Memory Management Units

This chapter describes the memory management units (MMUs) in the device.

Topic **Page**

20.1 MMU Overview

A memory management unit (MMU) is a hardware component responsible for handling accesses to memory requested by a processing unit, DMA controller, or other bus requestor. MMU functions include:

- Translation of initiator internal (virtual) addresses to physical addresses (that is, virtual memory management)
- Preventing an initiator from making accesses to unmapped pages of the system memory

This device includes the following MMUs:

- Two top-level (system) MMUs:
  - MMU1 dedicated to EDMA Transfer Controller 0 (TC0), and EDMA Transfer Controller 1 (TC1)
  - MMU2 dedicated to PCIe_SS1, and PCIeSS2
- One MMU inside the MPU (dual Cortex®-A15) subsystem – MPU_MMU. This MMU is integrated in the Cortex-A15 processor.
- Two MMUs inside each of the DSP1, and DSP2 subsystems:
  - DSP1 subsystem: DSP1_MMU0, and DSP1_MMU1
  - DSP2 subsystem: DSP2_MMU0, and DSP2_MMU1
- Two MMUs inside each of the EVE1, EVE2, EVE3, and EVE4 subsystems:
  - EVE1 subsystem: EVE1_MMU0, and EVE1_MMU1
  - EVE2 subsystem: EVE2_MMU0, and EVE2_MMU1
  - EVE3 subsystem: EVE3_MMU0, and EVE3_MMU1
  - EVE4 subsystem: EVE4_MMU0, and EVE4_MMU1
- Two MMUs inside each of the IPU1, and IPU2 subsystems:
  - L1 unicache MMU – IPU1_UNICACHE_MMU, and IPU2_UNICACHE_MMU
  - L2 MMU – IPU1_MMU, and IPU2_MMU
- One MMU inside the 3D GPU (dual SGX544 core) subsystem
- One MMU inside the BB2D subsystem

NOTE: EVE is not supported in this family of devices.

NOTE: There is a Physical Address Translator (PAT) module in the Dynamic Memory Manager (DMM), which has similar to the MMU functionality. For more information about this module, see Section 15.2, *Dynamic Memory Manager*.

NOTE: This chapter provides a detailed description of the following MMUs:

- System MMUs
- DSP MMUs
- EVE MMUs
- IPU L2 MMU

System MMUs and DSP MMUs are fully identical from functional perspective. IPU L2 MMU is different in that it does not support “bypass” functionality.

For more information about:

- Cortex-A15 MMU, see the ARM® *Cortex®-A15 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).
- IPU unicache MMU, see Chapter 7, *Dual Cortex-M4 IPU Subsystem*.

Figure 20-1 and Figure 20-2 show an overview of system MMU1 and MMU2, respectively. In summary, requests initiated by a given requestor (EDMA TC0 and TC1 [both read and write ports] for system MMU1; PCIe_SS1 and PCIe_SS2 for system MMU2) can optionally be routed through the corresponding system MMU. Each requestor's use (or not) of the MMU is independently controllable via the Control Module CTRL_CORE_SMA_SW_7 register bitfields. It is recommended that this register is set during system initialization and remain static.

If the MMU loopback path is enabled for a given requestor, requests will be routed via the L3_MAIN interconnect to the MMU and will again go through the L3 interconnect to the requested physical address. If the MMU loopback path is disabled for a given requestor, those bus requests go directly through the L3_MAIN interconnect to the requested physical address, thus minimizing bus request latency.

Figure 20-1. System MMU1 Overview

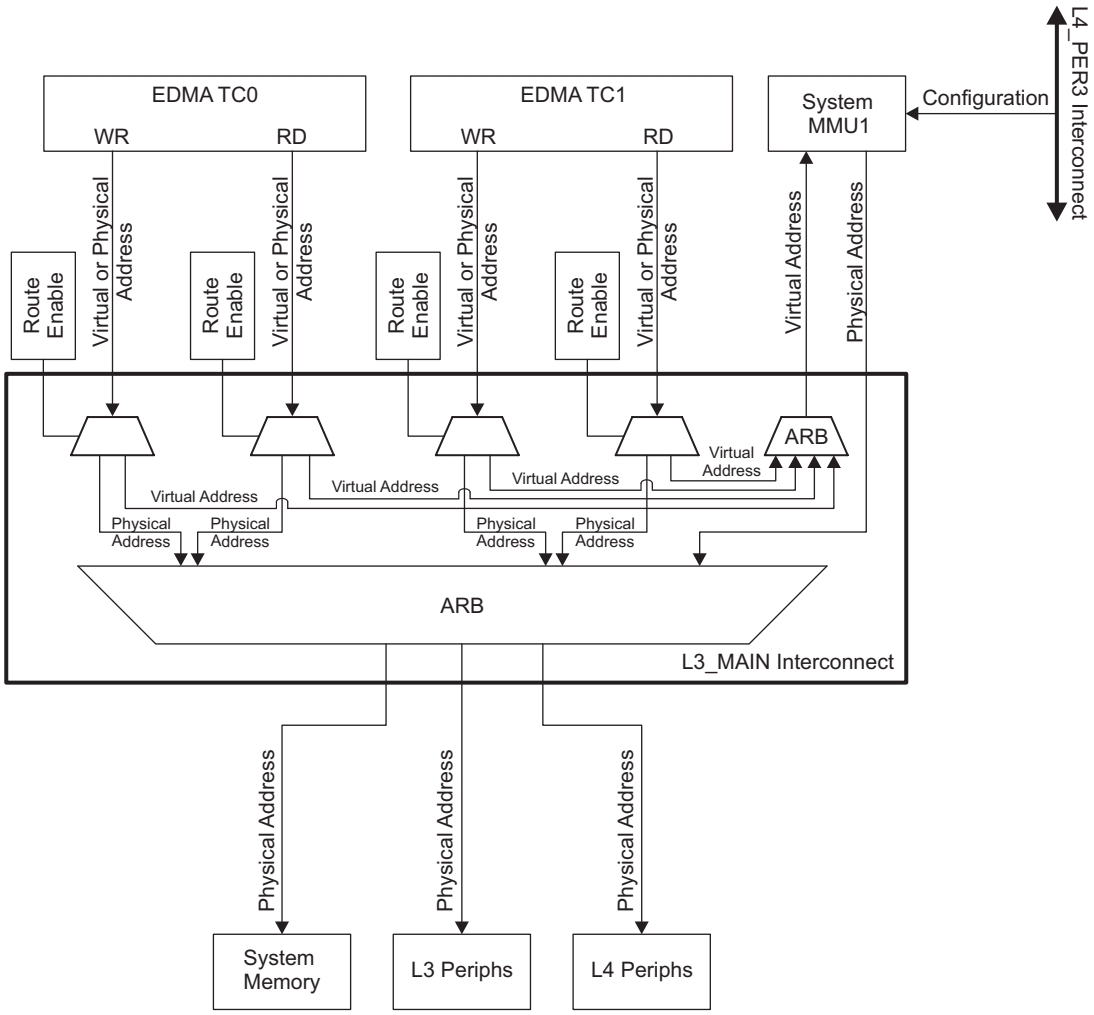
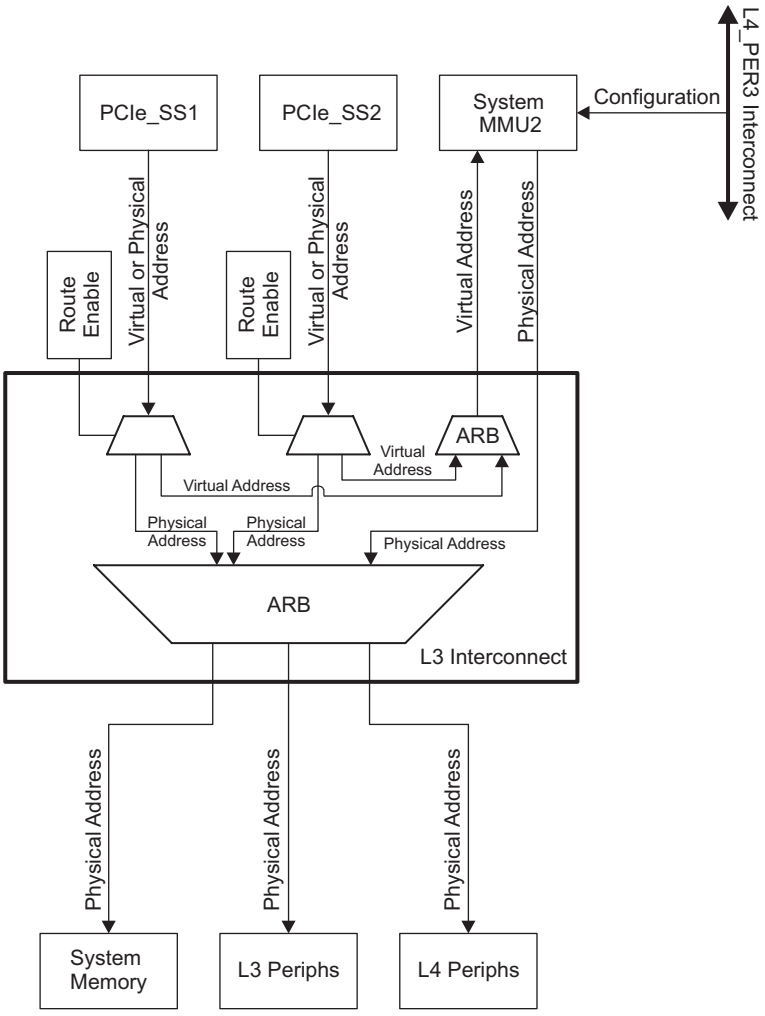


Figure 20-2. System MMU2 Overview



20.2 MMU Integration

This section describes the system MMUs integration in the device, including information about clocks, resets, and hardware requests. For more information about DSP, EVE, and IPU MMUs integration, refer to their respective chapters.

Figure 20-3 and Figure 20-4 show system MMU1, and MMU2 integration, respectively.

Figure 20-3. System MMU1 Integration

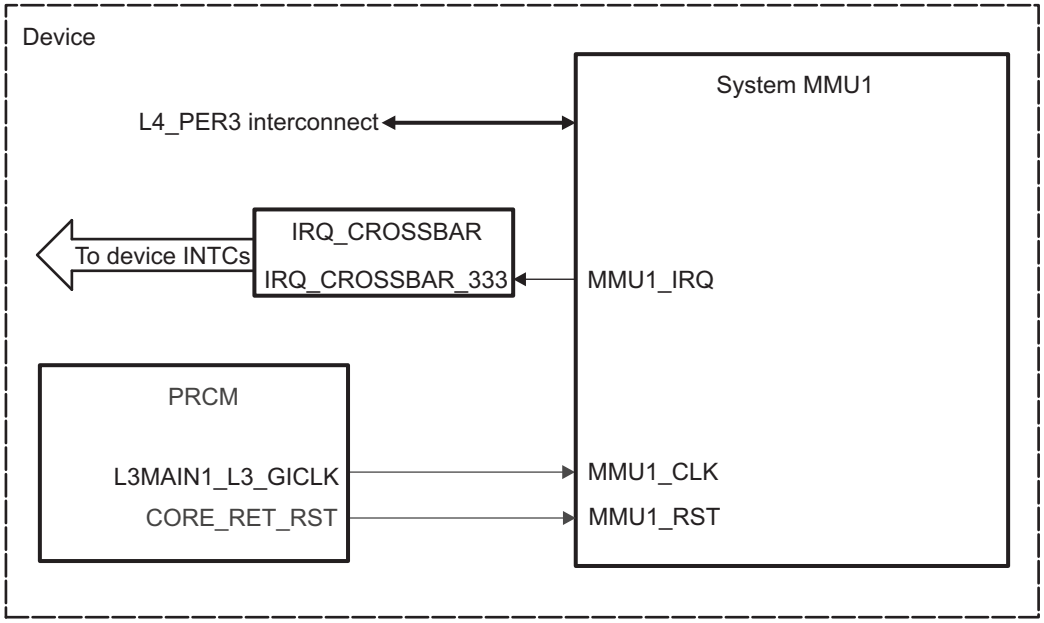


Figure 20-4. System MMU2 Integration

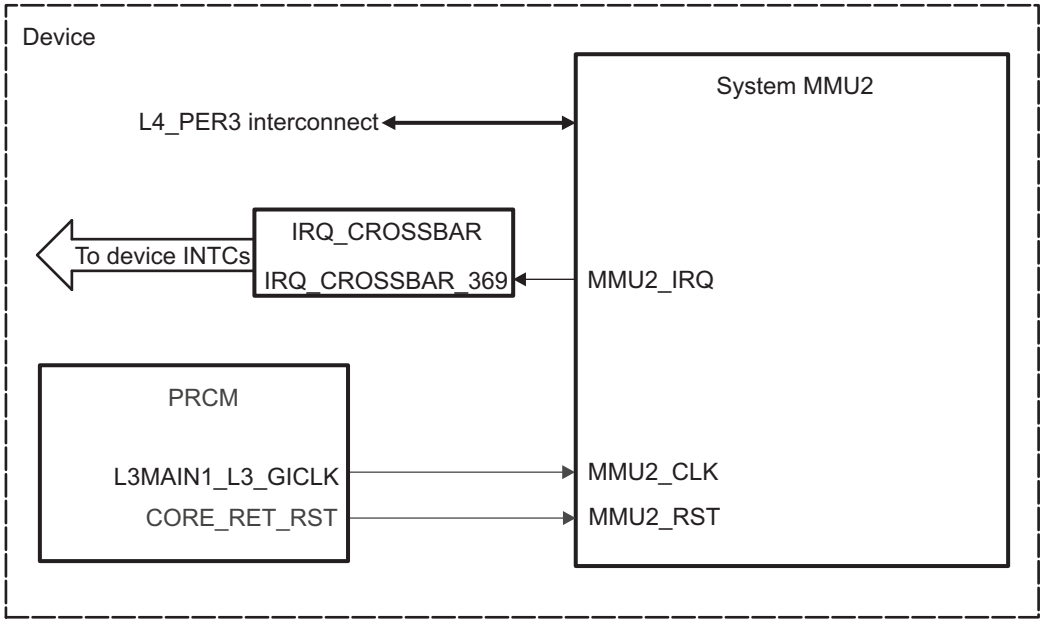


Table 20-1 through Table 20-3 summarize the system MMUs integration.

Table 20-1. MMU Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
System MMU1	PD_COREAON	L4_PER3
System MMU2	PD_COREAON	L4_PER3

Table 20-2. MMU Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
System MMU1	MMU1_CLK	L3MAIN1_L3_GICLK	PRCM	System MMU1 interface/functional clock. This clock is used for all interface and functional operations.
System MMU2	MMU2_CLK	L3MAIN1_L3_GICLK	PRCM	System MMU2 interface/functional clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
System MMU1	MMU1_RST	CORE_RET_RST	PRCM	System MMU1 hardware reset. This reset is asynchronously applied to the MMU1 internal registers.
System MMU2	MMU2_RST	CORE_RET_RST	PRCM	System MMU2 hardware reset. This reset is asynchronously applied to the MMU2 internal registers.

Table 20-3. MMU Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
System MMU1	MMU1_IRQ	IRQ_CROSSBAR_333	–	System MMU1 interrupt.
System MMU2	MMU2_IRQ	IRQ_CROSSBAR_369	–	System MMU2 interrupt.
No DMA Requests				

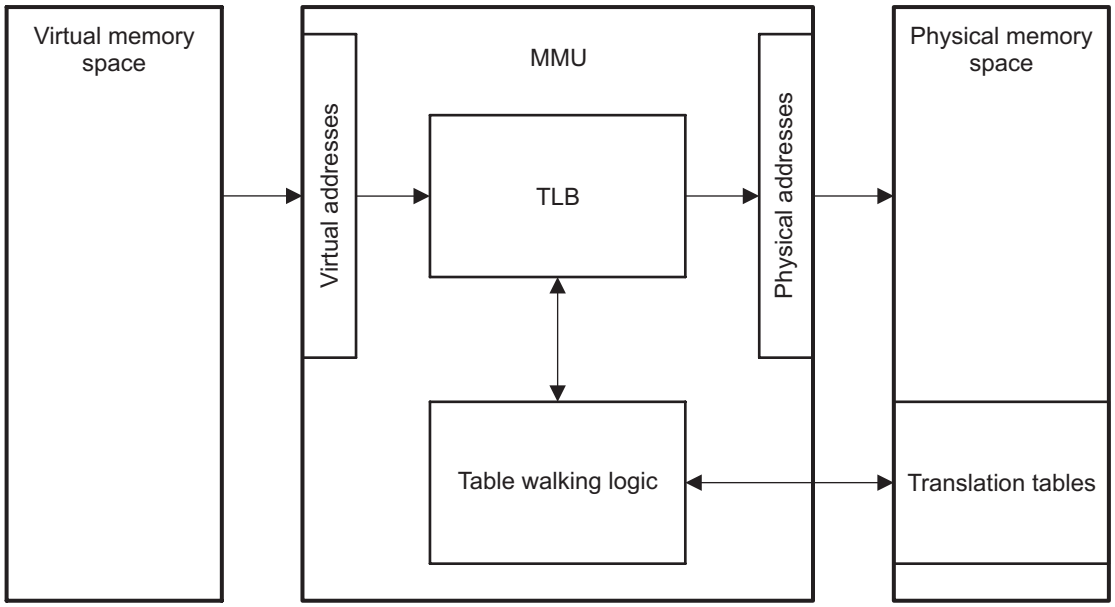
NOTE: For a description of the interrupt sources, see Section 20.3.4, *MMU Interrupt Requests*.

20.3 MMU Functional Description

20.3.1 MMU Block Diagram

The MMU manages the virtual to physical address translation for external addresses. Figure 20-5 shows the MMU block diagram.

Figure 20-5. MMU Block Diagram



Each table entry describes the translation of one contiguous memory region. For a description of the structure of these tables, see Section 20.3.1.2, *Translation Tables*.

Two major functional units exist in the MMU to provide address translation automatically based on the table entries:

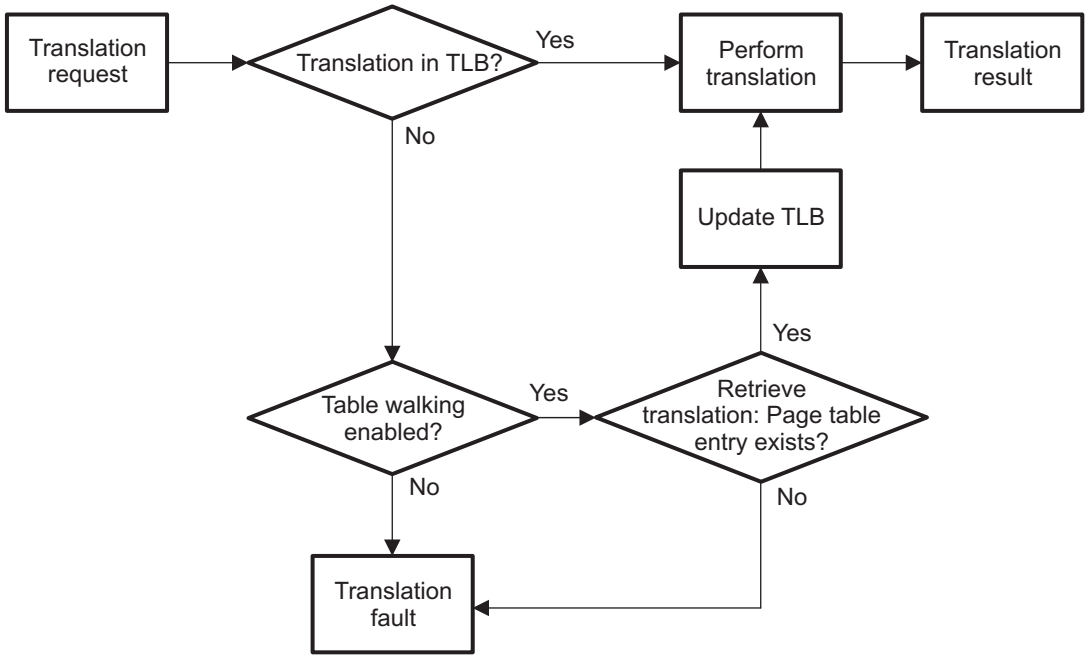
- The table walker automatically retrieves the correct translation table entry for a requested translation. If two-level translation is used (for the translation of small memory pages), the table walker also automatically reads the required second-level translation table entry. The two-level translation is described later in the chapter.
- The translation look-aside buffer (TLB) stores recently used translation entries, acting like a cache of the translation table.

20.3.1.1 MMU Address Translation Process

Whenever an address translation is requested (that is, for every access with the MMU enabled), the MMU first checks whether the translation is contained in the TLB, which acts like a cache storing recent translations. The TLB can also be programmed manually to ensure that time-critical data can be translated without delay.

If the requested translation is not in the TLB, the table-walking logic retrieves this translation from the translation table(s), and then updates the TLB. The address translation is then performed. Figure 20-6 summarizes the process.

Figure 20-6. Translation Process



20.3.1.2 Translation Tables

The translation of virtual to physical addresses is based on entries in translation tables that define the following properties:

- Address translation, that is, the correspondence between virtual and physical addresses
- Size of the memory region the entry translates

The virtual addresses index the translation tables. Each virtual address corresponds to exactly one entry in the translation table.

20.3.1.2.1 Translation Table Hierarchy

When developing a table-based address translation scheme, one of the most important design parameters is the memory page size described by each translation table entry. MMU instances support 4-KiB and 64-KiB pages, a 1-MiB section, and a 16-MiB supersection. Using bigger page sizes means a smaller translation table.

Using a smaller page size greatly increases the efficiency of dynamic memory allocation and defragmentation. That is why many operating systems (OSs) can operate on memory blocks as small as 4 KiB; however, the smaller size implies a more complex table structure.

A quick calculation shows that using 4 KiB memory pages with one translation table would require one million entries to span the entire 4-GiB address range. The table itself would be 32 MiB, a size that is not feasible.

However, using bigger pages reduces the flexibility of typical OS memory management. Implementing a two-level hierarchy reconciles these two requirements. Within this hierarchy, one first-level translation table describes the translation properties based on 1 MiB memory regions.

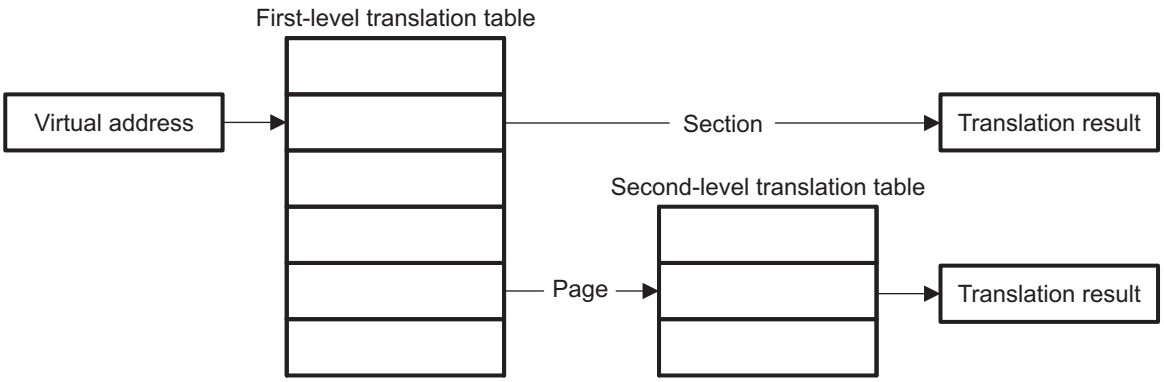
Each of the entries in this first-level translation table can specify the following:

- The translation properties for a big memory section. This memory section can be either 1 MiB (section) or 16 MiB (supersection). In this case, all translation parameters are specified in the first-level translation table entry.
- A pointer to a second-level translation table that specifies individual translation properties based on smaller pages within the 1-MiB page of memory. These pages can be either 64 KiB (large page) or 4

KiB (small page). In this case, the actual translation parameters are specified in the second-level translation table entry. The first-level translation table entry specifies only the base address of the second-level translation table.

This hierarchical approach means that additional translation information for smaller pages must be provided only when the pages are actually used. Figure 20-7 shows the hierarchy.

Figure 20-7. Translation Hierarchy



The structure of the first and second-level translation tables and their entries are described in more detail in Section 20.3.1.2.2, *First-Level Translation Table*, and Section 20.3.1.2.3, *Two-Level Translation*.

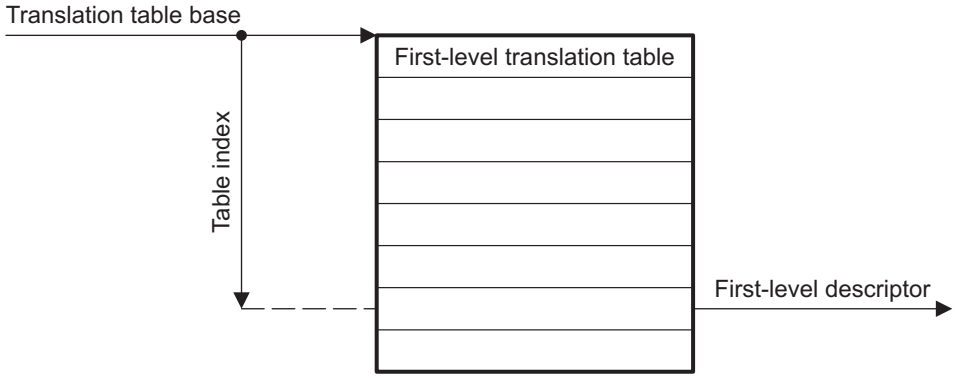
20.3.1.2.2 First-Level Translation Table

The first-level translation table describes the translation properties for 1-MiB sections. To describe a 4-GiB address range requires 4096 32-bit entries (so-called first-level descriptors).

The first-level translation table start address must be aligned on a multiple of the table size with a 128-byte minimum. Consequently, an alignment of at least 16K bytes is required for a complete 4096-entry table; that is, at least the last fourteen address bits must be zero.

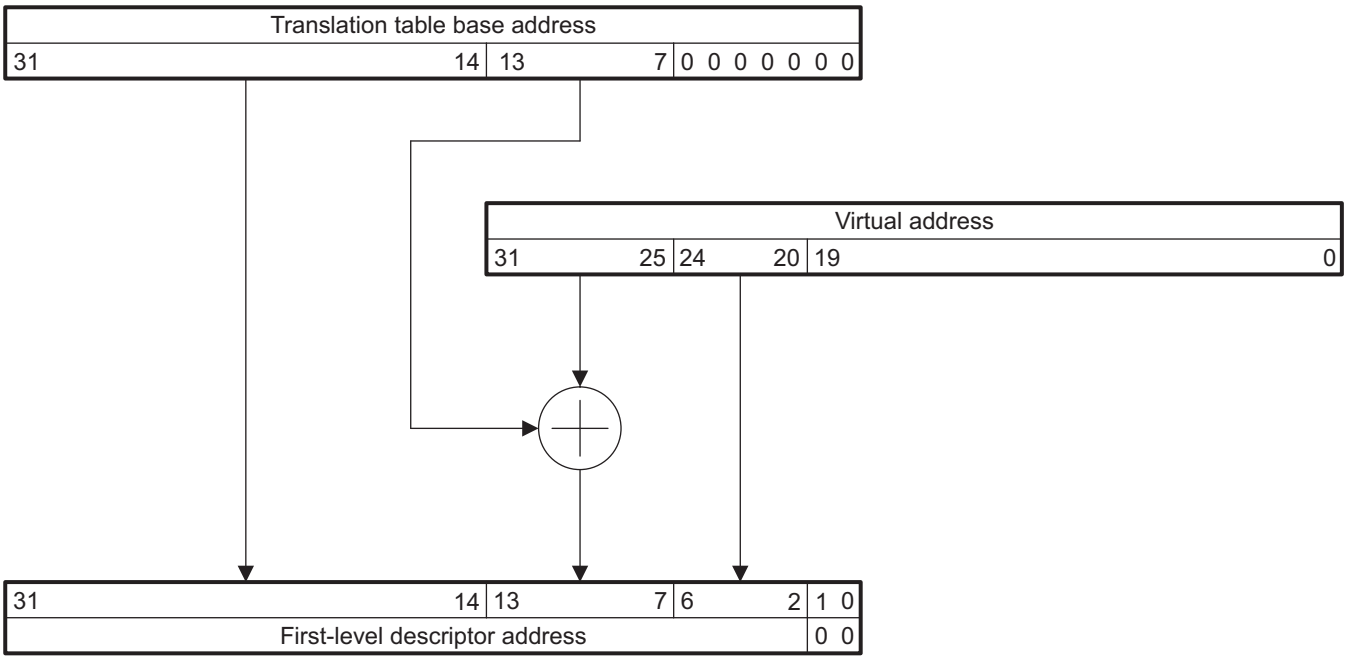
The start address of the first-level translation table is specified by the so-called translation table base. The table is indexed by the upper 12-bits of the virtual address. Figure 20-8 shows this mechanism.

Figure 20-8. First-level Descriptor Address Calculation



To summarize, the translation table base and the translation table index together define the first-level descriptor address. Figure 20-9 outlines the precise mechanism used to calculate this address.

Figure 20-9. Detailed First-Level Descriptor Address Calculation



As an example of this mechanism, consider a translation table base address of 0x8000:0000 and a virtual address of 0x1234:5678. In this case, the first-level descriptor address is 0x8000:0000 + (0x123 << 2) = 0x8000:048C.

20.3.1.2.2.1 First-Level Descriptor Format

Each first-level descriptor provides either the complete address translation for 1-MiB or 16-MiB sections or provides a pointer to a second-level translation table for 4 KiB or 64 KiB pages. Table 20-4 shows the first-level descriptor format.

Table 20-4. First-Level Descriptor Format

First-Level Descriptor Format								
31:24	23:20	19	18	17:10	9:2	1	0	
X						0	0	Fault
Second-Level Translation Table Base Address					X	0	1	Page
Section Base Address		X	0	X		1	0	Section
Supersection Base Address	X		1	X		1	0	Supersection
X						1	1	Fault

X = Don't care. Set to 0 for future compatibility.

20.3.1.2.2.2 First-Level Page Descriptor Format

If a translation granularity smaller than 1 MiB is required, a two-level translation process is used. In this case, the first-level block descriptor specifies only the start address of a second-level translation table. The second-level translation table entries specify the actual translation properties.

20.3.1.2.2.3 First-Level Section Descriptor Format

Each section descriptor in the first-level translation table specifies the complete translation properties for a 1-MiB section or a 16-MiB supersection.

NOTE: Supersection descriptors must be repeated 16 times, because each descriptor in the first-level translation table describes 1 MiB of memory. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

20.3.1.2.2.4 Section Translation Summary

Sections and supersections can be translated based solely on the information in the first-level translation table. Figure 20-10 summarizes the address translation process for a section.

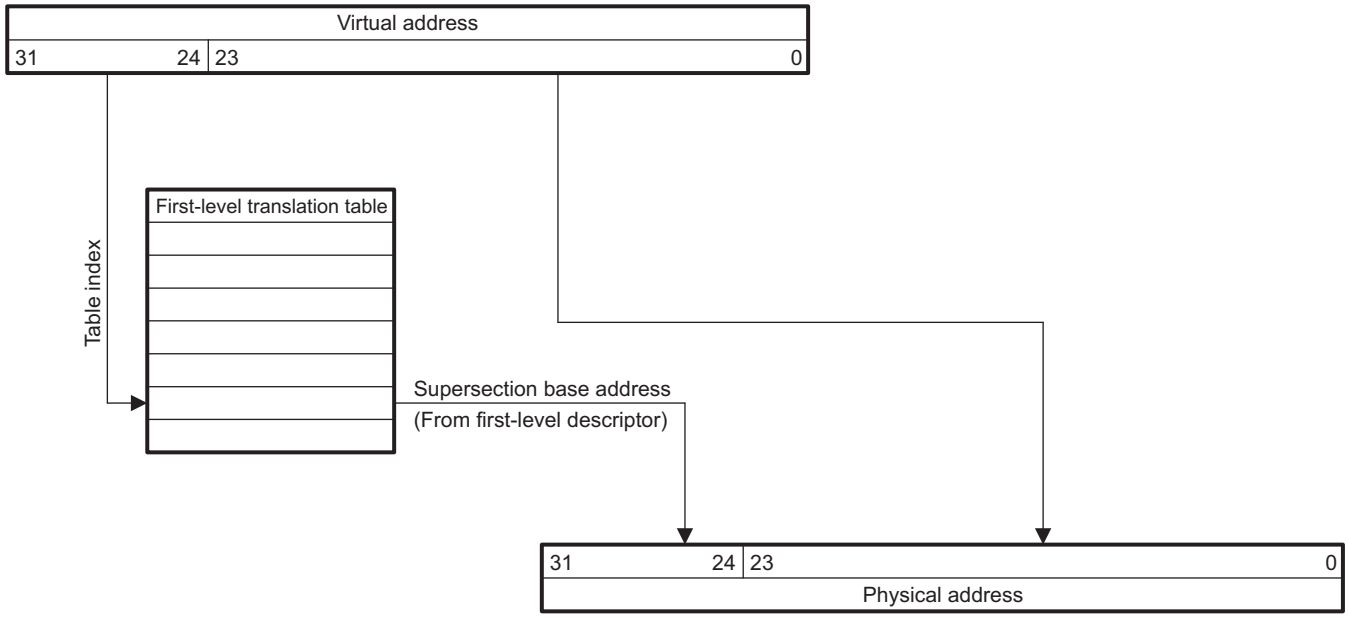
Figure 20-10. Section Translation Summary

20.3.1.2.2.5 Supersection Translation Summary

The translation of a supersection is similar to the translation of a section. The difference is that for a supersection only bits 31 to 24 index into the first-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a supersection.

Figure 20-11 shows the translation mechanism for a supersection.

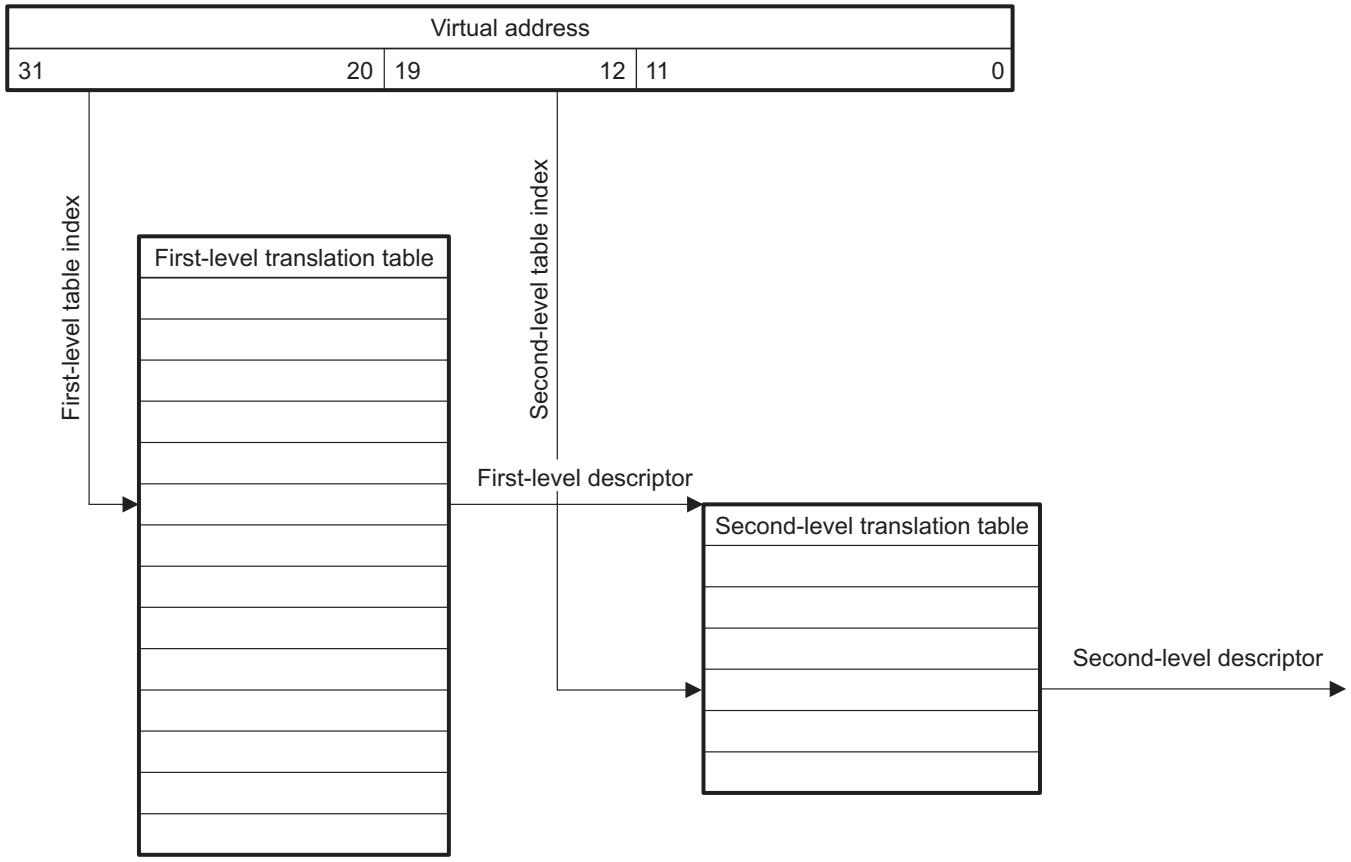
Figure 20-11. Supersection Translation Summary



20.3.1.2.3 Two-Level Translation

Two-level translation is used when fine-grain granularity is required, that is, when memory sections smaller than 1 MiB are needed. In this case, the first-level descriptor provides a pointer to the base address of a second-level translation table. This second-level table is indexed by bits 19 to 12 of the virtual address. Figure 20-12 shows this indexing mechanism.

Figure 20-12. Two-Level Translation



Each second-level translation table describes the translation of 1 MiB of address space in pages of 64 KiB (large page) or 4 KiB (small page). It consists of 256 second-level descriptors describing 4 KiB each.

NOTE: In the case of a large page, the same descriptor must be repeated 16 times. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

20.3.1.2.3.1 Second-Level Descriptor Format

Similar to first-level section descriptors, second-level descriptors provide all of the necessary information for the translation of a large or small page. Table 20-5 shows the format of second-level descriptors.

Table 20-5. Second-Level Descriptor Format

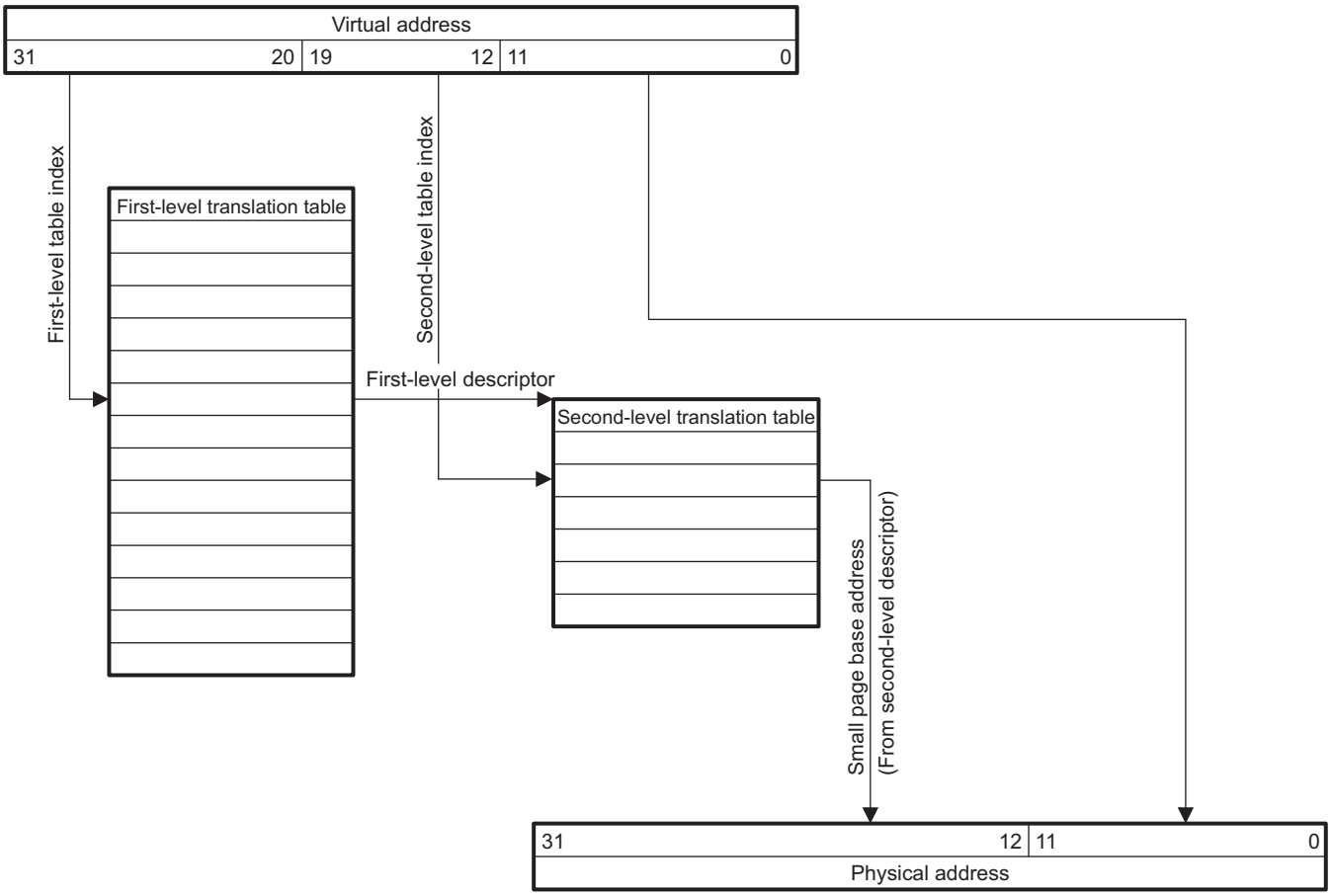
Second-Level Descriptor Format					
31:16	15:12	11:2	1	0	
X			0	0	Fault
Large Page Base Address	X		0	1	Large Page
Small Page Base Address		X	1	X	Small Page

X = Don't care. Set to 0 for future compatibility.

20.3.1.2.3.2 Small Page Translation Summary

Figure 20-13 summarizes the translation process for small pages.

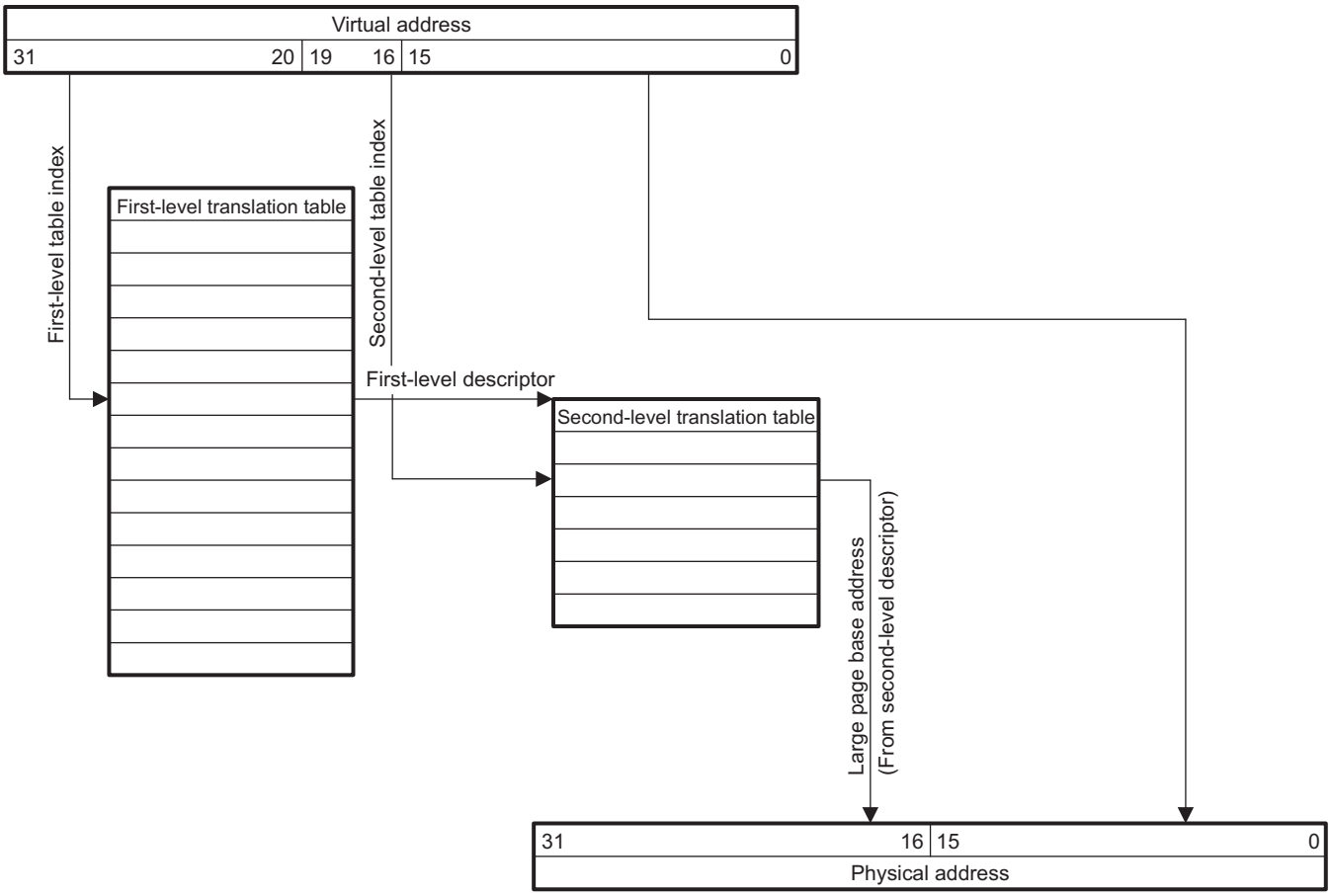
Figure 20-13. Small Page Translation Summary



20.3.1.2.3.3 Large Page Translation Summary

The translation of a large page is similar to the translation of a small page. The difference is that, for a large page, only bits 19 to 16 index into the second-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a large page. This is shown in Figure 20-14.

Figure 20-14. Large Page Translation Summary



20.3.1.3 Translation Lookaside Buffer

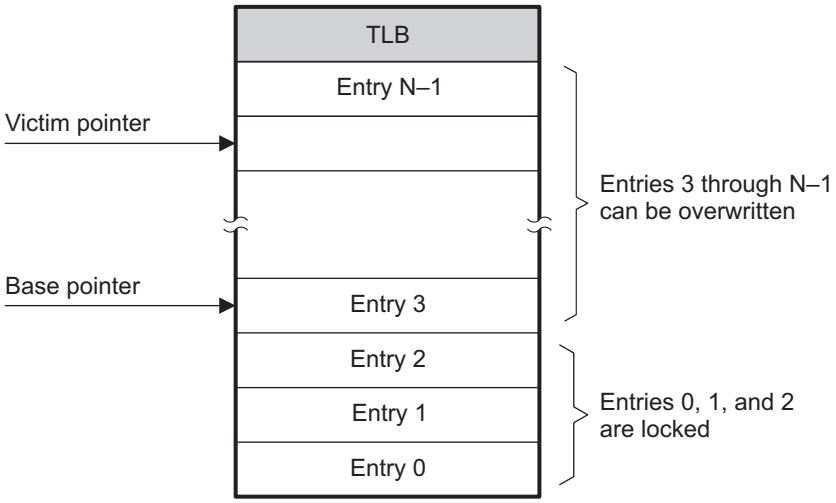
Translating virtual addresses to physical addresses is required for each memory access in systems using an MMU. To accelerate this translation process, a cache, or TLB, holds the result of recent translations.

For every translation, the MMU internal logic first checks whether the requested translation is already cached in the TLB. If the translation is cached, this translation is used; otherwise the translation is retrieved from the translation tables and the TLB is updated. If the TLB is full, one of its entries must be replaced. This entry is selected on a random basis.

The first *n* TLB entries, where *n* < *Total Number N of TLB Entries*, can be protected (locked) against being overwritten by setting the TLB base pointer to *n*. When this mechanism is used, only unprotected entries can be overwritten. The victim pointer indicates the next TLB entry to be written. Figure 20-15 shows an example of the TLB with N TLB entries (ranging from 0 to N-1). The base pointer contains the value "3" protecting Entry 0, Entry 1, and Entry 2 and the victim pointer points to the next TLB entry to be updated.

NOTE: The last TLB entry (Entry N-1) always remains unprotected.

Figure 20-15. TLB Entry Lock Mechanism



The table walking logic automatically writes the TLB entries. The entries can also be manually written, which is done typically to ensure that the translation of time-critical data accesses is already present in the TLB so that they execute as fast as possible. The entries must be locked to prevent them from being overwritten.

20.3.1.3.1 TLB Entry Format

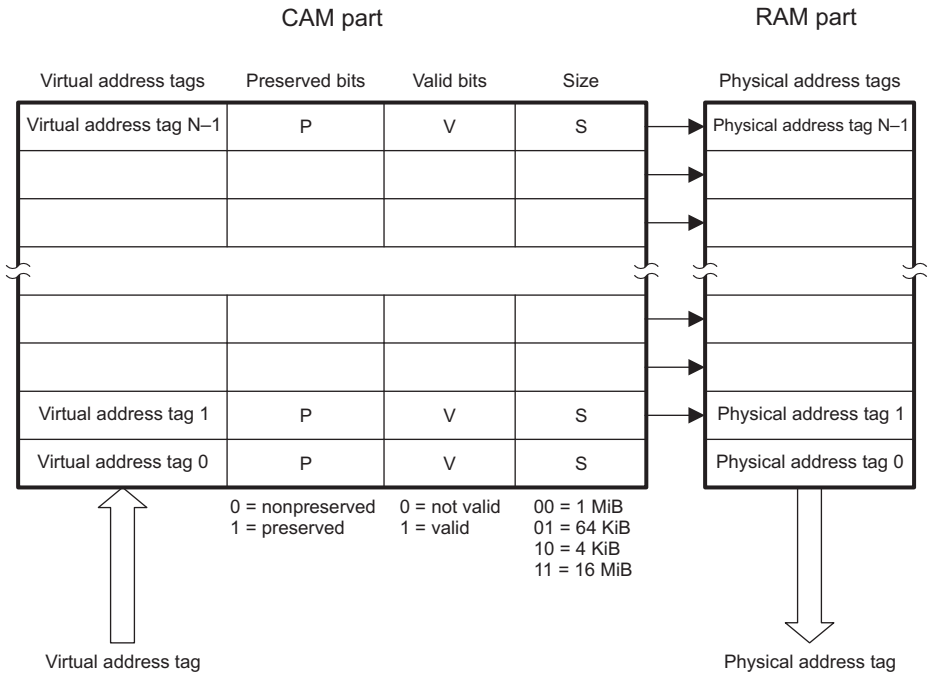
TLB entries consist of two parts:

- The Content Addressable Memory (CAM) part contains the virtual address tag used to determine if a virtual address translation is in the TLB. The TLB acts like a fully associative cache addressed by the virtual address tag. The CAM part also contains the section/page size, as well as the preserved and the valid parameters. See the MMU_CAM register table for more details.
- The Random Addressable Memory (RAM) part contains the address translation that belongs to the virtual address tag. See the MMU_RAM register table for more details.

The valid parameter specifies whether an entry is valid or not. The preserved parameter determines the behavior of an entry in the event of a TLB flush. If an entry is set as preserved, it is not deleted when a TLB is flushed, that is, when MMU_GFLUSH[0] GLOBALFLUSH is set to 1. Preserved entries must be deleted manually. Section 20.3.1.2.2, *First-Level Translation Table* describes the procedure to delete TLB entries.

Figure 20-17 shows the TLB entry structure.

Figure 20-16. TLB Entry Structure



20.3.1.4 No Translation (Bypass) Regions

The MMU provides support for up to four user programmable regions where there is no address translation. Any access to a region specified by the MMU_BYPASS_REGIONx_ADDR and MMU_BYPASS_REGIONx_SIZE registers (where x = 1 to 4) will have no virtual to physical address translation.

20.3.2 MMU Software Reset

To perform a software reset, write 1 in the MMU_SYSCONFIG[1] SOFTRESET bit. The MMU_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the MMU_SYSCONFIG[1] SOFTRESET bit is automatically reset. The software must ensure that the software reset completes before doing MMU operations. When an MMU instance is released from reset, its TLB is empty and the MMU is disabled.

20.3.3 MMU Power Management

Table 20-6 describes the power-management features available for the MMU modules.

Table 20-6. MMU Local Power Management Features

Feature	Register
Idle modes	MMU_SYSCONFIG[4:3] IDLEMODE
Clock activity	MMU_SYSCONFIG[9:8] CLOCKACTIVITY
Clock autogating	MMU_SYSCONFIG[0] AUTOIDLE

NOTE: The MMU_SYSCONFIG[9:8] CLOCKACTIVITY bit field is read only.

20.3.4 MMU Interrupt Requests

Table 20-7. MMU Events

Event Flag	Event Mask	Description
MMU_IRQSTATUS[4] MULTIHITFAULT	MMU_IRQENABLE[4] MULTIHITFAULT	Error due to multiple matches in the TLB
MMU_IRQSTATUS[3] TABLEWALKFAULT	MMU_IRQENABLE[3] TABLEWALKFAULT	Error due to error response received during a Table Walk
MMU_IRQSTATUS[2] EMUMISS	MMU_IRQENABLE[2] EMUMISS	Error due to unrecoverable TLB miss during debug (hardware TWL disabled)
MMU_IRQSTATUS[1] TRANSLATIONFAULT	MMU_IRQENABLE[1] TRANSLATIONFAULT	Error due to invalid descriptor in the translation tables (translation fault)
MMU_IRQSTATUS[0] TLBMISS	MMU_IRQENABLE[0] TLBMISS	Error due to unrecoverable TLB miss (hardware TWL disabled)

20.3.5 MMU Error Handling

Table 20-8 summarizes the intended operation for real and potential error conditions.

Table 20-8. Error Handling

Item	Condition	Action
1	Table-walk read has an error response.	Treat generally the same as a translation fault, but set the TableWalkFault interrupt status bit to aid in diagnosis
2	MMU is disabled during table-walk.	Not permitted; can result in loss of the current transaction but must not deadlock the MMU. Avoid this condition by first disabling the table-walk logic and then polling the TWLRunning bit to ensure that no table walk is pending
3	MMU is disabled during an address translation.	Not permitted; can result in access to an unintended location, but must not deadlock MMU. This condition should be avoided by ensuring that no accesses are pending.
4	TLB is accessed during an address translation or a table walk.	Reading permitted; write should be done with care to ensure that the TLB is self-consistent at all times that a translation can occur.
5	TLB is flushed during address translation or a table walk.	Permitted; the flush is processed first, followed by the TWL update.
6	MMU is disabled while an interrupt is pending.	Not permitted; all pending interrupts should be processed before disabling the MMU.
7	Interrupt is not enabled and a fault/miss happens during translation.	If MMU_GPR[0] FAULT_INTR_DIS = 1 : Error response is sent back. If MMU_GPR[0] FAULT_INTR_DIS = 0 : • Mreqdebug = 1 (debug access) : Error response is sent back • Mreqdebug = 0 (application access) : Error response is not sent. MMU is waiting for TLB to be updated through config port. But since the interrupt is not asserted, system does not know there was a fault. This results in deadlock. Software must take care of this by enabling interrupts.

20.4 MMU Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

20.4.1 Global Initialization

20.4.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the MMU module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMU. For more information, see Section 20.2, *MMU Module Integration* .

Table 20-9. Global Initialization of Surrounding Modules

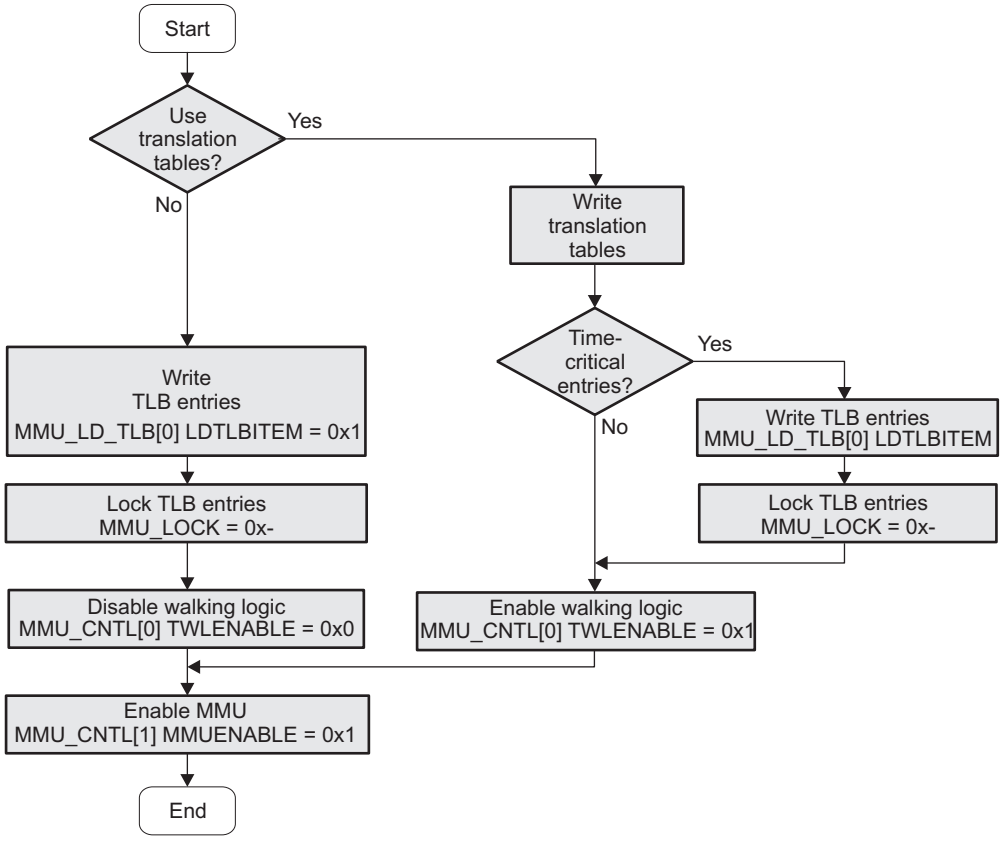
Surrounding Modules	Comments
PRCM	Enable MMU interface/functional clock.
(optional) Interrupt controller(s)	Configure device interrupt controller(s) to enable the interrupts from MMU.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.

20.4.1.2 MMU Global Initialization

20.4.1.2.1 Main Sequence - MMU Global Initialization

Figure 20-17 shows the procedure to initialize the MMU after a power-on or software reset.

Figure 20-17. MMU Global Initialization



20.4.1.2.2 Subsequence - Configure a TLB entry

Table 20-10. Configure a TLB Entry

Step	Register / Bit Field / Programming Model	Value
Load the Virtual Address Tag	MMU_CAM[31:12] VATAG	0x-
Protect the TLB entry against flush	MMU_CAM[3] P	0x1
Validate the TLB entry	MMU_CAM[2] V	0x1
Define the page size	MMU_CAM[1:0] PAGESIZE	0x-

20.4.1.3 Operational Modes Configuration

20.4.1.3.1 Main Sequence - Writing TLB Entries Statically

Writing TLB entries statically avoids the need to write translation tables in memory and is commonly used for relatively small address spaces. This method ensures that the translation of time-critical data accesses execute as fast as possible with entries already present in the TLB. These entries must be locked to prevent them from being overwritten.

Table 20-11. MMU Writing TLB Entries Statically

Step	Register/ Bit Field / Programming Model	Value
Execute software reset	MMU_SYSCONFIG[1] SOFTRESET	0x1
Wait for reset to complete	MMU_SYSSTATUS[0] RESETDONE	=0x1
Enable power saving via automatic interface clock gating	MMU_SYSCONFIG[0] AUTOIDLE	0x1
Configure TLB entries	See Table 20-10	
Load the physical Address of the page	MMU_RAM[31:12] PHYSICALADDRESS	0x-
Specify the TLB entry you want to write	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Load the specified entry in the TLB	MMU_LD_TLB[0] LDTLBITEM	0x1
Enable multihit fault and TLB miss	MMU_IRQENABLE[4] MULTIHITFAULT	0x1
	MMU_IRQENABLE[0] TLBMISS	0x1
Enable memory translations	MMU_CNTL[1] MMUENABLE	0x1

20.4.1.3.2 Main Sequence - Protecting TLB Entries

The first *n* TLB entries (with *n* < total number of TLB entries) can be protected from being overwritten with new translations. This is useful to ensure that certain commonly used or time-critical translations are always in the TLB and do not require retrieval using the table walking process.

Table 20-12. Protecting TLB Entries

Step	Registe/Bit Field/Programming Model	Value
Locks the TLB entries	MMU_LOCK[14:10] BASEVALUE	0x-

20.4.1.3.3 Main Sequence - Deleting TLB Entries

Two mechanisms exist to delete TLB entries. All unpreserved TLB entries, that is, TLB entries written with the preserved bit set to zero, can be deleted by invoking a TLB flush. The preserved bit should only be used on protected TLB entries, as it does not prevent replacement by the table walking logic.

Table 20-13. Deleting TLB Entries

Step	Registet / Bit Field / Programming Model	Value
Flush all nonprotected TLB entries	MMU_GFLUSH[0] GLOBALFLUSH	0x1
Flush all TLB entries specified by the CAM register	MMU_FLUSH_ENTRY[0] FLUSHENTRY	0x1

20.4.1.3.4 Main Sequence - Read TLB Entries

TLB entries can be read by the programmer to determine the TLB content at runtime.

Table 20-14. Read TLB Entries

Step	Register / Bit Field / Programming Model	Value
Set the current victim pointer	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Read RAM parts of the TLB entry	MMU_READ_RAM	

Table 20-14. Read TLB Entries (continued)

Step	Register / Bit Field / Programming Model	Value
Read CAM parts of the TLB entry	MMU_READ_CAM	

20.5 MMU Register Manual

20.5.1 MMU Instance Summary

Table 20-15. MMU Instance Summary

Module Name	Base Address (L3/L4 Access)	Base Address (CPU Private Access)	Size
System MMU1	0x4881 C000	–	176 Bytes
System MMU2	0x4881 E000	–	176 Bytes
DSP1_MMU0	0x40D0 1000	0x01D0 1000	176 Bytes
DSP1_MMU1	0x40D0 2000	0x01D0 2000	176 Bytes
DSP2_MMU0	0x4150 1000	0x01D0 1000	176 Bytes
DSP2_MMU1	0x4150 2000	0x01D0 2000	176 Bytes
EVE1_MMU0(1)	0x4208 1000	0x4008 1000	176 Bytes
EVE1_MMU1(1)	0x4208 2000	0x4008 2000	176 Bytes
EVE2_MMU0(1)	0x4218 1000	0x4008 1000	176 Bytes
EVE2_MMU1(1)	0x4218 2000	0x4008 2000	176 Bytes
EVE3_MMU0(1)	0x4228 1000	0x4008 1000	176 Bytes
EVE3_MMU1(1)	0x4228 2000	0x4008 2000	176 Bytes
EVE4_MMU0(1)	0x4238 1000	0x4008 1000	176 Bytes
EVE4_MMU1(1)	0x4238 2000	0x4008 2000	176 Bytes
IPU1_MMU	0x5888 2000	0x5508 2000	176 Bytes
IPU2_MMU	0x5508 2000	0x5508 2000	176 Bytes

(1) EVE is not supported in this family of devices.

20.5.2 MMU Registers

20.5.2.1 MMU Register Summary

Table 20-16. System MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	System MMU1 Physical Address (L4_PER3 Access)	System MMU2 Physical Address (L4_PER3 Access)
MMU_REVISION	R	32	0x0000 0000	0x4881 C000	0x4881 E000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4881 C010	0x4881 E010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4881 C014	0x4881 E014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4881 C018	0x4881 E018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4881 C01C	0x4881 E01C
MMU_WALKING_ST	R	32	0x0000 0040	0x4881 C040	0x4881 E040
MMU_CNTL	RW	32	0x0000 0044	0x4881 C044	0x4881 E044
MMU_FAULT_AD	R	32	0x0000 0048	0x4881 C048	0x4881 E048
MMU_TTB	RW	32	0x0000 004C	0x4881 C04C	0x4881 E04C
MMU_LOCK	RW	32	0x0000 0050	0x4881 C050	0x4881 E050
MMU_LD_TLB	RW	32	0x0000 0054	0x4881 C054	0x4881 E054
MMU_CAM	RW	32	0x0000 0058	0x4881 C058	0x4881 E058
MMU_RAM	RW	32	0x0000 005C	0x4881 C05C	0x4881 E05C
MMU_GFLUSH	RW	32	0x0000 0060	0x4881 C060	0x4881 E060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4881 C064	0x4881 E064

Table 20-16. System MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	System MMU1 Physical Address (L4_PER3 Access)	System MMU2 Physical Address (L4_PER3 Access)
MMU_READ_CAM	R	32	0x0000 0068	0x4881 C068	0x4881 E068
MMU_READ_RAM	R	32	0x0000 006C	0x4881 C06C	0x4881 E06C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4881 C070	0x4881 E070
RESERVED	R	32	0x0000 0074	0x4881 C074	0x4881 E074
RESERVED	R	32	0x0000 0078	0x4881 C078	0x4881 E078
RESERVED	R	32	0x0000 007C	0x4881 C07C	0x4881 E07C
MMU_FAULT_PC	R	32	0x0000 0080	0x4881 C080	0x4881 E080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4881 C084	0x4881 E084
MMU_GPR	RW	32	0x0000 0088	0x4881 C088	0x4881 E088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4881 C090	0x4881 E090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4881 C094	0x4881 E094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4881 C098	0x4881 E098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4881 C09C	0x4881 E09C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4881 C0A0	0x4881 E0A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4881 C0A4	0x4881 E0A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4881 C0A8	0x4881 E0A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4881 C0AC	0x4881 E0AC

Table 20-17. DSP1 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_MMU0 Physical Address (L3_MAIN Access)	DSP1_MMU0 Physical Address (DSP1 Private Access)	DSP1_MMU1 Physical Address (L3_MAIN Access)	DSP1_MMU1 Physical Address (DSP1 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x40D0 1000	0x01D0 1000	0x40D0 2000	0x01D0 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x40D0 1010	0x01D0 1010	0x40D0 2010	0x01D0 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x40D0 1014	0x01D0 1014	0x40D0 2014	0x01D0 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x40D0 1018	0x01D0 1018	0x40D0 2018	0x01D0 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x40D0 101C	0x01D0 101C	0x40D0 201C	0x01D0 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x40D0 1040	0x01D0 1040	0x40D0 2040	0x01D0 2040
MMU_CNTL	RW	32	0x0000 0044	0x40D0 1044	0x01D0 1044	0x40D0 2044	0x01D0 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x40D0 1048	0x01D0 1048	0x40D0 2048	0x01D0 2048
MMU_TTB	RW	32	0x0000 004C	0x40D0 104C	0x01D0 104C	0x40D0 204C	0x01D0 204C
MMU_LOCK	RW	32	0x0000 0050	0x40D0 1050	0x01D0 1050	0x40D0 2050	0x01D0 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x40D0 1054	0x01D0 1054	0x40D0 2054	0x01D0 2054
MMU_CAM	RW	32	0x0000 0058	0x40D0 1058	0x01D0 1058	0x40D0 2058	0x01D0 2058
MMU_RAM	RW	32	0x0000 005C	0x40D0 105C	0x01D0 105C	0x40D0 205C	0x01D0 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x40D0 1060	0x01D0 1060	0x40D0 2060	0x01D0 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x40D0 1064	0x01D0 1064	0x40D0 2064	0x01D0 2064
MMU_READ_CAM	R	32	0x0000 0068	0x40D0 1068	0x01D0 1068	0x40D0 2068	0x01D0 2068
MMU_READ_RAM	R	32	0x0000 006C	0x40D0 106C	0x01D0 106C	0x40D0 206C	0x01D0 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x40D0 1070	0x01D0 1070	0x40D0 2070	0x01D0 2070
RESERVED	R	32	0x0000 0074	0x40D0 1074	0x01D0 1074	0x40D0 2074	0x01D0 2074
RESERVED	R	32	0x0000 0078	0x40D0 1078	0x01D0 1078	0x40D0 2078	0x01D0 2078

Table 20-17. DSP1 MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_MMU0 Physical Address (L3_MAIN Access)	DSP1_MMU0 Physical Address (DSP1 Private Access)	DSP1_MMU1 Physical Address (L3_MAIN Access)	DSP1_MMU1 Physical Address (DSP1 Private Access)
RESERVED	R	32	0x0000 007C	0x40D0 107C	0x01D0 107C	0x40D0 207C	0x01D0 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x40D0 1080	0x01D0 1080	0x40D0 2080	0x01D0 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x40D0 1084	0x01D0 1084	0x40D0 2084	0x01D0 2084
MMU_GPR	RW	32	0x0000 0088	0x40D0 1088	0x01D0 1088	0x40D0 2088	0x01D0 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x40D0 1090	0x01D0 1090	0x40D0 2090	0x01D0 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x40D0 1094	0x01D0 1094	0x40D0 2094	0x01D0 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x40D0 1098	0x01D0 1098	0x40D0 2098	0x01D0 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x40D0 109C	0x01D0 109C	0x40D0 209C	0x01D0 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x40D0 10A0	0x01D0 10A0	0x40D0 20A0	0x01D0 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x40D0 10A4	0x01D0 10A4	0x40D0 20A4	0x01D0 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x40D0 10A8	0x01D0 10A8	0x40D0 20A8	0x01D0 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x40D0 10AC	0x01D0 10AC	0x40D0 20AC	0x01D0 20AC

Table 20-18. DSP2 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_MMU0 Physical Address (L3_MAIN Access)	DSP2_MMU0 Physical Address (DSP2 Private Access)	DSP2_MMU1 Physical Address (L3_MAIN Access)	DSP2_MMU1 Physical Address (DSP2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4150 1000	0x01D0 1000	0x4150 2000	0x01D0 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4150 1010	0x01D0 1010	0x4150 2010	0x01D0 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4150 1014	0x01D0 1014	0x4150 2014	0x01D0 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4150 1018	0x01D0 1018	0x4150 2018	0x01D0 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4150 101C	0x01D0 101C	0x4150 201C	0x01D0 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4150 1040	0x01D0 1040	0x4150 2040	0x01D0 2040
MMU_CNTL	RW	32	0x0000 0044	0x4150 1044	0x01D0 1044	0x4150 2044	0x01D0 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4150 1048	0x01D0 1048	0x4150 2048	0x01D0 2048
MMU_TTB	RW	32	0x0000 004C	0x4150 104C	0x01D0 104C	0x4150 204C	0x01D0 204C
MMU_LOCK	RW	32	0x0000 0050	0x4150 1050	0x01D0 1050	0x4150 2050	0x01D0 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4150 1054	0x01D0 1054	0x4150 2054	0x01D0 2054
MMU_CAM	RW	32	0x0000 0058	0x4150 1058	0x01D0 1058	0x4150 2058	0x01D0 2058
MMU_RAM	RW	32	0x0000 005C	0x4150 105C	0x01D0 105C	0x4150 205C	0x01D0 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4150 1060	0x01D0 1060	0x4150 2060	0x01D0 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4150 1064	0x01D0 1064	0x4150 2064	0x01D0 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4150 1068	0x01D0 1068	0x4150 2068	0x01D0 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4150 106C	0x01D0 106C	0x4150 206C	0x01D0 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4150 1070	0x01D0 1070	0x4150 2070	0x01D0 2070
RESERVED	R	32	0x0000 0074	0x4150 1074	0x01D0 1074	0x4150 2074	0x01D0 2074
RESERVED	R	32	0x0000 0078	0x4150 1078	0x01D0 1078	0x4150 2078	0x01D0 2078
RESERVED	R	32	0x0000 007C	0x4150 107C	0x01D0 107C	0x4150 207C	0x01D0 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4150 1080	0x01D0 1080	0x4150 2080	0x01D0 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4150 1084	0x01D0 1084	0x4150 2084	0x01D0 2084
MMU_GPR	RW	32	0x0000 0088	0x4150 1088	0x01D0 1088	0x4150 2088	0x01D0 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4150 1090	0x01D0 1090	0x4150 2090	0x01D0 2090

Table 20-18. DSP2 MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_MMU0 Physical Address (L3_MAIN Access)	DSP2_MMU0 Physical Address (DSP2 Private Access)	DSP2_MMU1 Physical Address (L3_MAIN Access)	DSP2_MMU1 Physical Address (DSP2 Private Access)
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4150 1094	0x01D0 1094	0x4150 2094	0x01D0 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4150 1098	0x01D0 1098	0x4150 2098	0x01D0 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4150 109C	0x01D0 109C	0x4150 209C	0x01D0 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4150 10A0	0x01D0 10A0	0x4150 20A0	0x01D0 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4150 10A4	0x01D0 10A4	0x4150 20A4	0x01D0 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4150 10A8	0x01D0 10A8	0x4150 20A8	0x01D0 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4150 10AC	0x01D0 10AC	0x4150 20AC	0x01D0 20AC

Table 20-19. EVE1 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_MMU0 Physical Address (L3_MAIN Access)	EVE1_MMU0 Physical Address (EVE1 Private Access)	EVE1_MMU1 Physical Address (L3_MAIN Access)	EVE1_MMU1 Physical Address (EVE1 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4208 1000	0x4008 1000	0x4208 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4208 1010	0x4008 1010	0x4208 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4208 1014	0x4008 1014	0x4208 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4208 1018	0x4008 1018	0x4208 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4208 101C	0x4008 101C	0x4208 201C	0x4008 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4208 1040	0x4008 1040	0x4208 2040	0x4008 2040
MMU_CNTL	RW	32	0x0000 0044	0x4208 1044	0x4008 1044	0x4208 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4208 1048	0x4008 1048	0x4208 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4208 104C	0x4008 104C	0x4208 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4208 1050	0x4008 1050	0x4208 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4208 1054	0x4008 1054	0x4208 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4208 1058	0x4008 1058	0x4208 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4208 105C	0x4008 105C	0x4208 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4208 1060	0x4008 1060	0x4208 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4208 1064	0x4008 1064	0x4208 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4208 1068	0x4008 1068	0x4208 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4208 106C	0x4008 106C	0x4208 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4208 1070	0x4008 1070	0x4208 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4208 1074	0x4008 1074	0x4208 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4208 1078	0x4008 1078	0x4208 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4208 107C	0x4008 107C	0x4208 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4208 1080	0x4008 1080	0x4208 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4208 1084	0x4008 1084	0x4208 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4208 1088	0x4008 1088	0x4208 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4208 1090	0x4008 1090	0x4208 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4208 1094	0x4008 1094	0x4208 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4208 1098	0x4008 1098	0x4208 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4208 109C	0x4008 109C	0x4208 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4208 10A0	0x4008 10A0	0x4208 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4208 10A4	0x4008 10A4	0x4208 20A4	0x4008 20A4

Table 20-19. EVE1 MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_MMU0 Physical Address (L3_MAIN Access)	EVE1_MMU0 Physical Address (EVE1 Private Access)	EVE1_MMU1 Physical Address (L3_MAIN Access)	EVE1_MMU1 Physical Address (EVE1 Private Access)
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4208 10A8	0x4008 10A8	0x4208 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4208 10AC	0x4008 10AC	0x4208 20AC	0x4008 20AC

Table 20-20. EVE2 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_MMU0 Physical Address (L3_MAIN Access)	EVE2_MMU0 Physical Address (EVE2 Private Access)	EVE2_MMU1 Physical Address (L3_MAIN Access)	EVE2_MMU1 Physical Address (EVE2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4218 1000	0x4008 1000	0x4218 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4218 1010	0x4008 1010	0x4218 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4218 1014	0x4008 1014	0x4218 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4218 1018	0x4008 1018	0x4218 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4218 101C	0x4008 101C	0x4218 201C	0x4008 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4218 1040	0x4008 1040	0x4218 2040	0x4008 2040
MMU_CNTL	RW	32	0x0000 0044	0x4218 1044	0x4008 1044	0x4218 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4218 1048	0x4008 1048	0x4218 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4218 104C	0x4008 104C	0x4218 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4218 1050	0x4008 1050	0x4218 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4218 1054	0x4008 1054	0x4218 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4218 1058	0x4008 1058	0x4218 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4218 105C	0x4008 105C	0x4218 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4218 1060	0x4008 1060	0x4218 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4218 1064	0x4008 1064	0x4218 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4218 1068	0x4008 1068	0x4218 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4218 106C	0x4008 106C	0x4218 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4218 1070	0x4008 1070	0x4218 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4218 1074	0x4008 1074	0x4218 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4218 1078	0x4008 1078	0x4218 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4218 107C	0x4008 107C	0x4218 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4218 1080	0x4008 1080	0x4218 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4218 1084	0x4008 1084	0x4218 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4218 1088	0x4008 1088	0x4218 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4218 1090	0x4008 1090	0x4218 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4218 1094	0x4008 1094	0x4218 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4218 1098	0x4008 1098	0x4218 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4218 109C	0x4008 109C	0x4218 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4218 10A0	0x4008 10A0	0x4218 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4218 10A4	0x4008 10A4	0x4218 20A4	0x4008 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4218 10A8	0x4008 10A8	0x4218 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4218 10AC	0x4008 10AC	0x4218 20AC	0x4008 20AC

Table 20-21. EVE3 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE3_MMU0 Physical Address (L3_MAIN Access)	EVE3_MMU0 Physical Address (EVE3 Private Access)	EVE3_MMU1 Physical Address (L3_MAIN Access)	EVE3_MMU1 Physical Address (EVE3 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4228 1000	0x4008 1000	0x4228 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4228 1010	0x4008 1010	0x4228 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4228 1014	0x4008 1014	0x4228 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4228 1018	0x4008 1018	0x4228 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4228 101C	0x4008 101C	0x4228 201C	0x4008 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4228 1040	0x4008 1040	0x4228 2040	0x4008 2040
MMU_CNTL	RW	32	0x0000 0044	0x4228 1044	0x4008 1044	0x4228 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4228 1048	0x4008 1048	0x4228 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4228 104C	0x4008 104C	0x4228 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4228 1050	0x4008 1050	0x4228 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4228 1054	0x4008 1054	0x4228 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4228 1058	0x4008 1058	0x4228 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4228 105C	0x4008 105C	0x4228 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4228 1060	0x4008 1060	0x4228 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4228 1064	0x4008 1064	0x4228 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4228 1068	0x4008 1068	0x4228 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4228 106C	0x4008 106C	0x4228 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4228 1070	0x4008 1070	0x4228 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4228 1074	0x4008 1074	0x4228 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4228 1078	0x4008 1078	0x4228 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4228 107C	0x4008 107C	0x4228 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4228 1080	0x4008 1080	0x4228 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4228 1084	0x4008 1084	0x4228 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4228 1088	0x4008 1088	0x4228 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4228 1090	0x4008 1090	0x4228 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4228 1094	0x4008 1094	0x4228 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4228 1098	0x4008 1098	0x4228 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4228 109C	0x4008 109C	0x4228 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4228 10A0	0x4008 10A0	0x4228 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4228 10A4	0x4008 10A4	0x4228 20A4	0x4008 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4228 10A8	0x4008 10A8	0x4228 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4228 10AC	0x4008 10AC	0x4228 20AC	0x4008 20AC

Table 20-22. EVE4 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE4_MMU0 Physical Address (L3_MAIN Access)	EVE4_MMU0 Physical Address (EVE4 Private Access)	EVE4_MMU1 Physical Address (L3_MAIN Access)	EVE4_MMU1 Physical Address (EVE4 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4238 1000	0x4008 1000	0x4238 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4238 1010	0x4008 1010	0x4238 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4238 1014	0x4008 1014	0x4238 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4238 1018	0x4008 1018	0x4238 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4238 101C	0x4008 101C	0x4238 201C	0x4008 201C

Table 20-22. EVE4 MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE4_MMU0 Physical Address (L3_MAIN Access)	EVE4_MMU0 Physical Address (EVE4 Private Access)	EVE4_MMU1 Physical Address (L3_MAIN Access)	EVE4_MMU1 Physical Address (EVE4 Private Access)
MMU_WALKING_ST	R	32	0x0000 0040	0x4238 1040	0x4008 1040	0x4238 2040	0x4008 2040
MMU_CNTL	RW	32	0x0000 0044	0x4238 1044	0x4008 1044	0x4238 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4238 1048	0x4008 1048	0x4238 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4238 104C	0x4008 104C	0x4238 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4238 1050	0x4008 1050	0x4238 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4238 1054	0x4008 1054	0x4238 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4238 1058	0x4008 1058	0x4238 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4238 105C	0x4008 105C	0x4238 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4238 1060	0x4008 1060	0x4238 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4238 1064	0x4008 1064	0x4238 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4238 1068	0x4008 1068	0x4238 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4238 106C	0x4008 106C	0x4238 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4238 1070	0x4008 1070	0x4238 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4238 1074	0x4008 1074	0x4238 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4238 1078	0x4008 1078	0x4238 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4238 107C	0x4008 107C	0x4238 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4238 1080	0x4008 1080	0x4238 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4238 1084	0x4008 1084	0x4238 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4238 1088	0x4008 1088	0x4238 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4238 1090	0x4008 1090	0x4238 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4238 1094	0x4008 1094	0x4238 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4238 1098	0x4008 1098	0x4238 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4238 109C	0x4008 109C	0x4238 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4238 10A0	0x4008 10A0	0x4238 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4238 10A4	0x4008 10A4	0x4238 20A4	0x4008 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4238 10A8	0x4008 10A8	0x4238 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4238 10AC	0x4008 10AC	0x4238 20AC	0x4008 20AC

Table 20-23. IPU MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU1_MMU Physical Address (L3_MAIN Access)	IPU1_MMU Physical Address (IPU1 Private Access)	IPU2_MMU Physical Address (L3_MAIN Access)	IPU2_MMU Physical Address (IPU2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x5888 2000	0x5508 2000	0x5508 2000	0x5508 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x5888 2010	0x5508 2010	0x5508 2010	0x5508 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x5888 2014	0x5508 2014	0x5508 2014	0x5508 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x5888 2018	0x5508 2018	0x5508 2018	0x5508 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x5888 201C	0x5508 201C	0x5508 201C	0x5508 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x5888 2040	0x5508 2040	0x5508 2040	0x5508 2040
MMU_CNTL	RW	32	0x0000 0044	0x5888 2044	0x5508 2044	0x5508 2044	0x5508 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x5888 2048	0x5508 2048	0x5508 2048	0x5508 2048
MMU_TTB	RW	32	0x0000 004C	0x5888 204C	0x5508 204C	0x5508 204C	0x5508 204C
MMU_LOCK	RW	32	0x0000 0050	0x5888 2050	0x5508 2050	0x5508 2050	0x5508 2050

Table 20-23. IPU MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IPU1_MMU Physical Address (L3_MAIN Access)	IPU1_MMU Physical Address (IPU1 Private Access)	IPU2_MMU Physical Address (L3_MAIN Access)	IPU2_MMU Physical Address (IPU2 Private Access)
MMU_LD_TLB	RW	32	0x0000 0054	0x5888 2054	0x5508 2054	0x5508 2054	0x5508 2054
MMU_CAM	RW	32	0x0000 0058	0x5888 2058	0x5508 2058	0x5508 2058	0x5508 2058
MMU_RAM	RW	32	0x0000 005C	0x5888 205C	0x5508 205C	0x5508 205C	0x5508 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x5888 2060	0x5508 2060	0x5508 2060	0x5508 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x5888 2064	0x5508 2064	0x5508 2064	0x5508 2064
MMU_READ_CAM	R	32	0x0000 0068	0x5888 2068	0x5508 2068	0x5508 2068	0x5508 2068
MMU_READ_RAM	R	32	0x0000 006C	0x5888 206C	0x5508 206C	0x5508 206C	0x5508 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x5888 2070	0x5508 2070	0x5508 2070	0x5508 2070
RESERVED	R	32	0x0000 0074	0x5888 2074	0x5508 2074	0x5508 2074	0x5508 2074
RESERVED	R	32	0x0000 0078	0x5888 2078	0x5508 2078	0x5508 2078	0x5508 2078
RESERVED	R	32	0x0000 007C	0x5888 207C	0x5508 207C	0x5508 207C	0x5508 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x5888 2080	0x5508 2080	0x5508 2080	0x5508 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x5888 2084	0x5508 2084	0x5508 2084	0x5508 2084
MMU_GPR	RW	32	0x0000 0088	0x5888 2088	0x5508 2088	0x5508 2088	0x5508 2088

20.5.2.2 MMU Register Description

Table 20-24. MMU_REVISION

Address Offset	0x0000 0000		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the IP revision code		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See (1)

(1) TI internal data

Table 20-25. Register Call Summary for Register MMU_REVISION

MMU Register Manual
- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]

Table 20-26. MMU_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register controls the various parameters of the OCP interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLOCKACTIVITY		RESERVED				IDLEMODE		RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x000000
9:8	CLOCKACTIVITY	Clock activity during wake-up mode	R	0x0
		0x0: Functional and OCP clocks can be switched off		
7:5	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
4:3	IDLEMODE	Idle mode	RW	0x0
		0x0: Force-idle. An idle request is acknowledged unconditionally		
		0x1: No-idle. An idle request is never acknowledged		
		0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module		
		0x3: Reserved. Do not use		
2	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
1	SOFTRESET	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0	W	0x0
		Write 0x0: No functional effect		
		Write 0x1: The module is reset		
0	AUTOIDLE	Internal OCP clock gating strategy	RW	0x0
		0x0: OCP clock is free-running		
		0x1: Automatic interconnect clock gating strategy is applied, based on the interconnect interface activity		

Table 20-27. Register Call Summary for Register MMU_SYSCONFIG

MMU Functional Description
- MMU Software Reset: [0] [1]
- MMU Power Management: [2] [3] [4] [5]

MMU Low-level Programming Models
- Operational Modes Configuration: [6] [7]

MMU Register Manual
- MMU Register Summary: [8] [9] [10] [11] [12] [13] [14] [15]

Table 20-28. MMU_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register provides status information about the module, excluding the interrupt status information		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x000000
0	RESETDONE	Internal reset monitoring	R	-
		Read 0x0: Internal module reset in on-going		
		Read 0x1: Reset completed		

Table 20-29. Register Call Summary for Register MMU_SYSSTATUS

MMU Functional Description • MMU Software Reset: [0]
MMU Low-level Programming Models • Operational Modes Configuration: [1]
MMU Register Manual • MMU Register Summary: [2] [3] [4] [5] [6] [7] [8] [9]

Table 20-30. MMU_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											MULTIHITFAULT	TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT	TLBMISS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB	RW (W1toClr)	0x0
		Read 0x0: MultiHitFault false		
		Write 0x0: MultiHitFault status bit unchanged		
		Write 0x1: MultiHitFault status bit is reset		
		Read 0x1: MultiHitFault is true ('pending')		

Bits	Field Name	Description	Type	Reset
3	TABLEWALKFAULT	Error response received during a Table Walk Read 0x0: TableWalkFault false Write 0x0: TableWalkFault status bit unchanged Write 0x1: TableWalkFault status bit is reset Read 0x1: TableWalkFault is true ('pending')	RW (W1toClr)	0x0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) Read 0x0: EMUMiss false Write 0x0: EMUMiss status bit unchanged Write 0x1: EMUMiss status bit is reset Read 0x1: EMUMiss is true ('pending')	RW (W1toClr)	0x0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) Read 0x0: TranslationFault false Write 0x0: TranslationFault status bit unchanged Write 0x1: TranslationFault status bit is reset Read 0x1: TranslationFault is true ('pending')	RW (W1toClr)	0x0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) Read 0x0: TLBMiss false Write 0x0: TLBMiss status bit unchanged Write 0x1: TLBMiss status bit is reset Read 0x1: TLBMiss is true ('pending')	RW (W1toClr)	0x0

Table 20-31. Register Call Summary for Register MMU_IRQSTATUS

MMU Functional Description
- MMU Interrupt Requests: [0] [1] [2] [3] [4]

MMU Register Manual
- MMU Register Summary: [5] [6] [7] [8] [9] [10] [11] [12]

Table 20-32. MMU_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											MULTIHITFAULT	TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT	TLBMISS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB 0x0: MultiHitFault is masked 0x1: MultiHitFault event generates an interrupt if occurs	RW	0x0

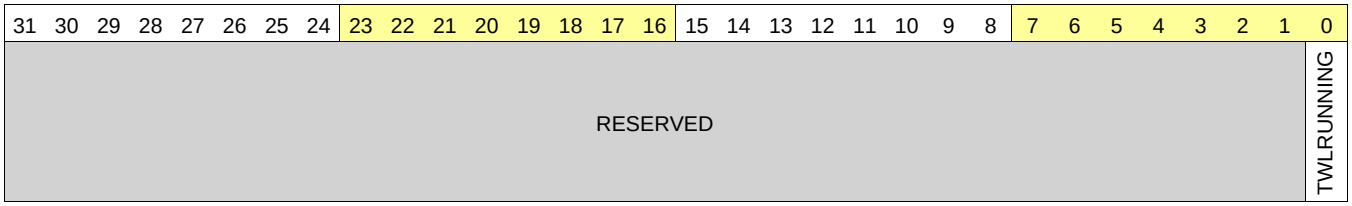
Bits	Field Name	Description	Type	Reset
3	TABLEWALKFAULT	Error response received during a Table Walk 0x0: TableWalkFault is masked 0x1: TableWalkFault event generates an interrupt if occurs	RW	0x0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) 0x0: EMUMiss interrupt is masked 0x1: EMUMiss event generates an interrupt when it occurs	RW	0x0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) 0x0: TranslationFault is masked 0x1: TranslationFault event generates an interrupt if occurs	RW	0x0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) 0x0: TLBMiss interrupt is masked 0x1: TLBMiss event generates an interrupt when if occurs	RW	0x0

Table 20-33. Register Call Summary for Register MMU_IRQENABLE

MMU Functional Description • MMU Interrupt Requests: [0] [1] [2] [3] [4]
MMU Low-level Programming Models • Operational Modes Configuration: [5] [6]
MMU Register Manual • MMU Register Summary: [7] [8] [9] [10] [11] [12] [13] [14]

Table 20-34. MMU_WALKING_ST

Address Offset	0x0000 0040		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register provides status information about the table walking logic		
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0	R	0x0000 0000
0	TWLRUNNING	Table Walking Logic is running Read 0x0: TWL Completed Read 0x1: TWL Running	R	0x0

Table 20-35. Register Call Summary for Register MMU_WALKING_ST

MMU Register Manual • MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]

Table 20-36. MMU_CNTL

Address Offset	0x0000 0044		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register programs the MMU features		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EMUTLBUPDATE	TWLENABLE	MMUENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000000
3	EMUTLBUPDATE	Enable TLB update on emulator table walk 0x0: Emulator TLB update disabled 0x1: Emulator TLB update enabled	RW	0x0
2	TWLENABLE	Table Walking Logic enable 0x0: TWL disabled 0x1: TWL enabled	RW	0x0
1	MMUENABLE	MMU enable 0x0: MMU disabled 0x1: MMU enabled	RW	0x0
0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 20-37. Register Call Summary for Register MMU_CNTL

MMU Low-level Programming Models
- Operational Modes Configuration: [0]

MMU Register Manual
- MMU Register Summary: [1] [2] [3] [4] [5] [6] [7] [8]

Table 20-38. MMU_FAULT_AD

Address Offset	0x0000 0048		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the virtual address that generated the interrupt		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAULTADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	FAULTADDRESS	Virtual address of the access that generated a fault	R	0x0000 0000

Table 20-39. Register Call Summary for Register MMU_FAULT_AD

MMU Register Manual
- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]

Table 20-40. MMU_TTB

Address Offset	0x0000 004C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the Translation Table Base address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTBADDRESS																								RESERVED							

Bits	Field Name	Description	Type	Reset
31:7	TTBADDRESS	Translation Table Base Address	RW	0x0000000
6:0	RESERVED	Write 0's for future compatibility. Reads retiurn 0	R	0x00

Table 20-41. Register Call Summary for Register MMU_TTB

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]

Table 20-42. MMU_LOCK

Address Offset	0x0000 0050		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register locks some of the TLB entries		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASEVALUE						RESERVED		CURRENTVICTIM				RESERVED			

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00000
14:10	BASEVALUE	Locked entries base value.	RW	0x00
9	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
8:4	CURRENTVICTIM	Current entry to be updated either by the TWL or by the software. Write value : TLB entry to be updated by software Read value : TLB entry that will be updated by table walk logic. This will be same as BASEVALUE when there are no tablewalks.	RW	0x00
3:0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 20-43. Register Call Summary for Register MMU_LOCK

MMU Low-level Programming Models

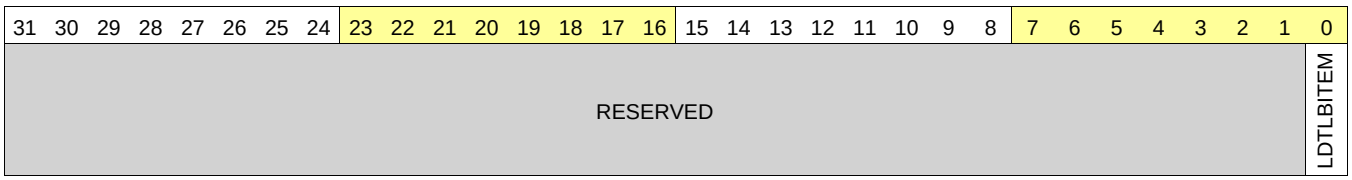
- Operational Modes Configuration: [0] [1] [2]

MMU Register Manual

- MMU Register Summary: [3] [4] [5] [6] [7] [8] [9] [10]
- MMU Register Description: [12] [13]

Table 20-44. MMU_LD_TLB

Address Offset	0x0000 0054		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register loads a TLB entry (CAM+RAM)		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	LDTLBITEM	Write (load) data in the TLB. Reads return 0. Write 0x0: No functional effect Write 0x1: Load TLB data	W	0x0

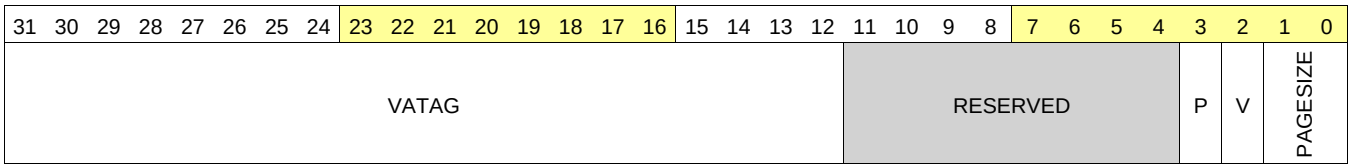
Table 20-45. Register Call Summary for Register MMU_LD_TLB

MMU Low-level Programming Models
- Operational Modes Configuration: [0]

MMU Register Manual
- MMU Register Summary: [1] [2] [3] [4] [5] [6] [7] [8]

Table 20-46. MMU_CAM

Address Offset	0x0000 0058		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register holds a CAM entry		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	RW	0x00000
11:4	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
3	P	Preserved bit 0x0: TLB entry may be flushed 0x1: TLB entry is protected against flush	RW	0x0
2	V	Valid bit 0x0: TLB entry is invalid 0x1: TLB entry is valid	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	PAGESIZE	Page size 0x0: Section (1 MiB) 0x1: Large page (64 KiB) 0x2: Small page (4 KiB) 0x3: Supersection (16 MiB)	RW	0x0

Table 20-47. Register Call Summary for Register MMU_CAM

MMU Functional Description • Translation Lookaside Buffer: [0]
MMU Low-level Programming Models • MMU Global Initialization: [1] [2] [3] [4]
MMU Register Manual • MMU Register Summary: [5] [6] [7] [8] [9] [10] [11] [12] • MMU Register Description: [14]

Table 20-48. MMU_RAM

Address Offset	0x0000 005C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains bits [31:12] of the physical address to be written to a TLB entry pointed to by CURRENTVICTIM field of MMU_LOCK register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYSICALADDRESS																				RESERVED											

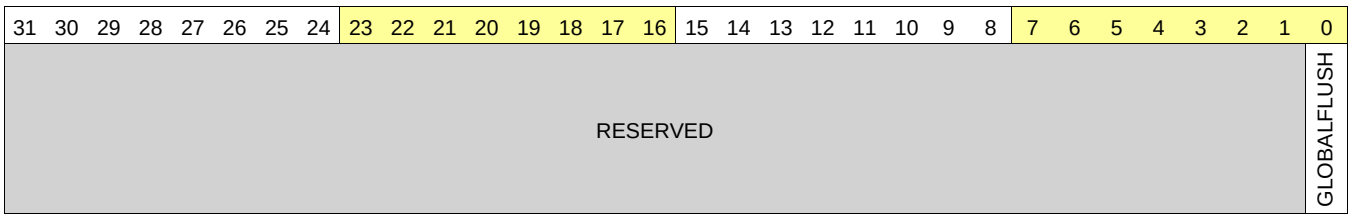
Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	RW	0x00000
11:0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 20-49. Register Call Summary for Register MMU_RAM

MMU Functional Description • Translation Lookaside Buffer: [0]
MMU Low-level Programming Models • Operational Modes Configuration: [1]
MMU Register Manual • MMU Register Summary: [2] [3] [4] [5] [6] [7] [8] [9]

Table 20-50. MMU_GFLUSH

Address Offset	0x0000 0060		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register flushes all the non-protected TLB entries		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	GLOBALFLUSH	Flush all the non-protected TLB entries when set. Reads return 0. Write 0x0: No functional effect Write 0x1: Flush all the non-protected TLB entries	W	0x0

Table 20-51. Register Call Summary for Register MMU_GFLUSH

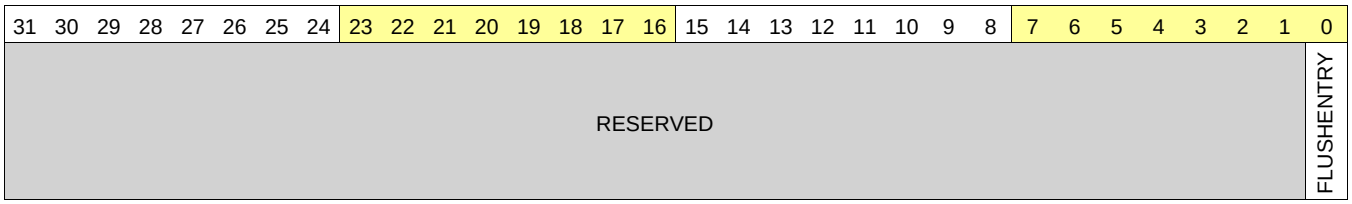
MMU Functional Description
- Translation Lookaside Buffer: [0]

MMU Low-level Programming Models
- Operational Modes Configuration: [1]

MMU Register Manual
- MMU Register Summary: [2] [3] [4] [5] [6] [7] [8] [9]

Table 20-52. MMU_FLUSH_ENTRY

Address Offset	0x0000 0064		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register flushes the entry pointed to by the CAM virtual address		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	FLUSHENTRY	Flush the TLB entry pointed by the virtual address (VATag) in MMU_CAM register, even if this entry is set protected. Reads return 0. Write 0x0: No functional effect Write 0x1: Flush all the TLB entries specified by the CAM register	W	0x0

Table 20-53. Register Call Summary for Register MMU_FLUSH_ENTRY

MMU Low-level Programming Models • Operational Modes Configuration: [0]
MMU Register Manual • MMU Register Summary: [1] [2] [3] [4] [5] [6] [7] [8]

Table 20-54. MMU_READ_CAM

Address Offset	0x0000 0068		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register reads CAM data from a CAM entry		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VATAG																RESERVED								P	V	PAGESIZE					

Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	R	0x00000
11:4	RESERVED	Reads return 0	R	0x00
3	P	Preserved bit	R	0x0
		Read 0x0: TLB entry may be flushed		
		Read 0x1: TLB entry is protected against flush		
2	V	Valid bit	R	0x0
		Read 0x0: TLB entry is invalid		
		Read 0x1: TLB entry is valid		
1:0	PAGESIZE	Page size	R	0x0
		Read 0x0: Section (1 MiB)		
		Read 0x1: Large page (64 KiB)		
		Read 0x2: Small page (4 KiB)		
		Read 0x3: Supersection (16 MiB)		

Table 20-55. Register Call Summary for Register MMU_READ_CAM

MMU Low-level Programming Models • Operational Modes Configuration: [0]
MMU Register Manual • MMU Register Summary: [1] [2] [3] [4] [5] [6] [7] [8]

Table 20-56. MMU_READ_RAM

Address Offset	0x0000 006C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register reads bits [31:12] of the physical address from the TLB entry pointed to by CURRENTVICTIM field of the MMU_LOCK register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYSICALADDRESS																				RESERVED											

Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	R	0x00000
11:0	RESERVED	Reads return 0	R	0x0

Table 20-57. Register Call Summary for Register MMU_READ_RAM

MMU Low-level Programming Models
- Operational Modes Configuration: [0]

MMU Register Manual
- MMU Register Summary: [1] [2] [3] [4] [5] [6] [7] [8]

Table 20-58. MMU_EMU_FAULT_AD

Address Offset	0x0000 0070		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the last virtual address of a fault caused by the debugger		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMUFAULTADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	EMUFAULTADDRESS	Virtual address of the last emulator access that generated a fault	R	0x0000 0000

Table 20-59. Register Call Summary for Register MMU_EMU_FAULT_AD

MMU Register Manual
- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]
- MMU Register Description: [9]

Table 20-60. MMU_FAULT_PC

Address Offset	0x0000 0080		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	Typically CPU program counter value of instruction generating MMU fault. The address value is captured at MMU_EMU_FAULT_AD[31:0] EMUFAULTADDRESS. Data-Read-access : corresponding PC. Data-write-access : not perfect accuracy due to posted-write.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC																															

Bits	Field Name	Description	Type	Reset
31:0	PC	Typically CPU program counter value of instruction generating MMU fault	R	0x0000 0000

Table 20-61. Register Call Summary for Register MMU_FAULT_PC

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]

Table 20-62. MMU_FAULT_STATUS

Address Offset	0x0000 0084		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	Fault status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MMU_FAULT_TRANS_ID				RD_WR	MMU_FAULT_TYPE		FAULTINDICATION

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8:4	MMU_FAULT_TRANS_ID	MtagID of the transaction that caused fault	R	0x0
3	RD_WR	Indicates read or write 0x0: Write 0x1: Read	R	0x0
2:1	MMU_FAULT_TYPE	MReqInfo[1:0] is captured as fault type	R	0x0
0	FAULTINDICATION	Indicates an MMU fault	RW (W1toClr)	0x0

Table 20-63. Register Call Summary for Register MMU_FAULT_STATUS

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6] [7]

Table 20-64. MMU_GPR

Address Offset	0x0000 0088		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	General purpose register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPO																RESERVED															FAULT_INTR_DIS

Bits	Field Name	Description	Type	Reset
31:16	GPO	General purpose output sent out as MMU output	RW	0x0000
15:1	RESERVED	Reserved	R	0x0000
0	FAULT_INTR_DIS	Disable generation of interrupt on fault. Error response is returned instead on the slave port	RW	0x0

Table 20-65. Register Call Summary for Register MMU_GPR

MMU Functional Description
- MMU Error Handling: [0] [1]

MMU Register Manual
- MMU Register Summary: [2] [3] [4] [5] [6] [7] [8] [9]

Table 20-66. MMU_BYPASS_REGION1_ADDR

Address Offset	0x0000 0090		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the first NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION for 2D bursts. This has to be aligned to SIZE in MMU_BYPASS_REGION1_SIZE	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-67. Register Call Summary for Register MMU_BYPASS_REGION1_ADDR

MMU Register Manual
- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]

Table 20-68. MMU_BYPASS_REGION1_SIZE

Address Offset	0x0000 0094		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of first NO TRANSLATION REGION for		

Table 20-68. MMU_BYPASS_REGION1_SIZE (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION of 2D bursts. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-69. Register Call Summary for Register MMU_BYPASS_REGION1_SIZE

MMU Register Manual
- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]
- MMU Register Description: [8]

Table 20-70. MMU_BYPASS_REGION2_ADDR

Address Offset	0x0000 0098		
Physical Address	See Section 20.5.2.1	Instance	See Table 20-15
Description	This register contains the start address of the second NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION for 2D bursts. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-71. Register Call Summary for Register MMU_BYPASS_REGION2_ADDR

MMU Register Manual
- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]

Table 20-72. MMU_BYPASS_REGION2_SIZE

Address Offset	0x0000 009C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of second NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION of 2D bursts. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-73. Register Call Summary for Register MMU_BYPASS_REGION2_SIZE

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]
- MMU Register Description: [8] [9] [10]

Table 20-74. MMU_BYPASS_REGION3_ADDR

Address Offset	0x0000 00A0		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the third NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION for 2D bursts. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-75. Register Call Summary for Register MMU_BYPASS_REGION3_ADDR

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]

Table 20-76. MMU_BYPASS_REGION3_SIZE

Address Offset	0x0000 00A4		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of third NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION of 2D bursts. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-77. Register Call Summary for Register MMU_BYPASS_REGION3_SIZE

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]

Table 20-78. MMU_BYPASS_REGION4_ADDR

Address Offset	0x0000 00A8		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the fourth NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION for 2D bursts. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-79. Register Call Summary for Register MMU_BYPASS_REGION4_ADDR

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]

Table 20-80. MMU_BYPASS_REGION4_SIZE

Address Offset	0x0000 00AC		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of fourth NO TRANSLATION REGION for 2D bursts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION of 2D bursts. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-81. Register Call Summary for Register MMU_BYPASS_REGION4_SIZE

MMU Register Manual

- MMU Register Summary: [0] [1] [2] [3] [4] [5] [6]