ExynosAutoV9

SYSMMU Fault Handling

Revision 1.10

May 2021

Application Note

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	April, 2021	• Initial version of the document	S.LSI
1.10	May, 2021	• Added audio fault handling information	S.LSI

Table of Contents

List of Figures

Figure Number	Title	Page Number

목차 항목을 찾을 수 없습니다.

List of Tables

Table Number	Title	Page Number

목차 항목을 찾을 수 없습니다.

List of Examples

Example Number	Title	Page Number

목차 항목을 찾을 수 없습니다.

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 SYSMMU recovery fault handler support

1.1 Background

Customer is asking the way not to be reset the SYS domain by SYSMMU fault.

1.1.1 Word

PTW(Page Table Walk)	The read operation to receive the page descriptor.
Reserved_pgtable	The original page table which is mapped Master IP VA <-> PA information.
Recovery_pgtable	The page table that makes all PTW success
Recovery_memory	It is accessed by recovery_pgtable (It must have to allowed in audio s2mpu)

1.1.2 Problem

- sysMMU occurs panic in fault handler.

- System domain will be reset by panic.

1.1.3 Solution

1) Audio registers fault handler using iommu_register_device_fault_handler()

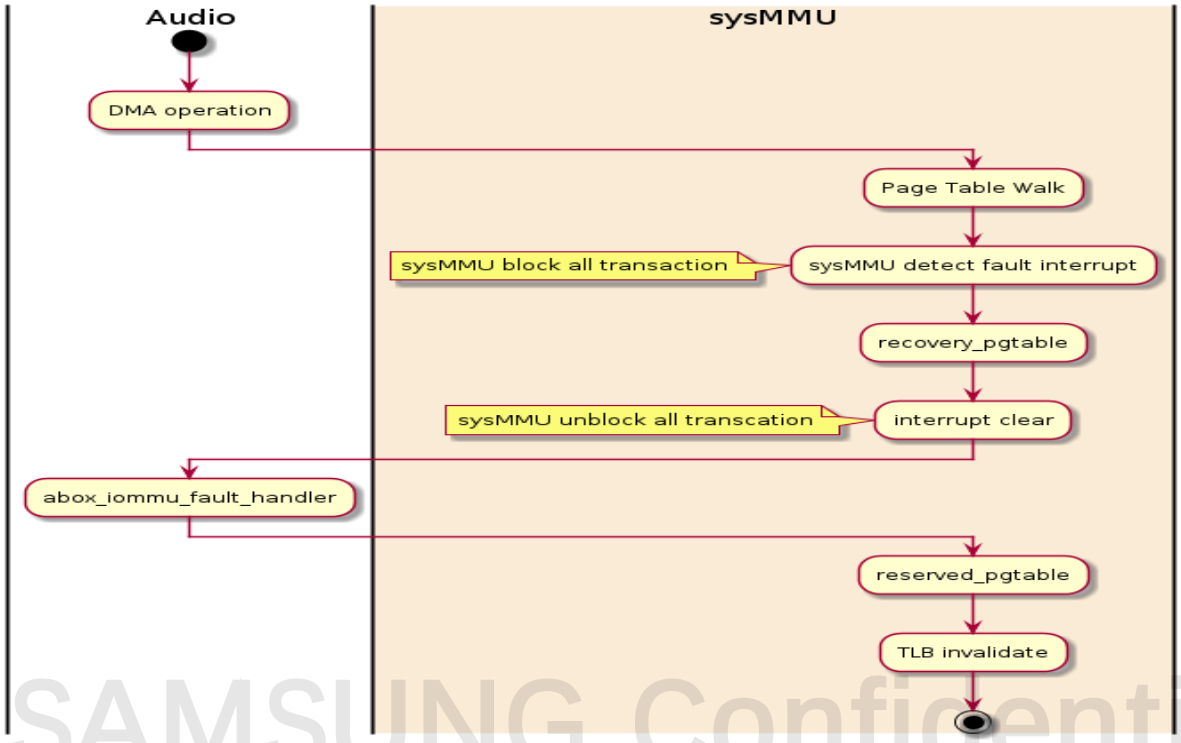
When sysMMU fault is occurred, the fault handler will be callback.

```
int iommu_register_device_fault_handler(struct device *dev, iommu_dev_fault_handler_t handler, void *data)
```

2) After finishing audio fault handler, sysMMU change recovery_pgtable to reserved_pgtable to support normal HIFI core operation.

.

1.1.4 Work flow after solution



1) sysMMU detect fault interrupt
2) sysMMU changes from reserved_pgtable to recovery_pgtable.
3) sysMMU interrupt clear.
   - After sysMMU interrupt clear, sysMMU try to do transaction to make unblock.

   ※ If sysMMU success PTW, restore block status to unblock.
4) sysMMU callback the registered audio fault handler.
5) Audio fault handler resets hifi core.
6) After audio fault handler works, sysMMU changse from recovery_pgtable to reserved_pgtable to support normal HIFI core operation.
7) SysMMU invalidate TLB
8) sysMMU return IRQ_HANDLED

1.1.5 recovery_pgtable/ reserved_pgtable / recovery_mem

- recovery_pgtable / reserved_pgtable / recovery_mem should be controlled by memory-region each project.

Please see the guide below and modify the appropriate address.

1) The reason about recovery_pgtable / reserved_pgtable is needed

- Recovery page table will be used to avoid fault occur again.
- After sysmmu interrupt clear, sysMMU status changes to unblock and retries address translation of the fault request.
- If sysMMU leave the normal page table, same fault interrupt will be occured. Because fault address is not mapped normal page table.
- But, Recovery page table is mapped to recovery_mem about all translation address. It makes retry address translation will be successed

2) Information about page table address in SADK

reserved_pgtable = 0xAFEFC000

recovery_pgtable = dynamic allocated each domain.

recovery_mem = 0xD3B00000 (It will be adpted)

3) How to set recovery_pgtable / reserved_pgtable / recovery_mem

kernel 4.14

ITEM	location	Descriptions
Reserved_pgtable	sources/lk-boot/platform/ exynosauto-v9/ include/platform/rev/evt1/ exynosauto_v9_evt1.h	#define SYSMMU_PG_TABLE 0xB9EFC000

	arch/arm64/boot/dts/exynos /linux_sys-and_ivi/linux_sys/ exynosauto9-sysmmu/ sadkauto9-evt1-sys_si.dts	&iommu_domain_aud {
		…
		lk_rsvd_pgtable = <0xB9EFC000>;
		…
		};

Recovery_pgtable		Dynamic allocated each domain
Recovery_mem		It will be updated in RC6

kernel 5.4

ITEM	location	Descriptions
Reserved_pgtable	sources/lk-boot/platform/ exynosauto-v9/ include/platform/rev/evt1/ exynosauto_v9_evt1.h	#define SYSMMU_PG_TABLE 0xAFEFC000

	sources/linux_sys_dts-la/linux_sys/
	exynosauto9-sadk-en-sysmmu.dtsi
		&reserved_memory {
		……
		sysmmu_aud_rmem: sysmmu_aud_rmem {
		compatible = "exynos-sysmmu,rmem";
		reg = <0x0 0xAFEFC000 0x104000>;
		};
		};

Recovery_pgtable		Dynamic allocated each domain

Recovery_mem	sources/linux_sys_dts-la/linux_sys/
	exynosauto9-sadk-en-sysmmu.dtsi
		&sysmmu_aud {
		……
		recovery_mem = <0xD3B00000>;
		……
		};

1.1.6 How to reset HiFi core

If you need to reset hifi core when you get sysmmu fault irq, you should follow below sequence.

All Abox's resource must to release before reset hifi core.
Resource List for reset :
1. DMA
- All running DMA must to stop
  - RDMA, WDMA

2. Clock
- Enabling Clock must to disable
  - UAIF : bclk, bclk_gate
  - ABOX : cpu, audif

3. GPIO
- All GPIO must to set 'IDLE'
  - UAIF : aud_i2s_bus

HiFi core reset sequence

- if you need reference code, you can refer to S2R code

1) DMA, GPIO, UAIF Clock Disable
- DMA : you can use below two options
    - send Stop IPC to hifi core. You can refer 'abox_rdma_trigger'
    - directly set register to dma
- GPIO : you should use pinctrl
    - UAIF has pinctrl in uaif private data. You should use it.
    - you can refer in 'abox_uaif_cfg_gpio function'.
- UAIF Clock : you should use 'clk_disable'
    - you can refer in 'abox_uaif_shutdown'.

2) Send 'CPU power off' to hifi core using IPC
- you should use 'abox_cpu_power_all'
- you can refer 'abox_suspend'.

3) HiFi core power off using PMU
- you can refer 'abox_suspend'.

4) Disable Abox Clock
- you should use 'abox_clock_enable'
- you can refer 'abox_suspend'.

5) Disable Mailbox IRQ
- you should use 'abox_mailbox_disable_irq'
- you can refer 'abox_suspend'.

6) HiFi core image reload
- you should use 'abox_download_all_firmwares_from_backup'
- you can refer 'abox_resume'.

7) HiFi core power on using PMU
- you should use 'abox_cpu_power_all'
- you can refer 'abox_resume'.

8) Call abox_enable function
- you should call ‘abox_enable’
- you can refer ‘abox_resume’.

9) Enable Abox Clock
- you should use ‘abox_clock_enable’
- you can refer ‘abox_resume’.

<End of Document>