ARM® System Memory Management Unit Architecture Specification

64KB Translation Granule Supplement

ARM®

ARM IHI 0067A.b (ID073113)

ARM System Memory Management Unit Architecture Specification
64KB Translation Granule Supplement

Release Information

The following changes have been made to this document.

Change History

Date	Issue	Confidentiality	Change
24 January 2013	A	Confidential	Beta release
23 July 2013	A.b	Non-Confidential	Final release

Proprietary Notice

ARM System MMU (SMMU) Architecture Specification Licence

This end user licence agreement ("licence") is a legal agreement between you (either a single individual, or single legal entity) and ARM Limited ("ARM") for the use of the relevant SMMU Architecture Specification accompanying this licence. ARM is only willing to license the relevant SMMU Architecture Specification to you on condition that you accept all of the terms in this licence. By clicking "I agree" or otherwise using or copying the relevant SMMU Architecture Specification you indicate that you agree to be bound by all the terms of this licence. If you do not agree to the terms of this licence, ARM is unwilling to license the relevant SMMU Architecture Specification to you and you may not use or copy the relevant SMMU Architecture Specification and you should promptly return the relevant SMMU Architecture Specification to ARM.

"LICENSEE" means You and your Subsidiaries.

"Subsidiary" means, if You are a single entity, any company the majority of whose voting shares is now or hereafter owned or controlled, directly or indirectly, by You. A company shall be a Subsidiary only for the period during which such control exists.

1. Subject to the provisions of Clauses 2, 3 and 4, ARM hereby grants to LICENSEE a perpetual, non-exclusive, non-transferable, royalty free, worldwide licence to:

   (i) use and copy the relevant SMMU Architecture Specification for the purpose of developing and having developed products that comply with the relevant SMMU Architecture Specification;

   (ii) manufacture and have manufactured products which either: (a) have been created by or for LICENSEE under the licence granted in Clause 1(i); or (b) incorporate a product(s) which has been created by a third party(s) under a licence granted by ARM in Clause 1(i) of such third party's ARM SMMU Architecture Specification Licence; and

   (iii) offer to sell, sell, supply or otherwise distribute products which have either been (a) created by or for LICENSEE under the licence granted in Clause 1(i); or (b) manufactured by or for LICENSEE under the licence granted in Clause 1(ii).

2. LICENSEE hereby agrees that the licence granted in Clause 1 is subject to the following restrictions:

   (i) where a product is created under Clause 1(i) or manufactured under Clause 1(ii) it must contain at least one processor core which has either been (a) developed by or for ARM; or (b) developed under licence from ARM;

   (ii) the licences granted in Clause 1(iii) shall not extend to any portion or function of a product that is not itself compliant with part of the relevant SMMU Architecture Specification; and

   (iii) no right is granted to LICENSEE to sublicense the rights granted to LICENSEE under this Agreement.

3. Except as specifically licensed in accordance with Clause 1, LICENSEE acquires no right, title or interest in any ARM technology or any intellectual property embodied therein. In no event shall the licences granted in accordance with Clause 1 be construed as granting LICENSEE, expressly or by implication, estoppel or otherwise, a licence to use any ARM technology except the relevant SMMU Architecture Specification.

4. **The relevant SMMU Architecture Specification is provided "as is" with no warranties express, implied or statutory, including but not limited to any warranty of satisfactory quality, merchantability, noninfringement or fitness for a particular purpose.**

5. No licence, express, implied or otherwise, is granted to LICENSEE, under the provisions of Clause 1, to use the ARM tradename in connection with the relevant SMMU Architecture Specification or any products based thereon. Nothing in Clause 1 shall be construed as authority for LICENSEE to make any representations on behalf of ARM in respect of the relevant SMMU Architecture Specification.

6. This Licence shall remain in force until terminated by you or by ARM. Without prejudice to any of its other rights if LICENSEE is in breach of any of the terms and conditions of this Licence then ARM may terminate this Licence immediately upon giving written notice to You. You may terminate this Licence at any time. Upon expiry or termination of this Licence by You or by ARM LICENSEE shall stop using the relevant SMMU Architecture Specification and destroy all copies of the relevant SMMU Architecture Specification in your possession together with all documentation and related materials. Upon expiry or termination of this Licence, the provisions of clauses 6 and 7 shall survive.

7. The validity, construction and performance of this Agreement shall be governed by English Law.

ARM contract references: **LES-PRE-20225 ARM System MMU (SMMU) Architecture Specification Licence**

ARM Limited, 110 Fulbourn Road, Cambridge, England CB1 9NJ.

Note

The term ARM is also used to refer to versions of the ARM architecture, for example ARMv7 refers to version 7 of the ARM architecture. The context makes it clear when the term is used in this way.

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Note

The only changes between issue A and issue A.b are the status and confidentiality. The only changes to pages 5-20 of this document are the document version and confidentiality status in the footer of each page.

Contents

ARM System Memory Management Unit Architecture Specification 64KB Translation Granule Supplement

Preface

This preface introduces the *ARM System Memory Management Unit (SMMU) Architecture Specification 64KB Translation Granule Supplement*. It contains the following sections:

About this supplement

This supplement describes the option to support a 64KB translation granule, in an implementation of version 1 of the SMMU architecture, for Non-secure stage 2 translations. It provides information that is supplementary to Issue B of the *ARM System Memory Management Unit Architecture Specification*.

Intended audience

This supplement is written for readers who are familiar with Issue B of the *ARM System Memory Management Unit Architecture Specification*.

Using this supplement

The information in this supplement is organized into the following chapters:

Chapter 1 ***Introduction***

Read this for a brief introduction to the permitted SMMUv1 architecture use of a 64KB translation granule.

Chapter 2 ***Stage 2 Translation Context Bank Format***

Read this for information about the differences that apply to the Stage 2 translation context bank format in an SMMUv1 implementation that supports the 64KB translation granule.

Conventions

The following sections describe conventions that this book can use:

- *Typographic conventions*
- *Numbers*.

Typographic conventions

The typographical conventions are:

italic	Introduces special terminology, and denotes citations.
bold	Denotes signal names, and is used for terms in descriptive lists, where appropriate.
`monospace`	Used for assembler syntax descriptions, pseudocode, and source code examples. Also used in the main text for instruction mnemonics and for references to other items appearing in assembler syntax descriptions, pseudocode, and source code examples.
SMALL CAPITALS	Used for a few terms that have specific technical meanings, and are included in the glossary.
Colored text	Indicates a link. This can be: • A URL, for example, `http://infocenter.arm.com`. • A cross-reference, that includes the page number of the referenced information if it is not on the current page, for example, *ARM publications* on page xi. • A link, to a chapter or appendix, or to a glossary entry, or to the section of the document that defines the colored term, for example SMMU_CBA2Rn.

Numbers

Numbers are normally written in decimal. Binary numbers are preceded by `0b`, and hexadecimal numbers by `0x`. In both cases, the prefix and the associated value are written in a `monospace` font, for example `0xFFFF0000`.

Additional reading

This section lists relevant publications from ARM and third parties.

See the Infocenter, http://infocenter.arm.com, for access to ARM documentation.

ARM publications

- Issue B of the *ARM System Memory Management Unit Architecture Specification* (ARM IHI 0062B).
- *ARM Architecture Reference Manual, ARMv7-A and ARMv7-R edition* (ARM DDI 0406).
- *CoreSight™ Architecture Specification* (ARM IHI 0029).

Feedback

ARM welcomes feedback on its documentation.

Feedback on this book

If you have comments on the content of this book, send an e-mail to `errata@arm.com`. Give:

- The title.
- The number, ARM IHI 0067A.b.
- The page numbers to which your comments apply.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

ARM tests PDFs only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the appearance or behavior of any document when viewed with any other PDF reader.

Chapter 1
Introduction

This chapter introduces the SMMUv1 architecture support for a 64KB translation granule. It contains the following sections:

1.1 Translation regimes, translation stages, and translation table formats

The ARMv7 architecture, with the Virtualization Extensions and the Large Physical Address Extension, supports the translation regimes and stages shown in Figure 1-1:

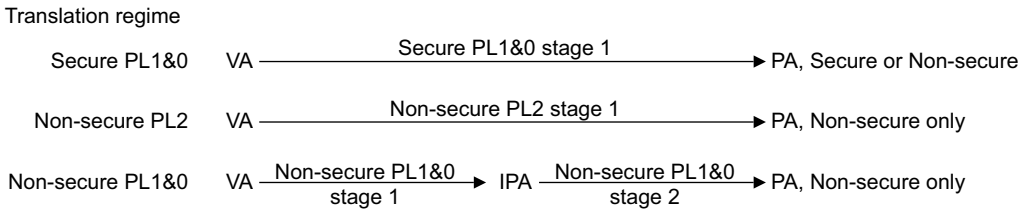


Figure 1-1 ARM VMSAv7 translation regimes and translation stages

In this figure:

- VA is Virtual Address.
- PA is Physical Address.
- IPA is Intermediate Physical Address.

As Figure 1-1 shows:

- The Non-secure PL1&0 translation regime comprises two stages of translation.
- The other translation regimes comprise only a single stage of translation.

The SMMUv1 architecture specification, defined in issues A and B of the *ARM System Memory Management Unit Architecture Specification*, supports these translation stages and the associated VMSAv7 translation table formats. These formats are:

- The VMSAv7 Short-descriptor format, that:
  - Uses 32-bit entries, or *descriptors*, in its translation tables.
  - Is compatible with earlier versions of the ARM architecture.
  - Supports input addresses of up to 32 bits.
  - Supports output addresses of up to 32 bits.
- The VMSAv7 Long-descriptor format, that:
  - Uses 64-bit entries, or *descriptors*, in its translation tables.
  - Is added to the ARMv7 architecture by the Large Physical Address Extension.
  - Supports input addresses of up to 40 bits.
  - Supports output addresses of up to 40 bits.

  Note

  In ARMv7, when using the Long-descriptor translation table format:
  - Any stage of address translation can generate output addresses of up to 40 bits.
  - Only Non-secure PL1&0 stage 2 address translations support input addresses larger than 32 bits.

These formats use a *translation granule* of 4KB, meaning:

- The size of a complete translation table is 4KB.
- The translation tables can define the attributes of memory regions at a granularity of 4KB.

1.1.1 Translation regime changes introduced by this 64KB translation granule supplement

The ARMv8 architecture:

- Introduces support for VAs and PAs of up to 48 bits. To support these, it extends the VMSAv7 Long-descriptor translation table format, to provide support for:
  - Input addresses of up to 48 bits.
  - Output addresses of up to 48 bits.
- Introduces support for a 64KB translation granule, as an alternative to the 4KB translation granule. When using this granule:
  - The size of a complete translation table is 64KB.
  - The translation tables can define the attributes of memory regions at a granularity of 64KB.

Use of the 64KB translation granule affects the translation table descriptor formats, because it:

- Reduces the number of low-order bits required to specify the address of a translation table.
- Increases the number of address bits required to index the required descriptor within a translation table.

The extended version of the VMSAv7 Long-descriptor translation table format is called the ARMv8 Long-descriptor translation table format, or the VMSAv8 Long-descriptor translation table format.

This 64KB translation granule supplement adds support for the following address translation option to the SMMUv1 architecture specification:

- 64KB translation granule.
- Input addresses of up to 40 bits.
- Output addresses of up to 40 bits.

This translation option is supported only for Non-secure PL1&0 stage 2 translations.

1.1.2 Summary of the translation table options with the 64KB translation granule extension

For an SMMUv1 implementation that includes support for the 64KB translation granule extension described in this supplement, Table 1-1 shows the supported translation table formats.

Table 1-1 SMMUv1 with 64KB translation granule extension translation table formats

Format	Translation granule	Abbreviation	Notes
ARMv7 Short-descriptor	4KB	V7S	-
ARMv7 Long-descriptor	4KB	V7L	-
Constrained ARMv8 Long-descriptor[a]	64KB	V8L	Supported only for Non-secure PL1&0 stage 2 translations

a. The ARMv8 Long-descriptor format supports input and output addresses of up to 48 bits.

The V8L short name is used both for:

- The constrained ARMv8 Long-descriptor format introduced by the SMMUv1 extension described in this supplement.
- The full ARMv8 Long-descriptor format supported by SMMUv2.

Note

The SMMU configuration registers do not distinguish between the ARMv8 Long-descriptor format and the constrained version of that format introduced by this extension.

1.2 About SMMUv1 support for the V8L translation table format

The SMMU_CBA2Rn.RW64 field controls whether a transaction uses the 4KB or the 64KB translation granule. When a transaction uses the 64KB translation granule, the transaction uses the constrained V8L translation table format.

The 64KB translation granule is supported only when a Translation context bank provides stage 2 translation only. This applies when the value of the SMMU_CBARn.TYPE field is `0b00`.

Table 1-2 shows the SMMU_CBA2Rn registers in the System MMU Global Register space 1, including the offset from SMMU_GR1_BASE.

Table 1-2 SMMU_CBA2Rn registers in the System MMU Global Register Space 1

Offset	Name	Type	Description
`0x00_800`	SMMU_CBA2R0	RW	*SMMU_CBA2Rn, Context Bank Attribute Registers 2* on page 2-20
`0x00_804`	SMMU_CBA2R1		
`0x00_808-0x00_9FC`	SMMU_CBA2R2 to SMMU_CBA2R127		

Chapter 2
Stage 2 Translation Context Bank Format

This chapter describes the differences that apply to the Stage 2 translation context bank format that the *ARM System Memory Management Unit Architecture Specification* describes. It contains the following sections:

2.1 SMMU_CBn_TTBCR, Translation Table Base Control Register

The SMMU_CB*n*_TTBCR characteristics are:

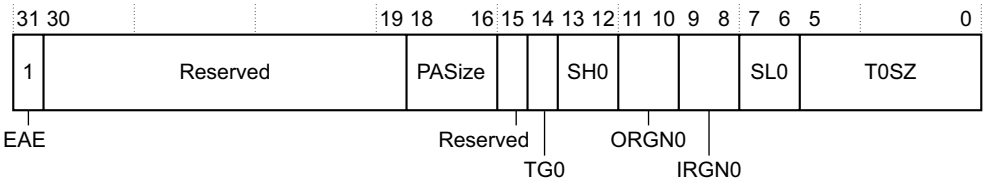
Purpose Provides additional configuration for stage 2 translations.

Usage constraints The register format described in this section applies only when the SMMU_CBA2Rn.RW64 bit is set to 1. When the RW64 bit is set to 0, the V7S or V7L format is used, and the format of this register is as described in the *ARM System Memory Management Unit Architecture Specification*.

Configurations It is IMPLEMENTATION DEFINED whether the system implements stage 2 translation, as defined by the SMMU_IDR0 register. See the *ARM System Memory Management Unit Architecture Specification* for more information.

Attributes A 32-bit RW register.

The SMMU_CB*n*_TTBCR bit assignments are:



EAE, bit[31]

Extended Address Enable.

For a Stage 2 translation context entry, this field is RAO/WI.

A value of 1 means the translation context uses a Long-descriptor translation table format.

Bits[30:19] Reserved.

PASize, bits[18:16]

Physical address size. The encoding of this field is:

0 32-bit.

1 36-bit.

2 40-bit.

3-7 Reserved.

Bit[15] Reserved.

TG0, Bit[14] Translation granule size for SMMU_CBn_TTBR0.

0 4KB.

1 64KB.

SH0, bits[13:12]

Shareability attributes for the memory associated with the translation table walks using SMMU_CBn_TTBR0.

ORGN0, bits[11:10]

Outer cacheability attributes for the memory associated with the translation table walks using SMMU_CBn_TTBR0.

IRGN0, bits[9:8]

Inner cacheability attributes for the memory associated with the translation table walks using SMMU_CBn_TTBR0.

SL0, bits[7:6] Lookup Start Level for the SMMU_CBn_TTBR0 addressed region. The encoding of this field depends on the translation granule size:

- 4KB translation granule:
  - **0** Level 2.
  - **1** Level 1.
  - **2** Level 0.
  - **3** Reserved.
- 64KB translation granule:
  - **0** Level 3.
  - **1** Level 2.
  - **2** Reserved.
  - **3** Reserved.

T0SZ, bits[5:0]

The Size offset of the SMMU_CBn_TTBR0 addressed region, encoded as a 6-bit signed number giving the size of the region as $2^{(64-T0SZ)}$.

2.2 SMMU_CBA2Rn, Context Bank Attribute Registers 2

The SMMU_CBA2Rn characteristics are:

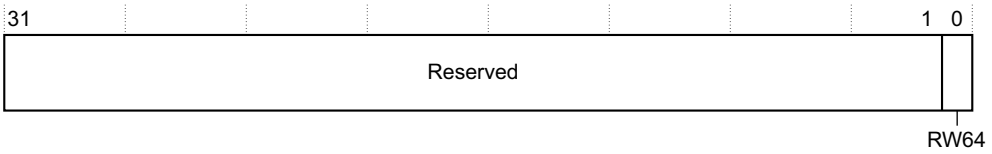
Purpose This register extends the information provided in the associated SMMU_CBARn register, by specifying additional configuration attributes for Translation context bank *n*.

Usage constraints This register is used when a Translation context bank provides stage 2 translation only, that is, when the value of the SMMU_CBARn.TYPE field is `0b00`.

Configurations The number of SMMU_CBA2Rn registers is IMPLEMENTATION DEFINED.

Attributes A 32-bit RW register. See Table 1-2 on page 1-16.

The SMMU_CBA2Rn bit assignments are:



Bits[31:1] Reserved.

RW64, bit[0]

Input address width supported, as indicated by the supported translation table format:

0 V7S or V7L.

1 V8L.