ExynosAutoV9

SYSMMU

Revision 1.00
August 2020

Application Note

SAMSUNG ELECTRONICS RESERVES THE RIGHT TO CHANGE PRODUCTS, INFORMATION AND SPECIFICATIONS WITHOUT NOTICE.

Products and specifications discussed herein are for reference purposes only. All information discussed herein is provided on an "AS IS" basis, without warranties of any kind.

This document and all information discussed herein remain the sole and exclusive property of Samsung Electronics. No license of any patent, copyright, mask work, trademark or any other intellectual property right is granted by one party to the other party under this document, by implication, estoppel or otherwise.

Samsung products are not intended for use in life support, medical, or safety equipment or any military or defense application, or any governmental procurement to which special terms or provisions may apply. Samsung products intended for automotive application are not fail-safe or error-free and that a driving assistance system or other similar system incorporation Samsung products may be prone to failures of safety-critical functions if used without proper safety technology measures in the form of hardware, software, information, and time redundancy. Accordingly, Samsung disclaims all liability arising from any and all failures of a safety-critical function caused by any error or failure of Samsung products.

For updates or additional information about Samsung products, contact your nearest Samsung office.
All brand names, trademarks and registered trademarks belong to their respective owners

Important Notice

Samsung Electronics Co. Ltd. (“Samsung”) reserves the right to make changes to the information in this publication at any time without prior notice. All information provided is for reference purpose only. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

This publication on its own does not convey any license, either express or implied, relating to any Samsung and/or third-party products, under the intellectual property rights of Samsung and/or any third parties.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

Customers are responsible for their own products and applications. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could reasonably be expected to create a situation where personal injury or death may occur. Customers acknowledge and agree that they are solely responsible to meet all other legal and regulatory requirements regarding their applications using Samsung products notwithstanding any information provided in this publication. Customer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim (including but not limited to personal injury or death) that may be associated with such unintended, unauthorized and/or illegal use.

Samsung Electronics Co., Ltd.
1-1,Samsungjeonja-ro,Hwaseong-si,
Gyeonggi-do Korea 18448

Contact Us: *myunggeun.ji@samsung.com*

Home Page: *http://www.samsungsemi.com*

Trademarks

All brand names, trademarks and registered trademarks belong to their respective owners.

- Exynos, FlexOneNAND, and OneNAND are trademarks of Samsung Electronics.
- ARM, Jazelle, TrustZone, and Thumb are registered trademarks of ARM Limited.
- Cortex, ETM, ETB, Coresight, ISA, and Neon are trademarks of ARM Limited.
- Java is a trademark of Sun Microsystems, Inc.
- SD is a registered trademark of Toshiba Corporation.
- MMC and eMMC are trademarks of MultiMediaCard Association.
- JTAG is a registered trademark of JTAG Technologies, Inc.
- Synopsys is a registered trademark of Synopsys, Inc.
- I2S is a trademark of Phillips Electronics.
- I2C is a trademark of Phillips Semiconductor Corp.
- MIPI and Slimbus are registered trademarks of the Mobile Industry Processor Interface (MIPI) Alliance.

All other trademarks used in this publication are the property of their respective owners.

Chip Handling Guide

Precaution against Electrostatic Discharge

When using semiconductor devices, ensure that the environment is protected against static electricity:

1. Wear antistatic clothes and use earth band.
2. All objects that are in direct contact with devices must be made up of materials that do not produce static electricity.
3. Ensure that the equipment and work table are earthed.
4. Use ionizer to remove electron charge.

Contamination

Do not use semiconductor products in an environment exposed to dust or dirt adhesion.

Temperature/Humidity

Semiconductor devices are sensitive to:

- Environment
- Temperature
- Humidity

High temperature or humidity deteriorates the characteristics of semiconductor devices. Therefore, do not store or use semiconductor devices in such conditions.

Mechanical Shock

Do not to apply excessive mechanical shock or force on semiconductor devices.

Chemical

Do not expose semiconductor devices to chemicals because exposure to chemicals leads to reactions that deteriorate the characteristics of the devices.

Light Protection

In non- Epoxy Molding Compound (EMC) package, do not expose semiconductor IC to bright light. Exposure to bright light causes malfunctioning of the devices. However, a few special products that utilize light or with security functions are exempted from this guide.

Radioactive, Cosmic and X-ray

Radioactive substances, cosmic ray, or X-ray may influence semiconductor devices. These substances or rays may cause a soft error during a device operation. Therefore, ensure to shield the semiconductor devices under environment that may be exposed to radioactive substances, cosmic ray, or X-ray.

EMS (Electromagnetic Susceptibility)

Strong electromagnetic wave or magnetic field may affect the characteristic of semiconductor devices during the operation under insufficient PCB circuit design for Electromagnetic Susceptibility (EMS).

Revision History

Revision No.	Date	Description	Author(s)
1.00	August 24, 2020	• Initial version of the document	Myunggeun Ji

Table of Contents

List of Figures

List of Tables

Table Number	Title	Page Number

목차 항목을 찾을 수 없습니다.

List of Examples

List of Conventions

Register RW Access Type Conventions

Type	Definition	Description
R	Read Only	The application has permission to read the Register field. Writes to read-only fields have no effect.
W	Write Only	The application has permission to write in the Register field.
RW	Read & Write	The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0.

Register Value Conventions

Expression	Description
x	Undefined bit
X	Undefined multiple bits
?	Undefined, but depends on the device or pin status
Device dependent	The value depends on the device
Pin value	The value depends on the pin status

Reset Value Conventions

Expression	Description
0	Clears the register field
1	Sets the register field
x	Don't care condition

Warning: Some bits of control registers are driven by hardware or write operation only. As a result the indicated reset value and the read value after reset might be different.

List of Terms

Terms	Descriptions

List of Acronyms

Acronyms	Descriptions

1 Introduction

1.1 Overview

System Memory Management Unit (SYSMMU) translates the Virtual Address (VA) in the transactions initiated by master IPs in a system to Physical Address (PA) that is accepted by slaves in System-on-Chips (SoCs).

Figure 1-1 illustrates that SYSMMU must be placed between VA and PA domain in an AXI-compliant interconnect.

SYSMMU is intended for those masters that do not contain an embedded MMU such as multimedia masters.

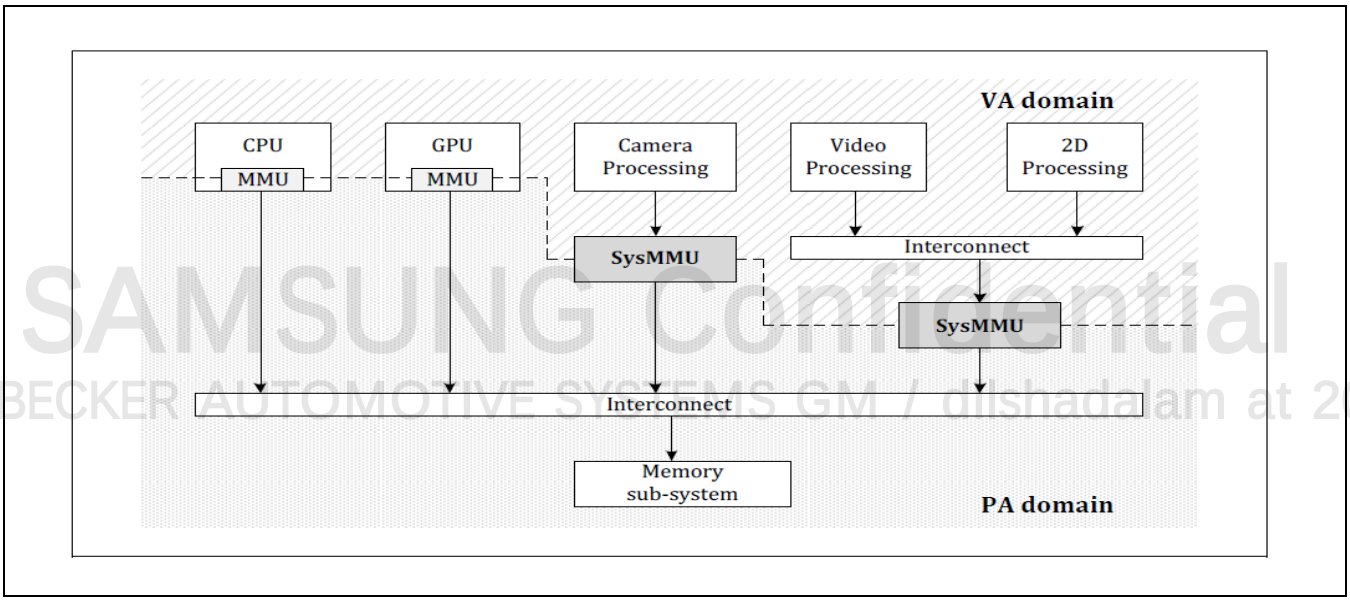

Figure 1-1 Placement of SYSMMU in SoCs

For efficient memory management, the Operating System (OS) allocates the memory resources in pages to the masters. Each page has 4 KB of memory fragment. Therefore, buffers used by masters consist of multiple distant pages. However, masters require contiguous memory view for efficient DMA implementation. SYSMMU provides a virtual contiguous memory view irrespective of the actual placement of allocated pages in the memory.

Page table consists of the mappings between virtual memory view and physical memory view. Therefore, SYSMMU must always refer to the page table to translate the VAs to PAs and this process is called Page Table Walk (PTW). However, PTW takes relatively longer time because SYSMMU accesses the main memory in which the page table is stored.

To reduce the translation time, SYSMMU uses Translation Look-ahead Buffer (TLB). TLB contains the mapping information recently read by PTW. This information is called page descriptor and it enables temporal locality. If TLB contains the page descriptor for an address translation request, then the translation is performed immediately without accessing the main memory for PTW.

Figure 1-2 illustrates the three major tasks performed by the SYSMMU, namely PTW, address translation, and cache.

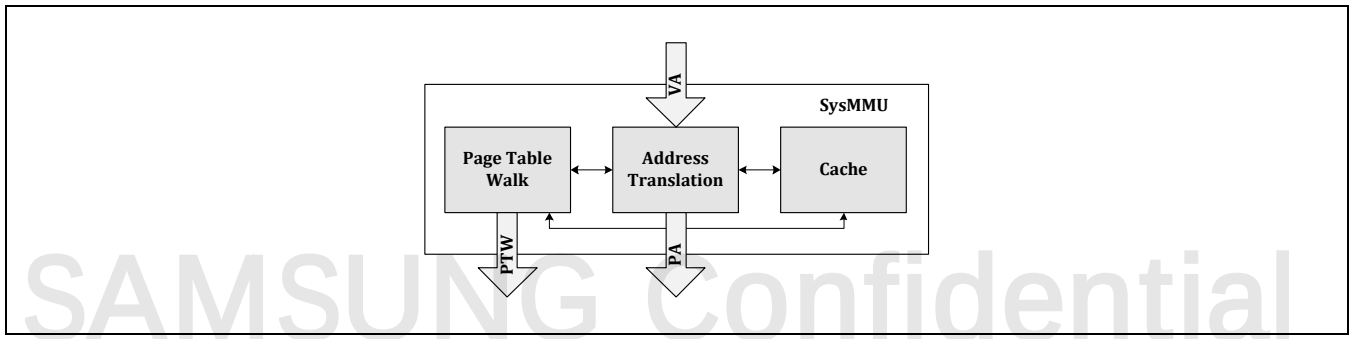

Figure 1-2 Tasks Performed by SYSMMU

Figure 1-3 illustrates a static view design of the SYSMMU driver based on the aforementioned content.

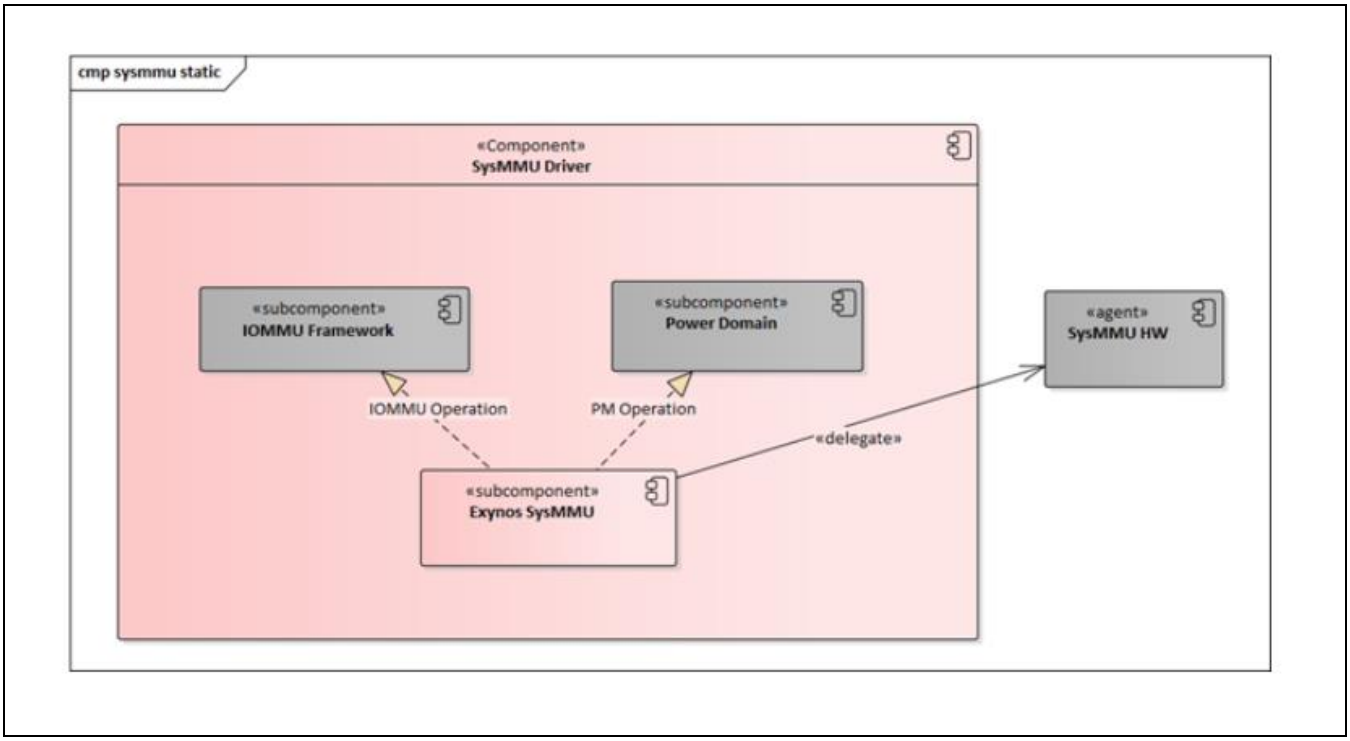

Figure 1-3 Static View Design for SYSMMU Driver

2 Driver Bring-up

This chapter describes the method to operate SYSMMU by setting kernel driver and Device-Tree (DT).

2.1 Kernel Driver

2.1.1 Kernel Driver Source Code

Kernel driver code is responsible for initializing and controlling the SYSMMU H/W.

It is divided into IOMMU common code and Exynos specific code, and the location of the source code is as follows:

```
# Common driver code
{Kernel}/driver/iommu/iommu-xxx

# Exynos specific code
{Kernel}/driver/iommu/exynos-xxx
```

2.1.2 Kernel Configuration

2.1.2.1 Required Configuration

CONFIG_EXYNOS_IOMMU=y
Enable this option to build support for SYSMMU

CONFIG_ EXYNOS_IOVMM=y (default y)
Enable this option for Exynos IOMMU users to allocate and map an IO virtual memory region with a physical memory region and manage the allocated virtual memory regions.

CONFIG_EXYNOS_VGEN=y
Enable Exynos IOMMU VGEN support feature to parse vid in DT and set virt_sfrbase according to vid.

2.1.2.2 Optional Configuration

CONFIG_EXYNOS_IOMMU_EMULATION=y
Enable Exynos IOMMU emulation feature

2.2 DT Configuration

The Open Firmware Device Tree or DT is a data structure and language for describing the hardware. DT stores the hardware description in OS readable format.

2.2.1 DT for SYSMMU

DT code of ExynosAutoV9 is located in {Kernel}/arch/arm64/boot/dts/exynos/. The exynosauto9-sysmmu.dtsi file defines the SYSMMU.

2.2.2 SYSMMU Node

The following example describes the various properties of one of the SYSMMUs of ExynosAutoV9 SoC:

```
sysmmu _dpum_0: sysmmu@18C80000 {
# Compatible using SYSMMU driver
        compatible = "samsung,exynos- sysmmu ";
# Register address (1st base, 2nd virtual base addr, 3rd GIC base address)
        reg = <0x0 0x18C80000 0x8000>,
              <0x0 0x18C88000 0x8000>,
              <0x0 0x10030000 0x1000>;
# Interrupts for SYSMMU, All of SYSMMUs share same interrupt number.
        interrupts = <0 INTREQ_COMB_NONSECURE_SYSREG_PERIS 0>,
                     <0 INTREQ_COMB_SECURE_SYSREG_PERIS 0>;
# dintc_port number is used to distinguish each SYSMMU interrupt.
  sysmmu,dintc_port = <37>;
# QoS value control of PTW request
        qos = <15>;

# information of clock which is used by SYSMMU
        clock-names = "aclk";
        clocks = <&clock GATE_SYSMMU_D0_DPUM_QCH_S0>;

        port-name = "GF0, G0";
# Secure IRQ use
        sysmmu,secure-irq;
#Secure Register address
        sysmmu,secure_base = <0x18CC0000>;
#SYSMMU tlb configuration (※ It is optimization value)
        sysmmu,tlb_property =
            <(SYSMMU_PORT_PREFETCH_PREDICTION_READ(0x0) | SYSMMU_BL1) SYSMMU_NOID>,
<(SYSMMU_PORT_PREFETCH_PREDICTION_READ(0x1) | SYSMMU_BL1) SYSMMU_ID_MASK(0x4,     0xC07)>,
            <(SYSMMU_PORT_PREFETCH_PREDICTION_READ(0x1) | SYSMMU_BL1) SYSMMU_ID_MASK(0x400,
0xC00)>,
            <(SYSMMU_PORT_PREFETCH_PREDICTION_READ(0x1) | SYSMMU_BL1) SYSMMU_ID_MASK(0x800,
0xC00)>;
        #iommu-cells = <0>;
#Power-domain is used by SYSMMU.
        samsung,power-domain = <&pd_dpum>;
#default status of SYSMMU
        status = "disabled";
};
```

Caution: The tlb_property is an optimization value. Do not change this value.

2.2.3 SYSMMU on Hypervisor

To enable SYSMMU in Hypervisor OS, add the settings described in section 2.2.3.1 Host OS and 2.2.3.2 Guest OS to the DT.

2.2.3.1 Host OS

1. In the host OS, change status of SYSMMU from disable to enable, and then add the vid value (specific domain ID).

NOTE: DT code of ExynosAutoV9 is located in {Kernel}/arch/arm64/boot/dts/exynos/{virtual domain}/{Host OS domain}, of which the exynosauto9-sadk-en-sysmmu.dtsi file defines the SYSMMU.

The following example describes a SYSMMU node. (Location: {kernel}/arch/arm64/boot/dts/exynos/linux_sys-and_ivi/linux_sys/exynosauto9-sadk-en-sysmmu.dtsi)

```
&sysmmu_dpus0_0 {
        status = "okay";
        vid = <2>;
};

&sysmmu_dpus0_1 {
        status = "okay";
        vid = <2>;
};

&sysmmu_dpus0_2 {
        status = "okay";
        vid = <2>;
};
```

2. The SYSMMU used in guest OS should be blocked in host OS.

The host OS allows access to all the MMU resources (register and IRQ) by default.

Therefore, if the guest OS needs to access a specific resource, it needs to block the host OS.

NOTE: DT code of ExynosAutoV9 is located in {Kernel}/arch/arm64/boot/dts/exynos/{virtual domain}/{Host OS domain}, of which exynosauto9-sadk-vplatform-sysmmu.dtsi file defines the SYSMMU.

The following example describes the method to block the host OS from accessing the SYSMMU in guest OS.

(File location: kernel/arch/arm64/boot/dts/exynos/linux_sys-and_ivi/linux_sys/exynosauto9-sadk-vplatform-sysmmu.dtsi)

```
&vplatform {

// sysmmu_dpum_2
vl,io-memory@18CA0000 {
        compatible = "vl,io-memory";
        reg = <0x0 0x18CA0000 0x10000>;
        vl,disabled;
};

// sysmmu_dpum_3
vl,io-memory@18CB0000 {
        compatible = "vl,io-memory";
        reg = <0x0 0x18CB0000 0x10000>;
        vl,disabled;
};
……
}
```

2.2.3.2 Guest OS

1. In the guest OS, change status of SYSMMU from disable to enable, and then add the vid value (specific domain ID).

2. Ensure that the power-domain is not controlled by the guest OS. If you delete the power domain node from the DT, guest OS cannot control the power domain.

NOTE: DT code of ExynosAutoV9 is located in {Kernel}/arch/arm64/boot/dts/exynos/{virtual domain}/{Guest OS domain}, of which exynosauto9-sadk-en-sysmmu.dtsi file defines the SYSMMU.

The following example describes the method of changing the setting of SYSMMU in the guest OS:

(File location: kernel/arch/arm64/boot/dts/exynos/linux_sys-and_ivi/and_ivi/exynosauto9-sadk-en-sysmmu.dtsi)

```
&sysmmu_dpum_2 {
        status = "okay";
        vid = <3>;
        /delete-property/ samsung,power-domain;
};

&sysmmu_dpum_3 {
        status = "okay";
        vid = <3>;
        /delete-property/ samsung,power-domain;
};
```

3. To use the SYSMMU, the guest OS must have the SYSMMU resource information in the DT.

DT code of ExynosAutoV9 is located in {Kernel}/arch/arm64/boot/dts/exynos/{virtual domain}/{guest OS domain}, of which exynosauto9-sadk-vplatform-sysmmu.dtsi file defines the SYSMMU.

The following example describes a method by which the guest OS can allow for SYSMMU resource:

(File location: kernel/arch/arm64/boot/dts/exynos/linux_sys-and_ivi/and_ivi/exynosauto9-sadk-vplatform-sysmmu.dtsi)

```
&vplatform {
        // sysmmu_dpum_2
        vl,io-memory@18CA0000 {
                compatible = "vl,io-memory";
                reg = <0x0 0x18CA0000 0x10000>;
        };

        // sysmmu_dpum_3
        vl,io-memory@18CB0000 {
                compatible = "vl,io-memory";
                reg = <0x0 0x18CB0000 0x10000>;
        };
    ……
}
```

2.2.4 SYSMMU Log

The following log is displayed, when SYSMMU probe is successful:

```
exynos-sysmmu 1aa80000.sysmmu: get    specific vid = 2
exynos-sysmmu 1aa80000.sysmmu: Registering secure irq 41
exynos-sysmmu 1aa80000.sysmmu: Secure base = 0x1aac0000
exynos-sysmmu 1aa80000.sysmmu: is probed. Version 7.4.0
```

3 Fault Handling in SYSMMU

3.1.1 Operating Principle

SYSMMU is a Samsung specific implementation of IOMMU. It prevents DMAable IPs from accessing unauthorized memory area. *Figure 3-1* illustrates the mechanism used in SYSMMU to translate VA to PA. DMAable IPs must refer to the SYSMMU page table for address translation. Therefore, DMAable IPs cannot access PAs that are not mapped in the page table.

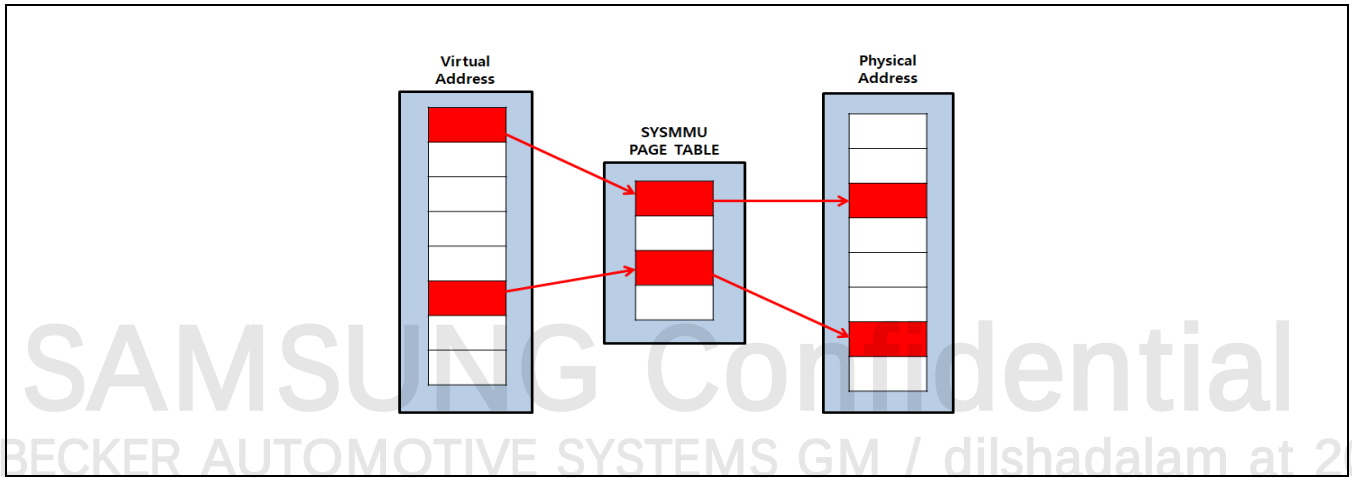

Figure 3-1 Address Translation in SYSMMU

3.1.2 Address Translation Faults and their Detection Mechanisms

SYSMMU can detect the following faults during address translation:

- Page fault: Request has VA that is not mapped to any PA
- Access to a page table for PTW: An illegal page table address is specified for the PTW transaction.
- Access permission: Physical page cannot be accessed due to resource permission issues.

NOTE: Access permission faults are detected only when ENABLE_ACCESS_PROT is enabled.

SYSMMU detects the address translation faults as follows:

- Page fault: When a read/write is requested through VA, SYSMMU H/W tries to find a mapping between VA and PA in the page table. If the page table does not contain any mapping information, SYSMMU triggers a page fault.
- Access to a page table for PTW: When SYSMMU accesses the page table with an invalid address for the PTW, SYSMMU triggers this fault
- Access permission: A page table contains page descriptors that comprises of mapping address and R/W permissions. Whenever a DMAable IP accesses DRAM with a VA through SYSMMU, SYSMMU checks if the data transaction has the valid R/W permission for DRAM memory, and trigger an access permission fault if it does not have valid R/W permission.

3.1.3 Fault Handling

When SYSMMU detects a fault, the following fault handling routine is executed:

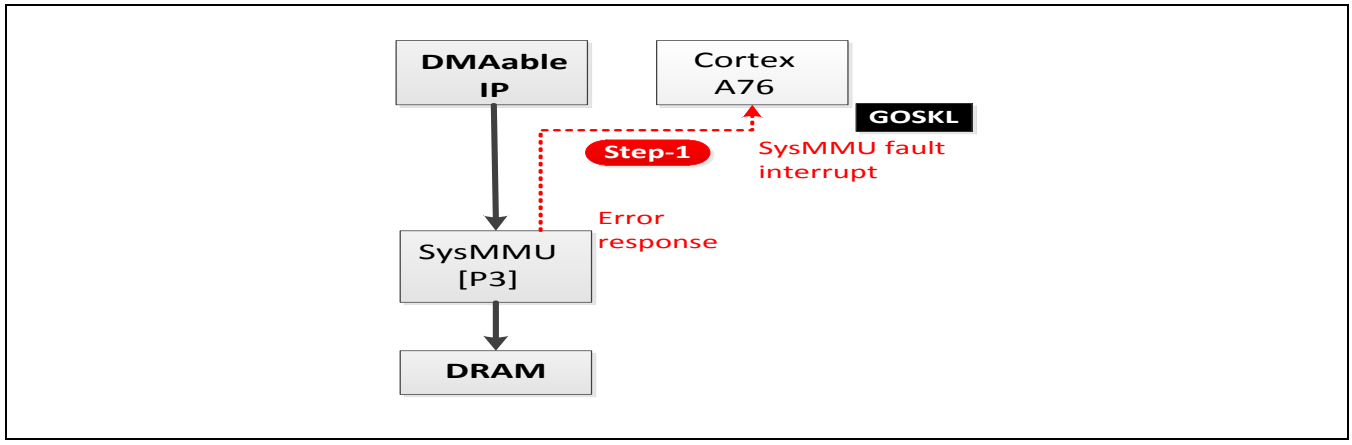

Figure 3-2 Error Response for Fault Interrupt

1. SYSMMU blocks the subsequent requests and it enters into block-mode.

2. SYSMMU generates a SYSMMU fault interrupt, and then the interrupt handler of SYSMMU driver is invoked. The SYSMMU driver is located within the guest OS.

3. Interrupt handler of SYSMMU driver prints the fault information on the console.

4. Interrupt handler of SYSMMU driver triggers kernel panic for further debugging.

3.1.4 Analyzing SYSMMU Fault Log

Example 3-1 describes the SYSMMU log that is available in a guest OS kernel log when a fault occurs. The READ PAGE FAULT occurred because a Display Processing Unit (DPU) running in VM2 tried to access with VA 0x10730000, but LV2 (level-2) entry 0x00000000 indicates that no PA is mapped to the VA in a LV2 page table.

NOTE: 0x1aa8000 refers to DPU in ExynosAuto9_UM. 13.3 Register Description.

Additionally, SYSMMU SFR and TLB status information is printed when the fault occurs.

Example 3-1 SYSMMU Fault Log in Kernel

```
exynos-sysmmu 1aa80000.sysmmu: exynos_sysmmu_irq:407: irq(52) happened
------------------------------------------------------------
From [GF0, G0], SYSMMU READ PAGE FAULT at 0x10730000 (page table @ 0x00000000cffd0000) in vid 2
AxID: 0x0, AxLEN: 0x0
Lv1 entry: 0xbe8d72c1
Lv2 entry: 0x00000000
GLOBAL ADDR: 0x0000000018c80000(VA: ffffff800c2c0000), MMU_CTRL: 0x00000005
VIRTUAL ADDR: 0x0000000018c8b000(VA: ffffff800c2d3000), MMU_CTRL: 0x00000009, PT_BASE: 0x000cffd0
VERSION 7.4.0
MMU_CFG(G): 0x00000f80, MMU_CFG(V) : 0x00000000, MMU_STATUS: 0x00000009
SYSMMU has 5 TLBs, 1 ports, 8 sbb entries
TLB.0 has 1 way, 2 set.
------------ TLB[WAY][SET][ENTRY] ------------
TLB.1 has 1 way, 2 set.
------------ TLB[WAY][SET][ENTRY] ------------
TLB.2 has 1 way, 2 set.
------------ TLB[WAY][SET][ENTRY] ------------
TLB.3 has 1 way, 2 set.
------------ TLB[WAY][SET][ENTRY] ------------
TLB.4 has 1 way, 2 set.
------------ TLB[WAY][SET][ENTRY] ------------
>> No Valid TLB Entries
--- SBB(Second-Level Page Table Base Address Buffer) ---
>> No Valid SBB Entries
------------------------------------------------------------
Kernel panic - not syncing: Unrecoverable System MMU Fault!!
```

4 SYSMMU sysfs guide

Console Command	Description	Example and explanation
/sys/kernel/debug/iommu/eventlog/iommu-xxx-domain	Information about device attached in iommu_domain, virtual address and page descriptor.	$ cat iommu-dpu_domain IOMMU_ATTACH of DPUS0_GF0 IOMMU_ATTACH of DPUS0_G0 IOMMU_ATTACH of DPUS0_G1 …… IOMMU_ALLOCSLPD @ [iova:0x10000000, entry:0xbedbe601) IOMMU_ALLOCSLPD @ [iova:0x10100000, entry:0xbedbe501) ……
/sys/kernel/debug/iovmm/eventlog/iommu-xxx-domain	Information about map / unmap virtual address	cat iommu-dpu_domain IOVMM_MAP [0x10000000, 0x10004000(+0x3c000)) IOVMM_MAP [0x10100000, 0x10104000(+0x3c000)) IOVMM_MAP [0x10200000, 0x109e9000(+0x37000)) IOVMM_MAP [0x10b00000, 0x112e9000(+0x37000)) IOVMM_MAP [0x11400000, 0x11784000(+0x3c000)) IOVMM_MAP [0x11800000, 0x11b84000(+0x3c000)) ……