KeyStone II Architecture
Multicore Shared Memory Controller (MSMC)

User Guide

Literature Number: SPRUHJ6
November 2012

Release History

Release	Date	Description/Comments
SPRUHJ6	November 2012	Initial Release.

Contents

List of Tables

List of Figures

Preface

About This Manual

The Multicore Shared Memory Controller (MSMC) in KeyStone II devices manages traffic among ARM CorePacs, multiple C66x CorePacs, DMA, other mastering peripherals, and the EMIF in a multicore device. MSMC also provides a shared on-chip SRAM that is accessible by all the CorePacs and the mastering peripherals on the device. MSMC provides memory protection for accesses to the MSMC SRAM and DDR3 memory from system masters.

This manual should be used in conjunction with the device-specific data manual.

Notational Conventions

This document uses the following conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.
- Terminal sessions and information the system displays are in `screen font`.
- Information you must enter is in **`boldface screen font`**.
- Elements in square brackets ([ ]) are optional.

Notes use the following conventions:

Note—Means reader take note. Notes contain helpful suggestions or references to material not covered in the publication.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

CAUTION—Indicates the possibility of service interruption if precautions are not taken.

WARNING—Indicates the possibility of damage to equipment if precautions are not taken.

Related Documentation from Texas Instruments

ARM CorePac User Guide for KeyStone II Device	SPRUHJ4
C66x CorePac User Guide	SPRUGW0
DDR3 Memory Controller for KeyStone Devices User Guide	SPRUGV8

Trademarks

Chapter 1

Introduction

1.1 Overview

The Multicore Shared Memory Controller (MSMC) in KeyStone II devices manages traffic among ARM CorePacs, multiple C66x CorePacs, DMA, other mastering peripherals, and the EMIF in a multicore device. MSMC also provides a shared on-chip SRAM that is accessible by all the CorePacs and the mastering peripherals on the device. MSMC provides memory protection for accesses to the MSMC SRAM and DDR3 memory from system masters.

1.2 Terminology

Term	Definition
MPAX	Memory Protection and Address Extension
MSMC	Multicore Shared Memory Controller used in KeyStone II devices
PrivID	Privilege Identification for the System Masters
SES	System EMIF Access Slave Interface
SMS	System MSMC SRAM Access Slave Interface
TeraNet	Switch fabric

1.3 Features

The MSMC module provides the following:

- CPU/1 frequency of operation
- Coherency between ARM CorePac L1/L2 cache and EDMA/IO peripherals (through SES/SMS ports) in MSMC SRAM and DDR3A space
- Level 2 or Level 3 shared MSMC SRAM that is accessible by all the CorePacs and the mastering peripherals
- Memory protection for accesses to MSMC SRAM and DDR3 memory from system masters
- 32-bit to 36-bit address extension for larger addressing space

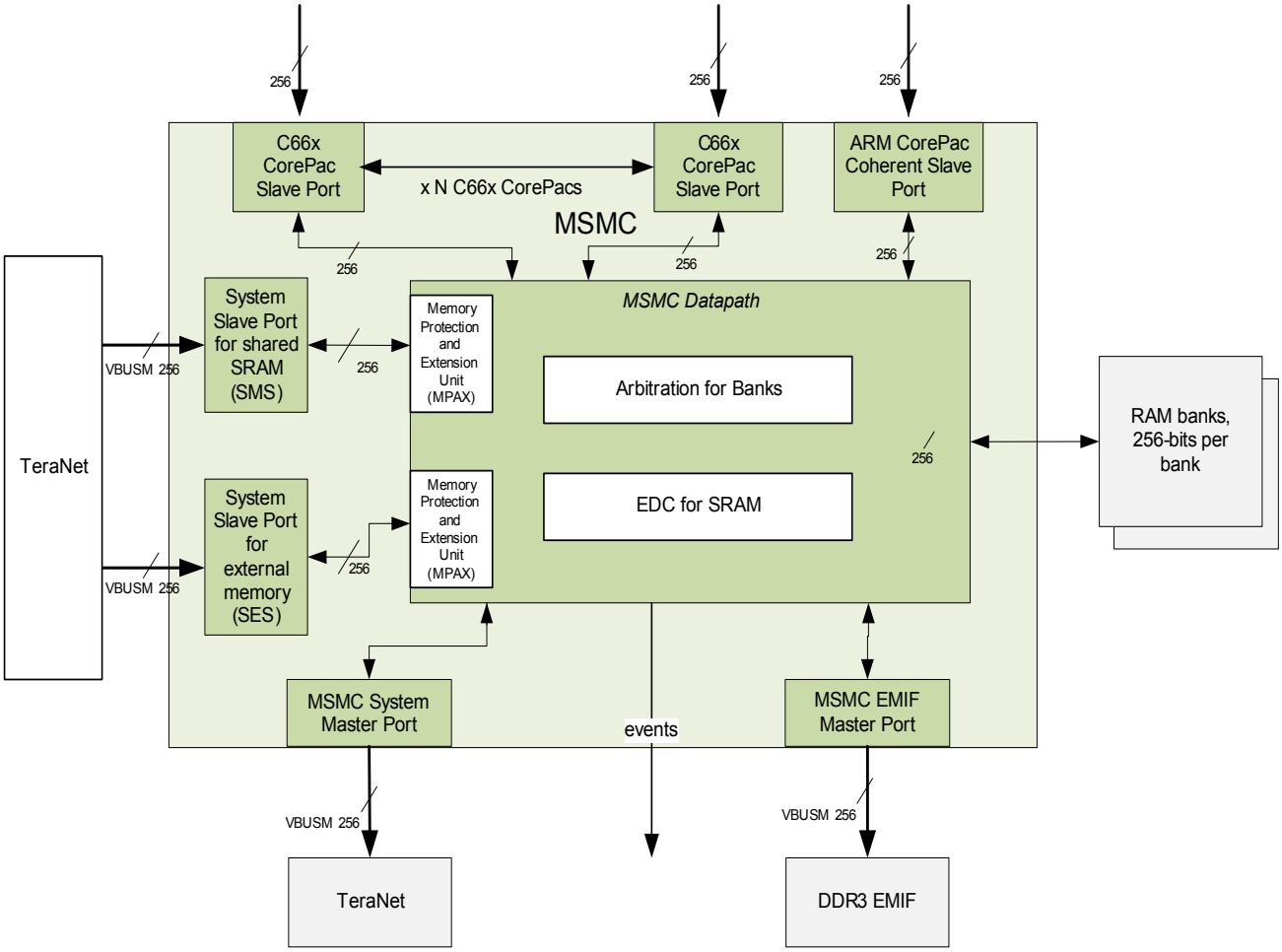
Chapter 2

MSMC Architecture

2.1 Functional Overview

Figure 2-1 shows a high-level view of the MSMC module that includes the main interfaces, memory, and subunits.

Figure 2-1 MSMC Functional Block Diagram



The MSMC has slave interfaces to connect to the ARM CorePac (one slave interface per ARM cluster), C66x CorePacs (one slave interface per CorePac), two slave interfaces to connect to the system interconnect (TeraNet), one master port to connect to the EMIF, and one master port to connect to the system interconnect (TeraNet).

2.1.1 CorePac Slave Interfaces

The MSMC has slave interfaces to connect to the ACE (AXI Coherence Extensions) port of ARM CorePac and MDMA port of C66x CorePacs. The CorePacs use this interface to access the MSMC on-chip memory, the external memory, and EMIF memory-mapped registers through the MSMC-EMIF master port or system level resources through the MSMC-system master port.

2.1.2 System Slave Interfaces

The MSMC has two slave interfaces to handle accesses from mastering peripherals in the system for the MSMC SRAM and EMIF.

2.1.2.1 System EMIF Access Slave Interface (SES)

The SES interface handles accesses to external DDR3 memory and memory-mapped registers inside the EMIF module that originate from a system master that is not a C66x or ARM CorePac.

Accesses presented on this interface to any addresses outside of the address range mapped to the external memory, or the EMIF memory mapped registers, result in an addressing error returned to the requesting master.

Note—When MSMC SRAM is remapped to an external address space using MPAX, such accesses will not result in an addressing error as long as the accesses are within the valid external memory address range.

The address width on this interface is 32 bits. Address extension to a 36-bit external memory address is done inside the MSMC as described in "Memory Protection and Address Extension (MPAX)" on page 2-4.

2.1.2.2 System MSMC SRAM Access Slave Interface (SMS)

The SMS interface handles accesses to MSMC SRAM that originate from a system master that is not a C66x or ARM CorePac. Accesses from masters in the system to MSMC configuration registers are also expected to be presented at this interface.

Any accesses from the SMS interface that do not address the MSMC SRAM or configuration registers result in an addressing error returned to the requesting master.

2.1.3 System Master Interface

The MSMC features one master interface for the CorePacs to access system resources other than the MSMC SRAM, MSMC MMRs, DDR3 memory, and the EMIF MMRs.

Traffic from the system slave interfaces does not pass through the master interface.

2.1.4 External Memory Master Interface

The external memory interface (EMIF) module is connected to the MSMC through the external memory master interface. The address width for this interface is 36 bits because it supports the extended memory addressing space beyond 4 GB. The MSMC implements an address extension to 36 bits as described in "Memory Protection and Address Extension (MPAX)" on page 2-4.

2.2 Memory Protection and Address Extension (MPAX)

The MSMC module MPAX supports an external memory addressing space of up to 32 GBytes addressable with a 36-bit address, even though the SoC addressing remains at 32-bits. Some KeyStone II devices (see device-specific data sheet) can support only up to 8 GBytes of external memory space. The C66x CorePac uses its own MPAX units to extend 32-bit addresses to 36-bit addresses before presenting them to the MSMC module. The ARM CorePac can optionally use the MMU with LPAE (Large Physical Address Extension) to support 40 bit physical addressing. However the 4 MSBs of the physical address should be set to 0x0 in KeyStone II devices in the ARM MMU.

The slave interfaces on the MSMC that receive addresses from all other masters in the system must extend the address inside the MSMC. These interfaces also provide support for memory protection for accesses from system masters to MSMC SRAM, external memory, and memory-mapped registers in the EMIF.

Both system slave interfaces (SES and SMS) feature an MPAX unit similar to the MPAX unit inside the C66x CorePac that combines these functions.

2.2.1 MPAX Segment Operation

The MPAX process is performed for a variable-sized segment of memory and is controlled with a register pair for each segment: MPAXH and MPAXL control registers.

- The MPAXH specifies the base address and size of the segment to match.
- The MPAXL specifies the replacement address and permissions for the segment.

Each MPAX unit provides eight control register pairs per Privilege ID (PrivID) of the system masters, which allows eight independent and potentially overlapping variable-size memory segments to be operated upon. See the device-specific data manual for the Privilege ID (PrivID) values assigned for various system masters.

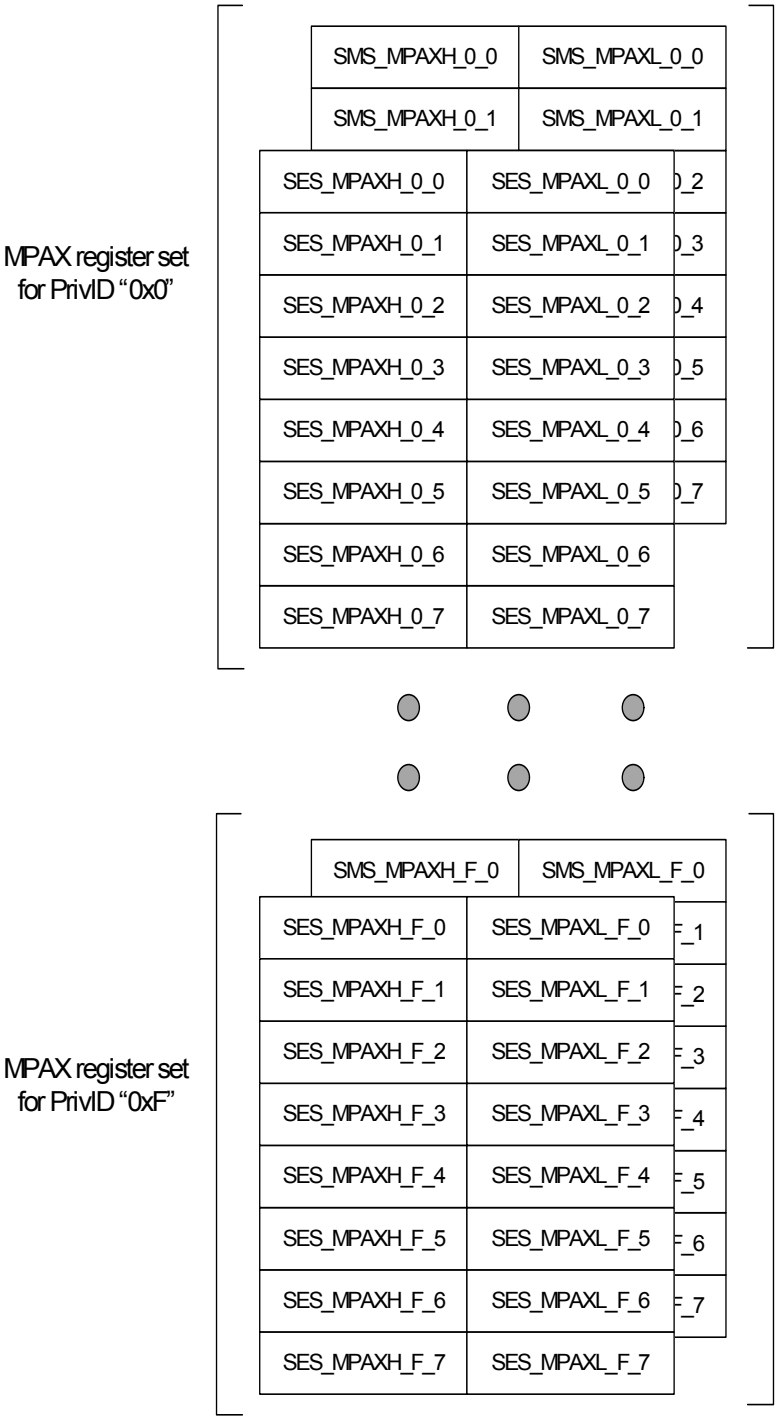
The control register memory-mapped space for the MSMC is not operated upon by the MPAX hardware. Addresses presented for any system accesses to the MSMC control registers through the SMS port are used as such.

Figure 2-2 shows how the MPAX segment registers are organized. The MPAX segment registers are described in "MPAX Segment Registers" on page 3-11.

All of the MPAX registers in the MSMC are readable by any of the connected CorePacs and also by system masters through the SMS port. Write-access control to these registers can be coordinated with the aid of semaphores external to the MSMC as well as through the locking mechanism for MSMC configuration registers.

Accesses to the segment registers through system masters is automatically protected as these registers are addressed by PrivID such that each PrivID has access to only its set of eight pairs of segment registers. Accesses to MPAX segment registers with a mismatched PrivID will generate a protection error.

Figure 2-2 MPAX Segment Register Set Layout



2.2.2 MPAX Segment Register Reset Values

At reset, the MPAX segment 0 register pair has initial values that set up unrestricted access to the full MSMC SRAM address space and 2 GB of the EMIF address space. All other segments come up with the permission bits and size set to 0 (that is, no access map to those segments).

- For each PrivID, SMS_MPAXH[0] is reset to 0x0C000017 and SMS_MPAXL[0] is reset to 0x00C000BF, (i.e., segment 0 is sized to 16 MB and matches any accesses to the address range 0x0CXXXXXX).
- For each PrivID, SES_MPAXH[0] is reset to 0x8000001E and SES_MPAXL[0] is reset to 0x800000BF, (i.e., the segment 0 is sized to 2 GB and matches any accesses to the address range 0x8XXXXXXX). This 2 GB space starts at the external memory base address of 0x80000000.
- SMS_MPAXH and SMS_MPAXL for segments 1 through 7 come out of reset as 0x0C000000 and 0x00C00080 respectively. SES_MPAXH and SES_MPAXL for segments 1 through 7 come out of reset as 0x00000000 and 0x00000080 respectively.

These reset settings are provided for the convenience of set-up code. It is recommended that boot code set up these registers as appropriate for the application.

2.2.3 Memory Protection

For the SES and SMS, the MPAXH register contains the Segment Base Address (BADDR) field that is matched to the incoming address on the SES port to identify the addressed segment. The BADDR and the Segment Size (SEGSZ) field describe the placement and size of the controlled segment. SEGSZ is a five-bit field that can be used to specify segment sizes ranging from 4 KB to 4 GB in powers of two, as listed in Table 2-1 "MPAX Segment Size Encoding".

Table 2-1 MPAX Segment Size Encoding

SEGSZ	Meaning	SEGSZ	Meaning	SEGSZ	Meaning	SEGSZ	Meaning
00000b	Segment disabled	01000b	Reserved	10000b	128KB	11000b	32MB
00001b	Reserved	01001b	Reserved	10001b	256KB	11001b	64MB
00010b	Reserved	01010b	Reserved	10010b	512KB	11010b	128MB
00011b	Reserved	01011b	4KB	10011b	1MB	11011b	256MB
00100b	Reserved	01100b	8KB	10100b	2MB	11100b	512MB
00101b	Reserved	01101b	16KB	10101b	4MB	11101b	1GB
00110b	Reserved	01110b	32KB	10110b	8MB	11110b	2GB
00111b	Reserved	01111b	64KB	10111b	16MB	11111b	4GB

Segments are always sized to a power of two and start on a corresponding power-of-two boundary (for example, a 4 KB segment always starts on a 4 K boundary, and a 4 GB segment corresponds to the entire 32-bit address space). For the SMS port MPAX unit, the maximum allowed segment size is 16 MB (SEGSZ=10111b), which would cover the full MSMC storage space.

For the MPAX unit in the SES port, a sufficient number of upper bits (depending on the size specified in the SEGSZ field) of the incoming address are matched against the BADDR field of all the MPAXH registers corresponding to the access PrivID to select the MPAXH control register pair to use for memory protection and extension. For

example, for 4 KB segments, all 20 bits of the BADDR field must match the upper 20 bits of the system address. For 16 MB segments, the upper eight bits of the BADDR field must match the upper eight bits of the C66x CorePac address; the remaining bits are ignored. For 4 GB segments, no BADDR bits are consulted and all addresses match.

- If an address does not match any of the programmed MPAXH registers and is not a MSMC configuration register address, the access permissions for the access are considered to be 0; this results in a protection fault.
- If an address matches multiple MPAXH registers (overlapping segment descriptors), the highest numbered MPAX register-pair is selected (that is, if the address matches BADDR in MPAXH5 and in MPAXH2, MPAXH5 is selected). Using this priority-based matching behavior, multiple MPAX entries can be programmed for overlapping segments to achieve either non-power-of-two sized segments or subsegments with different memory protection (and/or extension) parameters.

2.2.3.1 Memory Protection Fault Reporting

The MSMC memory protection fault reporting registers are listed in Table 2-2 and described in "Memory Protection Fault Reporting Registers" on page 3-14.

Table 2-2 MSMC Protection Fault Reporting Register List

Acronym	Register Description
SMPFAR	Shared Memory Protection Fault Address Register
SMPFXR	Shared Memory Protection Fault Extension Register
SMPFR	Shared Memory Protection Fault Requestor Register
SMPFCR	Shared Memory Protection Fault Control Register

The MPAXL register contains the permission attributes for supervisor and user accesses. If an access mismatches MPAXL with the corresponding permission attributes, a memory fault is triggered that results in the following actions:

- A memory-protection-fault interrupt (MSMC_mpf_error*n* - where *n* is the PrivID of the requestor making the offending access) is generated on the faulting access (if the corresponding interrupt is enabled in the SMIESTAT register). This is also recorded in the SMIRSTAT register (see "MSMC Interrupt Control" on page 2-22) that contains one event bit per PrivID that is set when an access from the PrivID faults and is cleared on writing 1 to the associated bit in the SMIRC register. Only one fault is notified per PrivID; until the interrupt status is cleared in SMIRC, the next interrupt for that PrivID is not generated.
- The access address causing the fault is captured in the SMPFAR register. If the fault is the result of the address not matching any of the segment BADDR, the NM field in the SMPFXR register is set. The master ID and PrivID associated with the access is also recorded in the SMPFR register. See the device-specific data manual for the Master ID and Privilege ID (PrivID) values assigned for various system masters. A bus-protection-error status is returned to the requesting master for the faulting access.
- There is only one fault recorded in the SMPFAR and SMPFXR. The master is notified with a fault interrupt and receives a bus error for the access. In response, the master must clear the recorded fault by writing 1 to the CLR bitfield in SMPFCR before further faults can be recorded.

2.2.4 Address Extension

For access to the SES interface, the 32-bit address is extended to 36 bits by replacing the appropriate number of upper bits of the address (based on segment size) with a corresponding portion of the replacement address field (RADDR) in the MPAXL register that has four more upper bits, as shown in Table 2-3 "Replacement Address Used as Per-Segment Size".

Table 2-3 Replacement Address Used as Per-Segment Size

SEGSZ	SES / SMS RADDR bits	SEGSZ	SES RADDR bits	SMS RADDR bits	SEGSZ	SES RADDR bits	SMS RADDR bits	SEGSZ	SES RADDR bits
00000b	Segment disabled	01000b	Reserved	Reserved	10000b	[23:5]	[19:5]	11000b	[23:13]
00001b	Reserved	01001b	Reserved	Reserved	10001b	[23:6]	[19:6]	11001b	[23:14]
00010b	Reserved	01010b	Reserved	Reserved	10010b	[23:7]	[19:7]	11010b	[23:15]
00011b	Reserved	01011b	[23:0]	[19:0]	10011b	[23:8]	[19:8]	11011b	[23:16]
00100b	Reserved	01100b	[23:1]	[19:1]	10100b	[23:9]	[19:9]	11100b	[23:17]
00101b	Reserved	01101b	[23:2]	[19:2]	10101b	[23:10]	[19:10]	11101b	[23:18]
00110b	Reserved	01110b	[23:3]	[19:3]	10110b	[23:11]	[19:11]	11110b	[23:19]
00111b	Reserved	01111b	[23:4]	[19:4]	10111b	[23:12]	[19:12]	11111b	[23:20]
End of Table 2-3									

This operation is identical to that in the C66x CorePac MPAX unit. Programming the same values into the BADDR, RADDR, and SEGZ fields will result in an identical address-extension operation for a system master access.

2.2.4.1 SES Aliased Access to MSMC RAM

Using address remapping in the MPAX unit at the SES port, it is possible to map some or all of the MSMC SRAM into external memory space and access it through the SES port. This provides the same ability to alias MSMC memory to external address space for system masters as is available to the C66x CorePacs through the CorePac MPAX unit.

The ability to alias external addresses to the SRAM has the following constraints:

- Crossing between MSMC SRAM and external memory endpoints in the course of a transaction on SES is not supported and results in an addressing error for the endpoint first addressed in the command.
- Addressing errors get prioritized for reporting over protection errors.

2.2.4.2 SMS MPAX Address Extension

For the SMS interface, the address-extension operation is very similar except that the generated address maintains bits 31-24 to be the same as the original address (that is, the start address of the MSMC SRAM). The SMS_MPAXL contains a correspondingly smaller RADDR field. Note that this resultant address is still a 32-bit address within the address space of the MSMC SRAM and the operation is that of remapping the address segment to a different location within the MSMC SRAM.

It is possible to violate the permissions associated with a segment and checked as part of the memory-protection check by extending the address into a target segment that has more restrictive permissions. This is actually a valid configuration of the MPAX registers to carve out subsegments that overlap segments with differing permissions.

- If segment overlap is not desired or intended, it is a good practice to align the level of restriction in the segment permissions with the MPAX segment matching priority scheme; that is, configure a less restrictive segment A in a lower numbered MPAX pair than a more restrictive segment B.
- If the permissions of segment A are intended to be used for the overlap region between the two segments A and B, configure A in a higher numbered MPAX pair than the one for segment B.

2.2.4.3 Address Extension Error Reporting

If an SES address extension results in an address that is outside the range of the EMIF (memory or memory-mapped registers), the MSMC EMIF Master port receives an addressing error from the EMIF and relays this error to the SES interface.

2.3 MSMC Memory

2.3.1 MSMC SRAM

MSMC SRAM can serve as a Shared Level 2 or Level 3 memory

- Shared Level 2 memory—The MSMC memory is cacheable by C66x L1D and L1P caches; C66x L2 will not cache requests to MSMC SRAM.
- Shared Level 3 memory—The MSMC memory is not directly cacheable at the C66x L2, but is cacheable in C66x L1D and L1P. However, if it is remapped to an external address using the address-extension capabilities in the C66x CorePac MPAX, the MSMC memory can be cached as a shared L3 memory both in the L1 and L2 caches of C66x CorePac. To achieve this, the caching must be enabled in MAR registers (using MAR.PC bit) for the remapped region. The MSMC memory is directly cacheable in ARM L2 memory by defining the MSMC SRAM region as normal cacheable memory in ARM MMU.

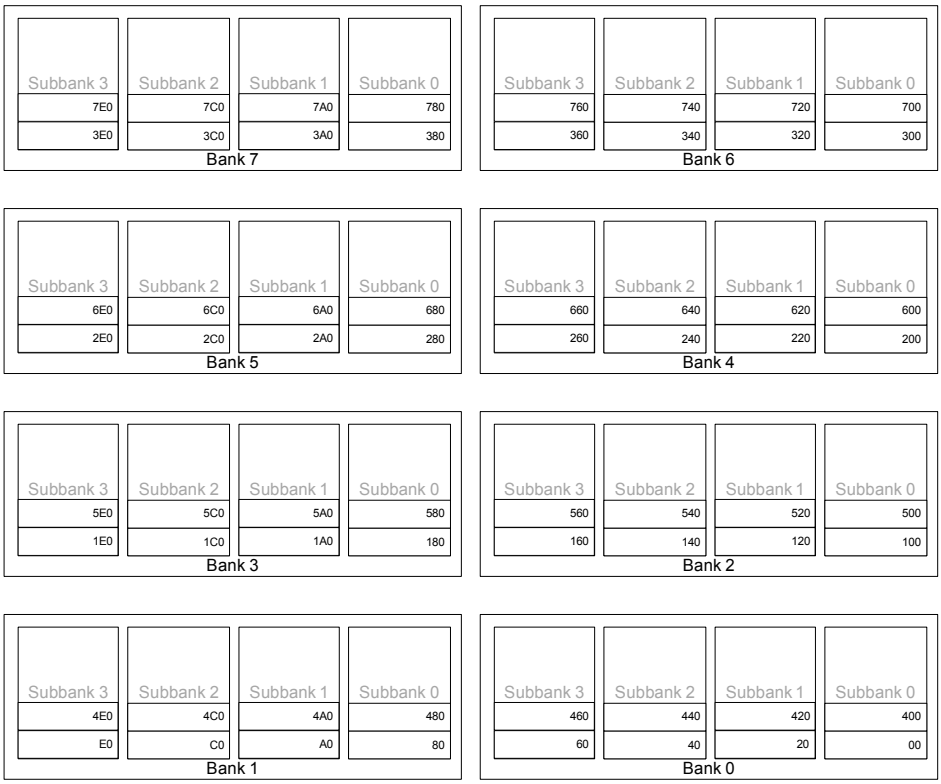
2.3.2 MSMC Memory Banking

The memory is organized using banks that consist internally of four subbanks that hold adjacently-addressed locations. The addressing of the banks is done so that 128-byte aligned, 128-byte chunks of memory are located in different banks. In addition, 32-byte aligned addresses within each 128-byte chunk that are addressed in a bank are located in different subbanks. The 128 byte banking structure aligns with the CGEM1 L2 cache line size.

Figure 2-3 "MSMC SRAM Bank Addressing " shows the location of consecutive byte addresses in the eight-bank organization of the MSMC memory as they are addressed—bit 7,8 and 9 of the byte address are used to select between the eight banks and bit 5 and 6 of the address is used to select the subbank in the selected bank.

See the device-specific data manual for MSMC SRAM sizes and configurations.

Figure 2-3 MSMC SRAM Bank Addressing



2.3.3 MSMC Bandwidth Management

The arbitration scheme attempts to allocate accesses fairly to requestors at the same priority level. However, it is not sufficient to ensure a bound on the wait times experienced by lower priority requests. Consequently, requestors could be starved for access when there is heavy traffic at higher priority levels. To avoid indefinite starvation for lower priority requests, the MSMC features a bandwidth management scheme that limits starvation times.

The MSMC features a starvation bound register (SBND) per requestor. These memory-mapped configuration registers are SBNDC0-SBNDC*n* for the CorePac slaves, and SBNDM for the SMS port and SBNDE for the SES port. The registers are programmed with a desired starvation bound in MSMC cycles for the requestor's accesses.

MSMC bandwidth management registers are described in "Bandwidth Management Control Registers" on page 3-9.

The central arbiters for the memory banks and the EMIF master port contain a starvation counter for each of the requestors being tracked. Whenever a SBND register SCNT bitfield is programmed, the counter for the requestor is initialized with the same value.

Table 2-4 Starvation Counters per Requestor

Register	Bitfield	Function
SBNDCn	SCNTCE	Reload value (pre-scaled by 16) for the starvation counter for CorePac slave n requests at the EMIF arbiter.
SBNDE	SCNTEE	Reload value (pre-scaled by 16) for the starvation counter for SES requests at the EMIF arbiter.
SBNDCn	SCNTCM	Reload value for the starvation counters for CorePac slave n requests at the RAM bank arbiters.
SBNDM	SCNTMM	Reload value for the starvation counters for SMS requests at the RAM bank arbiters.
SBNDE	SCNTEM	Reload value for the starvation counters for SES requests at the RAM bank arbiters.

For the EMIF arbiter, if the interface to the EMIF is stalled, the SCNT countdown is suspended for that cycle.

When the SCNT reaches zero, the priority of the request is elevated to zero (the highest priority level). After the elevated priority is serviced, the SCNT counter is reloaded from the SBND register for the requestor. Further accesses from the requestor are based on the original priority.

2.4 MSMC IO Coherence

MSMC supports hardware cache coherence between the ARM CorePac L1/L2 caches and EDMA/IO peripherals for shared SRAM and DDR spaces. This feature allows the sharing of MSMC SRAM and DDR data spaces by these masters on the chip, without having to use explicit software cache maintenance techniques.

MSMC does not support memory coherence for these spaces:

- EMIF Configuration Space
- MSMC Configuration Space
- System Master Port Peripherals/Memory
- Any memory not directly connected to MSMC

2.4.1 Cache Coherence Operation

Coherent masters typically track more states than the traditional valid/dirty combinations to maintain coherence. MSMC uses the ACE (AXI Coherence Extensions) coherence protocol which supports the following five states (MOESI)

- Modified - Cache line is unique (no other caches currently have it) and dirty
- Owned - Cache line is shared (other caches may have it) and dirty
- Exclusive - Cache line is unique and clean
- Shared - Cache line is shared and clean
- Invalid - No allocated line

In order to maintain coherence between cached masters, MSMC can initiate requests to coherent masters for shared regions of memory. These requests are called as snoop requests and are performed as per the ACE protocol. MSMC maintains coherence by snooping coherent caches for the latest value when an access hits a shared memory region (SES/SMS MPAXHn.US=0).

The following example demonstrates how coherence is maintained between EDMA and ARM, when EDMA writes to shareable memory space.

1. EDMA (master with no cache) issues a write to shareable space
2. MSMC coherence controller issues Invalidate and Writeback snoop to the CPU
3. CPU invalidates the line from its cache and responds with the dirty data
4. MSMC merges the EDMA write data with the victim and commits to memory

The following example demonstrates how coherence is maintained between EDMA and ARM, when EDMA reads from shareable memory space.

1. EDMA issues a read to shareable space
2. MSMC coherence controller issues a ReadOnce snoop to the CPU
3. CPU responds with cached data
4. MSMC returns the read data to EDMA

Software has the ability to control which memory regions are shared among certain sets of coherent masters using ARM MMU (in ARM CorePac) and using SMS_MPAXH or SES_MPAXH register for EDMA/IO peripherals. Software should ensure that the shareability mappings between the types of masters are consistent to avoid unexpected behavior.

For example, a memory segment marked as outer shared in ARM MMU and marked as unshared in the respective SMS_MPAXH.US or SES_MPAXH.US is an inconsistent mapping and can cause unpredictable behavior.

2.5 MSMC Register Access Control

The memory-mapped configuration registers for MSMC are accessible by all the CorePacs connected to the CorePac slave ports as well as all system masters that can access MSMC through the SMS port. It is required that updates to memory-mapped registers are not made while accesses are being made through any of the system ports that utilize the registers being written to. For example, software would need to ensure that updates to an SES MPAX register are done only when no transfers are going on through the SES interface from the same PrivID.

Note—The MSMC memory-mapped register hardware cannot guarantee the determinism of a transfer if the configuration registers used by the transfer are being modified simultaneously. If such an update is done to the registers, the outcome of the transfer is not deterministic and undefined.

While reads to these registers from any of the masters are not restricted, it is often useful to restrict write access to the registers to prevent runaway code on one of the cores or an incorrectly-configured system master from corrupting the state of MSMC configuration. While a semaphore-based mutex access control is still expected to be used by software on the cores (and any masters that may program the registers), MSMC features a simple lock mechanism to protect against runaway pointer writes.

MSMC register access control is managed by the lock registers listed in Table 2-5 and described in "MSMC Configuration Write Lock Registers" on page 3-16.

For the purpose of write-access control, the MSMC registers are grouped into three categories:

- SMS MPAX registers
- SES MPAX registers
- Non-MPAX registers

For each category, a write-lock bit (one per PrivID for the MPAX registers) controls whether writes are enabled.

Table 2-5 MSMC Configuration Write Lock Register List

Acronym	Register Description
CFGLCK	Configuration Lock control for non-MPAX registers
CFGULCK	Configuration Unlock control for non-MPAX registers
CFGLCKSTAT	Configuration Lock Status for non-MPAX registers
SMS_MPAX_LCK	Configuration Lock control for SMS MPAX registers
SMS_MPAX_ULCK	Configuration Unlock control for SMS MPAX registers
SMS_MPAX_LCKSTAT	Configuration Lock Status for SMS MPAX registers
SES_MPAX_LCK	Configuration Lock control for SES MPAX registers
SES_MPAX_ULCK	Configuration Unlock control for SES MPAX registers
SES_MPAX_LCKSTAT	Configuration Lock Status for SES MPAX registers

- **CFGLCK, CFGULCK, and CFGLCKSTAT registers for non-MPAX registers (with the exception of the write lock registers for all three categories)** - These manage one lock-bit with the bitfields WLCK (set to 1 to lock), WEN (set to 1 to unlock) and WSTAT (read to query the state of the lock).

- **SMS_MPAX_LCK, SMS_MPAX_ULCK, and SMS_MPAX_LCKSTAT registers for all the SMS MPAX registers** - These manage 16 lock-bits (one bit per PrivID) with the bitfields WLCK (set to 1 to lock), WEN (set to 1 to unlock) and WSTAT (read to query the state of the lock).
- **SES_MPAX_LCK, SES_MPAX_ULCK, and SES_MPAX_LCKSTAT registers for all the SES MPAX registers** - These manage 16 lock-bits (one bit per PrivID) with the bitfields WLCK (set to 1 to lock), WEN (set to 1 to unlock) and WSTAT (read to query the state of the lock).

All the LCK and ULCK registers are modified only when the upper 16 bits of the write data match the fixed MGCID key field of these registers. Any other value used in the upper 16 bits of the write data does not modify its WLCK/WEN bitfield. For each category, the sequence of steps is as follows:

- To lock, write MGCID_WLCK[15:0] into the LCK register (for example, write 2CD1_0440 to lock the SMS MPAX registers for PrivIDs 7 and 11). For non-MPAX registers, write 2CD0_0001 to the CFGLCK register. Until it is unlocked, all further writes to that category registers generate protection errors.
- To unlock, write MGCID_WEN[15:0] into the ULCK register (for example, write 2CD1_0440 to unlock the SMS MPAX registers for PrivIDs 7 and 11). For non-MPAX registers, write 2CD0_0001 to the CFGULCK register. Until it is locked, all further writes to that category of configuration registers generate protection errors.

Note—This provides a simple write-protection mechanism that protects against unintentional modification only. It is recommended that the software uses a semaphore to ensure exclusive access when modifying the MSMC lock registers.

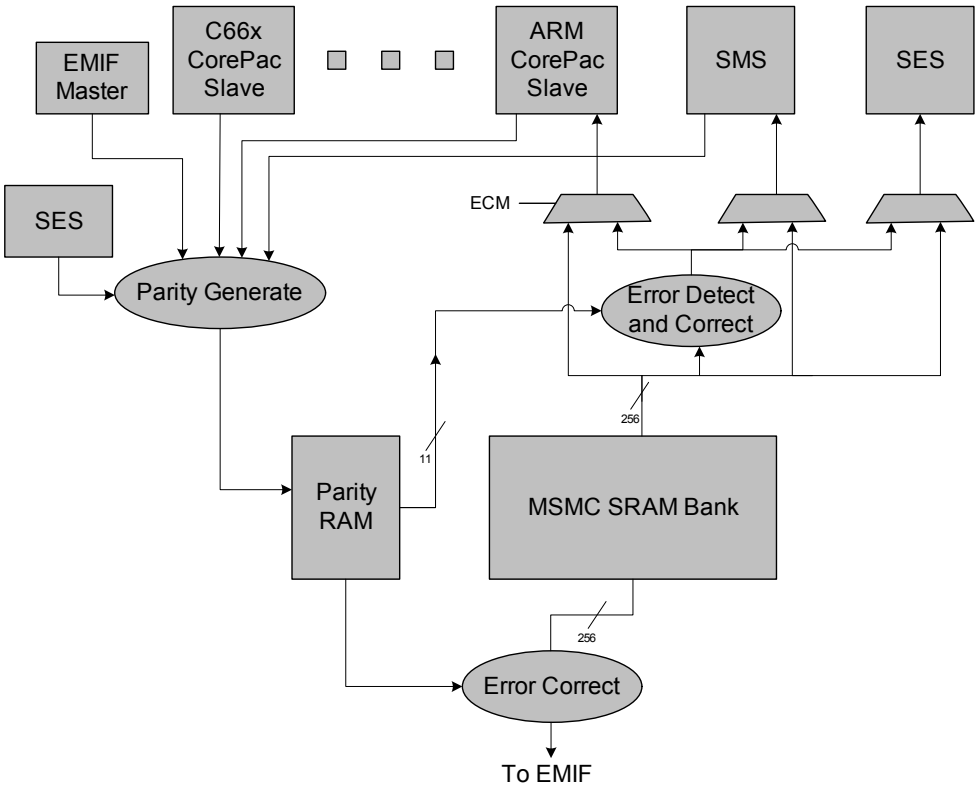
2.6 Error Detection and Correction Support

The MSMC has error detection and correction hardware to protect the contents of the MSMC memory storage against corruption due to transient (soft) errors. The level of protection provided and the scheme used is the same as that of the C66x CorePacs (that is, one-bit error correction, two-bit error detection, with the parity codes calculated over a 256 bit datum).

2.6.1 Parity Generation and Checking

Figure 2-4 shows the hardware available per MSMC bank that supports parity generation, storage, and check/correction. The parity code generated for each location in a MSMC SRAM bank is stored in a corresponding location in the associated parity RAM. The parity RAM location also stores the parity of the parity RAM location to allow detection of transient errors in the parity RAM alone.

Figure 2-4 Error Detection and Correction



2.6.2 EDC Operation

In the MSMC EDC scheme, parity is generated and tracked for each 256-bit datum which corresponds to a RAM location and a data phase for slave port access. A parity RAM entry may become invalid when data is written to a bank with a granularity smaller than 256 bits (for example, when the L2 cache for a C66x CorePac is off/frozen/bypassed and a write miss to the MSMC SRAM region occurs in its L1D cache). In this case, the valid bit in the corresponding parity RAM entry is reset.

The parity code for a line is recalculated when the line is transferred through any of the outgoing paths shown in Figure 4-4. The assumption on the reliability of the data being stored and transferred to/from the MSMC is that any incoming data for MSMC storage is reliable and a parity code generated from it is valid.

The EDC hardware consists of the registers listed in Table 2-6 and described in "EDC Registers" on page 3-5.

Table 2-6 MSMC EDC Register List

Acronym	Register Description
SMCERRAR	MSMC SRAM correctable EDC error address register
SMCERRXR	MSMC SRAM correctable EDC extended error register
SMEDCC	MSMC SRAM EDC control register
SMCEA	MSMC Scrubbing Error Corrected Address register
SMSECC	MSMC Scrubbing Error Corrected Counter register
SMNCERRAR	MSMC SRAM non-correctable EDC error address register
SMNCERRXR	MSMC SRAM non-correctable EDC extended error register
SMNCEA	MSMC Scrubbing Non-Correctable Address register

2.6.2.1 Error Correction Mode

Error correction adds latency for data-read accesses from slave ports. The ECM (Error Correction Mode) sticky bit in the SMEDCC register controls if error correction is enabled on all the slave port read paths. ECM is '0' at reset and corresponds to selecting no error correction for slave accesses (the direct return path in Figure 2-4 "Error Detection and Correction") resulting in the fastest read configuration by default. Setting ECM=1 enables error correction; however, any further writes to the ECM bit are ignored (that is, once enabled, error correction stays enabled until the MSMC is reset).

2.6.2.1.1 Soft Error Correction

For data being transferred out of the MSMC storage, the EDC hardware generates the parity for the data and checks against the corresponding stored parity for errors and log information in the EDC registers. The actions and reporting are summarized in Table 2-7, more details about the interrupts are described in "MSMC Interrupt Control" on page 2-22.

Table 2-7 Soft Error Correction Actions

Read Access Port	ECM state	Soft Error Type	Interrupt Recorded	Data Returned
CorePac Slave, SES, SMS	1	Correctable	CEI	Corrected
	1	Non-Correctable	NCEI	0
	1	Parity RAM	-	Uncorrected
	0	Correctable	CEI	Uncorrected
	0	Non-Correctable	NCEI	Uncorrected
	0	Parity RAM	-	Uncorrected

2.6.3 EDC Error Reporting

When an error is detected, the EDC hardware reports the error details in the EDC error reporting registers as follows:

- For a correctable (one-bit) error:
  - The lower 32 bits of the 36-bit address used in accessing the corrupted location are stored in the SMCERRAR register and the upper 4 bits of the 36-bit address in the SEEADDR field of the SMCERRXR register.
  - The syndrome for the corrected location is logged in the ESYN field of SMCERRXR.
  - The PrivID associated with the access is saved in the SEPID field of the SMCERRXR register. If the request came in from the SMS or the SES port, the SER bit is set.
  - Generates MSMC_dedc_cerror interrupt if it is enabled in the SMIESTAT register.
- For a non-correctable (two-bit) error:
  - The lower 32 bits of the 36-bit address used in accessing the corrupted location are stored in the SMNCERRAR register and the upper 4 bits of the 36-bit address in the SEEADDR field of the SMNCERRXR register.
  - The PrivID associated with the access is saved in the SEPID field of the SMNCERRXR register. If the request came in from the SMS or the SES port, the SER bit is set.
  - Generates MSMC_dedc_nc_error interrupt if it is enabled in the SMIESTAT register.
- In the unusual case that an error is detected in multiple banks accessed during the same cycle, the hardware captures the address and related information for only one bank selected in the order of priority, where priority of bank (N)> bank (N+1) (N = 0..6)
- After they are written, the error-logging registers SMCERRAR/SMNCERRAR and SMCERRXR/SMNCERRXR store the contents until further error logging is re-enabled by clearing the associated interrupt status with a write to the SMIRC register (see "MSMC Interrupt Control" on page 2-22).

2.6.4 Background Parity Refresh-Scrubbing

As parity is tracked at a granularity equal to the width of the banks (32 bytes), writes that are smaller than 32 bytes can invalidate the parity information for a line. MSMC contains a background error correction hardware called the Scrubbing Engine that periodically refreshes the parity bits for the memory.

The MSMC scrubbing engine is a state machine that periodically cycles through each location of each memory bank in the MSMC, reading and correcting the data, recalculating the parity bits for the data, and storing the data and parity information. Each such "scrubbing cycle" consists of a series of read-modify-write "scrub bursts" to the memory banks.

2.6.4.1 Scrubbing Rate

The frequency with which each scrubbing cycle is initiated and the delay between each burst by the scrubbing engine is programmed using the SMEDCC register.

- The bitfield REFDEL is programmed to control the number of MSMC clock cycles between each scrub burst. To prevent the bursts of the scrubbing engine from posing a significant performance impact, the value in the REFDEL register is prescaled by 1024.
- When prescaling of the REFDEL is enabled, a value of 0 is interpreted to be the same as a value of 1. When the prescaling is disabled, a value of REFDEL=0 is interpreted as 0, i.e., no delay between scrubbing bursts.
- The operation of the scrubbing engine is enabled by default after reset and parity RAM initialization, but may be disabled by setting the SEN bit in the SMEDCC register. There is no setup required to enable the scrubbing operation aside from checking for an end of the reset sequence as described in "Parity RAM Initialization at Reset" on page 2-21.

2.6.4.2 Scrubbing Error Logging and Statistics Collection

The MSMC scrubbing engine can log errors and collect statistics about scrubbing errors locally:

- If the scrubbing engine access detects a location where contents have been corrected, it logs the address in the SMCEA register, logs the syndrome value (that identifies the erroneous bit in the data) in the ESYN field of the SMCEA register, and increments the SCEC (Scrub Correctable Error Counter) field in the SMSECC register. It also generates MSMC_scrub_cerror interrupt (if it is enabled in the SMIESTAT register) indicating correctable (1-bit) soft error detected during scrub cycle.
- If the scrubbing engine access detects a two-bit error that is not correctable, it logs the address in the SMNCEA register and increments the SNCEC (Scrub Non-Correctable Error Counter) in the SMSECC register. It also generates MSMC_scrub_nc_error interrupt (if it is enabled in the SMIESTAT register) indicating non-correctable (2-bit) soft error detected during scrub cycle.
- Once written, the error logging registers SMCEA and SMNCEA store the contents till further error logging is re-enabled and these registers are cleared by clearing the associated interrupt status with a write to the SMIRC register (see "MSMC Interrupt Control" on page 2-22).
- The SMCEA and SMNCEA registers are both saturating counters and count all correctable and non-correctable errors, respectively, independent of the logging-enable state of the SMCEA and SMNCEA registers.

Periodically, the software can read the statistics collected to analyze persistent soft errors. Reading saturated counts in the error counters should indicate to the software that the error counters need to be checked more frequently to get a better indication of the error rate. If the error rate is high, it may be warranted to initiate scrub cycles more frequently to decrease the probability of an un-repairable two-bit error. If the error rate is low, less frequent scrubbing may be needed to reduce power consumption and reduce interference with functional memory-access traffic.

2.6.5 Parity RAM Initialization at Reset

The MSMC EDC hardware invalidates all the parity RAM entries at reset. During this initialization cycle, accesses are not stalled from the slaves or from the scrubbing engine, but are provided no error correction or detection:

- Read accesses to the memory are serviced but no data check is done, i.e., the memory locations are not protected from soft errors during this period. These initialization accesses are done with the prescaler disabled. Consequently, no spurious events are generated till the parity RAM is initialized completely. At the end of the initialization, the default prescaler delay of 1024 is applied.
- The PRR (Parity RAM Ready) bit in the SMEDCC shows the status of this process and is 1 whenever the Parity RAM is ready for use in EDC and scrubbing operations; PRR is 0 following a reset while the entries are being initialized. Software must consult this bit before making the first read accesses to the MSMC RAM after reset.
- Write accesses of a full 32-byte location during this period generates the corresponding parity value into the parity RAM location. When a write access arrives in the same cycle as the initialization write of the associated parity RAM location, the former takes precedence.

2.7 MSMC Interrupt Control

The MSMC module generates the following interrupts; see the device-specific data manual for details on how these interrupts are routed at the chip level.

Table 2-8 MSMC Interrupt Table

Interrupt	Description	Section
MSMC_mpf_error[15-0]	Memory protection fault interrupt for each system master Privilege ID (PrivID).	See Section 2.2.3.1
MSMC_dedc_cerror	EDC interrupt - Correctable (1-bit) soft error detected on SRAM read.	See Section 2.6.3
MSMC_dedc_nc_error	EDC interrupt - Non-correctable (2-bit) soft error detected on SRAM read.	
MSMC_scrub_cerror	EDC interrupt - Correctable (1-bit) soft error detected during scrub cycle.	See Section 2.6.4.2
MSMC_scrub_nc_error	EDC interrupt - Non-correctable (2-bit) soft error detected during scrub cycle.	

The MSMC features a set of interrupt status and enable registers that can control the generation of interrupts at the MSMC module boundary.

The MSMC interrupt operation is controlled by the registers listed in Table 2-9 and described in "MSMC Interrupt Control Registers" on page 3-21.

Table 2-9 MSMC Interrupt Control Register List

Acronym	Register Description
SMESTAT	Interrupt Enabled Status register. ANDed value of SMIRSTAT and SMIESTAT registers
SMIRSTAT	Interrupt Raw Status register
SMIRC	Interrupt Raw Status Clear register. Writes of 0 have no effect.
SMIESTAT	Interrupt Enable Status register
SMIEC	Interrupt Enable Clear register

The SMIRSTAT register stores the raw interrupt status for each interrupt line, a bitfield is set if the associated interrupt has occurred. An interrupt pulse can be generated and its raw status recorded in SMIRSTAT either when the event occurs in hardware (hardware interrupt) or when software writes to the associated bitfield in the SMIRSTAT register (software interrupt for simulating the event). Regardless of how the interrupt is generated, it is cleared by writing to the associated bit in the SMIRC register.

Note—The memory-protection-fault-interrupt event for each PrivID is recorded in the SMIRSTAT and the associated address and master information is logged into the SMPFAR and SMPFXR registers (see "Memory Protection Fault Reporting Registers" on page 3-14). The software must clear the SMPFAR and SMPFXR before clearing the corresponding bits for the PrivID in the SMIRSTAT so that the event status and associated master information are kept synchronized. This order is not enforced in hardware.

The SMIESTAT register stores the interrupt enable state for each of the interrupt lines. A bitfield is set if the associated interrupt is enabled. The enable state can be set by writing to the SMIESTAT register and cleared by writing to the SMIEC register.

For interrupt signalling and status capture the following rules apply:

- For a given interrupt line, once an interrupt is signaled, its status is set and needs to be cleared by the servicing host before another occurrence can be signaled. The status for an interrupt line is set even if it is not enabled in the SMIESTAT. Therefore, it is recommended that the software clears an interrupt status while enabling it. This can be done atomically with a double-word-store writing to the SMIESTAT and the SMIRC registers together.
- Because there are three sources that can modify a bit in the SMIRSTAT, the following precedence rules apply:
  - A hardware interrupt takes precedence over a software interrupt if they arrive in the same cycle. This is not an intended usage mode as one would either enable a hardware interrupt to signal a real hardware event or use a software write to simulate the same event.
  - A write to set a event in the SMIRSTAT register takes precedence over clearing an event in SMIRC register. This situation generates the interrupt pulse corresponding to the event that is set.

2.8 Reset Considerations

The MSMC module resets on all device resets. Upon reset, the following sequence is followed:

- All MSMC configuration registers are reset to their initial default state.
- The parity RAM entries associated with the MSMC SRAM are invalidated as described in "Parity RAM Initialization at Reset" on page 2-21.
- The MSMC pipeline is reset to an idle state terminating any outstanding accesses.

2.9 Memory Map

For memory map details, see the data manual for your specific KeyStone Architecture device.

Chapter 3

MSMC Registers

This chapter describes the memory-mapped registers associated with the MSMC.

3.1 MSMC Memory Mapped Registers

Table 3-1 lists the MSMC memory-mapped registers. See the device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 3-1 should be considered as reserved locations and the register contents should not be modified.

Table 3-1 MSMC Memory Mapped Registers (Part 1 of 2)

Offset	Acronym	Register Description	Section
0x00	PID	MSMC Peripheral ID register	See 3.2
0x04	Reserved	Reserved	—
0x08	SMCERRAR	MSMC SRAM correctable EDC error address register	See 3.3.2
0x0C	SMCERRXR	MSMC SRAM correctable EDC extended error register	See 3.3.3
0x10	SMEDCC	MSMC SRAM EDC control register	See 3.3.1
0x14	SMCEA	MSMC Scrubbing Error Corrected Address register	See 3.3.6
0x18	SMSECC	MSMC Scrubbing Error Counter register	See 3.3.8
0x1C	SMPFAR	MSMC Memory Protection Fault Address register	See 3.6.1
0x20	SMPFXR	MSMC Memory Protection Fault Extension register	See 3.6.2
0x24	SMPFR	MSMC Memory Protection Fault Requestor register	See 3.6.3
0x28	SMPFCR	MSMC Memory Protection Fault Control register	See 3.6.4
0x2C	Reserved	Reserved	—
0x30	SBNDC0	Starvation bound register for C66x CorePac slave port 0	See 3.4.1
0x34	SBNDC1	Starvation bound register for C66x CorePac slave port 1	
0x38	SBNDC2	Starvation bound register for C66x CorePac slave port 2	
0x3C	SBNDC3	Starvation bound register for C66x CorePac slave port 3	
0x40	SBNDC4	Starvation bound register for C66x CorePac slave port 4	
0x44	SBNDC5	Starvation bound register for C66x CorePac slave port 5	
0x48	SBNDC6	Starvation bound register for C66x CorePac slave port 6	
0x4C	SBNDC7	Starvation bound register for C66x CorePac slave port 7	
0x50	SBNDM	Starvation bound register for SMS port	See 3.4.2
0x54	SBNDE	Starvation bound register for SES port	See 3.4.3
0x58	Reserved	Reserved	—
0x5C	CFGLCK	Configuration Lock control for non-MPAX registers	See 3.7.1
0x60	CFGULCK	Configuration Unlock control for non-MPAX registers	See 3.7.2
0x64	CFGLCKSTAT	Configuration Lock Status for non-MPAX registers	See 3.7.3
0x68	SMS_MPAX_LCK	Configuration Lock control for SMS MPAX registers	See 3.7.4
0x6C	SMS_MPAX_ULCK	Configuration Unlock control for SMS MPAX registers	See 3.7.5
0x70	SMS_MPAX_LCKSTAT	Configuration Lock Status for SMS MPAX registers	See 3.7.6
0x74	SES_MPAX_LCK	Configuration Lock control for SES MPAX registers	See 3.7.7
0x78	SES_MPAX_ULCK	Configuration Unlock control for SES MPAX registers	See 3.7.8
0x7C	SES_MPAX_LCKSTAT	Configuration Lock Status for SES MPAX registers	See 3.7.9
0x80	SMESTAT	Interrupt Enabled Status register. ANDed value of SMIRSTAT and SMIESTAT registers	See 3.8.1
0x84	SMIRSTAT	Interrupt Raw Status register	See 3.8.2
0x88	SMIRC	Interrupt Raw Status Clear register. Writes of 0 have no effect	See 3.8.3
0x8C	SMIESTAT	Interrupt Enable Status register	See 3.8.4
0x90	SMIEC	Interrupt Enable Clear register	See 3.8.5
0x94 - 0xC0	Reserved	Reserved	—

Table 3-1 MSMC Memory Mapped Registers (Part 2 of 2)

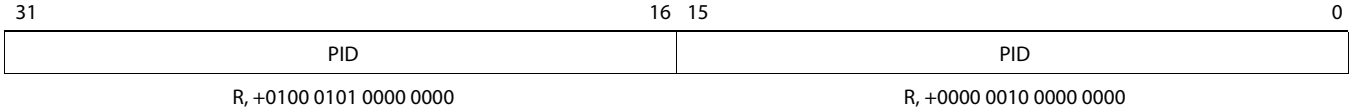
Offset	Acronym	Register Description	Section
0xC4	SMNCERRAR	MSMC SRAM non-correctable EDC error address register	See 3.3.4
0xC8	SMNCERRXR	MSMC SRAM non-correctable EDC extended error register	See 3.3.5
0xCC	SMNCEA	MSMC Scrubbing Non-Correctable Address register	See 3.3.7
0xD0 - 0x1FC	Reserved	Reserved	—
0x200:0x204	SMS_MPAXL_0_0:SMS_MPAXH_0_0	MPAX register pair 0 for SMS for PrivID 0	See 3.5.2 , 3.5.1
0x208:0x20C	SMS_MPAXL_0_1:SMS_MPAXH_0_1	MPAX register pair 1 for SMS for PrivID 0	See 3.5.2 , 3.5.1
…	…	…	—
0x5F8:0x5FC	SMS_MPAXL_F_7:SMS_MPAXH_F_7	MPAX register pair 7 for SMS for PrivID F	See 3.5.2 , 3.5.1
0x600:0x604	SES_MPAXL_0_0:SES_MPAXH_0_0	MPAX register pair 0 for SES for PrivID 0	See 3.5.2 , 3.5.3
0x608:0x60C	SES_MPAXL_0_1:SES_MPAXH_0_1	MPAX register pair 1 for SES for PrivID 0	See 3.5.2 , 3.5.3
…	…	…	—
0x9F8:0x9FC	SES_MPAXL_F_7:SES_MPAXH_F_7	MPAX register pair 7 for SES for PrivID F	See 3.5.2 , 3.5.3
End of Table 3-1			

3.2 Peripheral Identification Register (PID)

The peripheral identification register (PID) uniquely identifies the MSMC and the specific revision of the MSMC.

The PID is shown in Figure 3-1 and described in Table 3-2.

Figure 3-1 Peripheral ID Register (PID)



Legend: R = Read only; W = Write only; -*n* = value after reset; -x, value is indeterminate — see the device-specific data manual

Table 3-2 Peripheral ID Register (PID) Field Descriptions

Bit	Field	Description
31-0	PID	Peripheral identifier uniquely identifies the MSMC and the specific revision of the MSMC.

3.3 EDC Registers

3.3.1 MSMC SRAM EDC Control Register (SMEDCC)

EDC operation is controlled by SMEDCC.

The SMEDCC is shown in Figure 3-2 and described in Table 3-3.

Figure 3-2 MSMC SRAM EDC Control Register (SMEDCC)

31	30	29 27	26	25 8	7 0
SEN	ECM	Reserved	PRR	Reserved	REFDEL
RW+00	RW+00	R,+000	RW+0	R, +00 0000 0000 0000 0000	RW, +0000 0001

Legend: R = Read only; W = Write only

Table 3-3 MSMC SRAM EDC Control Register (SMEDCC) Field Descriptions

Bit	Field	Description
31	SEN	Scrubbing Engine Enable 0 = The operation of the scrubbing engine is enabled by default after reset and parity RAM initialization 1 = The operation of the scrubbing engine is disabled
30	ECM	Error Correction Mode 0 = Disables error correction 1 = Enables error correction
29-27	Reserved	Reads return 0 and writes have no effect.
26	PRR	The PRR (Parity RAM Ready) bit shows the status of Parity RAM. 0 = Parity RAM is not ready, this will be the state following a reset while the entries are being initialized. 1 = Parity RAM is ready for use in EDC and scrubbing operations. Software must consult this bit before making the first read access to MSMC RAM after reset.
25-8	Reserved	0 = Reads return 0 and writes have no effect.
7-0	REFDEL	Value = 0-FFh Controls the number of MSMC clock cycles between each scrub burst. To prevent the bursts from the scrubbing engine from posing a significant performance impact, the value in the REFDEL register is pre-scaled by 1024.
End of Table 3-3		

3.3.2 MSMC SRAM Correctable EDC Error Address Register (SMCERRAR)

On detection of a correctable one-bit error, the EDC hardware reports the error details in the correctable EDC error reporting registers.

The SMCERRAR is shown in Figure 3-3 and described in Table 3-4.

Figure 3-3 MSMC SRAM Correctable EDC Error Address Register (SMCERRAR)

31 0
SEADDR
R, +0000 0000 0000 0000 0000 0000 0000 0000

Legend: R = Read only

Table 3-4 MSMC SRAM Correctable EDC Error Address Register (SMCERRAR) Field Descriptions

Bit	Field	Description
31-0	SEADDR	Value = 0-FFFF FFFFh The lower 32 bits of the 36 bit address used in the accessing the corrupted location is stored here. The upper 4 bits of the 36 bit address in stored in SEEADDR field of the SMCERRXR register.

3.3.3 MSMC SRAM Correctable EDC Extended Error Register (SMCERRXR)

On detection of a correctable one-bit error, the EDC hardware reports the error details in the correctable EDC error reporting registers.

The SMCERRXR is shown in Figure 3-4 and described in Table 3-5.

Figure 3-4 MSMC SRAM Correctable EDC Extended Error Register (SMCERRXR)

31 24	23 16	15 9	8	7 4	3 0
Reserved	ESYN	Reserved	SER	SEPID	SEEADDR
R, +0000 0000	R, +0000 0000	R, +0000 000	R,+0	R, +0000	R, +0000

Legend: R = Read only

Table 3-5 MSMC SRAM Correctable EDC Extended Error Register (SMCERRXR) Field Descriptions

Bit	Field	Description
31-24	Reserved	0 = Reads return 0 and writes have no effect.
23-16	ESYN	Value = 0-FFh Logs the syndrome value that identifies the erroneous bit in the data.
15-9	Reserved	0 = Reads return 0 and writes have no effect.
8	SER	0 = Request came in from the C66x CorePacs. 1 = Request came in from the SMS or the SES port.
7-4	SEPID	Value = 0-Fh The PrivID associated with the access.
3-0	SEEADDR	Value = 0-Fh The upper 4 bits of the 36 bit address used in the accessing the corrupted location is stored here. The lower 32 bits of the 36 bit address in stored in SEADDR field of the SMCERRAR register.
End of Table 3-5		

3.3.4 MSMC SRAM Non-correctable EDC Error Address Register (SMNCERRAR)

On detection of a non-correctable two-bit error, the EDC hardware reports the error details in the non-correctable EDC error reporting registers.

The SMNCERRAR is shown in Figure 3-5 and described in Table 3-6.

Figure 3-5 MSMC SRAM Non-correctable EDC Error Address Register (SMNCERRAR)

31 0
SEADDR
R, +0000 0000 0000 0000 0000 0000 0000 0000

Legend: R = Read only

Table 3-6 MSMC SRAM Non-correctable EDC Error Address Register (SMNCERRAR) Field Descriptions

Bit	Field	Description
31-0	SEADDR	Value = 0-FFFF FFFFh The lower 32 bits of the 36 bit address used in the accessing the corrupted location is stored here. The upper 4 bits of the 36 bit address in stored in SEEADDR field of the SMNCERRXR register.

3.3.5 MSMC SRAM Non-correctable EDC Extended Error Register (SMNCERRXR)

On detection of a non-correctable two-bit error, the EDC hardware reports the error details in the non-correctable EDC error reporting registers.

The SMNCERRXR is shown in Figure 3-6 and described in Table 3-7.

Figure 3-6 MSMC SRAM Non-correctable EDC Extended Error Register (SMNCERRXR)

31 9	8	7 4	3 0
Reserved	SER	SEPID	SEEADDR
R, + 0000 0000 0000 0000 0000 000	R,+0	R, +0000	R, +0000

Legend: R = Read only

Table 3-7 MSMC SRAM Non-correctable EDC Extended Error Register (SMNCERRXR) Field Descriptions

Bit	Field	Description
31-9	Reserved	0 = Reads return 0 and writes have no effect.
8	SER	0 = Request came in from the C66x CorePacs. 1 = Request came in from the SMS or the SES port.
7-4	SEPID	Value = 0-Fh The PrivID associated with the access.
3-0	SEEADDR	Value = 0-Fh The upper 4 bits of the 36 bit address used in the accessing the corrupted location is stored here. The lower 32 bits of the 36 bit address in stored in SEADDR field of the SMNCERRAR register.
End of Table 3-7		

3.3.6 MSMC Scrubbing Error Corrected Address Register (SMCEA)

On detection of a correctable one-bit error, the scrubbing engine reports the address in the SMCEA register.

The SMCEA is shown in Figure 3-7 and described in Table 3-8.

Figure 3-7 MSMC Scrubbing Error Corrected Address Register (SMCEA)

31 24	23 0
ESYN	SECA
R, +0000 0000	R, +0000 0000 0000 0000 0000 0000

Legend: R = Read only

Table 3-8 MSMC Scrubbing Error Corrected Address Register (SMCEA) Field Descriptions

Bit	Field	Description
31-24	ESYN	Value = 0-FFh Logs the syndrome value that identifies the erroneous bit in the data.
23-0	SECA	Value = 0-FF FFFFh Scrubbing Error Correctable Address. If the scrubbing engine access detects that a location contents have been corrected, it logs the address.
End of Table 3-8		

3.3.7 MSMC Scrubbing Non-Correctable Address Register (SMNCEA)

On detection of a non-correctable two-bit error, the scrubbing engine reports the address in the SMNCEA register.

The SMNCEA is shown in Figure 3-8 and described in Table 3-9.

Figure 3-8 MSMC Scrubbing Non-Correctable Address Register (SMNCEA)

31	24	23	0
Reserved		SENCA	
R, +0000 0000		R, +0000 0000 0000 0000 0000 0000	

Legend: R = Read only

Table 3-9 MSMC Scrubbing Non-Correctable Address Register (SMNCEA) Field Descriptions

Bit	Field	Description
31-24	Reserved	0 = Reads return 0 and writes have no effect.
23-0	SENCA	0-FF FFFFh Value = Scrubbing Error Non-Correctable Address.
End of Table 3-9		

3.3.8 MSMC Scrubbing Error Counter Register (SMSECC)

On detection of a correctable one-bit error, the scrubbing engine increments the SCEC (Scrub Correctable Error Counter) field in the SMSECC register. On detection of a non-correctable two-bit error, the scrubbing engine increments the SNCEC (Scrub Non-Correctable Error Counter) field in the SMSECC register.

The SMSECC is shown in Figure 3-9 and described in Table 3-10.

Figure 3-9 MSMC Scrubbing Error Counter Register (SMSECC)

31	16	15	0
SNCEC		SCEC	
RW, +0000 0000 0000 0000		RW, +0000 0000 0000 0000	

Legend: R = Read only; W = Write only

Table 3-10 MSMC Scrubbing Error Counter Register (SMSECC) Field Descriptions

Bit	Field	Description
31-16	SNCEC	Value = 0-FFFFh Increments the counter on detection of a non-correctable two-bit error.
15-0	SCEC	Value = 0-FFFFh Increments the counter on detection of a correctable one-bit error.
End of Table 3-10		

3.4 Bandwidth Management Control Registers

3.4.1 Starvation Bound Register for C66x CorePac Slave Ports (SBNDC*n*)

The SBNDC*n* is shown in Figure 3-10 and described in Table 3-11.

Figure 3-10 Starvation Bound Register for C66x CorePac Slave Ports (SBNDC*n*)[1]

31 24	23 16	15 8	7 0
Reserved	SCNTCE	Reserved	SCNTCM
R,+0000 0000	RW,+0001 1111	R,+0000 0000	RW,+0001 1111

Legend: R = Read only; W = Write only

1. n = 0 to N-1, where N is the number of CorePacs available in the device.

Table 3-11 Starvation Bound Register for C66x CorePac Slave Ports (SBNDC*n*) Field Descriptions

Bit	Field	Description
31-24	Reserved	0 = Reads return 0 and writes have no effect.
23-16	SCNTCE	Value = 0-FFh Reload value (pre-scaled by 16) for the starvation counter for C66x CorePac slave n requests at the EMIF arbiter.
15-8	Reserved	0 = Reads return 0 and writes have no effect.
7-0	SCNTCM	Value = 0-FFh Reload value for the starvation counters for C66x CorePac slave n requests at the RAM bank arbiters.
End of Table 3-11		

3.4.2 Starvation Bound Register for SMS Port (SBNDM)

The SBNDM is shown in Figure 3-11 and described in Table 3-12.

Figure 3-11 Starvation Bound Register for SMS Port (SBNDM)

31 8	7 0
Reserved	SCNTMM
R,+0000 0000 0000 0000 0000 0000	RW,+0001 1111

Legend: R = Read only; W = Write only

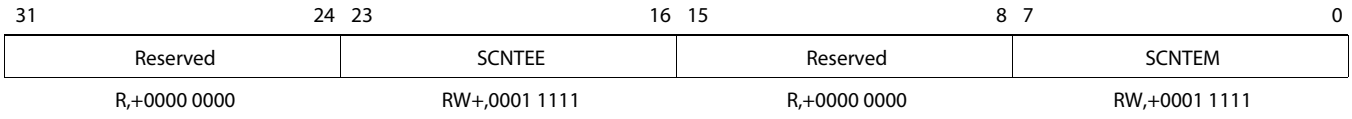
Table 3-12 Starvation Bound Register for SMS Port (SBNDM) Field Descriptions

Bit	Field	Description
31-8	Reserved	0 = Reads return 0 and writes have no effect.
7-0	SCNTMM	Value = 0-FFh Reload value for the starvation counters for SMS requests at the RAM bank arbiters.
End of Table 3-12		

3.4.3 Starvation Bound Register for SES Port (SBNDE)

The SBNDE is shown in Figure 3-12 and described in Table 3-13.

Figure 3-12 Starvation Bound Register for SES Port (SBNDE)



Legend: R = Read only; W = Write only

Table 3-13 Starvation Bound Register for SES Port (SBNDE) Field Descriptions

Bit	Field	Description
31-24	Reserved	0 = Reads return 0 and writes have no effect.
23-16	SCNTEE	Value = 0-FFh Reload value (prescaled by 16) for the starvation counter for SES requests at the EMIF arbiter.
15-8	Reserved	0 = Reads return 0 and writes have no effect.
7-0	SCNTEM	Value = 0-FFh Reload value for the starvation counters for SES requests at the RAM bank arbiters.
End of Table 3-13		

3.5 MPAX Segment Registers

3.5.1 SMS_MPAXH*n*

The SMS_MPAXH*n* is shown in Figure 3-13 and described in Table 3-14.

Figure 3-13 SMS_MPAXH*n*

31 24	23 12	11 8	7	6 5	4 0
constant	BADDR	Reserved	US	Reserved	SEGSZ
R, +0000 1100	RW, +0000 0000 0000	R, +0000	RW,+0	R,00	RW, +0 0000

Legend: R = Read only; W = Write only

Table 3-14 SMS_MPAXH*n* Field Descriptions

Bit	Field	Description
31-24	constant	0Ch = Constant value
23-12	BADDR	Value = 0-FFFh Base Address—The Segment Base Address field defines the placement of the controlled segment. This field is used to match against the incoming address on the system slave port to identify the addressed segment. For SMS and SES, MPAXH[31:12] is compared against the requested address.
11-8	Reserved	0 = Reads return 0 and writes have no effect.
7	US	0 = This memory page is shared and coherence actions will be spawned on access to this page 1 = This memory page is not shared and coherence actions will not be spawned on access to this page
6-5	Reserved	0 = Reads return 0 and writes have no effect.
4-0	SEGSZ	Value = 0-1Fh Segment Size—The Segment Size field defines the size of the controlled segment. Refer to the table Table 2-1 "MPAX Segment Size Encoding" on page 2-6 for segment size encoding details.
End of Table 3-14		

3.5.2 SMS_MPAXLn

The SMS_MPAXL*n* is shown in Figure 3-14 and described in Table 3-15.

Figure 3-14 SMS_MPAXL*n*

31 28	27 20	19 8	7 6	5	4	3	2	1	0
Reserved	constant	RADDR	Reserved	SR	SW	SX	UR	UW	UX
R, +0000	R, +0000 1100	RW, +0000 0000 0000			RW, +0000 0000				

Legend: R = Read only; W = Write only

Table 3-15 SMS_MPAXL*n* Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-28	Reserved	0 = Reads return 0 and writes have no effect.
27-20	constant	0Ch = Constant value
19-8	RADDR	0-FFFh = Replacement Address—Bits that replace and extend the upper address bits matched by BADDR.
7-6	Reserved	Reads return 0 and writes have no effect.
5	SR	Supervisor read access type. 0 = Normal operation. 1 = Indicates a supervisor read request.
4	SW	Supervisor write access type. 0 = Normal operation. 1 = Indicates a supervisor write request.

Table 3-15 SMS_MPAXL*n* Field Descriptions (Part 2 of 2)

Bit	Field	Description
3	SX	Supervisor execute access type. 0 = Normal operation. 1 = Indicates a supervisor execute request.
2	UR	User read access type. 0 = Normal operation. 1 = Indicates a user read request.
1	UW	User write access type. 0 = Normal operation. 1 = Indicates a user write request.
0	UX	User execute access type. 0 = Normal operation. 1 - Indicates a user execute request.
End of Table 3-15		

3.5.3 SES_MPAXH*n*

The SES_MPAXH*n* is shown in Figure 3-15 and described in Table 3-16.

Figure 3-15 SES_MPAXH*n*

31 12	11 8	7	6 5	4 0
BADDR	Reserved	US	Reserved	SEGSZ
RW, +0000 0000 0000 0000 0000	R, +0000	RW,+0	R,00	RW, +0 0000

Legend: R = Read only; W = Write only

Table 3-16 SES_MPAXH*n* Field Descriptions

Bit	Field	Description
31-12	BADDR	Value = 0-F FFFFh Base Address—The Segment Base Address field defines the placement of the controlled segment. This field is used to match against the incoming address on the system slave port to identify the addressed segment. For SMS and SES, MPAXH[31:12] is compared against the requested address.
11-8	Reserved	0 = Reads return 0 and writes have no effect.
7	US	0 = This memory page is shared and coherence actions will be spawned on access to this page 1 = This memory page is not shared and coherence actions will not be spawned on access to this page
6-5	Reserved	0 = Reads return 0 and writes have no effect.
4-0	SEGSZ	Value = 0-1Fh Segment Size—The Segment Size field defines the size of the controlled segment. Refer to the table Table 2-1 "MPAX Segment Size Encoding" on page 2-6 for segment size encoding details.
End of Table 3-16		

3.5.4 SES_MPAXL*n*

The SES_MPAXL*n* is shown in Figure 3-16 and described in Table 3-17.

Figure 3-16 SES_MPAXL*n*

31 8	7 6	5	4	3	2	1	0
RADDR	Reserved	SR	SW	SX	UR	UW	UX
RW, +0000 0000 0000 0000 0000 0000		RW, +0000 0000					

Legend: R = Read only; W = Write only

Table 3-17 SES_MPAXL*n* Field Descriptions

Bit	Field	Description
31-8	RADDR	Value = 0-FF FFFFh Replacement Address—Bits that replace and extend the upper address bits matched by BADDR.
7-6	Reserved	0 = Reads return 0 and writes have no effect.
5	SR	Supervisor read access type. 0 = Normal operation. 1 = Indicates a supervisor read request.
4	SW	Supervisor write access type. 0 = Normal operation. 1 = Indicates a supervisor write request.
3	SX	Supervisor execute access type. 0 = Normal operation. 1 = Indicates a supervisor execute request.
2	UR	User read access type. 0 = Normal operation. 1 = Indicates a user read request.
1	UW	User write access type. 0 = Normal operation. 1 = Indicates a user write request.
0	UX	User execute access type. 0 = Normal operation. 1 = Indicates a user execute request.
End of Table 3-17		

3.6 Memory Protection Fault Reporting Registers

3.6.1 MSMC Memory Protection Fault Address Register (SMPFAR)

The SMPFAR is shown in Figure 3-17 and described in Table 3-18.

Figure 3-17 MSMC Memory Protection Fault Address Register (SMPFAR)

31	0
Fault Address	
R, +0000 0000 0000 0000 0000 0000 0000 0000	

Legend: R = Read only

Table 3-18 MSMC Memory Protection Fault Address Register (SMPFAR) Field Descriptions

Bit	Field	Description
31-0	Fault Address	0-FFFF FFFFh = Fault Access Address

3.6.2 MSMC Memory Protection Fault Extension Register (SMPFXR)

The SMPFXR is shown in Figure 3-18 and described in Table 3-19.

Figure 3-18 MSMC Memory Protection Fault Extension Register (SMPFXR)

31	1	0
Reserved		NM
R, +0000 0000 0000 0000 0000 0000 0000 000		R, +0

Legend: R = Read only

Table 3-19 MSMC Memory Protection Fault Extension Register (SMPFXR) Field Descriptions

Bit	Field	Description
31-1	Reserved	0 = Reads return 0 and writes have no effect.
0	NM	0 = Fault is not caused by address mismatch of the segment BADDR. 1 = Fault is caused by the address not matching any of the segment BADDR.
End of Table 3-19		

3.6.3 MSMC Memory Protection Fault Requestor Register (SMPFR)

The SMPFR is shown in Figure 3-19 and described in Table 3-20.

Figure 3-19 MSMC Memory Protection Fault Requestor Register (SMPFR)

31	12	11	8	7	0
Reserved		FPID		FMSTID	
R, +0000 0000 0000 0000 0000		R, +0000		R, +0000 0000	

Legend: R = Read only

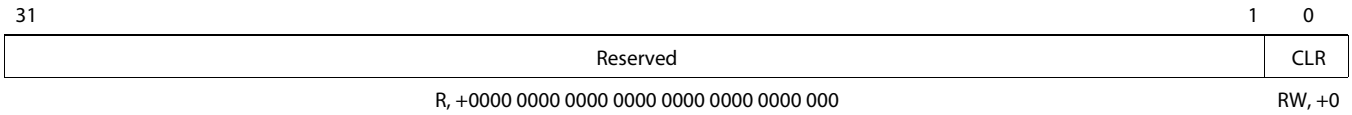
Table 3-20 MSMC Memory Protection Fault Requestor Register (SMPFR) Field Descriptions

Bit	Field	Description
31-12	Reserved	0 = Reads return 0 and writes have no effect.
11-8	FPID	0-Fh = PrivID associated with the fault access.
7-0	FMSTID	0-FFh = Master ID associated with the fault access.
End of Table 3-20		

3.6.4 MSMC Memory Protection Fault Control Register (SMPFCR)

The SMPFCR is shown in Figure 3-20 and described in Table 3-21.

Figure 3-20 MSMC Memory Protection Fault Control Register (SMPFCR)



Legend: R = Read only

Table 3-21 MSMC Memory Protection Fault Control Register (SMPFCR) Field Descriptions

Bit	Field	Description
31-1	Reserved	0 = Reads return 0 and writes have no effect.
0	CLR	0 = Writing 0 has no effect. 1 = Writing 1 to this bit clears the recorded fault in SMPFAR and SMPFXR registers, only after which further faults can be recorded.
End of Table 3-21		

3.7 MSMC Configuration Write Lock Registers

3.7.1 Configuration Lock Control for Non-MPAX Registers (CFGLCK)

The CFGLCK is shown in Figure 3-21 and described in Table 3-22.

Figure 3-21 Configuration Lock Control for Non-MPAX Registers (CFGLCK)

31 16	15 1	0
MGCID	Reserved	WLCK
W+0010 1100 1101 0000	R+0000 0000 0000 000	W+0

Legend: R = Read only; W = Write only

Table 3-22 Configuration Lock Control for Non-MPAX Registers (CFGLCK) Field Descriptions

Bit	Field	Description
31-16	MGCID	0x2CD0 = Writing this key value along with setting WLCK to 1 engages the lock.
15-1	Reserved	0 = Reads return 0 and writes have no effect.
0	WLCK	0 = Writing 0 has no effect. 1 = Writing this bit to 1 along with setting MGCID key = 0x2CD0 engages the lock.
End of Table 3-22		

3.7.2 Configuration Unlock Control for Non-MPAX Registers (CFGULCK)

The CFGULCK is shown in Figure 3-22 and described in Table 3-23.

Figure 3-22 Configuration Unlock Control for Non-MPAX Registers (CFGULCK)

31 16	15 1	0
MGCID	Reserved	WEN
W+0010 1100 1101 0000	R+0000 0000 0000 000	W+0

Legend: R = Read only; W = Write only

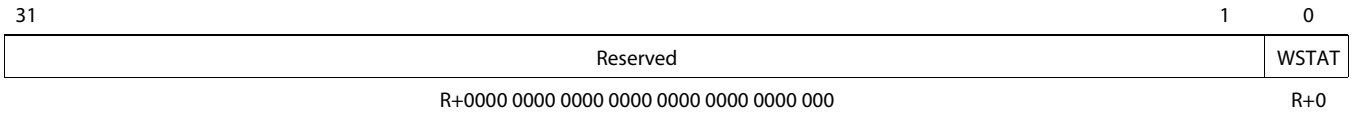
Table 3-23 Configuration Unlock Control for Non-MPAX Registers (CFGULCK) Field Descriptions

Bit	Field	Description
31-16	MGCID	0x2CD0 = Writing this key value along with setting WEN to 1 disengages the lock.
15-1	Reserved	0 = Reads return 0 and writes have no effect.
0	WEN	0 = Writing 0 has no effect. 1 = Writing this bit to 1 along with setting MGCID key = 0x2CD0 disengages the lock.
End of Table 3-23		

3.7.3 Configuration Lock Status for Non-MPAX Registers (CFGLCKSTAT)

The CFGLCKSTAT is shown in Figure 3-23 and described in Table 3-24.

Figure 3-23 Configuration Lock Status for Non-MPAX Registers (CFGLCKSTAT)



Legend: R = Read only

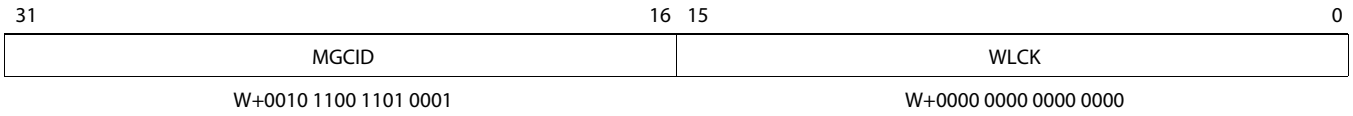
Table 3-24 Configuration Lock Status for Non-MPAX Registers (CFGLCKSTAT) Field Descriptions

Bit	Field	Description
31-1	Reserved	0 = Reads return 0 and writes have no effect.
0	WSTAT	Indicates the lock's current status. 0 = Lock is disengaged. 1 = Lock is engaged.
End of Table 3-24		

3.7.4 Configuration Lock Control for SMS MPAX Registers (SMS_MPAX_LCK)

The SMS_MPAX_LCK is shown in Figure 3-24 and described in Table 3-25.

Figure 3-24 Configuration Lock Control for SMS MPAX Registers (SMS_MPAX_LCK)



Legend: W = Write only

Table 3-25 Configuration Lock Control for SMS MPAX Registers (SMS_MPAX_LCK) Field Descriptions

Bit	Field	Description
31-16	MGCID	Value = 0x2CD1 Writing this key value along with setting WLCK[n] to 1 engages the lock for PrivID n.
15-0	WLCK	Value = 0-FFFFh Bit n denotes lock bit for PrivID n. Writing WLCK[n] bit to 0 has no effect. Writing WLCK[n] bit to 1 along with setting MGCID key = 0x2CD1 engages the lock for PrivID n.
End of Table 3-25		

3.7.5 Configuration Unlock Control for SMS MPAX Registers (SMS_MPAX_ULCK)

The SMS_MPAX_ULCK is shown in Figure 3-25 and described in Table 3-26.

Figure 3-25 Configuration Unlock Control for SMS MPAX Registers (SMS_MPAX_ULCK)

31	16	15	0
MGCID		WEN	
W+0010 1100 1101 0001		W+0000 0000 0000 0000	

Legend: W = Write only

Table 3-26 Configuration Unlock Control for SMS MPAX Registers (SMS_MPAX_ULCK) Field Descriptions

Bit	Field	Description
31-16	MGCID	Value = 0x2CD1 Writing this key value along with setting WEN[n] to 1 disengages the lock for PrivID n.
15-0	WEN	Value = 0-FFFFh Bit n denotes unlock bit for PrivID n. Writing WEN[n] bit to 0 has no effect. Writing WEN[n] bit to 1 along with setting MGCID key = 0x2CD1 disengages the lock for PrivID n.
End of Table 3-26		

3.7.6 Configuration Lock Status for SMS MPAX Registers (SMS_MPAX_LCKSTAT)

The SMS_MPAX_LCKSTAT is shown in Figure 3-26 and described in Table 3-27.

Figure 3-26 Configuration Lock Status for SMS MPAX Registers (SMS_MPAX_LCKSTAT)

31	16	15	0
Reserved		WSTAT	
R+0000 0000 0000 0000		R+0000 0000 0000 0000	

Legend: R = Read only

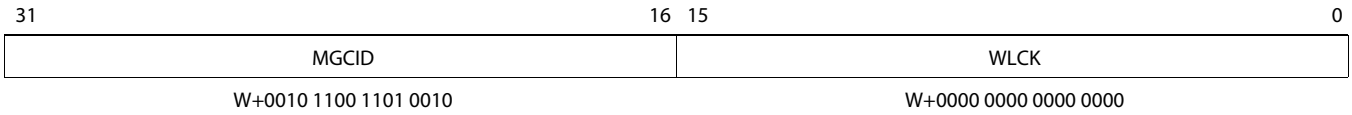
Table 3-27 Configuration Lock Status for SMS MPAX Registers (SMS_MPAX_LCKSTAT) Field Descriptions

Bit	Field	Description
31-16	Reserved	0 = Reads return 0 and writes have no effect.
15-0	WSTAT	Value = 0-FFFFh Bit n indicates the lock's current status for PrivID n. 0 - Lock is disengaged. 1 - Lock is engaged.
End of Table 3-27		

3.7.7 Configuration Lock Control for SES MPAX Registers (SES_MPAX_LCK)

The SES_MPAX_LCK is shown in Figure 3-27 and described in Table 3-28.

Figure 3-27 Configuration Lock Control for SES MPAX Registers (SES_MPAX_LCK)



Legend: W = Write only

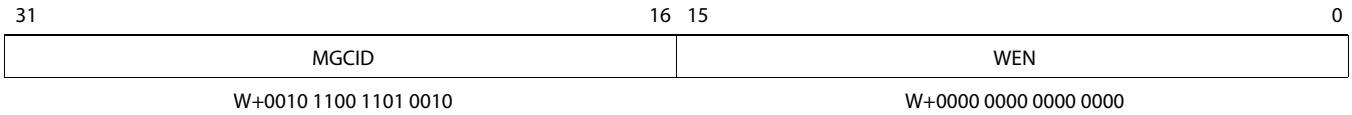
Table 3-28 Configuration Lock Control for SES MPAX Registers (SES_MPAX_LCK) Field Descriptions

Bit	Field	Description
31-16	MGCID	Value = 0x2CD2 Writing this key value along with setting WLCK[n] to 1 engages the lock for PrivID n.
15-0	WLCK	Value = 0-FFFFh Bit n denotes lock bit for PrivID n. Writing WLCK[n] bit to 0 has no effect. Writing WLCK[n] bit to 1 along with setting MGCID key = 0x2CD1 engages the lock for PrivID n.
End of Table 3-28		

3.7.8 Configuration Unlock Control for SES MPAX Registers (SES_MPAX_ULCK)

The SES_MPAX_ULCK is shown in Figure 3-28 and described in Table 3-29.

Figure 3-28 Configuration Unlock Control for SES MPAX Registers (SES_MPAX_ULCK)



Legend: W = Write only

Table 3-29 Configuration Unlock Control for SES MPAX Registers (SES_MPAX_ULCK) Field Descriptions

Bit	Field	Description
31-16	MGCID	Value = 0x2CD2 Writing this key value along with setting WEN[n] to 1 disengages the lock for PrivID n.
15-0	WEN	Value = 0-FFFFh Bit n denotes unlock bit for PrivID n. Writing WEN[n] bit to 0 has no effect. Writing WEN[n] bit to 1 along with setting MGCID key = 0x2CD1 disengages the lock for PrivID n.
End of Table 3-29		

3.7.9 Configuration Lock Status for SES MPAX Registers (SES_MPAX_LCKSTAT)

The SES_MPAX_LCKSTAT is shown in Figure 3-29 and described in Table 3-30.

Figure 3-29 Configuration Lock Status for SES MPAX Registers (SES_MPAX_LCKSTAT)

31	16	15	0
Reserved		WSTAT	
R+0000 0000 0000 0000		R+0000 0000 0000 0000	

Legend: R = Read only

Table 3-30 Configuration Lock Status for SES MPAX Registers (SES_MPAX_LCKSTAT) Field Descriptions

Bit	Field	Description
31-16	Reserved	0 = Reads return 0 and writes have no effect.
15-0	WSTAT	Value = 0-FFFFh Bit n indicates the lock's current status for PrivID n. 0 - Lock is disengaged. 1 - Lock is engaged.
End of Table 3-30		

3.8 MSMC Interrupt Control Registers

3.8.1 Interrupt Enabled Status Register (SMESTAT)

The SMESTAT register provides the status of those interrupts that are enabled; that is, a bitfield in the SMESTAT register is set if the associated interrupt is both enabled and has occurred.

The SMESTAT is shown in Figure 3-30 and described in Table 3-31.

Figure 3-30 Interrupt Enabled Status Register (SMESTAT)

31 16	15 4	3	2	1	0
PFESTAT	Reserved	CEES	NCEES	CSES	NCSES
R +0000 0000 0000 0000	R +0000 0000 0000	R +0	R +0	R +0	R +0

Legend: R = Read only

Table 3-31 Interrupt Enabled Status Register (SMESTAT) Field Descriptions

Bit	Field	Description
31-16	PFESTAT	Value = 0-FFFFh Bit n denotes the protection fault interrupt is enabled and pending for PrivID n.
15-4	Reserved	0 = Reads return 0 and writes have no effect.
3	CEES	0 = No EDC error. 1 = Correctable EDC error interrupt is enabled and pending.
2	NCEES	0 = No EDC error. 1 = Non-correctable EDC error interrupt is enabled and pending.
1	CSES	0 = No scrubbing error. 1 = Correctable scrubbing error interrupt is enabled and pending.
0	NCSES	0 - No scrubbing error. 1 = Non-correctable scrubbing error interrupt is enabled and pending.
End of Table 3-31		

3.8.2 Interrupt Raw Status Register (SMIRSTAT)

The SMIRSTAT register stores the raw interrupt status for each interrupt line, a bitfield is set if the associated interrupt has occurred. An interrupt pulse can be generated and its raw status recorded in SMIRSTAT either when the event occurs in hardware (hardware interrupt) or when software writes to the associated bitfield in the SMIRSTAT register (software interrupt for simulating the event).

The SMIRSTAT is shown in Figure 3-31 and described in Table 3-32.

Figure 3-31 Interrupt Raw Status Register (SMIRSTAT)

31 16	15 4	3	2	1	0
PFISTAT	Reserved	CEI	NCEI	CSI	NCSI
RW +0000 0000 0000 0000	R +0000 0000 0000	RW +0	RW +0	RW +0	RW +0

Legend: R = Read only; W = Write only

Table 3-32 Interrupt Raw Status Register (SMIRSTAT) Field Descriptions

Bit	Field	Description
31-16	PFISTAT	Value = 0-FFFFh Bit n denotes a protection fault for PrivID n. Write 1 to bit n sets the protection fault interrupt status for PrivID n.
15-4	Reserved	0 = Reads return 0 and writes have no effect.
3	CEI	0 = No EDC error. 1 - Correctable EDC error. Write 1 to set the correctable EDC error interrupt status.
2	NCEI	0 = No EDC error. 1 = Non-correctable EDC error. Write 1 to set the non-correctable EDC error interrupt status.
1	CSI	0 = No scrubbing error. 1 = Correctable scrubbing error. Write 1 to set the correctable scrubbing error interrupt status.
0	NCSI	0 = No scrubbing error. 1 = Non-correctable scrubbing error. Write 1 to set the non-correctable scrubbing error interrupt status.
End of Table 3-32		

3.8.3 Interrupt Raw Status Clear Register (SMIRC)

The SMIRC is shown in Figure 3-32 and described in Table 3-33.

Figure 3-32 Interrupt Raw Status Clear Register (SMIRC)

31 16	15 4	3	2	1	0
PFIC	Reserved	CEC	NCEC	CSC	NCSC
W +0000 0000 0000 0000	R +0000 0000 0000	W +0	W +0	W +0	W +0

Legend: R = Read only; W = Write only

Table 3-33 Interrupt Raw Status Clear Register (SMIRC) Field Descriptions

Bit	Field	Description
31-16	PFIC	Value = 0-FFFFh Write 1 to bit n clears the protection fault interrupt status for PrivID n.
15-4	Reserved	0 = Reads return 0 and writes have no effect.
3	CEC	0 = No effect. 1 = Clear the correctable EDC error interrupt status.
2	NCEC	0 = No effect. 1 = Clear the non-correctable EDC error interrupt status.
1	CSC	0 = No effect. 1 = Clear the correctable scrubbing error interrupt status.
0	NCSC	0 = No effect. 1 = Clear the non-correctable scrubbing error interrupt status.
End of Table 3-33		

3.8.4 Interrupt Enable Status Register (SMIESTAT)

The SMIESTAT is shown in Figure 3-33 and described in Table 3-34.

Figure 3-33 Interrupt Enable Status Register (SMIESTAT)

31 16	15 4	3	2	1	0
PFIESTAT	Reserved	CEIE	NCEIE	CSIE	NCSIE
RW +0000 0000 0000 0000	R +0000 0000 0000	RW +0	RW +0	RW +0	RW +0

Legend: R = Read only; W = Write only

Table 3-34 Interrupt Enable Status Register (SMIESTAT) Field Descriptions

Bit	Field	Description
31-16	PFIESTAT	Value = 0-FFFFh Bit n denotes the protection fault interrupt is enabled for PrivID n.
15-4	Reserved	0 = Reads return 0 and writes have no effect.
3	CEIE	0 = Correctable EDC error interrupt is disabled. 1 = Correctable EDC error interrupt is enabled.
2	NCEIE	0 = Non-correctable EDC error interrupt is disabled. 1 = Non-correctable EDC error interrupt is enabled.
1	CSIE	0 = Correctable scrubbing error interrupt is disabled. 1 = Correctable scrubbing error interrupt is enabled.
0	NCSIE	0 = Non-correctable scrubbing error interrupt is disabled. 1 = Non-correctable scrubbing error interrupt is enabled.
End of Table 3-34		

3.8.5 Interrupt Enable Clear Register (SMIEC)

The SMIEC is shown in Figure 3-34 and described in Table 3-35.

Figure 3-34 Interrupt Enable Clear Register (SMIEC)

31 16	15 4	3	2	1	0
PFIEC	Reserved	CEEC	NCEEC	CSEC	NCSEC
W +0000 0000 0000 0000	R +0000 0000 0000	W +0	W +0	W +0	W +0

Legend: R = Read only; W = Write only

Table 3-35 Interrupt Enable Clear Register (SMIEC) Field Descriptions

Bit	Field	Description
31-16	PFIEC	Value = 0-FFFFh Write 1 to bit n clears the protection fault interrupt enable for PrivID n.
15-4	Reserved	0 = Reads return 0 and writes have no effect.
3	CEEC	0 = No effect. 1 = Clear the correctable EDC error interrupt enable.
2	NCEEC	0 = No effect. 1 = Clear the non-correctable EDC error interrupt enable.
1	CSEC	0 = No effect. 1 = Clear the correctable scrubbing error interrupt enable.
0	NCSEC	0 = No effect. 1 = Clear the non-correctable scrubbing error interrupt enable.
End of Table 3-35		

Index

T

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components which meet ISO/TS16949 requirements, mainly for automotive use. Components which have not been so designated are neither designed nor intended for automotive use; and TI will not be responsible for any failure of such components to meet such requirements.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265