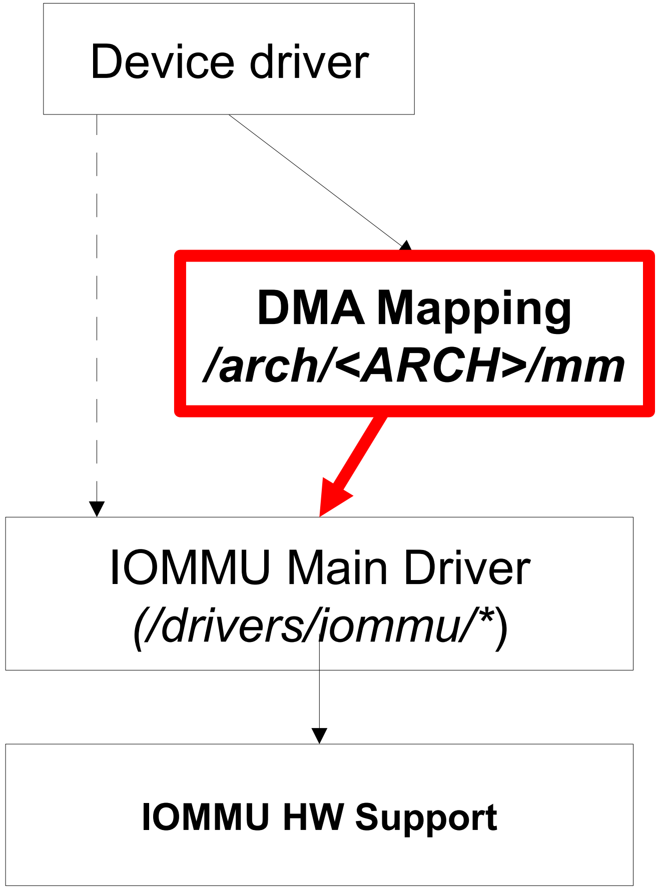
Almost all the features of IOMMU are abstracted at this layer (iommu.c, arm-smmu.c …)

ARM SMMU(IOMMU) HW Support

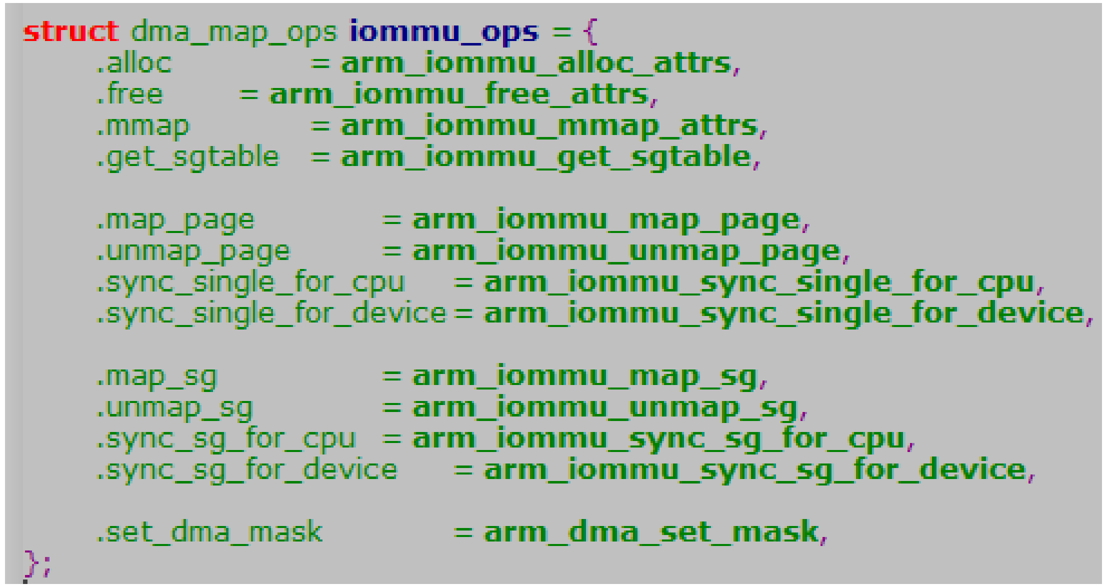
Support IOMMU support on Hardware

Code Flow: DMA API to IOMMU

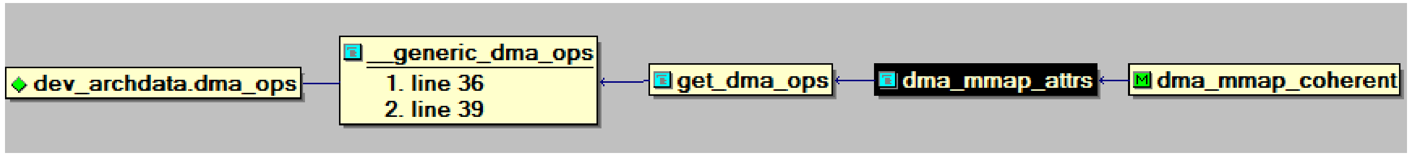
dma-mapping.c □ iommu.c □ arm_smmu.c



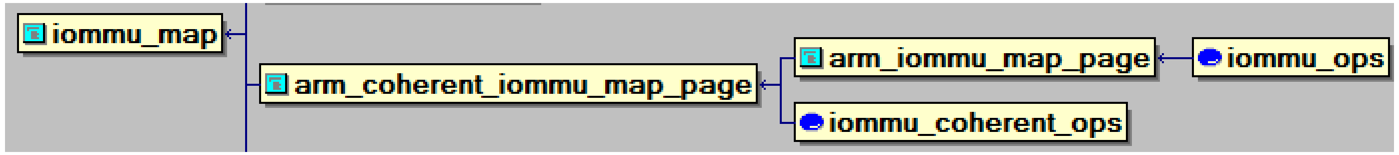
dma-mapping.c (arch\arm\mm)



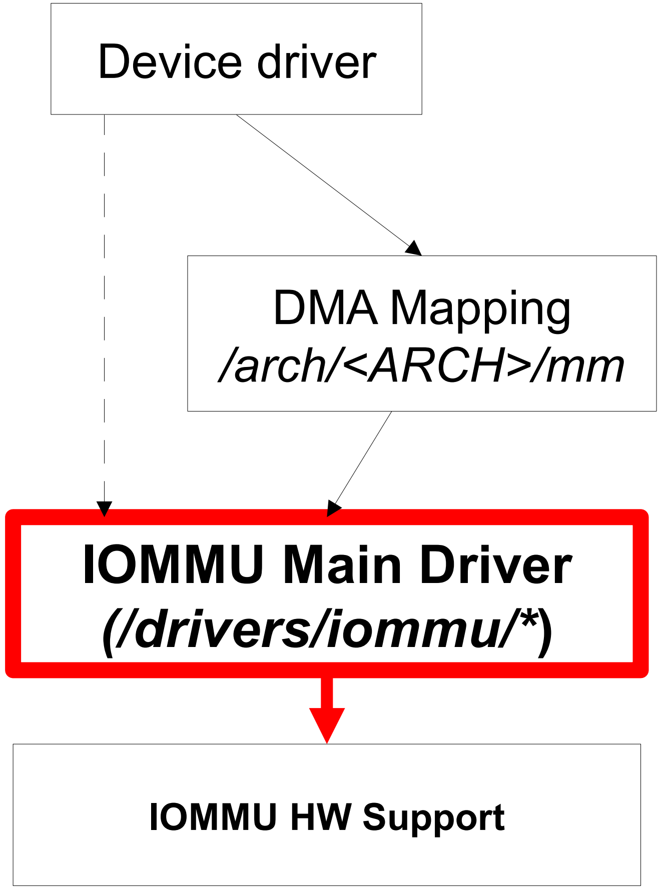
```
struct dma_map_ops iommu_ops = {
        .alloc              = arm_iommu_alloc_attrs,
        .free        = arm_iommu_free_attrs,
        .mmap               = arm_iommu_mmap_attrs,
        .get_sgtable    = arm_iommu_get_sgtable,

        .map_page           = arm_iommu_map_page,
        .unmap_page         = arm_iommu_unmap_page,
        .sync_single_for_cpu     = arm_iommu_sync_single_for_cpu,
        .sync_single_for_device = arm_iommu_sync_single_for_device,

        .map_sg             = arm_iommu_map_sg,
        .unmap_sg           = arm_iommu_unmap_sg,
        .sync_sg_for_cpu    = arm_iommu_sync_sg_for_cpu,
        .sync_sg_for_device      = arm_iommu_sync_sg_for_device,

        .set_dma_mask            = arm_dma_set_mask,
};
```



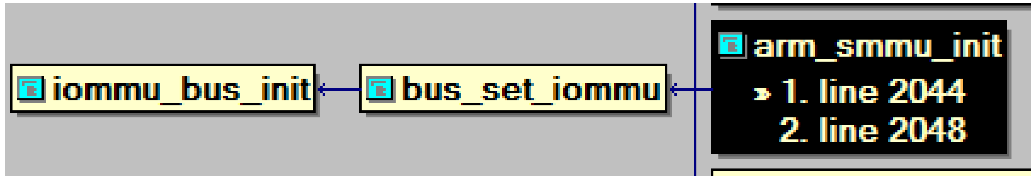
iommu.c



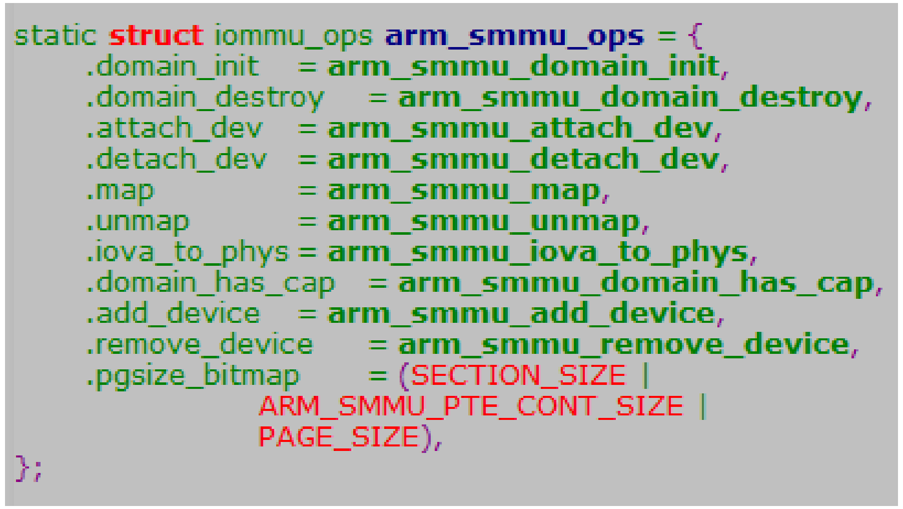
Code Flow: IOMMU to Specific IOMMU Drive r



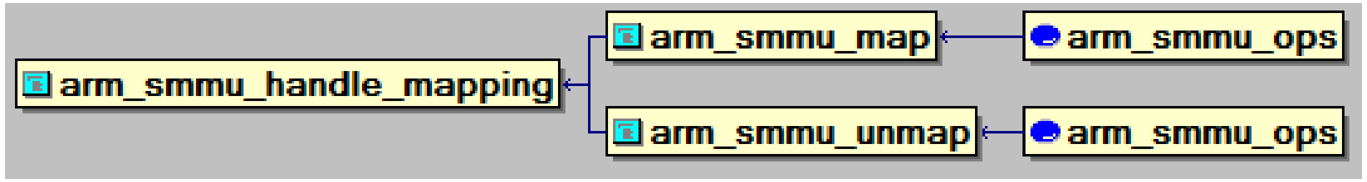
iommu.c



arm_smmu.c



```
static struct iommu_ops arm_smmu_ops = {
	.domain_init	= arm_smmu_domain_init,
	.domain_destroy	= arm_smmu_domain_destroy,
	.attach_dev	= arm_smmu_attach_dev,
	.detach_dev	= arm_smmu_detach_dev,
	.map		= arm_smmu_map,
	.unmap		= arm_smmu_unmap,
	.iova_to_phys = arm_smmu_iova_to_phys,
	.domain_has_cap	= arm_smmu_domain_has_cap,
	.add_device	= arm_smmu_add_device,
	.remove_device	= arm_smmu_remove_device,
	.pgsize_bitmap	= (SECTION_SIZE |
			   ARM_SMMU_PTE_CONT_SIZE |
			   PAGE_SIZE),
};
```



Contents

- IOMMU Introduction
- IOMMU for ARM
- Use cases
- Software Architecture
- **Summary**

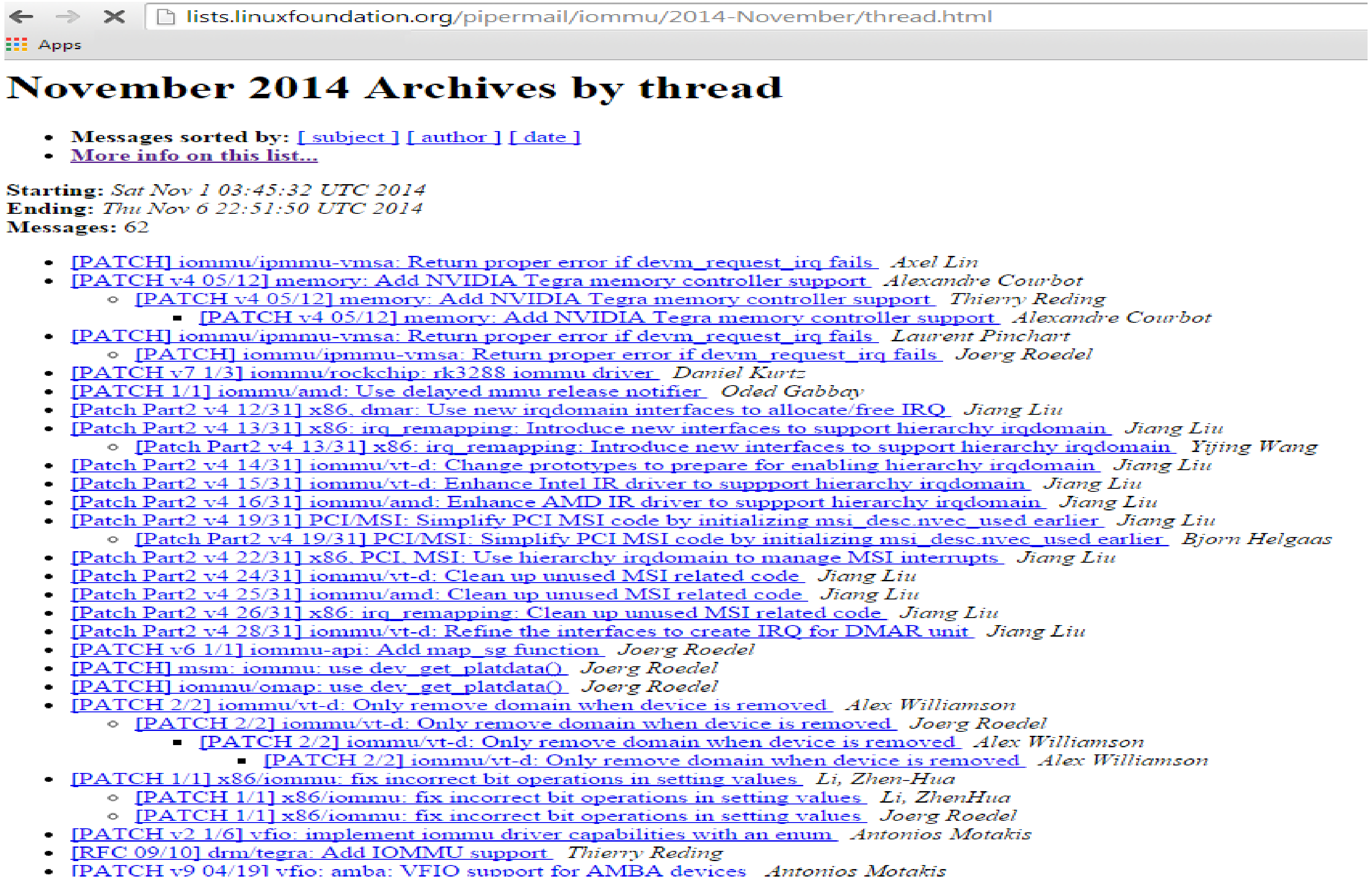
Summary

- ARM SMMU is getting refined and new versions with more features expected
- In Linux Kernel, iommu developments are active, especially for ARM SMMU
  - New drivers and features (PCIe/ATS, VFIO…)
  - Focus on Virtualization Support and extensions
  - Performance optimizations

anurup.m@huawei.com

sanil.kumar@huawei.com

IOMMU Mailing List


lists.linuxfoundation.org/pipermail/iommu/2014-November/thread.html
Apps

November 2014 Archives by thread

- **Messages sorted by:** [subject] [author] [date]
- More info on this list...

Starting: *Sat Nov 1 03:45:32 UTC 2014*
Ending: *Thu Nov 6 22:51:50 UTC 2014*
Messages: 62

- [PATCH] iommu/ipmmu-vmsa: Return proper error if devm_request_irq fails *Axel Lin*
- [PATCH v4 05/12] memory: Add NVIDIA Tegra memory controller support *Alexandre Courbot*
  - [PATCH v4 05/12] memory: Add NVIDIA Tegra memory controller support *Thierry Reding*
    - [PATCH v4 05/12] memory: Add NVIDIA Tegra memory controller support *Alexandre Courbot*
- [PATCH] iommu/ipmmu-vmsa: Return proper error if devm_request_irq fails *Laurent Pinchart*
  - [PATCH] iommu/ipmmu-vmsa: Return proper error if devm_request_irq fails *Joerg Roedel*
- [PATCH v7 1/3] iommu/rockchip: rk3288 iommu driver *Daniel Kurtz*
- [PATCH 1/1] iommu/amd: Use delayed mmu release notifier *Oded Gabbay*
- [Patch Part2 v4 12/31] x86, dmar: Use new irqdomain interfaces to allocate/free IRQ *Jiang Liu*
- [Patch Part2 v4 13/31] x86: irq_remapping: Introduce new interfaces to support hierarchy irqdomain *Jiang Liu*
  - [Patch Part2 v4 13/31] x86: irq_remapping: Introduce new interfaces to support hierarchy irqdomain *Yijing Wang*
- [Patch Part2 v4 14/31] iommu/vt-d: Change prototypes to prepare for enabling hierarchy irqdomain *Jiang Liu*
- [Patch Part2 v4 15/31] iommu/vt-d: Enhance Intel IR driver to suppport hierarchy irqdomain *Jiang Liu*
- [Patch Part2 v4 16/31] iommu/amd: Enhance AMD IR driver to suppport hierarchy irqdomain *Jiang Liu*
- [Patch Part2 v4 19/31] PCI/MSI: Simplify PCI MSI code by initializing msi_desc.nvec_used earlier *Jiang Liu*
  - [Patch Part2 v4 19/31] PCI/MSI: Simplify PCI MSI code by initializing msi_desc.nvec_used earlier *Bjorn Helgaas*
- [Patch Part2 v4 22/31] x86, PCI, MSI: Use hierarchy irqdomain to manage MSI interrupts *Jiang Liu*
- [Patch Part2 v4 24/31] iommu/vt-d: Clean up unused MSI related code *Jiang Liu*
- [Patch Part2 v4 25/31] iommu/amd: Clean up unused MSI related code *Jiang Liu*
- [Patch Part2 v4 26/31] x86: irq_remapping: Clean up unused MSI related code *Jiang Liu*
- [Patch Part2 v4 28/31] iommu/vt-d: Refine the interfaces to create IRQ for DMAR unit *Jiang Liu*
- [PATCH v6 1/1] iommu-api: Add map_sg function *Joerg Roedel*
- [PATCH] msm: iommu: use dev_get_platdata() *Joerg Roedel*
- [PATCH] iommu/omap: use dev_get_platdata() *Joerg Roedel*
- [PATCH 2/2] iommu/vt-d: Only remove domain when device is removed *Alex Williamson*
  - [PATCH 2/2] iommu/vt-d: Only remove domain when device is removed *Joerg Roedel*
    - [PATCH 2/2] iommu/vt-d: Only remove domain when device is removed *Alex Williamson*
      - [PATCH 2/2] iommu/vt-d: Only remove domain when device is removed *Alex Williamson*
- [PATCH 1/1] x86/iommu: fix incorrect bit operations in setting values *Li, Zhen-Hua*
  - [PATCH 1/1] x86/iommu: fix incorrect bit operations in setting values *Li, ZhenHua*
  - [PATCH 1/1] x86/iommu: fix incorrect bit operations in setting values *Joerg Roedel*
- [PATCH v2 1/6] vfio: implement iommu driver capabilities with an enum *Antonios Motakis*
- [RFC 09/10] drm/tegra: Add IOMMU support *Thierry Reding*
- [PATCH v9 04/19] vfio: amba: VFIO support for AMBA devices *Antonios Motakis*

Thank you

www.huawei.com

anurup.m@huawei.com

sanil.kumar@huawei.com

Reference

http://en.wikipedia.org/wiki/IOMMU - IOMMU information

http://infocenter.arm.com/help/topic/com.arm.doc.ihi0062c/IHI0062C_system_mmu_architecture_specification.pdf - ARM SMMU v2 specification

Linux kernel mainline source code 3.17