DSP Subsystems

This chapter describes the features and functions of the device integrated digital processing subsystems.

Topic **Page**

5.1 DSP Subsystems Overview

The device includes two identical instances (DSP1 and DSP2) of a digital signal processor (DSP) subsystem, based on the TI's standard TMS320C66x™ DSP CorePac core.

The TMS320C66x DSP core enhances the TMS320C674x™ core, which merges the C674x™ floating point and the C64x+™ fixed-point instruction set architectures. The C66x DSP is object-code compatible with the C64x+/C674x DSPs.

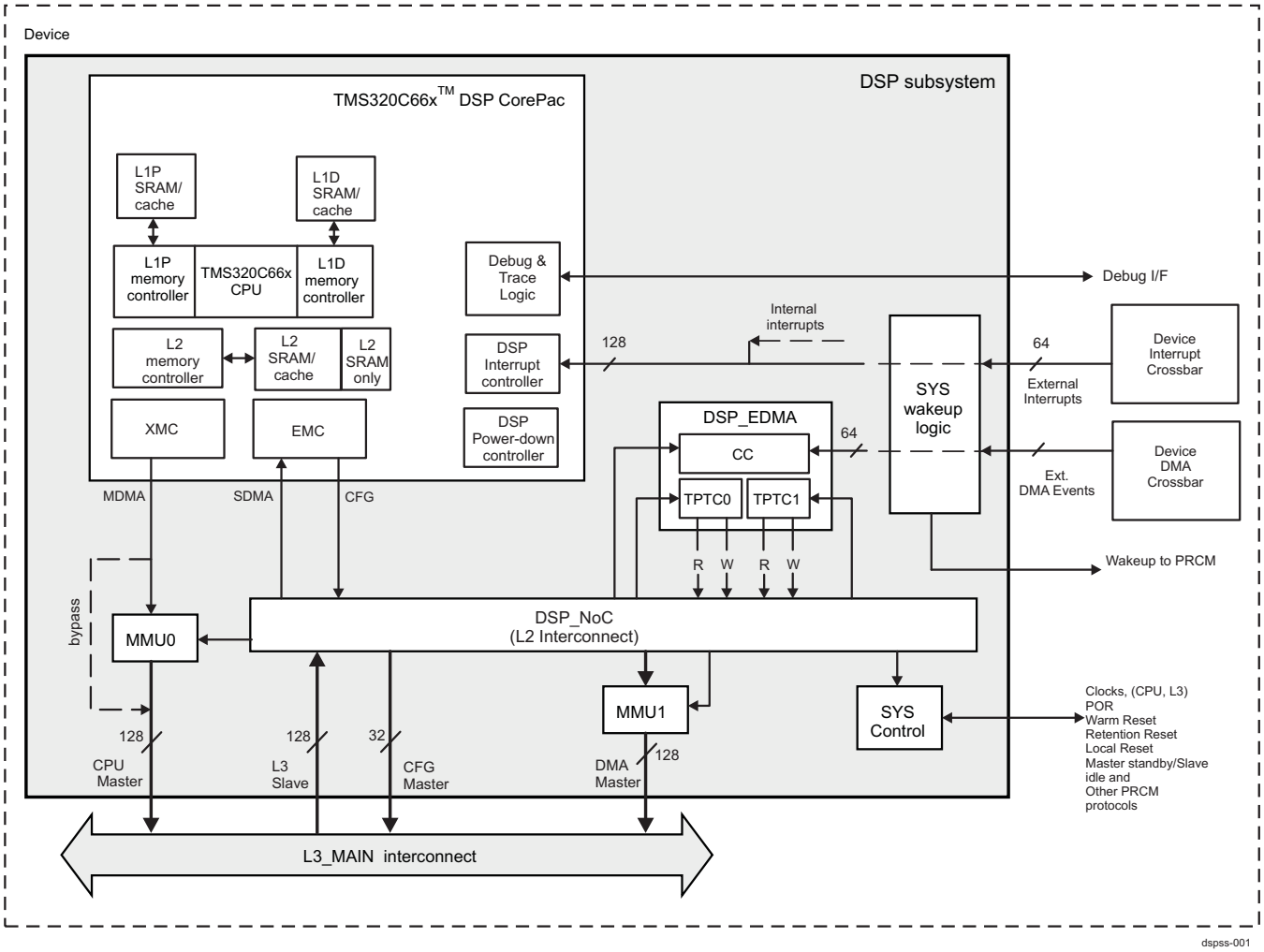
For more information on the TMS320C66x core CPU, see the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

Each of the two DSP subsystems integrated in the device includes the following components:

- A TMS320C66x™ CorePac DSP core that encompasses :
  - L1 program-dedicated (L1P) cacheable memory
  - L1 data-dedicated (L1D) cacheable memory
  - L2 (program and data) cacheable memory
  - Extended Memory Controller (XMC)
  - External Memory Controller (EMC)
  - DSP CorePac located interrupt controller (INTC)
  - DSP CorePac located power-down controller (PDC)
- Dedicated enhanced data memory access engine - EDMA, to transfer data from/to memories and peripherals external to the DSP subsystems and to local DSP memory (most commonly L2 SRAM). The external DMA requests are passed through DSP system level (SYS) wakeup logic, and collected from the DSP1 / DSP2 dedicated outputs of the device DMA Events Crossbar for each of the two subsystems.
- A level 2 (L2) interconnect network (DSP NoC) to allow connectivity between different modules of the subsystem or the remainder of the device via the device L3_MAIN interconnect.
- Two memory management units (on EDMA L2 interconnect and DSP MDMA paths) for accessing the device L3_MAIN interconnect address space
- Dedicated system control logic (DSP_SYSTEM) responsible for power management, clock generation, and connection to the device power, reset, and clock management (PRCM) module

Figure 5-1 is the DSP subsystem top-level architecture.

Figure 5-1. DSP Subsystem Highlight



5.1.1 DSP Subsystems Key Features

The TMS320C66x Instruction Set Architecture (ISA) is the latest for the C6000 family. As with its predecessors (C64x, C64x+ and C674x), the C66x is an advanced VLIW architecture with 8 functional units (two multiplier units and six arithmetic logic units) that operate in parallel. The C66x CPU has a total of 64 general-purpose 32-bit registers.

Some features of the DSP C6000 family devices are :

- Advanced VLIW CPU with eight functional units (two multipliers and six ALUs) which:
  - Executes up to eight instructions per cycle for up to ten times the performance of typical DSPs
  - Allows designers to develop highly effective RISC-like code for fast development time
- Instruction packing
  - Gives code size equivalence for eight instructions executed serially or in parallel
  - Reduces code size, program fetches, and power consumption
- Conditional execution of most instructions
  - Reduces costly branching
  - Increases parallelism for higher sustained performance
- Efficient code execution on independent functional units
  - Industry's most efficient C compiler on DSP benchmark suite

  - Industry's first assembly optimizer for fast development and improved parallelization
- 8-/16-/32-bit/64-bit data support, providing efficient memory support for a variety of applications
- 40-bit arithmetic options which add extra precision for vocoders and other computationally intensive applications
- Saturation and normalization to provide support for key arithmetic operations
- Field manipulation and instruction extract, set, clear, and bit counting support common operation found in control and data manipulation applications.

The C66x CPU has the following additional features :

- Each multiplier can perform two 16 × 16-bit or four 8 × 8 bit multiplies every clock cycle.
- Quad 8-bit and dual 16-bit instruction set extensions with data flow support
- Support for non-aligned 32-bit (word) and 64-bit (double word) memory accesses
- Special communication-specific instructions have been added to address common operations in error-correcting codes.
- Bit count and rotate hardware extends support for bit-level algorithms.
- Compact instructions: Common instructions (AND, ADD, LD, MPY) have 16-bit versions to reduce code size.
- Protected mode operation: A two-level system of privileged program execution to support higher-capability operating systems and system features such as memory protection.
- Exceptions support for error detection and program redirection to provide robust code execution
- Hardware support for modulo loop operation to reduce code size and allow interrupts during fully-pipelined code
- Each multiplier can perform 32 × 32 bit multiplies
- Additional instructions to support complex multiplies allowing up to eight 16-bit multiply/add/subtracts per clock cycle

The TMS320C66x has the following key improvements to the ISA :

- 4x Multiply Accumulate improvement for both fixed and floating point
- Improvement of the floating point arithmetic
- Enhancement of the vector processing capability for fixed and floating point
- Addition of domain-specific instructions for complex arithmetic and matrix operations

On the C66x ISA, the vector processing capability is improved by extending the width of the SIMD instructions. The C674x DSP supports 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. C66x enhances this capabilities with the addition of SIMD instructions for 32-bit data allowing operation on 128-bit vectors. For example the QMPY32 instruction is able to perform the element to element multiplication between two vectors of four 32-bit data each.

C66x ISA includes a set of specific instructions to handle complex arithmetic and matrix operations.

- **TMS320C66x DSP CorePac memory components :**
  - A 32-KiB L1 program memory (L1P) configurable as cache and / or SRAM:
    - When configured as a cache, the L1P is a 1-way set-associative cache with a 32-byte cache line
    - The DSP CorePac L1P memory controller provides bandwidth management, memory protection, and power-down functions
    - The L1P is capable of cache block and global coherence operations
    - The L1P controller has an Error Detection (ED) mechanism, including necessary SRAM
    - The L1P memory can be fully configured as a cache or SRAM
    - Page size for L1P memory is 2KB
  - A 32-KiB L1 data memory (L1D) configurable as cache and / or SRAM:
    - When configured as a cache, the L1D is a 2-way set-associative cache with a 64-byte cache line

        - The DSP CorePac L1D memory controller provides bandwidth management, memory protection, and power-down functions
        - The L1D memory can be fully configured as a cache or SRAM
        - No support for error correction or detection
        - Page size for L1D memory is 2KB
    - A 288-KiB (program and data) L2 memory, only part of which is cacheable:
        - When configured as a cache, the L2 memory is a 4-way set associative cache with a 128-byte cache line
        - Only 256 KiB of L2 memory can be configured as cache or SRAM
        - 32 KiB of the L2 memory is always mapped as SRAM
        - The L2 memory controller has an Error Correction Code (ECC) and ED mechanism, including necessary SRAM
        - The L2 memory controller supports hardware prefetching and also provides bandwidth management, memory protection, and power-down functions.
        - Page size for L2 memory is 16KB
- The **External Memory Controller (EMC)** is a bridge from the C66x CorePac to the rest of the DSP subsystem and device. It has :
    - a 32-bit configuration port (CFG) providing access to local subsystem resources (like DSP_EDMA, DSP_SYSTEM, etc) or to L3_MAIN resources accessible via the CFG address range.
    - a 128-bit slave-DMA port (SDMA) which provides accesses of system masters outside the DSP subsystem to resources inside the DSP subsystem or C66x DSP CorePac memories, i.e. when the DSP subsystem is the slave in a transaction.
- The Extended Memory Controller (XMC) processes requests from the L2 Cache Controller (which are a result of CPU instruction fetches, load/store commands, cache operations) to device resources via the C66x DSP CorePac 128-bit master DMA (MDMA) port :
    - Memory protection for addresses outside C66x DSP CorePac generated over device L3_MAIN on the MDMA port
    - Prefetch, multi-in-flight requests
- A DSP local **Interrupt Controller (INTC)** in the DSP C66x CorePac, interfaces the system events to the DSP C66x core CPU interrupt and exceptions inputs. Each DSP subsystem C66x CorePac interrupt controller supports up to 128 system events of which 64 interrupts are external to DSP subsystems, collected from the DSP1 /DSP2 dedicated outputs of the device Interrupt Crossbar.
- **Local Enhanced Direct Memory Access (EDMA) controller feaures:**
    - Channel controller (CC) : 64-channel, 128 PaRAM, 2 Queues
    - 2 x Third-party Transfer Controllers (TPTC0 and TPTC1):
        - Each TC has a 128-bit read port and a 128-bit write port
        - 2KiB FIFOs on each TPTC
    - 1-dimensional/2-dimensional (1D/2D) addressing
    - Chaining capability
- **DSP subsystems integrated MMUs**:
    - Two MMUs are integrated:
        - The MMU0 is located between DSP MDMA master port and the device L3_MAIN interconnect and can be optionally bypassed
        - The MMU1 is located between the EDMA master port and the device L3_MAIN interconnect
- A DSP local **Power-Down Controller (PDC)** is responsible to power-down various parts of the DSP C66x CorePac, or the entire DSP C66x CorePac.
- The DSP subsystems **System Control logic** provides:
    - Slave idle and master standby protocols with device PRCM for powerdown
    - OCP Disconnect handshake for init and target busses

  - Asynchronous reset
  - Power-down modes :
    - "Clockstop" mode featuring wake-up on interrupt event. The DMA event wake-up is managed in software.
- The device DSP subsystems are supplied by a PRCM DPLL, but each DSP1/2 **has integrated its own PLL module** outside the C66x CorePac for clock gating and division.
- **Each of the two device DSP subsystems has following port instances** to connect to remaining part of the device. See also Figure 5-1:
  - A 128-bit initiator (DSP MDMA master) port for MDMA/Cache requests
  - A 128-bit initiator (DSP EDMA master) port for EDMA requests
  - A 32-bit initiator (DSP CFG master) port for configuration requests
  - A 128-bit target (DSP slave) port for requests to DSP memories and various peripherals
- **C66x DSP subsystems (DSPSS) safety aspects** :
  - Above mentioned memory ECC/ED mechanisms
  - MMUs enable mapping of only the necessary application space to the processor
  - Memory Protection Units internal to the DSPSS (in L1P,L1D and L2 memory controllers) and external to DSPSS (firewalls) to help define legal accesses and raise exceptions on illegal accesses
  - Exceptions: Memory errors, various DSP errors, MMU errors and some system errors are detected and cause exceptions. The exceptions could be handled by the DSP or by a designated safety processor at the chip level. Note that it may not be possible for the safety processor to completely handle some exceptions

Unsupported features on the C66x DSP core for the device are:

- The Extended Memory Controller MPAX (memory protection and address extension) 36-bit addressing is NOT supported

Known DSP subsystem powermode restrictions for the device are :

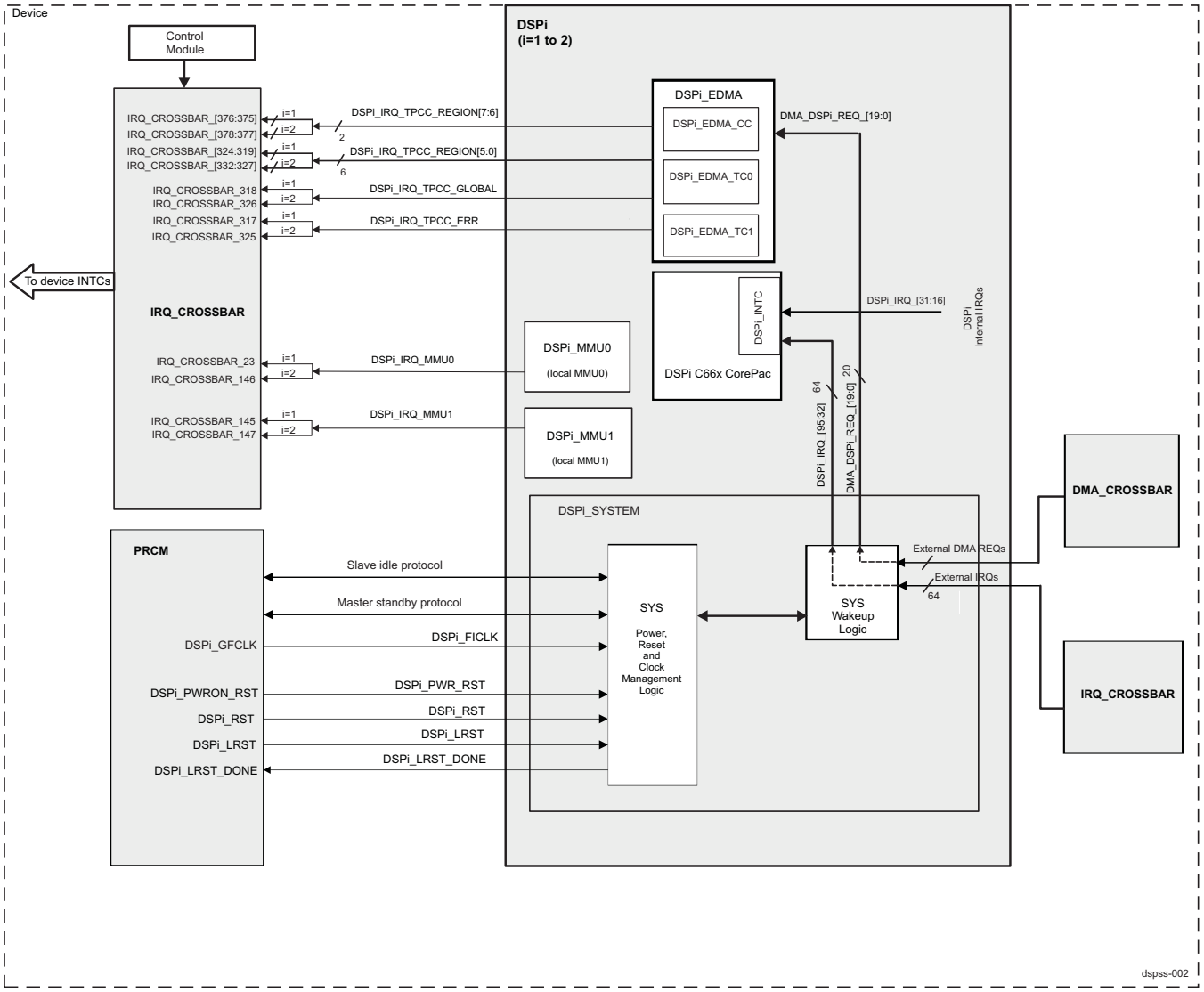
- "Full logic / RAM retention" mode featuring wake-up on both interrupt or DMA event (logic in “always on” domain). Only OFF mode is supported by DSP subsystem, **requiring full boot.**

5.2 DSP Subsystem Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 5-2 shows the integration of the DSP subsystem.

Figure 5-2. DSP Subsystem Integration



NOTE: For more information about the slave idle protocol and the wake-up request, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 5-1 through Table 5-3 summarize the integration of the module in the device.

Table 5-1. DSP Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DSP1	PD_DSP1	L3_MAIN
DSP2	PD_DSP2	L3_MAIN

Table 5-2. DSP Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSP1	DSP1_FICLK	DSP1_GFCLK	PRCM module	DSP1 subsystem gateable interface and functional clock.
DSP2	DSP2_FICLK	DSP2_GFCLK	PRCM module	DSP2 subsystem gateable interface and functional clock.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSP1	DSP1_PWR_RST	DSP1_PWRON_RST	PRCM module	For information about PRCM reset sources and distribution, see Section 3.10.2, *PD_DSP1 Description* in Chapter 3, *Power, Reset, and Clock Management*. For DSP1 local reset details see also the Section 5.3.3.2.
	DSP1_RST	DSP1_RST	PRCM module	
	DSP1_LRST	DSP1_LRST	PRCM module	
	DSP1_LRST_DONE(1)	DSP1_LRST_DONE	DSP1	
DSP2	DSP2_PWR_RST	DSP2_PWRON_RST	PRCM module	For information about PRCM reset sources and distribution, see Section 3.10.3, *PD_DSP2 Description* in Chapter 3, *Power, Reset, and Clock Management*. For DSP2 local reset details see also the Section 5.3.3.2.
	DSP2_RST	DSP2_RST	PRCM module	
	DSP2_LRST	DSP2_LRST	PRCM module	
	DSP2_LRST_DONE(1)	DSP2_LRST_DONE	DSP2	

(1) Destination of this local reset monitoring signal is the PRCM

NOTE: For information about PRCM clock gating and management, see Section 3.10.2, *PD_DSP1 Description* and Section 3.10.3, *PD_DSP2 Description* in Chapter 3, *Power, Reset, and Clock Management*.

The DSP1 / DSP2 generates a number of interrupt requests (IRQs) mapped via the device IRQ_CROSBBAR to other device interrupt controllers (outside DSP subsystem). They are described in Table 5-3.

Table 5-3. DSP Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DSP1	DSP1_IRQ_MMU0	IRQ_CROSSBAR_23	MPU_IRQ_28 DSP1_IRQ_54 DSP2_IRQ_54 PRUSS1_IRQ_54 PRUSS2_IRQ_54	Interrupt from the DSP1 subsystem local MMU0 (DSP1_MMU0CFG).
	DSP1_IRQ_MMU1	IRQ_CROSSBAR_145	-	DSP1 subsystem local MMU1 (DSP1_MMUCFG1) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_ERR	IRQ_CROSSBAR_317	-	DSP1 subsystem aggregated ("OR-ed") error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_GLOBAL	IRQ_CROSSBAR_318	-	DSP1 subsystem EDMA channel controller global interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION0	IRQ_CROSSBAR_319	-	DSP1 subsystem EDMA channel controller REGION0 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION1	IRQ_CROSSBAR_320	-	DSP1 subsystem EDMA channel controller REGION1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION2	IRQ_CROSSBAR_321	-	DSP1 subsystem EDMA channel controller REGION2 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 5-3. DSP Hardware Requests (continued)

	DSP1_IRQ_TPCC_REGION3	IRQ_CROSSBAR_322	-	DSP1 subsystem EDMA channel controller REGION3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION4	IRQ_CROSSBAR_323	-	DSP1 subsystem EDMA channel controller REGION4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION5	IRQ_CROSSBAR_324	-	DSP1 subsystem EDMA channel controller REGION5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION6	IRQ_CROSSBAR_375	-	DSP1 subsystem EDMA channel controller REGION6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION7	IRQ_CROSSBAR_376	-	DSP1 subsystem EDMA channel controller REGION7 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2	DSP2_IRQ_MMU0	IRQ_CROSSBAR_146	-	DSP2 subsystem local MMU0 (DSP2_MMU0CFG) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_MMU1	IRQ_CROSSBAR_147	-	DSP2 subsystem local MMU1 (DSP2_MMUCFG1) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_ERR	IRQ_CROSSBAR_325	-	DSP2 subsystem aggregated ("OR-ed") error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_GLOBAL	IRQ_CROSSBAR_326	-	DSP2 subsystem EDMA channel controller global interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_REGION0	IRQ_CROSSBAR_327	-	DSP2 subsystem EDMA channel controller REGION0 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 5-3. DSP Hardware Requests (continued)

DSP2_IRQ_TPCC_REGION1	IRQ_CROSSBAR_328	-	DSP2 subsystem EDMA channel controller REGION1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION2	IRQ_CROSSBAR_329	-	DSP2 subsystem EDMA channel controller REGION2 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION3	IRQ_CROSSBAR_330	-	DSP2 subsystem EDMA channel controller REGION3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION4	IRQ_CROSSBAR_331	-	DSP2 subsystem EDMA channel controller REGION4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION5	IRQ_CROSSBAR_332	-	DSP2 subsystem EDMA channel controller REGION5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION6	IRQ_CROSSBAR_377	-	DSP2 subsystem EDMA channel controller REGION6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION7	IRQ_CROSSBAR_378	-	DSP2 subsystem EDMA channel controller REGION7 interrupt. This IRQ source signal is not mapped by default to any device INTC.

NOTE: The DSP1 / DSP2 does NOT generate any DMA requests towards other device DMA controllers outside DSP1 / DSP2 (EDMA, DMA_SYSTEM, etc.).

NOTE: The "**Default Mapping**" column in Table 5-3, *DSP Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

NOTE:

- For a description of the interrupt source controls at DSP subsystem level, see Section 5.3.4, *DSP Interrupt Requests*.

DSP1/DSP2 subsystem external interrupt sources: The default interrupt sources mapped by the device IRQ_CROSSBAR to the DSP1 / DSP2 interrupt controller lines are described in the Chapter 17, *Interrupt Controllers*. The programmable muxing of various external interrupt sources to the DSP1_INTC.DSP1_IRQ_x and DSP2_INTC.DSP2_IRQ_x input lines (where x=32 to 95) is covered in the Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, of the Chapter 18, *Control Module*.

DSP1/DSP2 subsystem internal interrupt sources: The mapping of DSP subsystem internal IRQ sources to DSP1_IRQ_x / DSP2_IRQ_x lines (where x=0 to 31 and x=96 to 127 for DSP subsystem internal event sources) is described in the Section 5.3.4, *DSP Interrupt Requests*.

DSP1/DSP2 subsystem external DMA request sources: The Table 5-6 and Table 5-7 lists the default DSP1 and DSP2 external DMA request sources, respectively, routed via the device DMA_CROSSBAR to the DSP1_EDMA / DSP2_EDMA channel controller inputs (DMA_DSP1_DREQ_i / DMA_DSP2_DREQ_i).

5.3 DSP Subsystems Functional Description

5.3.1 DSP Subsystems Block Diagram

Each of the device two DSP subsystems - DSP1 and DSP2 is composed of a DSP C66x CorePac coupled with several other submodules that enable its integration in the device architecture. Device DSP subsystem provides :

- a 128-bit master data port (MDMA) on device L3_MAIN with a dedicated DSP subsystem local MMU (MMU0) on the path.
- a 32-bit master configuration port (CFG) on device L3_MAIN through which DSP host configures various device located peripherals (external to the DSP subsystem).
- a 128-bit slave DMA port (SDMA) on device L3_MAIN which allows external initiators (masters) to DSP to manipulate some portion of its config / status registers (those which are mapped in the L3_MAIN space) in the device
- a 128-bit master EDMA port - which allows the DSP_EDMA traffic controllers to initiate transfers on L3_MAIN.

The C66x DSP subsystem is illustrated in the Figure 5-1.

5.3.2 DSP Subsystem Components

5.3.2.1 C66x DSP Subsystem Introduction

The key component of the C66x DSP subsystem is built on the TI's high performance TMS320C66x DSP CorePac which consists of a single TMS320C66x CPU (DSP_C0) processor along with a level 1 (L1P and L1D) cacheable SRAM and a level 2 (L2) cacheable SRAM memories interfaced via associated local L1P, L1D and L2 memory controllers, respectively. A DSP C66x CorePac includes also some other internal peripheral components, see Section 5.3.2.2.3 for details.

This chapter provides an overview of the DSP subsystem and the following considerations associated with it :

- DSP C66x CorePac Core and L1 / L2 Memories
- DSP System control and configuration :
  - clock management
  - wake-up event generation
  - interrupt masking
- DSP Booting
- DSP subsystem internal memory and external memory (L3_MAIN) space views
- DSP INTC interrupts mapping, event combining and exception generation
- DMA requests mapping to EDMA channels and EDMA traffic routing
- Others

For more information on the TMS320C66x DSP CorePac, refer to the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C), the *TMS320C66x DSP Cache User Guide*, (SPRUGY8) and the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

5.3.2.2 DSP TMS320C66x CorePac

The TMS320C66x DSP CorePac is illustrated on Figure 5-1. It consists of a single DSP C66x CPU (DSP_C0) processor tightly coupled with level 1 - L1P (program), L1D (data) cacheable SRAM memories and level 2 (L2) cacheable SRAM memories. The C66x CorePac integrated memories are interfaced via associated local L1P, L1D and L2 memory controllers, respectively.

Additionally, the DSP C66x CorePac contains the following internal peripherals:

- an interrupt controller (DSP_INTC) to service DSP C66x CorePac internal and external interrupt events
- a power-down controller (DSP_PDC)
- an external memory controller - DSP_EMC
- an extended memory controller - DSP_XMC_CTRL
- a bandwidth manager - BWM with local controls to the L1P-, L1D- and L2-memories
- an internal direct memory access controller - IDMA

The C66x CorePac DSP also instantiates Debug and Trace logic, part of which is implemented in the DSP core C66x CPU. For more details, refer to the Chapter 34, *On-Chip Debug Support* .

5.3.2.2.1 DSP TMS320C66x CorePac CPU

The DSP C66x CorePac CPU includes following key components :

- A program fetch unit
- 16-/32-bit instruction dispatch unit, advanced instruction packing
- Instruction decode unit
- Two data paths, each with four functional units
- 64 x 32-bit general purpose registers
- Control logic and associated registers
- An internal interrupt and exception controller
- Test, emulation logic
- Internal DMA (IDMA) for transfers between internal memories

For more information on the TMS320C66x central processing unit, see the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

5.3.2.2.2 DSP TMS320C66x CorePac Internal Memory Controllers and Memories

The TMS320C66x DSP CorePac implements a two-level internal cache-based memory architecture.

5.3.2.2.2.1 Level 1 Memories

Level 1 memory (L1) is split into separate program memory (L1P memory) and data memory (L1D memory). Each of the memories can be split into static RAM (normal addressable on-chip memory) and cache.

L1P memory is dedicated to TMS320C66x CPU program words storage and is interfaced via a dedicated L1P memory controller in the DSP C66x CorePac. User can choose to initialize one part of the L1P memory as cache and the other as SRAM. The entire 32 KiB memory is cacheable. The L1P features a dynamically configurable cache size (4 KiB, 8 KiB, 16 KiB and 32 KiB) defined via L1P controller configuration register - L1PCFG[2:0] L1PMODE bitfield. **Note that, the L1P controller maps the cache space by starting at the top of the L1P memory map (i.e. from most significant address) and working downwards. The L1P mapped SRAM size is 32 KiB minus the configured cache size.**

NOTE: The L1P cache / SRAM is ONLY read-accessible by the C66x CPU processor. The DSP C66x CorePac external DMAs (SDMA and EDMA) and internal DMA (IDMA) are the only initiators which can write to the L1P memory. The CPU may however write access and modify certain L1P cache/SRAM controller registers if such access is allowed for the register.

NOTE: In the device integrated DSP, at reset, the entire 32 KiB- L1P memory is initialized as a cache (reset value of L1PMODE=0x7).

For more information on the L1P cache/SRAM memories, refer to the *TMS320C66x DSP Cache User Guide*, (SPRUGY8).

L1D memory is used for level 1 CPU data storage and is interfaced via a dedicated L1D memory controller in the DSP C66x CorePac. User can choose to initialize one part of the L1D memory as cache and the other as SRAM. The entire 32 KiB memory is cacheable. The L1D features a dynamically configurable cache size (4 KiB, 8 KiB, 16 KiB and 32 KiB) defined via L1D configuration register - L1DCFG[2:0] L1DMODE bitfield. **Note that, the L1D controller maps the cache space by starting at the top of the L1D memory map (i.e. from most significant address) and working downwards. The L1D mapped SRAM size is 32 KiB minus the configured cache size.**

NOTE: In the device integrated DSP, at reset, the entire 32 KiB- L1D memory is initialized as a cache (i.e. reset value of L1DMODE = 0x7).

For more information on the L1D cache/SRAM memories, refer to the *TMS320C66x DSP Cache User Guide*, (SPRUGY8).

5.3.2.2.2 Level 2 Memory

The L2 memory can also be split into L2 RAM (normal addressable on-chip memory) and L2-cache for caching external to the DSP meagmodule memory locations. The on-chip integrated L2 memory total size is 288 KiB. The L2 memory is shared between data and program word sources within and outside the DSP C66x CorePac. The L2 memory is divided into two physical 128 bit-wide banks, accesses to which are interleaved on address LSB. Each of the two L2 banks is further split into 4 subbanks.

NOTE: Only 256 KiB of the L2 memory are cacheable in the device DSP. The remaining 32 KiB are always mapped as static RAM.

The L2 memory features a dynamically configurable cache size (32 KiB, 64 KiB, 128 KiB and 256 KiB) defined via L2 configuration register - L2CFG[2:0] L2MODE bitfield. The additional (to the 32KiB fixed SRAM L2) SRAM available is 256-KiB minus the cache size.

L2 memory controller is responsible for MDMA bus error events reporting. The DSP C66x CorePac MDMA bus error event is exported outside the C66x DSP CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"MDMAERREVT"** event in the Table 5-5.

NOTE: The MDMA bus error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

5.3.2.2.3 DSP C66x CorePac Internal Peripherals

The DSP C66x CorePac includes the following internal peripherals:

- DSP interrupt controller (DSP_INTC)
- DSP power-down controller (DSP_PDC)
- Bandwidth manager (DSP_BWM)
- Memory Protection Hardware
- Internal DMA (DSP_IDMA) controller

- External Memory Controller (DSP_EMC)
- Extended Memory Controller (DSP_XMC_CTRL) including prefetch buffer logic
- Error Detection logic for the L1P memory
- Error Detection and Correction (ECC) logic for the L2 memory

This section briefly describes the DSP_INTC, DSP_PDC, DSP_BWM, DSP_IDMA, DSP_EMC, DSP_XMC_CTRL controller, L1P Error detection and L2 ECC logic. For more information on the TMS320C66x DSP CorePac, see the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.1 DSP C66x CorePac Interrupt Controller (DSP INTC)

The DSP C66x CorePac includes an interrupt controller (DSP_INTC) and can receive a total of 128 system events as inputs. They include DSP-generated events and chip-level events.

In addition to these 128 events, a non-maskable (NMI) event (see the Section 5.3.4.1.1) and reset events are mapped to the DSP_INTC as well, and are routed straight through to the DSP CPU core.

For more details on the DSP_INTC functionalities and its corresponding control / status registers (part of the DSP_ICFG local configuration space), refer to the section *Interrupt Controller* within the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

For more details on input interrupt mappings and associated IRQ wake-up events, refer to the Section 5.3.4.1.

For more information about the device DSP_INTC, see Chapter 17, *Interrupt Controllers*. For more information on chip level IRQ mapping via the device IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.

5.3.2.2.3.2 DSP C66x CorePac Power-Down Controller (DSP PDC)

The DSP C66x CorePac includes a power-down controller (PDC). The PDC can power-down all of the following components of the DSP C66x CorePac and internal memories of the DSP subsystem:

- C66x CPU
- L1P Memory
- L2 Memory
- Cache controllers
- Entire TMS320C66x DSP CorePac

Refer to the *Power-Down Controller* in the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

5.3.2.2.3.3 DSP C66x CorePac Bandwidth Manager (BWM)

The DSP C66x CorePac implements a bandwidth manager (BWM) to assure that some requestors do NOT block resources in the C66x CorePac DSP for extended periods of time.

Refer to the *Bandiwdth Management Architecture* in the*TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

5.3.2.2.3.4 DSP C66x CorePac Memory Protection Hardware

The C66x Core Pac memory protection architecture introduces in the DSP a combination of DSP privilege levels and a memory system permission structure. This provides several benefits to the system, as follows:

The DSP C66x CorePac MP events are exported outside the DSP C66x CorePac in DSP subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **memory protection fault** events listed in the Table 5-5.

NOTE: The memory protection exception events are not exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and associated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

NOTE: IDMA, DMA or System initiators should not issue read/write requests to regions of DSP L1P, L1D, or L2 memory configured as cache. In such cases the corresponding MPPA register should be set to 0x0 to disallow external read/write accesses.

Refer to the section *Memory Protection* in the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

5.3.2.2.3.5 DSP C66x CorePac Internal DMA (IDMA) Controller

The IDMA controller performs fast block transfers between any two memory locations local to the DSP C66x CorePac. Local memory locations are defined as those in Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories, or in the external peripheral configuration (CFG) port.

The IDMA configuration / status registers themselves are part of the DSP_ICFG and are visible only to C66x CPU.

NOTE: The IDMA cannot transfer data to or from the internal DSP memory-mapped register space (DSP_ICFG).

The IDMA exception is mapped to the system level ERRINT_IRQ interrupt event. For more details, refer to the Section 5.3.4.2.2 and the Table 5-5.

The DSP C66x CorePac IDMA error event is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"EMC_IDMAERR"** event in the Table 5-5.

NOTE: The IDMA exception error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

The IDMA is fully described in the section *Internal Direct Memory Access (IDMA) Controller* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.6 DSP C66x CorePac External Memory Controller

The DSP C66x CorePac has an embedded External Memory Controller which acts as a bridge between the DSP C66x CorePac CPU and the remaining part of DSP subsystem. It implements two ports interfacing the DSP C66x CorePac environment:

- An external peripheral 32-bit CFG port which acts as an (DSP CPU cfg) initiator on the DSP_NoC L2 interconnect. It is the root source of all C66x CPU external configuration traffic **(excluding the DSP_ICFG traffic which takes place only within the DSP C66x CorePac)** towards the subsystem.
- An SDMA slave port which is a target on the L2 DSP_NoC interconnect. It generally accepts traffic initiated on DSP_NoC by the:
  - DSP_EDMA_TC0 and DSP_EDMA_TC1
  - DSP external slave port on the L3_MAIN

In summary:

- The CPU CFG port provides access to the memory-mapped registers which control various peripherals

and resources within the DSP subsystem, such as the MMUs, DSP_EDMA controllers, DSP system control and wakeup logic, DSP NoC itself, DSP external peripherals, etc.

- The DSP system masters found outside the DSP C66x CorePac such as the DSP_EDMA controllers (TC0 and TC1) or L3_MAIN masters (device MPU, IPU1, IPU2, etc.) access the SDMA slave port to reach resources inside the DSP C66x CorePac. In respect to the SDMA port, the DSP C66x CorePac is the slave in the transaction.

The DSP_EMC controller adds following functionalities to the DSP C66x CorePac:

- Reporting errors related to the C66x CPU external peripheral configuration bus (associated registers)

NOTE: Regarding PrivID versus AID mapping functionality of the EMC, the C66x DSP CorePac is able to distinguish ONLY "local" vs "external" requests. The device integrated C66x DSP CorePac has the SDMA PrivID input tied-off to a value of 0x0. On DSP C66x CorePac SDMA port, this means that no distinguishing can be made between external requests which come over DSP_NoC interconnect from local DSP_EDMA and those coming from other initiators accesing DSP via the device L3_MAIN interconnect.

This limits the funcitionality of the internal memory protection registers.

The DSP C66x CorePac EMC error event is exported outside the DSP C66x CorePac in the subsystem, and is capable to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **EMC_BUSERR** event in the Table 5-5.

NOTE: The EMC configuration bus error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

For various DSP_NoC initiator vs target mappings, refer to the Table 5-8.

Note that, there are DSP_NoC pressure (Mflag bus) controls in DSP_SYSTEM logic related to the C66x CPU CFG and DSP_NoC SDMA init traffic. They are described in the Section 5.3.8.3.

The EMC functionalities / registers are fully described in the section *External Memory Controller (EMC)* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.7 DSP C66x CorePac Extended Memory Controller

The DSP C66x CorePac located extended memory controller (DSP_XMC_CTRL) implements a local DMA master port (MDMA) which provides the primary path for C66x CPU and cache requests to the device level memories such as (DDR or L3 SRAM) and peripheral / memory mapped register space. Via some additional logic, including DSP_SYSTEM controls and a local DSP MMU - DSP_MMU0 on the path (with option to bypass), C66x local MDMA port is mapped to the DSP subsystem CPU master port (i.e. MDMA master port of the DSP CPU on L3_MAIN). The DSP C66x Corepac MDMA port is mapped to the DSP Subsystem CPU Master Port (with DSP_MMU0 involved or not involved) to allow fast accesses (DSP_NoC not involved) to the external SDRAM (via the L3_MAIN and DMM) or to L3 SRAM (via the L3_MAIN).

The memory protection settings in MPAX defines types of the memory accesses permitted on various address ranges within DSP C66x CorePac 32-bit address map.

The DSP_XMC_CTRL also instantiates program and data prefetch buffer logic to reduce time during servicing read requests from the L1D, L1P and L2 memory controllers. The aim is to buffer program and data fetches from external L3_MAIN memory locations. While the program prefetch buffer is organized as 4 entry x 32 byte, the data prefetch buffer is organized in 8 slots, with 128 bytes per slot. The DSP_XMC_CTRL prefetch reduces the penalty associated with accesses to the L3_MAIN SDRAM upon L1P, L1D and L2-cache read-misses.

NOTE: the DSP_XMC_CTRL registers are part of the DSP_ICFG space, hence they are not accessible outside the DSP C66x CorePac (visible only to the C66x CPU).

In summary the DSP_XMC_CTRL provides :

- a master path from the DSP C66x CorePac level 2 cache / SRAM memory to device memory such as SDRAM or L3 SRAM or peripheral / MMR address space.
- memory protection on external address ranges related to L3_MAIN RAMs (via MPAX unit) :
  - 16 user-defined address ranges (MPAX segments) can be used to divide external memory space
- Address translation
- Data and Program prefetch buffer logic
- 12- address candidate buffer that acts as a "stream detection filter"

NOTE: The device DSP subsystem does NOT use the DSP C66x CorePac multicore shared memory controller (MSMC) port to add more static RAM within subsystem boundaries, i.e. no additional SRAM is available in the DSP except for the L1P, L1D and L2 memories. Only the DSP_XMC_CTRL controller MDMA port on the L3_MAIN interconnect is used to extend DSP available RAM memory via a direct or DSP_MMU0 translated access to the device EMIF DDR memories and OCMC RAMs.

NOTE: **Only the memory protection function of the DSP_XMC_CTRL MPAX unit is intergated** and used in the device DSP subsystem. The MDMA port 32-bit to 36-bit address extension function is NOT used in the device DSP because L3_MAIN address bus width is 32-bit. The DSP_MMU0 does NOT perform an address size (32b -> 36b) extension as well.

The XMC functionalities / registers are fully described in the section *Extended Memory Controller (XMC)* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.7.1 XMC MDMA Accesses at DSP System Level

5.3.2.2.3.7.1.1 DSP System MPAX Logic

The default configuration of MPAX registers provides a 32-bit view of system memory on L3_MAIN.

In summary, each MPAX segment (mentioned above) is programmed with a starting virtual base address, segment sizes from 4 GiB down to 4 KiB, replacement address (i.e., physical address); and permission attributes. Provided that DSP_MMU0 can be used to perform address translation, in most cases the replacement address will equal the base address (i.e., virtual == physical from DSP C66x CorePac perespective).

The system level implementation of MPAX logic allows the C66x CPU to change permission without being required to flush the cache.

The C66x CPU subsystem relies on the MPAXn.PERM field to properly configure the permissions for remote address ranges. The MDMA.rperm[6:0] signals are tie-off to a fixed value of 0x7F on the DSP C66x CorePac boundary.

5.3.2.2.3.7.1.2 MDMA Non-Post Override Control

The C66x corepac submits writes denoted as either “cacheable” or non-cacheable. Write accesses that are non-cacheable will be submitted as interconnect (L3_MAIN) non-posted writes; whereas write accesses that are cacheable are submitted as interconnect posted writes. An exception for the cache writes to L3_MAIN is that in the case of a cache block write-back operation (when actual cache evict busrts are actually issued towards L3_MAIN connected memory), a non-posted write is submitted.

NOTE: In order to provide a safety net for interconnects that may do aggressive reordering, a memory-mapped register SW control is provided - DSP_SYS_BUS_CONFIG[24] NOPOSTOVERRIDE. When set, this results in all write commands being issued as non-posted. This bit defaults to set, and thus the default behavior is for non-posted writes to be used exclusively.

5.3.2.2.3.8 L1P Memory Error Detection Logic

The L1P memory detection logic (no correction is implemented) uses a 4-bit parity per 256-bit location (1-bit parity per 64-bit line quadrant).

The L1P error detection logic features:

- L1P error detection command, status and address controls (registers)
- L1P_ED error detection exception / interrupt to the DSP_INTC upon DMA / IDMA access
- a direct exception event to C66x CPU (DSP_INTC not involved) upon parity error during a program fetch from L1P-cache
- L1P-cache error recovery

The L1P parity error detection event is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"PMC_ED"** event in the Table 5-5.

NOTE: The L1P error detection event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

For more details on L1P error detection logic, refer to the section *L1P Error Detection*, of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.9 L2 Memory Error Detection and Correction Logic

The L2 Memory error detection and correction logic (ECC) implements a distance-3 “detect 2, correct 1” Hamming code based error correction / detection algorithm. A 12-bit hamming code per 256-bit is used.

The L2 error detection and correction logic features:

- L2 error detection command, status and address controls (registers)
- L2 EDC enable
- L2 error detection event counter
- 2x L2 EDC exception / interrupts mapped to the DSP_INTC :
  - L2_ED1 = "error corrected" event
  - L2_ED2 = "error-not-corrected" event

The two L2 memory error correction events are exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"UMC_ED1" and "UMC_ED2"** events in the Table 5-5.

NOTE: The L2 error detection events are not exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

For more details on L2 error detection and correction logic, refer to the section *L2 Error Detection and Correction* of the*TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.3 DSP Debug and Trace Support

The DSP subsystem offers full support for the native DSP C66x CorePac debug features. This includes Advanced Event Triggering (AET) and Trace.

5.3.2.3.1 DSP Advanced Event Triggering (AET)

AET capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- Hardware Program Breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data Watchpoints: specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- Counters: count the occurrence of an event or cycles for performance monitoring.
- State Sequencing: allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

5.3.2.3.2 DSP Trace Support

Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system. Trace is supported via Code Composer Studio.

See also the Chapter 34, *On-Chip Debug Support* .

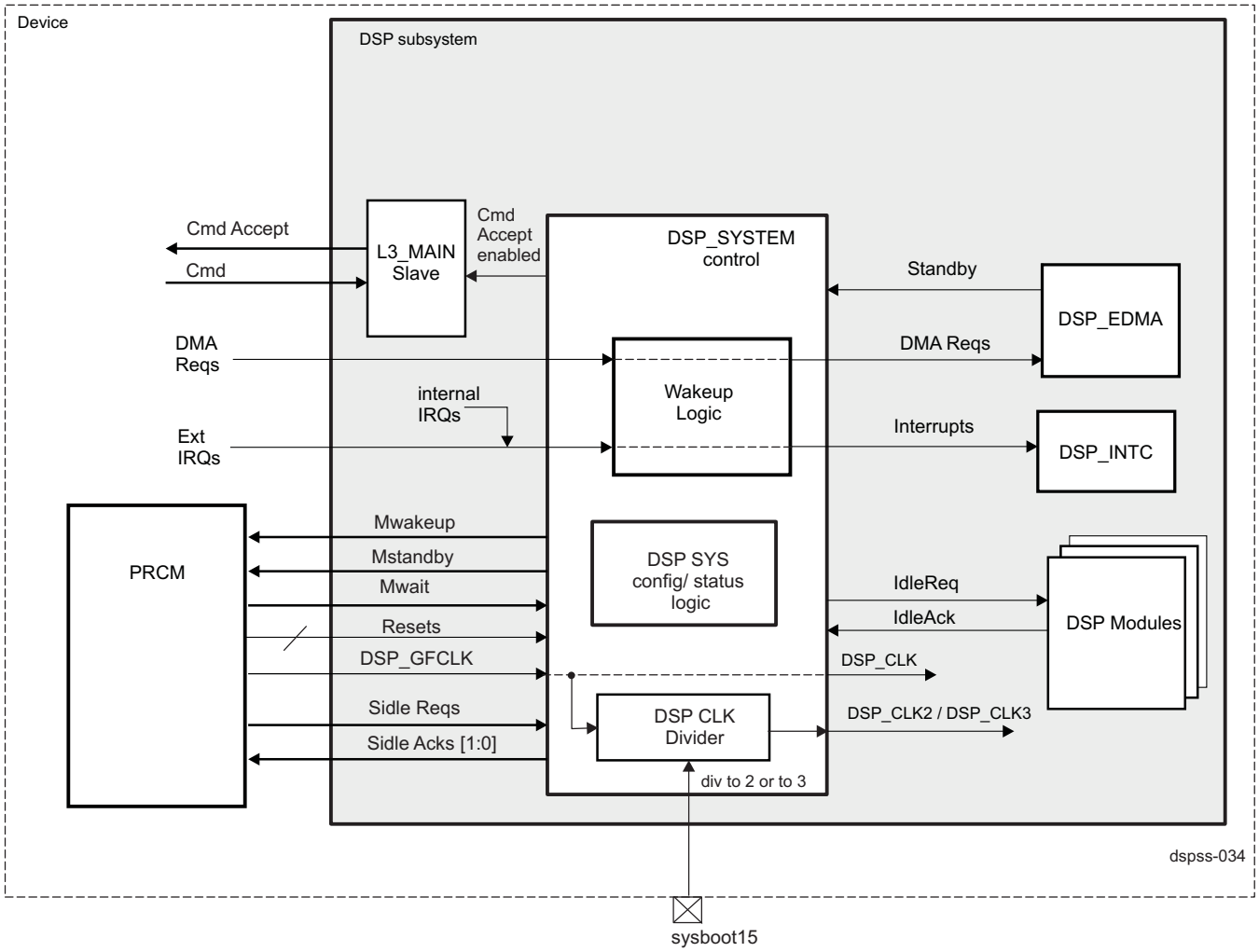
5.3.3 DSP System Control Logic

The DSP_SYSTEM module controls the following functions:

- Generation of the divided clocks (DSP_CLK2 or DSP_CLK3) to all components of the DSP subsystem
- Synchronization of the DSP divided clocks
- PRCM module power handshaking
- Reset input resynchronization of the active-to-inactive transition to the CD1_CLK clock
- DSP subsystem top level configuraiton registers and its access from the DSP core.

Figure 5-3 highlights the DSP_SYSTEM and its connectivities to the surrounding blocks within the subsystem and in the device.

Figure 5-3. DSP_SYSTEM Block Diagram



5.3.3.1 DSP System Clocks

The DSP1 and DSP2 subsystems inputs a primary non-divided clock (DSP1_FICLK / DSP2_FICLK) and based on it (DSP_CLK1), internally generates either **a divided by 2 clock** (DSP_CLK2) version or **a divided by 3 clock** (DSP_CLK3). The divided clock determines the operation rate of the DSP subsystem logic and bus interfaces. The division is defined upon device boot time through signal level externally applied on the device **sysboot15** input. The actual bit configuration is latched upon power-on reset in Control Module register CTRL_CORE_BOOTSTRAP[15] SYS_BOOT_15_CLOCK_DIVIDER boot status bit. For more details, refer to the Section 18.4.6.14.1, *System Boot Status Settings* of the chapter, *Control Module*.

NOTE: Upon boot time, sysboot15 set at '0' selects a DSP_CLK2 and a sysboot15 set at '1' selects a DSP_CLK3 divided clock version for the DSP subsystem logic and bus interfaces.

NOTE: Only DSP_CLK3 clock is supported on this SoC. For SR1.1, sysboot15 must be tied to vdd, but for SR2.0 it is configurable. For more information, see Section 18.4.6.1.1.1, *Permanent PU/PD disabling (SR 2.0 only)* in Chapter 18, *Control Module*.

The clock operating mode setting (DSP_CLK2 or DSP_CLK3) must be static just before and continually after reset deassertion. This signal will also drive the configuration to the DSP C66x CorePac for the XMC_MDMA_CLK, EMC_SDMA_CLK, and EMC_CFG_CLK configurations.

The DSPSS1 / DSPSS2 subsystem input clock frequency (DSP_CLK1) corresponds to the PRCM DSP1_GFCLK / DSP2_GFCLK frequency that is configured in the device PRCM registers.

NOTE: For valid DSP_CLK1 (and hence for DSP_CLK3 = DSP_CLK1 / 3) frequency range, see the Operating Performance Points section of the device Data Manual.

The Section 5.3.2 also shows the distribution of the different DSP subsystems blocks within the two DSP local clock domains CD0_CLK (running on DSP_CLK frequency) and CD1_CLK (running on DSP_CLK2 or DSP_CLK3 frequency).

5.3.3.2 DSP Hardware Resets

The DSP uses the same reset sources than those mapped to the DSP C66x CorePac; i.e. DSP C66x CorePac reset inputs will be pinned out as DSP system reset inputs.

The Table 5-4 summarizes the DSP hardware reset inputs and their functional descriptions.

Table 5-4. Summary of the DSP1 and DSP2 Hardware Resets

DSP1 reset input	DSP1 reset "done" output to PRCM	Description
DSP1_PWR_RST	-	This is power-on reset signal used inside DSP1 to reset mainly the emulation logic. It resets the entire DSP1 logic.
DSP1_RST	-	Reset signal used to reset all logic inside DSP1 except Emulation logic.
DSP1_LRST	-	Reset applied ONLY to the C66x CPU inside DSP1
-	DSP1_LRST_DONE	Indicates completion of the DSP1 local C66x CPU reset to device PRCM
DSP2 reset input	**DSP2 reset "done" output to PRCM**	**Description**
DSP2_PWR_RST	-	This is power-on reset signal used inside DSP2 to reset mainly the emulation logic. It resets the entire DSP2 logic.
DSP2_RST	-	Reset signal used to reset all logic inside DSP2 except Emulation logic.
DSP2_LRST	-	Reset applied ONLY to the C66x CPU inside DSP2
-	DSP2_LRST_DONE	Indicates completion of the DSP2 local C66x CPU reset to device PRCM

See also the Section 5.2 for more information on the PRCM reset sources to DSP reset inputs connectivity.

Refer to the Section 3.5.6.6, *DSP1 Subsystem Power-on Reset Sequence* and the Section 3.5.6.8, *DSP2 Subsystem Power-on Reset Sequence* in the chapter, *Power, Reset and Clock Manamgement* for more details on the DSP1 and DSP2 power-on reset sequence, respectively.

NOTE: In the case of DSP1 / DSP2 recovery from the “Powerdown-grid off”, a full power-on-reset sequence is required before re-booting and resuming functional operation.

The DSP host (device MPU) software must ensure that the PRCM functional clock DSP1_GFCLK / DSP2_GFCLK is enabled to the DSP1 / DSP2, respectively, prior to starting the DSP1 / DSP2 power-on reset sequence.

5.3.3.3 DSP Software Resets

During a software reset on the DSP, all resets described in Table 5-4 are asserted, except for the power-on DSP_PWRON_RST signal which remains de-asserted in this case.

The DSP subsystem does NOT implement any local software reset controls. The software reset assertion and DSP_LRST completion monitoring is done in PRCM located registers (part of the DSP1_PRM / DSP2_PRM address space).

Refer to the Section 3.5.6.7, *DSP1 Subsystem Software Warm Reset Sequence* and the Section 3.5.6.9, *DSP2 Subsystem Software Warm Reset Sequence* in the chapter, *Power, Reset and Clock Manamgement* for more details on the DSP1 and DSP2 software reset sequence and related software controls, respectively.

5.3.3.4 DSP Power Management

The supported power-down modes are:

- Slave idle and master standby protocols for powerdown
- "Disconnect from interconnect" handshake for init and target busses
- Clock Stop mode - wakeup on interrupt or DMA event
- Grid OFF mode : No power supply is switched-on

NOTE: Powerdown-retention mode is NOT supported by DSP subsystem. **The DSP recovery from the Powerdown-grid OFF mode requires full boot.**

The DSP C66x CorePac natively supports “CLKSTOP/Static Powerdown” and “POWERDOWN (Grid off)” modes of operation. Once the device PRCM restores clocks and power supply, then the DSP C66x CorePac can exit static-powerdown.

The DSP_SYSTEM wakeup logic is implemented in the "always-on" clock / power supply domain. This logic monitors new interrupts / events and will drive the IDLE wakeup request when a new interrupt / event occurs.

5.3.3.4.1 DSP System Powerdown Protocols

For each of the powerdown modes – Static or Powerdown-Grid Off, the device PRCM will control whether clocks are gated, or whether supplies are reduced or removed.

NOTE: In the case of “Powerdown-grid off”, a full power-on-reset sequence is required before re-booting and resuming functional operation.

in the static powerdown modes - the DSP recognizes new level interrupts while in a clock-gated state (and drive wakeup request). During this powerdown mode all internal state will be retained, including DSP C66x CorePac, interconnect, EDMA, memories, etc.

The following protocols are implemented with PRCM:

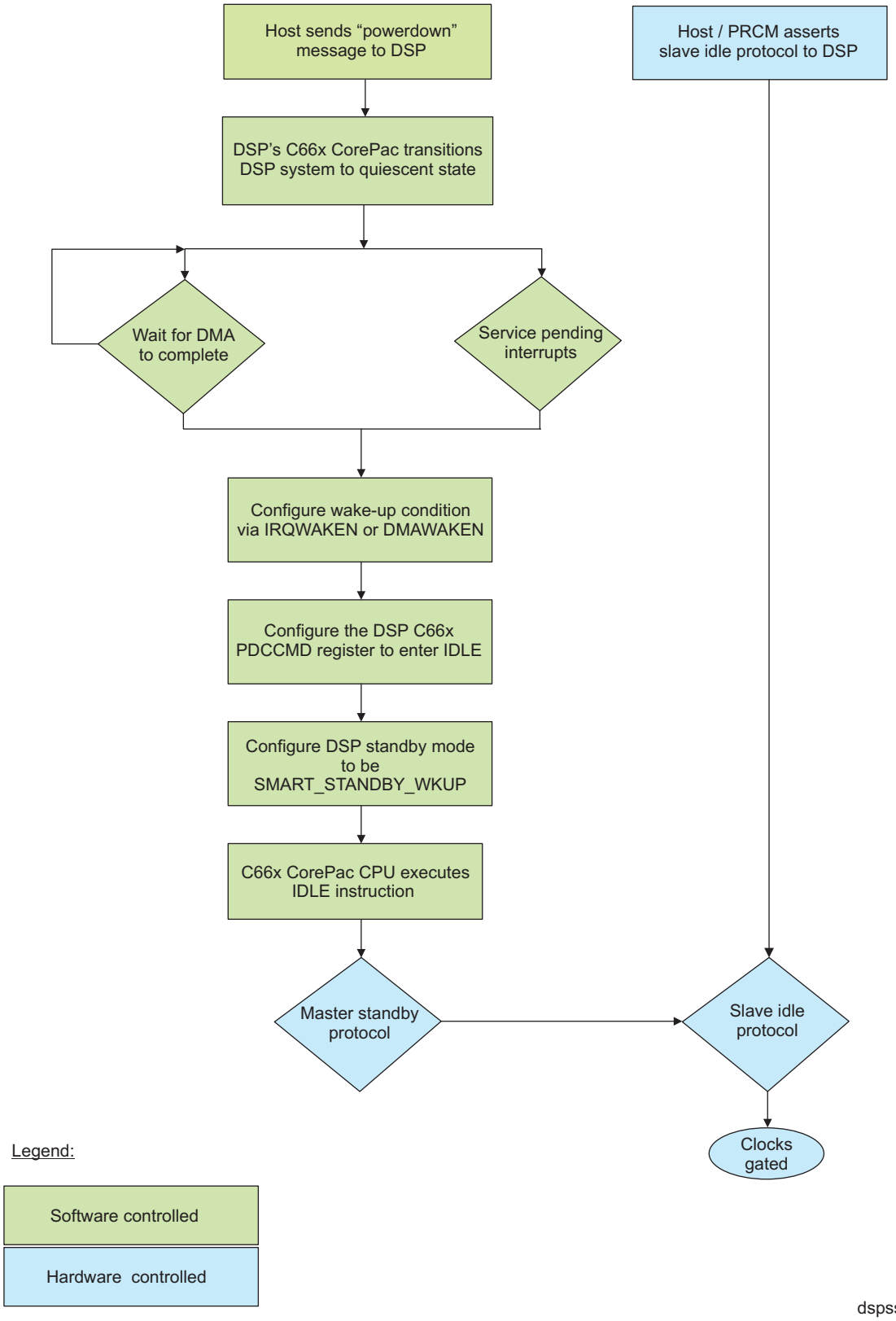
- Slave idle protocol with device PRCM for powerdown (wake-up capable)
- Master standby protocol with device PRCM for powerdown
- Interconnect disconnect for master and slave ports

The Master standby and slave idle protocols behaviour is controlled in the DSP_SYS_SYSCONFIG register.

5.3.3.4.2 DSP Software and Hardware Power Down Sequence Overview

Figure 5-4 highlights the high level flow-chart for entry into any of the DSP powerdown modes. The system host (typically) first informs the DSP that it should enter a powerdown mode. The host (g.h. device MPU) sends a software message (normally via system level mailbox+interrupt). In parallel, the PRCM (via host or DSP programmation) will hardware assert an SIdleReq request to the DSP via the IDLE Protocol connection. At the next stage, the C66x CPU, in general, performs any software bookkeeping necessary to transition the DSP subsystem to a quiescent state. This may include : waiting for outstanding DMA transfers to complete, waiting for outstanding DMA transfers to complete, etc. The C66x processor should finally execute the IDLE instruction when it is ready to be powered-down. Assuming the DSP_SYS_SYSCONFIG[5:4] STANDBYMODE is enabled, then the hardware will transition to an idle state and notify to the system the intention to enter powerdown state to the system via the master standby and slave idle protocols. After IDLE and MSTANDBY handshake is completed, the DSP clocks are optionally gated; and supply rails are optionally reduced or turned off.

Figure 5-4. Extended Duration Sleep Software and Hardware Sequence



NOTE: The PM_DSPx_PWRSTCTRL[1:0] POWERSTATE bit field in device PRCM must be set to 0x3 (ON state) prior to performing the sequence shown in Figure 5-4 for the transition to be successful.

5.3.3.4.3 DSP IDLE Wakeup

In order to facilitate auto-wakeup of DSP C66x, the IDLE protocol's wakeup capability is used. Wakeup operation is enabled if DSP_SYS_SYSCONFIG[3:2] IDLEMODE is set to 0x3.

In this mode, while in IDLE state, if an external input interrupt source is asserted (if enabled via the DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 mask) or if an external DMA event source is asserted (if enabled via the DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 mask) or if the DSP subsystem NMI input is asserted **(note that there is no wake enable mask for the non-maskable interrupt)** then the Mwakeup signal is asserted to the PRCM which is expected to observe the Mwakeup. Upon such assertion the PRCM enables the clocks, exiting the "Standby" and "Idle" states. at this point the C66x CPU is able to branch to the pending interrupt service routine. The MWakeup is deasserted when all IRQ or DMA requests enabled in the DSP_SYS_IRQWAKEEN0/1 and DSP_SYS_DMAWAKEEN0/1 are deasserted.

The Wakeup logic controlling assertion of the MWakeup request is completely asynchronous because in IDLE mode the clock may not be present. It relies on level sensitive interrupts.

NOTE: The DSP_EDMA must be manually removed from IDLE / Standby state. During that time, it is possible that the EDMA input event is no longer pending and may not have been recognized/latched as an EVENT to the EDMA. In that case, the user SW can enable the DSP EDMA_WAKE_INT (in associated DSP_SYS_EDMAWAKE0_IRQENABLE_SET register) to recognize in the (DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW /DSP_SYS_EDMAWAKE0_IRQSTATUS) which specific EDMA event was asserted and caused the wakeup condition. The DSP software can then trigger the corresponding DSP_EDMA channel manually (by setting the ESR) or by servicing the interrupt/event manually via reads and writes. For more details, refer to the Section 5.3.5.1.

5.3.3.4.4 DSP SYSTEM IRQWAKEEN registers

The DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 masking bits must be appropriately set for any valid interrupt mapped from device IRQ_CROSSBAR to the DSP subsystem boundary, to enable its path (passing through the DSP_SYSTEM wakeup logic) to the DSP local interrupt controller - DSP_INTC.

CAUTION

In order for a given interrupt to be serviced by the DSP (even when the Idle Instruction is NOT being executed), the Interrupt must be enabled in the coresponding DSP_SYS_IRQWAKEEN0 or DSP_SYS_IRQWAKEEN1 register.

5.3.3.4.5 DSP Automatic Power Transition

This section provides register details for configuring the DSP1 subsystem in automatic power transition mode. The same should be considered for DSP2 in corresponding PRCM and DSP2 related registers.

The DSP1 module is supposed to be configured to automatic management in PRCM.DSP1_CM_CORE_AON via setting the register CM_DSP1_DSP1_CLKCTRL[1:0] MODULEMODE bitfield to 0x1. The DSP1 clock domain is supposed to be configured in automatic "HW_AUTO" transition (setting bitfield CM_DSP1_CLKSTCTRL[1:0] CLKTRCTRL=0x3).
The power state (controls are in the PRCM.DSP1_PRM instance) to reach upon a sleep transition is configured in the PM_DSP1_PWRSTCTRL[1:0]POWERSTATE bitfield.

5.3.4 DSP Interrupt Requests

The DSP subsystem relies on the DSP C66x CorePac local interrupt controller - DSP_INTC for mapping the various input interrupts to the C66x CPU, that are :

- generated outside the DSP, by the device intergated modules and subsystems
- generated within the DSP subsystems but outside the DSP C66x CorePac
- generated by different components within the DSP C66x CorePac

In addition, a non-maskable input interrupt, direct mapped on a C66x processor NMI input is implemented. It is mapped via a register that resides within the device Control Module. Both the maskable interrupts and the non-maskable interrupts are synchronized internally.

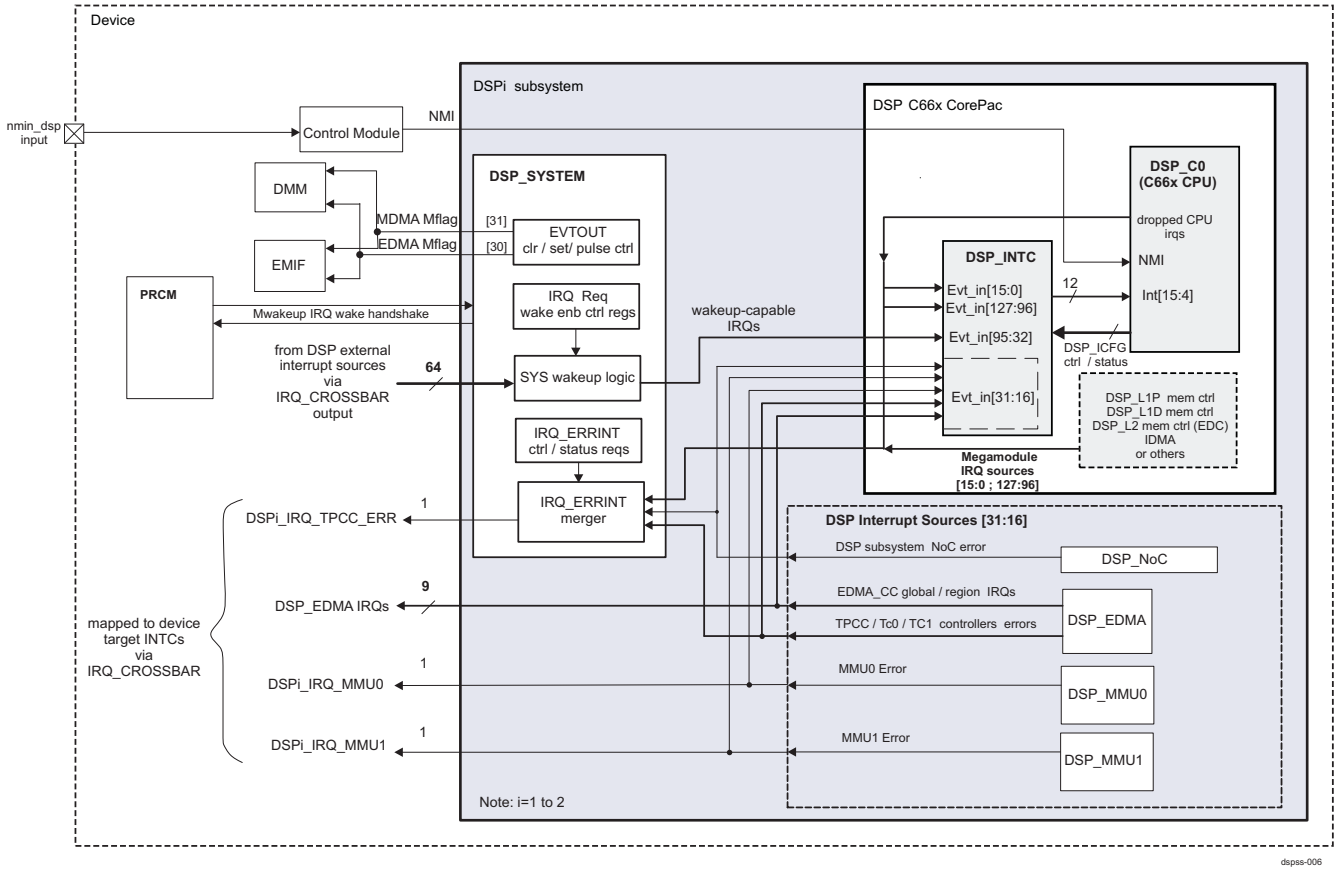
Part of the DSP subsystem module generated interrupts which are output as follows:

- DSP_EDMA interrupts
- DSP_MMU0 and DSP_MMU1 interrupts
- Error interrupts

Figure 5-5 shows how are the interrupt sources organized. To manage and expand the interrupt capabilities of the DSP C66x CorePac (internal and external interrupt requests), the DSP subsystem includes two levels of interrupt control :

- The DSP C66x CorePac local Interrupt controller - DSP_INTC
- The System control logic - DSP_SYSTEM

Figure 5-5. DSP Subsystem Interrupt Management



5.3.4.1 DSP Input Interrupts

In summary, the DSP_INTC accepts up to 124 event inputs, and flexibly maps those down to 12 interrupt inputs to the DSP. The mapping can be 1:1 (input:output), or can use the event combiner to map multiple interrupts (within a 32-bit group) to one of the DSP interrupt inputs. In general, many of the 124 interrupt controller inputs are collected within the DSP C66x CorePac, and are NOT available at the DSP C66x CorePac boundaries.

The C66x CPU dropped event is exported outside the C66x CorePac in DSP subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"INTERR" event** listed in the Table 5-5.

NOTE: The dropped CPU event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

Part of the input interrupts, generated by DSP peripherals that are located outside the DSP C66x CorePac - DSP_EDMA (DSP_EDMA_CC, DSP_EDMA_TC0, DSP_EDMA_TC1) , DSP_MMU0, DSP_MMU1 and DSP_NoC, are also mapped to outputs at the DSP subsystem boundary, such that they can be exported to system hosts (MPU, etc.) via the device IRQ_CROSSBAR.

Of particular interest, MMUs (DSP_MMU0 and DSP_MMU1) interrupts will typically be serviced by the device MPU instead of by the local DSP core.

Any interrupt input at DSP subsystem boundaries (i.e. excluding the DSP subsystem internal IRQ sources that reside in and outside the DSP C66x CorePac) can be used to wake-up the DSP subsystem from an IDLE state. This is described in the Section 5.3.3.4.3 and is controlled by the DSP_SYSTEM logic register DSP_SYS_SYSCONFIG [3:2] IDLEMODE bitfield along with the DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 registers.

CAUTION

The DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 bits MUST be enabled for externally mapped interrupts (DSP_INTC[95:32]) to be serviced by the DSP regardless of the DSP power state (IDLE or non-IDLE).

The DSP C66x CorePac DSP_INTC registers are NOT readable by any entity other than the C66x CPU, because they are part of the DSP_ICFG C66x CorePac internal configuration space (see also the Section 5.3.10). Hereby, only the C66x itself is able to service these interrupt events. The only way for these interrupts to be cleared is for the DSP CPU to clear the state in the EVTFLAGi (where i=0 to 3) register, or via reset assertion.

NOTE: For cases where the DSP maps an interrupt directly, the DSP is not strictly required to clear the EVTFLAGi register. User software must take the extra step of clearing the EVTFLAGi to cause the corresponding output interrupt to be cleared and re-asserted upon a new input event assertion.

5.3.4.1.1 DSP Non-maskable Interrupt Input

The device DSP also supports a non-maskable interrupt (NMI) directly mapped to the NMI input of the C66x CPU. This line is also mapped to the NMEVT input of the DSP local INTC, and can be used as an exception signal, too. At system level, the NMI interrupt mapping to the DSP_INTC is controlled via the **device core Control Module** register as follows:

- CTRL_CORE_NMI_DESTINATION_2 [15:8] DSP1 = 0x1 enables the DSP1 to receive the NMI coming from the device **nmin_dsp** input.
- CTRL_CORE_NMI_DESTINATION_2 [23:26] DSP2 = 0x1 enables the DSP2 to receive the NMI

coming from the device **nmin_dsp** input.

For more details on the NMI receive enable bit mapping, refer to the Section 18.5, *Control Module Register Manual* in the chapter, *Control Module*.

5.3.4.2 DSP Event and Interrupt Generation Outputs

5.3.4.2.1 DSP MDMA and DSP EDMA Mflag Event Outputs

The Mflag events generated by DSP subsystem EVTOUT bus are represented in the Figure 5-5.

A couple of the DSP EVTOUT bus outputs - EVTOUT[31] and EVTOUT[30] are used for generation of MFLAGs dedicated to the DSP MDMA and EDMA ports, respectively. DSP MFLAGs are connected directly to DMM and EMIF. The DSP MFLAGs participate in the DMM Emergency and EMIF MFLAG prioritization schemes. At the L3 Level Bandwidth regulators connected to the DSP MDMA and EDMA ports can be used to control DSP traffic versus other device traffic.

The device DSP subsystem is able to generate the 2 output Mflag events via the following DSP_SYSTEM module located registers :

- DSP_SYS_EVTOUT_SET[31:30]
- DSP_SYS_EVTOUT_CLR[31:30]

The current state of the outputs can be detected by reading any of these "pseudo" register bits.

NOTE: Only EVTOUT[31:30] outputs are implemented in the device, hence only bits [31:30] of the mentioned DSP_SYS_EVTOUT_x registers are used.

The DSP_SYS_EVTOUT_SET register unconditionally drives the corresponding output event to '1'. The DSP_SYS_EVTOUT_CLR register unconditionally drives the corresponding output event to a '0'.

5.3.4.2.2 DSP Aggregated Error Interrupt Output

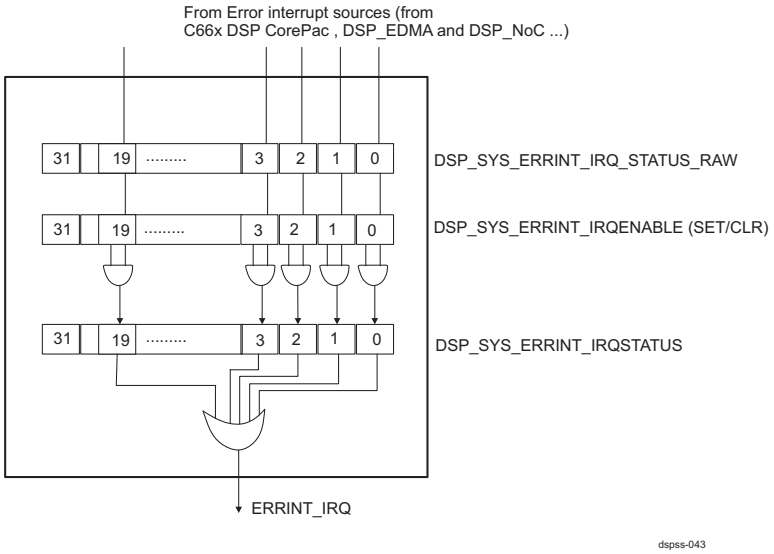
The aggregated error interrupt of the DSP subsystem is shown in the Figure 5-5.

The subset of those events that correspond to: **DSP C66x CorePac generated error events**, **DSP_EDMA error interrupts** and **L2 DSP_NoC interconnect error interrupt**, is reduced by an OR-schematic to a single ERRINT_IRQ output interrupt which is made available on DSP subsystem boundary. It is expected that one of the DSP system hosts monitors the interrupts/error conditions in safety conscious systems.

Figure 5-6 shows a functional representation of the DSP error interrupt "OR"-reduction logic. In summary, there exists **an unmasked status** (DSP_SYS_ERRINT_IRQSTATUS_RAW) register, two complementary enable bit-vector registers (DSP_SYS_ERRINT_IRQENABLE_SET / DSP_SYS_ERRINT_IRQENABLE_CLR), and a masked status register (DSP_SYS_ERRINT_IRQSTATUS). The ERRINT event is asserted when any enabled error interrupt input is asserted.

NOTE: The ERRINT_IRQ output can be programmatically mapped as the DSPi_IRQ_TPCC_ERR (where i=1 to 2) interrupt to all device (dsp hosts) interrupt controllers via the device IRQ_CROSSBAR. For more information on the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

Figure 5-6. ERRINT Diagram



The Table 5-5 details the mapping of error event output sources to the bit positions within the following DSP error event related registers :

- DSP_SYS_ERRINT_IRQSTATUS_RAW
- DSP_SYS_ERRINT_IRQSTATUS
- DSP_SYS_ERRINT_IRQENABLE_SET
- DSP_SYS_ERRINT_IRQENABLE_CLR

Following functional descriptions are valid for the above registers :

- **IRQ status raw register -** This register provides a per-event raw interrupt status vector. The Raw status is set even if the corresponding event is not enabled. Software can write '1' to set the (raw) status for debug purposes.
- **IRQ status register -** This register provides a per-event enabled interrupt status vector. The Enabled status is set if the corresponding event is enabled and the raw status is set. Software can write 1 to clear the (raw) status after the interrupt has been serviced. The clear takes effect even if the interrupt is not enabled.
- **IRQ enable register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to set (i.e., enable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Clear" - register)
- **IRQ clear register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to clear (i.e., disable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Set" register)

NOTE: A DSP_SYS_ERRINT_IRQSTATUS_RAW bit is set even if the corresponding event is NOT enabled in the DSP_SYS_ERRINT_IRQENABLE_SET.

Table 5-5. DSP ERRINT Interrupt Mapping

Interrupt Number	Name	Description
0	tpcc_errint_level	DSP EDMA CC error interrupt
1	tptc_errint0_level	DSP EDMA TC0 error interrupt
2	tptc_errint1_level	DSP EDMA TC1 error interrupt
3	noc_errint_level	DSP L2 Interconnect (DSP_NoC) error interrupt

Table 5-5. DSP ERRINT Interrupt Mapping (continued)

Interrupt Number	Name	Description
4	INTERR	DSP C66x CorePac Dropped CPU Interrupt event
5	EMC_IDMAERR	DSP C66x CorePac Invalid IDMA Parameters
6	MDMAERREVT	DSP C66x CorePac VbusM Error Event
7	PMC_ED	DSP C66x CorePac Single bit error detected during DMA read
8	UMC_ED1	DSP C66x CorePac Corrected bit error detected
9	UMC_ED2	DSP C66x CorePac Uncorrected bit error detected
10	SYS_CMPA	DSP C66x CorePac CPU memory protection fault
11	PMC_CMPA	DSP C66x CorePac CPU memory protection fault
12	PMC_DMPA	DSP C66x CorePac DMA memory protection fault
13	DMC_CMPA	DSP C66x CorePac CPU memory protection fault
14	DMC_DMPA	DSP C66x CorePac DMA memory protection fault
15	UMC_CMPA	DSP C66x CorePac CPU memory protection fault
16	UMC_DMPA	DSP C66x CorePac DMA memory protection fault
17	EMC_CMPA	DSP C66x CorePac CPU memory protection fault
18	EMC_BUSERR	DSP C66x CorePac Bus Error Interrupt
19	Reserved	-
20	Reserved	-
21	Reserved	-
22	Reserved	-

Note that neither of the events, listed in Table 5-5 , is exported as a separate hardware interrupt off the DSP boundary.

5.3.4.2.3 Non-DSP C66x CorePac Generated Peripheral Interrupt Outputs

The non-DSP C66x CorePac interrupts generated by peripherals within the DSP subsystem are also summarized in the Figure 5-5.

Besides the aggregated error event ERRINT_IRQ interrupts – DSPi_IRQ_TPCC_ERR (where i=1 to 2), interrupts (see also Figure 5-5) **generated individually by DSPSS peripherals located outside the DSP C66x CorePac**, are mapped as separate IRQ outputs at DSP boundaries. They are sourced by the DSP_EDMA_CC, DSP_EDMA_TC0, DSP_EDMA_TC1, DSP_MMU0, DSP_MMU1 and DSP_NoC and exported to other host INTCs via the device IRQ_CROSSBAR. Refer to the Section 5.2, for more information on these DSP interrupt outputs mapping.

5.3.5 DSP DMA Requests

The DSP_EDMA_CC (channel controller) supports 64 hardware event inputs, that can be used to synchronize the 64 DMA channels. These event inputs are provided at the DSP subsystem boundary via the device DMA_CROSSBAR and can be mapped to sources within the device.

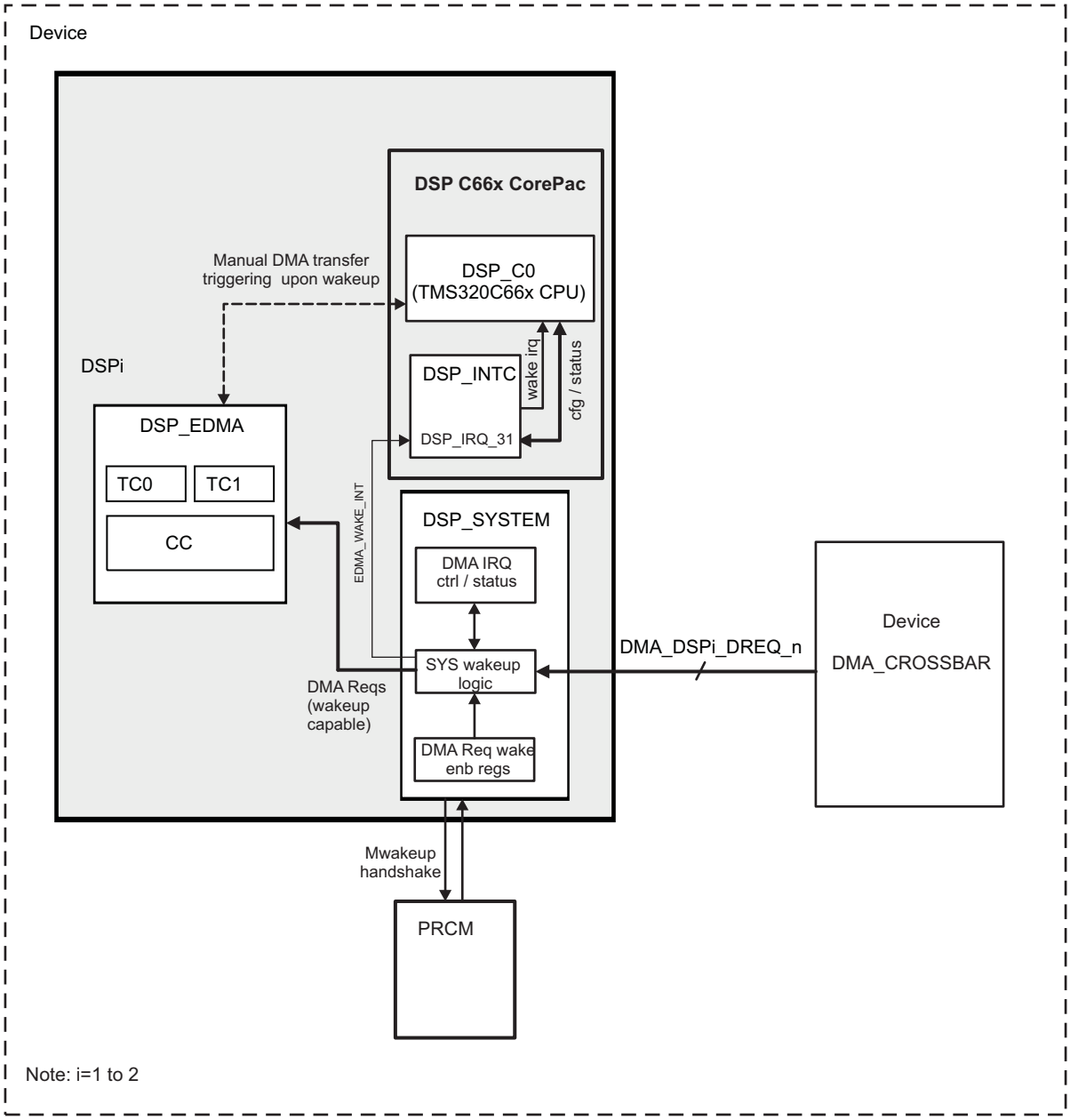
The DSP subsystem receives DMA requests from certain peripherals, such as the McASP modules. The DMA requests path through the DSP logic is shown in Figure 5-7.

Similar to the interrupts received at DSP subsystem boundary, the DSP EDMA requests are first routed through the wakeup generation logic of the DSP_SYSTEM module, hence, each DMA request received by the DSP subsystem can wakeup the system from DSP low power modes (including wakeup from DSP OFF mode). To enable the DMA requests mapped via the DMA_CROSSBAR to **DSP_EDMA_CC [19:0]** inputs, corresponding bits in range [19:0] of the register DSP_SYS_DMAWAKEEN0 must be enabled in software.

CAUTION

The DMA request corresponding DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 MUST be enabled, for the DMA requests to be serviced by the DSP regardless of the DSP being in IDLE or active state.

Figure 5-7. DSP DMA Requests



dspss-005

Table 5-6 and Table 5-7 list the default DMA sources for the DSP1_EDMA and DSP2_EDMA controllers. In addition, DSP1_EDMA / DSP2_EDMA inputs (DMA_DSP1_DREQ_[19:0] / DMA_DSP2_DREQ_[19:0]) can alternatively be sourced through the associated DMA_CROSSBAR from one of the 256 multiplexed device DMA sources listed in Table 16-6. The CTRL_CORE_DMA_DSP1_DREQ_y_z / CTRL_CORE_DMA_DSP2_DREQ_y_z registers (where *y* and *z* are indexes of DSP1_EDMA / DSP2_EDMA input lines) in the Control Module are used to select between the default DMA sources and the multiplexed DMA sources.

For more details on the device DMA_CROSSBAR multiplexing registers structure, refer to Section 18.4.6.5, *DMA_CROSSBAR Module Functional Description* of chapter, *Control Module*.

Table 5-6. DSP1_EDMA Default Request Mapping

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP1_DREQ_0	1	CTRL_CORE_DMA_DSP1_DREQ_0_1[7:0]	128	McASP1_DREQ_RX	McASP1 receive event
DMA_DSP1_DREQ_1	2	CTRL_CORE_DMA_DSP1_DREQ_0_1[23:16]	129	McASP1_DREQ_TX	McASP1 transmit event
DMA_DSP1_DREQ_2	3	CTRL_CORE_DMA_DSP1_DREQ_2_3[7:0]	130	McASP2_DREQ_RX	McASP2 receive event
DMA_DSP1_DREQ_3	4	CTRL_CORE_DMA_DSP1_DREQ_2_3[23:16]	131	McASP2_DREQ_TX	McASP2 transmit event
DMA_DSP1_DREQ_4	5	CTRL_CORE_DMA_DSP1_DREQ_4_5[7:0]	132	McASP3_DREQ_RX	McASP3 receive event
DMA_DSP1_DREQ_5	6	CTRL_CORE_DMA_DSP1_DREQ_4_5[23:16]	133	McASP3_DREQ_TX	McASP3 transmit event
DMA_DSP1_DREQ_6	7	CTRL_CORE_DMA_DSP1_DREQ_6_7[7:0]	134	McASP4_DREQ_RX	McASP4 receive event
DMA_DSP1_DREQ_7	8	CTRL_CORE_DMA_DSP1_DREQ_6_7[23:16]	135	McASP4_DREQ_TX	McASP4 transmit event
DMA_DSP1_DREQ_8	9	CTRL_CORE_DMA_DSP1_DREQ_8_9[7:0]	136	McASP5_DREQ_RX	McASP5 receive event
DMA_DSP1_DREQ_9	10	CTRL_CORE_DMA_DSP1_DREQ_8_9[23:16]	137	McASP5_DREQ_TX	McASP5 transmit event
DMA_DSP1_DREQ_10	11	CTRL_CORE_DMA_DSP1_DREQ_10_11[7:0]	138	McASP6_DREQ_RX	McASP6 receive event
DMA_DSP1_DREQ_11	12	CTRL_CORE_DMA_DSP1_DREQ_10_11[23:16]	139	McASP6_DREQ_TX	McASP6 transmit event
DMA_DSP1_DREQ_12	13	CTRL_CORE_DMA_DSP1_DREQ_12_13[7:0]	140	McASP7_DREQ_RX	McASP7 receive event
DMA_DSP1_DREQ_13	14	CTRL_CORE_DMA_DSP1_DREQ_12_13[23:16]	141	McASP7_DREQ_TX	McASP7 transmit event
DMA_DSP1_DREQ_14	15	CTRL_CORE_DMA_DSP1_DREQ_14_15[7:0]	142	McASP8_DREQ_RX	McASP8 receive event
DMA_DSP1_DREQ_15	16	CTRL_CORE_DMA_DSP1_DREQ_14_15[23:16]	143	McASP8_DREQ_TX	McASP8 transmit event
DMA_DSP1_DREQ_16	17	CTRL_CORE_DMA_DSP1_DREQ_16_17[7:0]	154	Reserved	Reserved
DMA_DSP1_DREQ_17	18	CTRL_CORE_DMA_DSP1_DREQ_16_17[23:16]	155	Reserved	Reserved
DMA_DSP1_DREQ_18	19	CTRL_CORE_DMA_DSP1_DREQ_18_19[7:0]	156	Reserved	Reserved
DMA_DSP1_DREQ_19	20	CTRL_CORE_DMA_DSP1_DREQ_18_19[23:16]	157	Reserved	Reserved
DMA_DSP1_DREQ_20 - DMA_DSP1_DREQ_63	N/A	N/A	N/A	Reserved	Reserved

Table 5-7. DSP2_EDMA Default Request Mapping

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP2_DREQ_0	1	CTRL_CORE_DMA_DSP2_DREQ_0_1[7:0]	128	McASP1_DREQ_RX	McASP1 receive event
DMA_DSP2_DREQ_1	2	CTRL_CORE_DMA_DSP2_DREQ_0_1[23:16]	129	McASP1_DREQ_TX	McASP1 transmit event
DMA_DSP2_DREQ_2	3	CTRL_CORE_DMA_DSP2_DREQ_2_3[7:0]	130	McASP2_DREQ_RX	McASP2 receive event
DMA_DSP2_DREQ_3	4	CTRL_CORE_DMA_DSP2_DREQ_2_3[23:16]	131	McASP2_DREQ_TX	McASP2 transmit event
DMA_DSP2_DREQ_4	5	CTRL_CORE_DMA_DSP2_DREQ_4_5[7:0]	132	McASP3_DREQ_RX	McASP3 receive event

Table 5-7. DSP2_EDMA Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP2_DREQ_5	6	CTRL_CORE_DMA_DSP2_DREQ_4_5[23:16]	133	McASP3_DREQ_TX	McASP3 transmit event
DMA_DSP2_DREQ_6	7	CTRL_CORE_DMA_DSP2_DREQ_6_7[7:0]	134	McASP4_DREQ_RX	McASP4 receive event
DMA_DSP2_DREQ_7	8	CTRL_CORE_DMA_DSP2_DREQ_6_7[23:16]	135	McASP4_DREQ_TX	McASP4 transmit event
DMA_DSP2_DREQ_8	9	CTRL_CORE_DMA_DSP2_DREQ_8_9[7:0]	136	McASP5_DREQ_RX	McASP5 receive event
DMA_DSP2_DREQ_9	10	CTRL_CORE_DMA_DSP2_DREQ_8_9[23:16]	137	McASP5_DREQ_TX	McASP5 transmit event
DMA_DSP2_DREQ_10	11	CTRL_CORE_DMA_DSP2_DREQ_10_11[7:0]	138	McASP6_DREQ_RX	McASP6 receive event
DMA_DSP2_DREQ_11	12	CTRL_CORE_DMA_DSP2_DREQ_10_11[23:16]	139	McASP6_DREQ_TX	McASP6 transmit event
DMA_DSP2_DREQ_12	13	CTRL_CORE_DMA_DSP2_DREQ_12_13[7:0]	140	McASP7_DREQ_RX	McASP7 receive event
DMA_DSP2_DREQ_13	14	CTRL_CORE_DMA_DSP2_DREQ_12_13[23:16]	141	McASP7_DREQ_TX	McASP7 transmit event
DMA_DSP2_DREQ_14	15	CTRL_CORE_DMA_DSP2_DREQ_14_15[7:0]	142	McASP8_DREQ_RX	McASP8 receive event
DMA_DSP2_DREQ_15	16	CTRL_CORE_DMA_DSP2_DREQ_14_15[23:16]	143	McASP8_DREQ_TX	McASP8 transmit event
DMA_DSP2_DREQ_16	17	CTRL_CORE_DMA_DSP2_DREQ_16_17[7:0]	154	Reserved	Reserved
DMA_DSP2_DREQ_17	18	CTRL_CORE_DMA_DSP2_DREQ_16_17[23:16]	155	Reserved	Reserved
DMA_DSP2_DREQ_18	19	CTRL_CORE_DMA_DSP2_DREQ_18_19[7:0]	156	Reserved	Reserved
DMA_DSP2_DREQ_19	20	CTRL_CORE_DMA_DSP2_DREQ_18_19[23:16]	157	Reserved	Reserved
DMA_DSP2_DREQ_20 - DMA_DSP2_DREQ_63	N/A	N/A	N/A	Reserved	Reserved

5.3.5.1 DSP EDMA Wakeup Interrupt

This section provides description of the registers used for the **EDMA wakeup interrupt** functionality, including the EDMA_WAKE_INT IRQ status and enable fields. The EDMA Wakeup Interrupt allows incoming EDMA events to be latched and an interrupt sent to the DSP (if enabled). This interrupt is generated in the DSP_SYSTEM as an single "OR-ed" output of all external DMA requests latched in DSP subsystem. This output is further synchronized to DSP_FCLK and mapped as the EDMA_WAKE_INT event to the DSP_IRQ_31 input of the C66x DSP CorePac DSP_INTC. The C66x CPU is expected to service the interrupt by triggering the corresponding EDMA channel manually, or by servicing the request via normal reads and writes (instead of using the EDMA). This functionality is required since the EDMA is not capable of following the smart wakeup protocol.

NOTE: The DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 registers are used for enabling the assertion of the 'Mwakeup' asynchronous wakeup request to the device PRCM upon DMA requests reception. The interrupt functionality of the registers: DSP_SYS_EDMAWAKE0_x covered in this subsection is specifically for generating **an wake interrupt** to the DSP. In most cases, the enable mask for the two sets of registers should be set to the same value.

The EDMAWAKE0 registers corresponding to the EDMA Events 19 thru 0 (msbit to lsbit) are as follows:

- DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW[19:0]
- DSP_SYS_EDMAWAKE0_IRQSTATUS[19:0]
- DSP_SYS_EDMAWAKE0_IRQENABLE_SET[19:0]
- DSP_SYS_EDMAWAKE0_IRQENABLE_CLR[19:0]

Following functional descriptions are valid for the above registers :

- **IRQ status raw register -** This register provides a per-event raw interrupt status vector. The Raw status is set even if the corresponding event is not enabled. Software can write '1' to set the (raw) status for debug purposes.
- **IRQ status register -** This register provides a per-event enabled interrupt status vector. The Enabled status is set if the corresponding event is enabled and the raw status is set. Software can write 1 to clear the (raw) status after the interrupt has been serviced. The clear takes effect even if the interrupt is not enabled.
- **IRQ enable register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to set (i.e., enable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Clear" - register)
- **IRQ clear register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to clear (i.e., disable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Set" register)

NOTE: A DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW bit is set even if the corresponding event is NOT enabled in the DSP_SYS_EDMAWAKE0_IRQENABLE_SET[19:0].

5.3.6 DSP Intergated Memory Management Units

5.3.6.1 DSP MMUs Overview

A standalone memory management unit (DSP_MMU0) is included within the DSP1 (DSP1_MMU0) and DSP2 (DSP2_MMU0) subsystems boundaries. The DSP_MMU0 is integrated on the C66x CPU MDMA path to the device L3_MAIN interconnect. This provides several benefits including protection of the system memories from corruption by DSP1 and DSP2 accidental accesses.

A standalone memory management unit (DSP_MMU1) is included within the DSP1 (DSP1_MMU1) and DSP2 (DSP2_MMU1) subsystems boundaries. The DSP_MMU1 is integrated on the EDMA data path which starts from the L2 DSP_NoC interconnect and leaves the DSP subsystem on the DSP EDMA master port. This provides several benefits including protection of the device L3_MAIN memory space from corruption by DSP1 and DSP2 DMA (DSP1_EDMA and DSP2_EDMA, respectively) accidental accesses.

Both DSP MMUs generate interrupts which are internally mapped to the DSP C66x CorePac DSP_INTC and output to the device IRQ_CROSSBAR. See also the Section 5.2 and Section 5.3.4.

CAUTION

In the case of a page fault, a DSP C66x CorePac CPU is unable to service it's own DSP_MMU0 and DSP_MMU1 interrupts . The device MPU (Cortex-A15) is expected to manage any TLB patches as necessary.

Both DSP MMUs (on MDMA and EDMA paths respectively) have identical functionalities.

- 32-bit input and ouput address width (to match L3_MAIN address width)
- 32 TLB cache entries
- 32 + 1 tags
- 128-bit data bus for MDMA and EDMA

5.3.6.2 Routing MDMA Traffic through DSP MMU0

DSP C66x CPU traffic initiated on the DSP MDMA port can be optionally routed through the DSP_MMU0 on the 32-bit MDMA address path. This is controlled in two levels :

- **global** by the DSP_SYSTEM register DSP_SYS_MMU_CONFIG [0] MMU0_EN bit. This bit acts as a mux-select : setting it to 0b1 enables requests to use the DSP_MMU0; clearing this bit to 0b0 disables MMU table lookup and causes accesses to use the non-translated address (MMU bypass). By default the DSP_MMU0 is disabled in DSP_SYSTEM and MDMA port traffic bypasses the DSP_MMU0.
- **local** by MMU enable control in a dedicated DSP_MMU0 memory mapped register. It is used to enable the MMU functionality after the page tables are programmed for MMU operation. For details, refer to the Chapter 20, *Memory Management Units*.

NOTE: For the DSP_MMU0 to operate, SW should enable it both at the DSP_SYSTEM global level and DSP_MMU0 local register level.

When enabling the DSP_MMU0, software must take care that no transactions are in flight through that MMU. This is typically handled by issueing a DSP "MFENCE" instruction operation. Note that the local enable bit inside the DSP_MMU0 must be configured as normal (refer to the Chapter 20, *Memory Management Units*.)

For more information on the MFENCE operation, refer to the section, *C66x CPU Instruction Set* of the *TMS320C66x DSP CPU and Instruction Set*).

In addition, the DSP_MMU0 traffic can be aborted in case of a lockup via the DSP_SYS_MMU_CONFIG [8] MMU0_ABORT bit. In other words, this bit can be used to clear a hang condition that may occur if the DSP_MMU0 encounters a page fault that cannot be serviced.

For more information on device DSP_MMU0 functionality and register settings, refer to the Section 20.3, *MMU Functional Description* and Section 20.5, *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

5.3.6.3 Routing EDMA Traffic thorugh DSP MMU1

The DSP_EDMA traffics initiated on the DSP EDMA master port can be optionally routed through the DSP_MMU1 on the EDMA address path. This is controlled in two levels :

- **global** by the DSP_SYSTEM register DSP_SYS_MMU_CONFIG [4] MMU1_EN bit. This bit acts as a

mux-select : setting it to 0b1 enables requests to use the DSP_MMU1; clearing this bit to 0b0 disables MMU table lookup and causes accesses to use the non-translated address (MMU bypass). By default the DSP_MMU1 is disabled in DSP_SYSTEM and EDMA traffic bypasses the DSP_MMU1.

- **local** by MMU enable control in a dedicated DSP_MMU1 memory mapped register. It is used to enable the MMU functionality after the page tables are programmed for MMU operation. For details, refer to the Section 20.5, *MMU Register Manual*, in the Chapter 20,*Memory Management Units*.

NOTE: For the DSP_MMU1 to operate, SW should enable it both at the DSP_SYSTEM global level and DSP_MMU1 local register level.

When enabling the DSP_MMU1, software must take care that no transactions are in flight through that MMU. This is typically handled by disabling any EDMA transactions prior to enabling the MMU. Note that the local enable bit inside the DSP_MMU1 must be configured as normal (refer to the Chapter 20,*Memory Management Units*.)

For more information on the MFENCE operation, refer to the section, *C66x CPU Instruction Set* of the *TMS320C66x DSP CPU and Instruction Set*).

In addition, the DSP_MMU1 traffic can be aborted in case of a lockup via the DSP_SYS_MMU_CONFIG [12] MMU1_ABORT bit. In other words, this bit can be used to clear a hang condition that may occur if the DSP_MMU1 encounters a page fault that cannot be serviced.

For more information on device DSP_MMU1 functionality and settings, refer to the Section 20.3, *MMU Functional Description* and the Section 20.5, *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

5.3.7 DSP Integrated EDMA Subsystem

This section represents an overview of the DSP integrated EDMA functionalities, as well as the subsystem level and device related register controls. For more details on the EDMA functionalities and programming registers, refer to the Section 16.2, *Enhanced DMA*.

5.3.7.1 DSP EDMA Overview

The enhanced-DMA subsystem which is part of the DSP1 (DSP1_EDMA) and the DSP2 (DSP2_EDMA) subsystems is the primary DMA engine for transfers between system memory (DDR and/or L3_MAIN SRAM) and DSP internal memories (L1s and L2).

The Channel Controller - DSP_EDMA_CC serves as the “user interface” of the DSP_EDMA. The two Transfer Controllers - DSP_EDMA_TC0 and DSP_EDMA_TC1 serve as the data transfer engines of the DSP_EDMA. The C66x CPU tipically programs the Channel Controller, which in turn submits Transfer Requests (TR) to the appropriate Transfer Controller. Interrupts are posted in the DSP_EDMA_CC upon transfer completion (if requested), and signaled to the C66x. The EDMA TC completion interrupt is not supported/connected.

The DSP_EDMA is primarily used to perform block transfers between DSP C66x CorePac memories (mostly L2 memory) and system memory (mostly DDR or L3 SRAM).

The DSP_EDMA is configured with 2 Queues (in the CC). Two DSP_EDMA traffic controllers (TC) offer high performance and preemptability of transfers. For typical use cases, it is expected that low latency/small payload transfers) use Queue0/TC0 and high bandwidth/large payload transfers (e.g., DDR on L3_MAIN or DSP local L2 SRAM) will use Queue1/TC1.

DSP_EDMA_CC configuration in the device features :

- 64x EDMA channels
- 8x QDMA channels
- 64x interrupt channels
- 128x PaRAM entries
- 2x Event Queues
- 2x Traffic controllers

- memory protection support
- channel mapping capability
- 8x memory protected and Shadow Regions

DSP_EDMA_TC0/TC1 configuration in the device features :

- 2048 Byte FIFO support
- multitag support
- 16-bit data bus
- 4-level destination register depth
- 7-bit address for internal FIFOs
- 16 IDs for Read commands
- 16 IDs for Write commands

NOTE: The device DSP integrated EDMA controller instances (DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1) are functionally identical with the device EDMA controller instances (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1). The only difference is that the DSP_EDMA instances are located at different physical addresses.

For more details on the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 controllers functionalities, refer to Section 16.2.5, *EDMA Controller Functional Description*, in Section 16.2, *Enhanced DMA*.

The DSP_EDMA instances, their corresponding registers summary and descriptions are covered in the Section 16.2.8, *EDMA Register Manual* of the Section 16.2, *Enhanced DMA*.

5.3.7.2 DSP System and Device Level Settings of DSP EDMA

DSP_EDMA traffic TC0 and TC1 controllers "Active" or "Idle" status: can be monitored in DSP_SYSTEM located :

- DSP_SYS_STAT[1] TC0_STAT bit
- DSP_SYS_STAT[2] TC1_STAT bit

The **default** burst size for both the DSP_EDMA_TC0 and DSP_EDMA_TC1 can be defined in DSP_SYSTEM register. This is achieved via programming :

- DSP_SYS_BUS_CONFIG [1:0] TC0_DBS
- DSP_SYS_BUS_CONFIG [5:4] TC1_DBS

There are also other DSP_EDMA controls associated with DSP_NoC interconnect pressure settings. For more details, refer to the Section 5.3.8.

The 3 error events associated with the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 are exported outside the DSP C66x CorePac in the subsystem, and are able to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"tpcc_errint_level", "tptc_errint0_level" and "tptc_errint1_level"** events, respectively in the Table 5-5.

NOTE: The DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 events are NOT exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

The DSP_SYSTEM logic is assigned to route the external DMA requests to the EDMA hardware request inputs. Additionally, EDMA events can conditionally wake-up the DSP system from a low power mode, via software enabling DSP_SYSTEM MWakeup handshake with the device PRCM. This mechanism is described in the Section 5.3.5.

The programmable muxing of various external DMA request sources to the DSP EDMA. DMA_DSP1_DREQ_x and DMA_DSP2_DREQ_x input lines (where x=0 to 19) is covered in the Section 18.4.6.5, *DMA_CROSSBAR Module Functional Description*, of the Chapter 18, *Control Module*.

DSP1/DSP2 subsystem external DMA request sources : For the default DSP1 / DSP2 external DMA request sources, routed via the device DMA_CROSSBAR to the DSP1_EDMA / DSP2_EDMA channel controller inputs (DMA_DSP1_DREQ_i / DMA_DSP2_DREQ_i), respectively, refer to the Section 5.3.5.

5.3.8 DSP L2 interconnect Network

A 128-bit level 2 (L2) Interconnect from Arteris - FlexNoC® is instantiated in the DSP subsystem, outside the DSP C66x CorePac. It is signified as "DSP_NoC" throughout this chapter.

NOTE: The C66x master MDMA data does NOT flow through the DSP_NoC.

The system and local initiators on DSP_NoC are as follows :

- local C66x CPU 32-bit CFG master port which traffic is split via DSP_NoC fabric into several configuration target traffics inside and outside the DSP subsystem.
- SDMA initiator port on DSP_NoC which conveys accesses towards DSP Memories and memory-mapped registers initiated outside the DSP subsystem via the L3_MAIN interconnect.
- EDMA traffic controllers - TC0 read / write initiator ports
- EDMA traffic controllers - TC1 read / write initiator ports

NOTE: The DSP_ICFG space is not visible to SDMA initiators (DSP_EDMA or DSP hosts on L3_MAIN) with CFG traffic.

The targets on the DSP_NoC are as follows :

- DSP C66x CorePac SDMA port
- Internal CFG targets on the DSP_NoC :
  - DSP_MMU0 Cfg
  - DSP_MMU1 Cfg
  - DSP_SYSTEM Cfg
  - DSP_EDMA_CC Cfg
  - DSP_EDMA_TC0 Cfg
  - DSP_EDMA_TC1 Cfg
- 32-bit CFG port on L3_MAIN (it acts as master on the L3_MAIN)
- EDMA Target port which conveys EDMA bidi transfers outside the DSP (through or bypassing DSP_MMU1).

The Table 5-8 summarizes the interconnections which can be established between DSP initiators and targets over the L2 DSP_NoC in the device. In this table HW implemented interconnections are marked with an asterics.

Table 5-8. DSP_NoC Defined Connectivities

		DSP_NoC Initiators			
		DSP C66x CorePac CFG init	EDMA_TC0 init	EDMA_TC1 init	SDMA init (mapped to SDMA port on L3_MAIN)
DSP_NoC Targets	DSP C66x CorePac SDMA (slave) port	n.a.	*	*	*
	DSP_MMU0 Cfg	*	n.a.	n.a.	*
	DSP_MMU1 Cfg	*	n.a.	n.a.	*
	DSP_SYSTEM Cfg	*	n.a.	n.a.	*
	DSP_EDMA_CC Cfg	*	n.a.	n.a.	*
	DSP_EDMA_TC0 Cfg	*	n.a.	n.a.	*
	DSP_EDMA_TC1 Cfg	*	n.a.	n.a.	*
	DSP_NoC Cfg	*	n.a.	n.a.	*
	Cfg port (Cfg Init on L3_MAIN)	*	n.a.	n.a.	n.a.
	Master DMA port (DSP DMA init on L3_MAIN)	n.a.	*	*	n.a.

A DSP_NoC error event (combination of several local to the interconnect events) is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"noc_errint_level"** event in the Table 5-5.

NOTE: The DSP_NoC event is NOT exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

5.3.8.1 DSP Public Firewall Settings

The DSP1 and DSP2 L2 Interconnect (DSP1_NoC and DSP2_NoC, respectively) implements two firewalls – dsp firewall0 (DSP_FW0) is used to protect DSP_MMU0's configuration space (which includes the TLB) and dsp firewall1 is used to protect DSP_MMU1's configuration space (which includes the TLB). Access permission is based on the privilege level, domain, ConnID, and access types of a request.

The default value of 0xFFFF_FFFF in the MRM region 0 permission registers:

- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0
- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0
- L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0
- L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

permits any requestor to access the DSP_MMU0 and DSP_MMU1 configuration space.

For more information on the access region definitions, public privilege access, public user access and initiator permission settings, which are identical between DSP_NoC firewalls and L3_MAIN interconnect firewalls, refer to the Section 14.2.3.7.3, *L3_MAIN Firewall Functionality*, in the, Section 14.2, *L3_MAIN Interconnect*.

There are also several other DSP_NoC registers - L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0, L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL used for error handling, firewall reset and other purposes. These are functionally identical with the corresponding L3_MAIN interconnect registers, described in the Section 14.2.3.8, *L3_MAIN Interconnect Error Handling*, in the Section 14.2, *L3_MAIN Interconnect*.

The various firewall access control registers are part of the C66x CPU local accessible - DSP_FW_L2_NOC_CFG address space, and L3_MAIN initiators accessible DSP1/2_FW_L2_NOC_CFG configuration space. The corresponding MMU0 and MMU1 configuration space firewall registers (DSP_FW0 starting at offset **0x0000_0000** , and DSP_FW1 starting at offset **0x0000_1000**) are summarized and described in the Section 5.4.4.

5.3.8.2 DSP NoC Flag Mux and Error Log Registers

The DSP_NoC registers (starting at offset >= 0x0000_4000) are used for error logging and flag muxing purposes. The status information stored in there can be used for example to resolve issues related to DSP_NoC access conflicts, for debug purposes, etc.

For more information, refer to the Section 14.2.3.5, *Flag Muxing*, in the Section 14.2, *L3_MAIN Interconnect*.

5.3.8.3 DSP NoC Arbitration

A pressure based arbitration is implemented for the DSP_NoC interconnect.

A DSP_NoC local MFlag mechanism is used but it is SW controlled in the DSP_SYSTEM configuration space.

This is done via the register DSP_SYS_BUS_CONFIG bitfields :

- TC0_L2PRES - for DSP_EDMA_TC0 pressure control
- TC1_L2PRES - for DSP_EDMA_TC1 pressure control
- CFG_L2PRES - for the DSP C66x CorePac 32-bit CFG pressure control
- SDMA_L2PRES - for pressure control of the DSP system and L3_MAIN accesses targetting the DSP C66x CorePac SDMA slave port

The pressure for each port is signaled on a MFlag[1:0] bus and conveys a value of 0 (lowest), 1 (medium), or 3 (highest). A value of 0x2 is reserved/undefined.

NOTE: The default pressure level for all ports is 0x0 and is recommended for most systems. This results in round-robin arbitration across active requests.

5.3.9 DSP Boot Configuration

DSP1 subsystem boot vector input which defines the 22-bit DSP1 Boot Address is mapped to the device core control module register CTRL_CORE_CONTROL_DSP1_RST_VECT[21:0] DSP1_RST_VECT bitfield. DSP2 subsystem boot vector input which defines the 22-bit DSP2 Boot Address is mapped to the device core control module register CTRL_CORE_CONTROL_DSP2_RST_VECT[21:0] DSP2_RST_VECT bitfield.In general, the device MPU (Cortex-A15) host loads code to a given address location in the device system memory, sets the DSP1_RST_VECT / DSP2_RST_VECT bitfield to the address value, and then release the DSP1 / DSP2 from reset. At that point, the DSP1 / DSP2 will begin fetching code from that location.

NOTE: If the values of the control core module CTRL_CORE_CONTROL_DSP1_RST_VECT[21:0] / CTRL_CORE_CONTROL_DSP2_RST_VECT [21:0] register change, the values will be taken into account by DSP upon the next reset.

NOTE: Upon device boot time (a power-on reset applied), the device "sysboot15" input latched in the Control Module bootstrap register defines the value of DSP functional clock divider (2 or 3). For more details, refer to the Section 5.3.3.1

5.3.10 DSP Internal and External Memory Views

5.3.10.1 C66x CPU View of the Address Space

The **C66x CPU View** represents the view from the DSP, which result from program fetches, or load / store instructions. Accesses to DSP memories (L1P, L1D, L2) and to DSP Internal configuration space (DSP_ICFG) are intercepted within the DSP C66x CorePac (whether using local or global addresses).

The DSP C66x CorePac CFG (C66x CPU 32-bit master port) interface is strictly for non-cacheable loads and stores, and is intended to be used for I/O space or memory mapped registers (MMR) space. DSP C66x CorePac CFG accesses are routed / arbitrated by the DSP_NoC L2 inteconnect. DSP C66x CorePac CFG accesses are mapped between 0x01C0_0000 and 0x0FFF_FFFF.

NOTE: The DSP C66x CorePac CFG initiator interface is strictly for non-cacheable loads and stores within MMR and I/O spaces.

DSP configuration accesses to **external to DSP subsystem** peripherals can be issued on either the DSP subsystem CFG Master port which is mapped to the device L3_MAIN or DSP 32-bit CFG "system" interface is connected to the chip level L3_MAIN interconnect and used to access L3_MAIN addressses that are mapped between 0x01C0_0000 and 0x0FFF_FFFF.

DSP **accesses (to non-DSP memories like SDRAM on L3_MAIN) for addresses above 0x1000_0000 are handled via the DSP (XMC) MDMA 128-bit master interface and are routed to the DSP subsystem MDMA Initiator port either through DSP_ MMU0 or bypassing MMU.**

NOTE: In some cases, L3_MAIN peripherals may be mapped to both the MDMA bus and the CFG bus. In that case, there may be a latency advantage of using the CFG address for those peripherals

Table 5-9 shows the DSP C66x CPU memory view of the various DSP C66x CorePac internal and external resources.

Table 5-9. C66x CPU View Map

C66x CPU View (DSP C66x CorePac Internal only, MDMA or CFG init ports)[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0080_0000	0x0084_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x00E0_0000	0x00E0_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D	DSP L1D SRAM (local)
0x0180_0000	0x01BF_FFFF	4096 KiB	DSP_ICFG	DSP Internal CFG [2]
0x01D0_0000	0x01D0_0FFF	4 KiB	DSP_SYSTEM	DSP_SYSTEM Memory Mapped Registers block
0x01D0_1000	0x01D0_1FFF	4 KiB	DSP_MMU0CFG	DSP MMU0 configuration / registers
0x01D0_2000	0x01D0_2FFF	4 KiB	DSP_MMU1CFG	DSP MMU1 configuration / registers
0x01D0_5000	0x01D0_5FFF	4 KiB	DSP_EDMA_TC0	DSP_EDMA Transfer Controller 0
0x01D0_6000	0x01D0_6FFF	4 KiB	DSP_EDMA_TC1	DSP_EDMA Transfer Controller 1
0x01D0_7000	0x01D0_7FFF	4 KiB	DSP_NoC	DSP L2 interconnect registers
0x01D1_0000	0x01D1_7FFF	32 KiB	DSP_EDMA_CC	DSP_EDMA Channel Controller
0x0200_0000	0x020F_FFFF	1 MiB	EVE1[3]	DSP configuration traffic to the EVE1 (mapped on the DSP CFG interface)
0x0210_0000	0x021F_FFFF	1 MiB	EVE2[3]	DSP configuration traffic to the EVE2 (mapped on the DSP CFG interface)
0x0220_0000	0x022F_FFFF	1 MiB	EVE3[3]	DSP configuration traffic to the EVE3 (mapped on the DSP CFG interface)
0x0230_0000	0x023F_FFFF	1 MiB	EVE4[3]	DSP configuration traffic to the EVE4 (mapped on the DSP CFG interface)
0x0330_0000	0x033F_FFFF	1 MiB	EDMA_TPCC	DSP configuration traffic to the EDMA_TPCC (mapped on the DSP CFG interface)
0x0340_0000	0x034F_FFFF	1 MiB	EDMA_TC0	DSP configuration traffic to the EDMA_TC0 (mapped on the DSP CFG interface)
0x0350_0000	0x035F_FFFF	1 MiB	EDMA_TC1	DSP configuration traffic to the EDMA_TC1 (mapped on the DSP CFG interface)
0x0800_0000	0x0800_FFFF	64 KiB	DSP_XMC_CTRL MMRs	DSP internal MMRs for XMC controller (non-cache)

(1) Only the C66x CPU view of device implemented functional memory regions are shown. The remaining regions are reserved.
(2) The internal configuration space registers are visible ONLY to the C66x CPU (DSP_C0) within the DSP C66x CorePac , i.e. they are NOT visible to initiators outside the DSP C66x CorePac .
(3) EVE is not supported in this family of devices.

Table 5-9. C66x CPU View Map (continued)

C66x CPU View (DSP C66x CorePac Internal only, MDMA or CFG init ports)[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0802_0000	0x080F_FFFF	896 KiB	MDMA non-cache	MDMA initiator (non-cache) to L3_MAIN (DSP_MMU0)
0x0810_0000	0x0BBF_FFFF	59 MiB		
0x1000_0000	0x10FF_FFFF	16 MiB	DSP1 L1P, L1D and L2 memories	An image of DSP1 C66x CorePac internal space - only L1P, L1D and L2 memories (global) [4]
0x1000_0000	0x10FF_FFFF	16 MiB	DSP2 MDMA (cached)	DSP2 MDMA initiator (cached) to L3_MAIN (through DSP2_MMU0)
0x1100_0000	0x11FF_FFFF	16 MiB	DSP2 L1P, L1D and L2 memories	An image of DSP2 C66x CorePac internal space - only L1P, L1D and L2 memories (global) [5]
			DSP1 MDMA (cached)	MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0)
0x1200_0000	0x1FFF_FFFF	224 MiB	MDMA (cached)	DSP1 and DSP2 MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0 / DSP2_MMU0, respectively)
0x2000_0000	0xFFFF_FFFF	3584 MiB	MDMA (cached)	DSP1 and DSP2 MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0 / DSP2_MMU0, respectively)

(4) The DSP1 CPU sees an image of its own memories in the 0x1000_0000 - 0x10FF_FFFF address range (the same mapped also at lower addresses 0x0080_0000 - 0x00F0_7FFF). On the other side, DSP2 CPU addresses, generated in the same range (with DSP2_MMU0 involved), are mapped to the DSP2 MDMA port (cached transfers to/from L3_MAIN connected system memories).

(5) The DSP2 CPU sees an image of its own memories in the 0x1100_0000 - 0x11FF_FFFF address range (the same mapped also at lower addresses 0x0080_0000 - 0x00F0_7FFF). On the other side, DSP1 CPU addresses, generated in the same range (with DSP1_MMU0 involved), are mapped to the DSP1 MDMA port (cached transfers to / from L3_MAIN connected system memories).

Refer to the Section 2.2, *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the Section 2.6, *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

5.3.10.2 DSP_EDMA View of the Address Space

EDMA is able to initiate internal accesses directly to the DSP memories via the DSP C66x CorePac SDMA bus. The access is conducted to the DSP C66x CorePac internal memories over the L2 DSP_NoC interconnect.

Table 5-10 shows the DSP integrated EDMA controller memory view of the various DSP C66x CorePac internal and external resources.

Table 5-10. DSP EDMA Controller View Map

DSP_EDMA Controller View (EDMA master internal / external port)		Size	DSP Memory Region	Function
Start Address	End Address			
0x0080_0000	0x0084_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x00E0_0000	0x00E0_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D	DSP L1D SRAM (local)
0x0802_0000	0x0BBF_FFFF	59 MiB	EDMA to L3_MAIN	EDMA initiator (DSP_MMU1)

Table 5-10. DSP EDMA Controller View Map (continued)

DSP_EDMA Controller View (EDMA master internal / external port)		Size	DSP Memory Region	Function
Start Address	End Address			
0x1000_0000	0x10FF_FFFF	16 MiB	DSP L1/L2	An image of DSP C66x CorePac internal space - only L1P, L1D and L2 memories (global) (1)
0x2000_0000	0xFFFF_FFFF	3584 MiB	DMA OCP	L3_MAIN interconnect memory via MMU1 / DMA OCP Initiator

(1) The internal configuration space registers DSP_ICFG are visible ONLY to the C66x CPU (DSP_C0) within the DSP C66x CorePac , i.e. they are NOT visible to the DSP_EDMA.

Access from EDMA to external resources on L3_MAIN are routed via DSP subsystem **EDMA initiator port**. **Note that these accesses are transferred through the DSP_MMU1 memory management unit.**

NOTE: The DSP_EDMA can NOT access the DSP_ICFG (DSP C66x CorePac internal) addresses.

With the DSP_MMU1 disabled, the subset of the memory map used for DSP_EDMA internal accesses will NOT be visible. Thus only addresse which equal 0x2000_0000 and above will be considered as valid 32-bit addresses (i.e. L3_MAIN space accesses only).

Refer to the Section 2.2, *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the Section 2.6, *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

5.3.10.3 L3_MAIN View of the DSP Address Space

System initiated accesses (i.e. external-to-DSP accesses over device L3_MAIN) to DSP are issued over the DSP SDMA target port.

NOTE: The MSB-bits of the address are truncated to only provide an 8 MiB view of the memory map within the DSP subsystem. Notice that the relative offsets of the DSP CFG space is different for the OCP SDMA target port relative to the DSP internal initiators.

The SDMA target bus is able to access internal subsystem address space (such as DSP_EDMA, DSP_MMU0, DPS_MMU1, etc.), or the DSP local memory address space. The SDMA target bus **is NOT able to access the DSP ICFG space (such as DSP_INTC, DSP_BWM, etc)** or the other initiator ports on the DSP subsystem boundary (i.e., accesses cannot go through DSPSS to get to the DSP_EDMA initiator port, L3_MAIN CFG initiator port).

The DSP slave DMA port memory map - Table 5-11 shows an 8 MiB window (23-bit address) both from the SDMA Target bus (0x0000_0000 through 0x007F_FFFF), as well as the EDMA (0x0080_0000 through 0x00FF_FFFF). The DSP C66x CorePac internally views itself as a 16 MiB window where 0x0000_0000 through 0x007F_FFFF is reserved, L2 SRAM starts at 0x0080_0000, L1P SRAM starts at 0x00E0_0000, and L1D SRAM starts at 0x00F0_0000).

Table 5-11. SDMA Target Port Memory Map

System L3_MAIN View(1)		Size	DSP Memory Region	Function
Start Address	End Address			
0x0000_0000	0x0004_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x0050_0000	0x0050_0FFF	4 KiB	DSP_SYSTEM	DSP SYSTEM MMR Block
0x0050_1000	0x0050_1FFF	4 KiB	DSP_MMU0CFG	DSP MMU0 configuration / regs

(1) Only system (L3_MAIN) view over functionally used regions are shown. The remaining regions are reserved.

Table 5-11. SDMA Target Port Memory Map (continued)

System L3_MAIN View[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0050_2000	0x0050_2FFF	4 KiB	DSP_MMU1CFG	DSP MMU1 configuration / regs
0x0050_5000	0x0050_5FFF	4 KiB	DSP_EDMA_TC0	DSP EDMA Transfer Controller 0
0x0050_6000	0x0050_6FFF	4 KiB	DSP_EDMA_TC1	DSP EDMA Transfer Controller 1
0x0050_7000	0x0050_7FFF	4 KiB	DSP_NoC	DSP L2 Interconnect registers
0x0051_0000	0x0051_7FFF	32 KiB	DSP_EDMA_CC	DSP EDMA Channel Controller
0x0060_0000	0x0060_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x0070_0000	0x0070_7FFF	32 KiB	DSP_L1D	DSP L1D SRAM (local)

Refer to the Section 2.2, *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the Section 2.6, *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

5.4 DSP Subsystem Register Manual

This section describes the DSP Subsystem instances registers.

5.4.1 DSP Subsystem Instance Summary

Table 5-12. DSP Subsystem Instance Summary

Module Name	Module Base Address	Size
DSP_ICFG	0x0180 0000[1]	4 KiB
DSP_SYSTEM	0x01D0 0000[1]	256 Bytes
DSP_FW_L2_NOC_CFG	0x01D0 3000[1]	8576 Bytes
DSP1_SYSTEM	0x40D0 0000	256 Bytes
DSP1_FW_L2_NOC_CFG	0x40D0 3000	8576 Bytes
DSP2_SYSTEM	0x4150 0000	256 Bytes
DSP2_FW_L2_NOC_CFG	0x4150 3000	8576 Bytes

(1) **The registers of DSP subsystem instances prefixed only with DSP in the name, and NOT DSP1 or DSP2, are NOT visible on the device L3_MAIN. They are visible only within the DSP_ICFG internal configuration space hence accessible only by the DSP C66x CPU.**

NOTE: For more details on the DSP_MMU0 and DSP_MMU1 registers, as well as their :

- DSP_MMU0CFG and DSP_MMU1CFG physical addresses accesible only by DSP_C0 CPU core in the DSP subsystem
- DSP1_MMU0CFG and DSP1_MMU1CFG physical addresses visible on L3_MAIN
- DSP2_MMU0CFG and DSP2_MMU1CFG physical addresses visible on L3_MAIN

refer to Section 20.5, *MMU Register Manual*, in Chapter 20, *Memory Management Units*.

NOTE: For more details on the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 registers, as well as their :

- DSP1_EDMA_CC, DSP1_EDMA_TC0 and DSP1_EDMA_TC1 physical addresses visible on L3_MAIN
- DSP2_EDMA_CC, DSP2_EDMA_TC0 and DSP2_EDMA_TC1 physical addresses visible on L3_MAIN

refer to Section 16.2.8, *EDMA Register Manual*, in Section 16.2, *Enhanced DMA*.

CAUTION

The L1P,L1D and L2- memory controller registers mapped in the L3_MAIN are limited to 32-bit data access; 16- and 8-bit access are not allowed and can corrupt register content.

5.4.2 DSP_ICFG Registers

5.4.2.1 DSP_ICFG Register Summary

NOTE: The DSP_ICFG addresses are visible only within the DSP core internal configuration space.

Table 5-13. DSP_ICFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
EVTFLAG0	R	32	0x0000 0000	0x0180 0000	0x0000 0000
EVTFLAG1	R	32	0x0000 0004	0x0180 0004	0x0000 0000
EVTFLAG2	R	32	0x0000 0008	0x0180 0008	0x0000 0000
EVTFLAG3	R	32	0x0000 000C	0x0180 000C	0x0000 0000
EVTSET0	W	32	0x0000 0020	0x0180 0020	0x0000 0000
EVTSET1	W	32	0x0000 0024	0x0180 0024	0x0000 0000
EVTSET2	W	32	0x0000 0028	0x0180 0028	0x0000 0000
EVTSET3	W	32	0x0000 002C	0x0180 002C	0x0000 0000
EVTCLR0	W	32	0x0000 0040	0x0180 0040	0x0000 0000
EVTCLR1	W	32	0x0000 0044	0x0180 0044	0x0000 0000
EVTCLR2	W	32	0x0000 0048	0x0180 0048	0x0000 0000
EVTCLR3	W	32	0x0000 004C	0x0180 004C	0x0000 0000
EVTMASK0	RW	32	0x0000 0080	0x0180 0080	0x0000 0000
EVTMASK1	RW	32	0x0000 0084	0x0180 0084	0x0000 0000
EVTMASK2	RW	32	0x0000 0088	0x0180 0088	0x0000 0000
EVTMASK3	RW	32	0x0000 008C	0x0180 008C	0x0000 0000
MEVTFLAG0	R	32	0x0000 00A0	0x0180 00A0	0x0000 0000
MEVTFLAG1	R	32	0x0000 00A4	0x0180 00A4	0x0000 0000
MEVTFLAG2	R	32	0x0000 00A8	0x0180 00A8	0x0000 0000
MEVTFLAG3	R	32	0x0000 00AC	0x0180 00AC	0x0000 0000
EXPMASK0	RW	32	0x0000 00C0	0x0180 00C0	0xFFFF FFFF
EXPMASK1	RW	32	0x0000 00C4	0x0180 00C4	0xFFFF FFFF
EXPMASK2	RW	32	0x0000 00C8	0x0180 00C8	0xFFFF FFFF
EXPMASK3	RW	32	0x0000 00CC	0x0180 00CC	0xFFFF FFFF
MEXPFLAG0	R	32	0x0000 00E0	0x0180 00E0	0x0000 0000

(1) This column considers not ONLY the DSP C66x CorePac specific reset values, but also the device level specific reset values.

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
MEXPFLAG1	R	32	0x0000 00E4	0x0180 00E4	0x0000 0000
MEXPFLAG2	R	32	0x0000 00E8	0x0180 00E8	0x0000 0000
MEXPFLAG3	R	32	0x0000 00EC	0x0180 00EC	0x0000 0000
INTMUX1	RW	32	0x0000 0104	0x0180 0104	0x0706 0504
INTMUX2	RW	32	0x0000 0108	0x0180 0108	0x0B0A 0908
INTMUX3	RW	32	0x0000 010C	0x0180 010C	0x0F0E 0D0C
AEGMUX0	RW	32	0x0000 0140	0x0180 0140	0x0302 0100
AEGMUX1	RW	32	0x0000 0144	0x0180 0144	0x0706 0504
INTXSTAT	RW	32	0x0000 0180	0x0180 0180	0x0000 0000
INTXCLR	RW	32	0x0000 0184	0x0180 0184	0x0000 0000
INTDMASK	RW	32	0x0000 0188	0x0180 0188	0x0000 0000
EVTASRT	RW	32	0x0000 01C0	0x0180 01C0	0x0302 0100
PDCCMD	RW	32	0x0001 0000	0x0181 0000	0x0000 0000
MM_REVID	RW	32	0x0001 2000	0x0181 2000	0x0000 0000
IDMA0_STAT	RW	32	0x0002 0000	0x0182 0000	0x0000 0000
IDMA0_MASK	RW	32	0x0002 0004	0x0182 0004	0x0000 0000
IDMA0_SOURCE	RW	32	0x0002 0008	0x0182 0008	0x0000 0000
IDMA0_DEST	RW	32	0x0002 000C	0x0182 000C	0x0000 0000
IDMA0_COUNT	RW	32	0x0002 0010	0x0182 0010	0x0000 0000
IDMA1_STAT	RW	32	0x0002 0100	0x0182 0100	0x0000 0000
IDMA1_SOURCE	RW	32	0x0002 0108	0x0182 0108	0x0000 0000
IDMA1_DEST	RW	32	0x0002 010C	0x0182 010C	0x0000 0000
IDMA1_COUNT	RW	32	0x0002 0110	0x0182 0110	0x0000 0000
CPUARBE	RW	32	0x0002 0200	0x0182 0200	0x0001 0010
IDMAARBE	RW	32	0x0002 0204	0x0182 0204	0x0000 0010
SDMAARBE	RW	32	0x0002 0208	0x0182 0208	0x0000 0001
ECFGARBE	RW	32	0x0002 0210	0x0182 0210	0x0007 0000
ICFGMPFAR	R	32	0x0002 0300	0x0182 0300	0x0000 0000
ICFGMPFSR	RW	32	0x0002 0304	0x0182 0304	0x0000 0000
ICFGMPFCR	RW	32	0x0002 0308	0x0182 0308	0x0000 0000
ECFGERR	RW	32	0x0002 0408	0x0182 0408	0x0000 0000
ECFGERRCLR	RW	32	0x0002 040C	0x0182 040C	0x0000 0000
PAMAP0	RW	32	0x0002 0500	0x0182 0500	0x0000 0000
PAMAP1	RW	32	0x0002 0504	0x0182 0504	0x0000 0001
PAMAP2	RW	32	0x0002 0508	0x0182 0508	0x0000 0002
PAMAP3	RW	32	0x0002 050C	0x0182 050C	0x0000 0003
PAMAP4	RW	32	0x0002 0510	0x0182 0510	0x0000 0004
PAMAP5	RW	32	0x0002 0514	0x0182 0514	0x0000 0005
PAMAP6	RW	32	0x0002 0518	0x0182 0518	0x0000 0006
PAMAP7	RW	32	0x0002 051C	0x0182 051C	0x0000 0007
PAMAP8	RW	32	0x0002 0520	0x0182 0520	0x0000 0007
PAMAP9	RW	32	0x0002 0524	0x0182 0524	0x0000 0007
PAMAP10	RW	32	0x0002 0528	0x0182 0528	0x0000 0007
PAMAP11	RW	32	0x0002 052C	0x0182 052C	0x0000 0007
PAMAP12	RW	32	0x0002 0530	0x0182 0530	0x0000 0007
PAMAP13	RW	32	0x0002 0534	0x0182 0534	0x0000 0007
PAMAP14	RW	32	0x0002 0538	0x0182 0538	0x0000 0007

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
PAMAP15	RW	32	0x0002 053C	0x0182 053C	0x0000 0007
L2CFG	RW	32	0x0004 0000	0x0184 0000	0x0100 0000
L1PCFG	RW	32	0x0004 0020	0x0184 0020	0x0000 0007
L1PCC	RW	32	0x0004 0024	0x0184 0024	0x0000 0000
L1DCFG	RW	32	0x0004 0040	0x0184 0040	0x0000 0007
L1DCC	RW	32	0x0004 0044	0x0184 0044	0x0000 0000
CPUARBU	RW	32	0x0004 1000	0x0184 1000	0x0001 0010
IDMAARBU	RW	32	0x0004 1004	0x0184 1004	0x0000 0010
SDMAARBU	RW	32	0x0004 1008	0x0184 1008	0x0000 0001
UCARBU	RW	32	0x0004 100C	0x0184 100C	0x0000 0020
MDMAARBU	RW	32	0x0004 1010	0x0184 1010	0x0607 0000
CPUARBD	RW	32	0x0004 1040	0x0184 1040	0x0001 0010
IDMAARBD	RW	32	0x0004 1044	0x0184 1044	0x0000 0010
SDMAARBD	RW	32	0x0004 1048	0x0184 1048	0x0000 0001
UCARBD	RW	32	0x0004 104C	0x0184 104C	0x0000 0020
L2WBAR	W	32	0x0004 4000	0x0184 4000	0x0000 0000
L2WWC	RW	32	0x0004 4004	0x0184 4004	0x0000 0000
L2WIBAR	W	32	0x0004 4010	0x0184 4010	0x0000 0000
L2WIWC	RW	32	0x0004 4014	0x0184 4014	0x0000 0000
L2IBAR	W	32	0x0004 4018	0x0184 4018	0x0000 0000
L2IWC	RW	32	0x0004 401C	0x0184 401C	0x0000 0000
L1PIBAR	W	32	0x0004 4020	0x0184 4020	0x0000 0000
L1PIWC	RW	32	0x0004 4024	0x0184 4024	0x0000 0000
L1DWIBAR	W	32	0x0004 4030	0x0184 4030	0x0000 0000
L1DWIWC	RW	32	0x0004 4034	0x0184 4034	0x0000 0000
L1DWBAR	W	32	0x0004 4040	0x0184 4040	0x0000 0000
L1DWWC	RW	32	0x0004 4044	0x0184 4044	0x0000 0000
L1DIBAR	W	32	0x0004 4048	0x0184 4048	0x0000 0000
L1DIWC	RW	32	0x0004 404C	0x0184 404C	0x0000 0000
L2WB	RW	32	0x0004 5000	0x0184 5000	0x0000 0000
L2WBINV	RW	32	0x0004 5004	0x0184 5004	0x0000 0000
L2INV	RW	32	0x0004 5008	0x0184 5008	0x0000 0000
L1PINV	RW	32	0x0004 5028	0x0184 5028	0x0000 0000
L1DWB	RW	32	0x0004 5040	0x0184 5040	0x0000 0000
L1DWBINV	RW	32	0x0004 5044	0x0184 5044	0x0000 0000
L1DINV	RW	32	0x0004 5048	0x0184 5048	0x0000 0000
L2EDSTAT	RW	32	0x0004 6004	0x0184 6004	0x0000 0001
L2EDCMD	RW	32	0x0004 6008	0x0184 6008	0x0000 0001
L2EDADDR	RW	32	0x0004 600C	0x0184 600C	0x0000 0000
L2EDCPEC	RW	32	0x0004 6018	0x0184 6018	0x0000 0000
L2EDCNEC	RW	32	0x0004 601C	0x0184 601C	0x0000 0000
MDMAERR	RW	32	0x0004 6020	0x0184 6020	0x0000 0000
MDMAERRCLR	RW	32	0x0004 6024	0x0184 6024	0x0000 0000
L2EDCEN	RW	32	0x0004 6030	0x0184 6030	0x0000 001F
L1PEDSTAT	RW	32	0x0004 6404	0x0184 6404	0x0000 0000
L1PEDCMD	RW	32	0x0004 6408	0x0184 6408	0x0000 0000
L1PEDADDR	RW	32	0x0004 640C	0x0184 640C	0x0000 0000

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
MARk[2]	RW	32	0x0004 8000 + (0x4*k)	0x0184 8000 + (0x4*k)	See [3]
L2MPFAR	R	32	0x0004 A000	0x0184 A000	0x0000 0000
L2MPFSR	RW	32	0x0004 A004	0x0184 A004	0x0000 0000
L2MPFCR	RW	32	0x0004 A008	0x0184 A008	0x0000 0000
L2MPPAm[4]	RW	32	0x0004 A200 + (0x4*m)	0x0184 A200 + (0x4*m)	0x0000 FFFF
L1PMPFAR	R	32	0x0004 A400	0x0184 A400	0x0000 0000
L1PMPFSR	RW	32	0x0004 A404	0x0184 A404	0x0000 0000
L1PMPFCR	RW	32	0x0004 A408	0x0184 A408	0x0000 0000
L1PMPPA16	RW	32	0x0004 A640	0x0184 A640	0x0000 FFFF
L1PMPPA17	RW	32	0x0004 A644	0x0184 A644	0x0000 FFFF
L1PMPPA18	RW	32	0x0004 A648	0x0184 A648	0x0000 FFFF
L1PMPPA19	RW	32	0x0004 A64C	0x0184 A64C	0x0000 FFFF
L1PMPPA20	RW	32	0x0004 A650	0x0184 A650	0x0000 FFFF
L1PMPPA21	RW	32	0x0004 A654	0x0184 A654	0x0000 FFFF
L1PMPPA22	RW	32	0x0004 A658	0x0184 A658	0x0000 FFFF
L1PMPPA23	RW	32	0x0004 A65C	0x0184 A65C	0x0000 FFFF
L1PMPPA24	RW	32	0x0004 A660	0x0184 A660	0x0000 FFFF
L1PMPPA25	RW	32	0x0004 A664	0x0184 A664	0x0000 FFFF
L1PMPPA26	RW	32	0x0004 A668	0x0184 A668	0x0000 FFFF
L1PMPPA27	RW	32	0x0004 A66C	0x0184 A66C	0x0000 FFFF
L1PMPPA28	RW	32	0x0004 A670	0x0184 A670	0x0000 FFFF
L1PMPPA29	RW	32	0x0004 A674	0x0184 A674	0x0000 FFFF
L1PMPPA30	RW	32	0x0004 A678	0x0184 A678	0x0000 FFFF
L1PMPPA31	RW	32	0x0004 A67C	0x0184 A67C	0x0000 FFFF
L1DMPFAR	R	32	0x0004 AC00	0x0184 AC00	0x0000 0000
L1DMPFSR	RW	32	0x0004 AC04	0x0184 AC04	0x0000 0000
L1DMPFCR	RW	32	0x0004 AC08	0x0184 AC08	0x0000 0000
MPLK0	W	32	0x0004 AD00	0x0184 AD00	0x0000 0000
MPLK1	W	32	0x0004 AD04	0x0184 AD04	0x0000 0000
MPLK2	W	32	0x0004 AD08	0x0184 AD08	0x0000 0000
MPLK3	W	32	0x0004 AD0C	0x0184 AD0C	0x0000 0000
MPLKCMD	RW	32	0x0004 AD10	0x0184 AD10	0x0000 0000
MPLKSTAT	RW	32	0x0004 AD14	0x0184 AD14	0x0000 0002
L1DMPPA16	RW	32	0x0004 AE40	0x0184 AE40	0x0000 FFF6
L1DMPPA17	RW	32	0x0004 AE44	0x0184 AE44	0x0000 FFF6
L1DMPPA18	RW	32	0x0004 AE48	0x0184 AE48	0x0000 FFF6
L1DMPPA19	RW	32	0x0004 AE4C	0x0184 AE4C	0x0000 FFF6
L1DMPPA20	RW	32	0x0004 AE50	0x0184 AE50	0x0000 FFF6
L1DMPPA21	RW	32	0x0004 AE54	0x0184 AE54	0x0000 FFF6
L1DMPPA22	RW	32	0x0004 AE58	0x0184 AE58	0x0000 FFF6
L1DMPPA23	RW	32	0x0004 AE5C	0x0184 AE5C	0x0000 FFF6

(2) k = 0 to 255

(3) MAR0 = 0x0000 0001;
MAR1…MAR11 = 0x0000 0000;
MAR12…MAR15 = 0x0000 000D;
MAR16…MAR255 = 0x0000 000C.

(4) m = 0 to 31

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
L1DMPPA24	RW	32	0x0004 AE60	0x0184 AE60	0x0000 FFF6
L1DMPPA25	RW	32	0x0004 AE64	0x0184 AE64	0x0000 FFF6
L1DMPPA26	RW	32	0x0004 AE68	0x0184 AE68	0x0000 FFF6
L1DMPPA27	RW	32	0x0004 AE6C	0x0184 AE6C	0x0000 FFF6
L1DMPPA28	RW	32	0x0004 AE70	0x0184 AE70	0x0000 FFF6
L1DMPPA29	RW	32	0x0004 AE74	0x0184 AE74	0x0000 FFF6
L1DMPPA30	RW	32	0x0004 AE78	0x0184 AE78	0x0000 FFF6
L1DMPPA31	RW	32	0x0004 AE7C	0x0184 AE7C	0x0000 FFF6

5.4.2.2 DSP_ICFG Register Description

For bitfield descriptions of all registers which reside within the DSP_ICFG configuration address space, refer to the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.4.3 DSP_SYSTEM Registers

5.4.3.1 DSP_SYSTEM Register Summary

NOTE: While the DSP1_SYSTEM and DSP2_SYSTEM addresses are part of the device L3_MAIN memory space, **the DSP_SYSTEM addresses are visible only within the DSP_ICFG internal configuration space (visible only to C66x CPU and debug logic).**

Table 5-14. DSP_SYSTEM and DSP1_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_SYSTEM Physical Address DSP1 and DSP2 private	DSP1_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_REVISION	R	32	0x0000 0000	0x01D0 0000	0x40D0 0000
DSP_SYS_HWINFO	R	32	0x0000 0004	0x01D0 0004	0x40D0 0004
DSP_SYS_SYSCONFIG	RW	32	0x0000 0008	0x01D0 0008	0x40D0 0008
DSP_SYS_STAT	R	32	0x0000 000C	0x01D0 000C	0x40D0 000C
DSP_SYS_DISC_CONFIG	RW	32	0x0000 0010	0x01D0 0010	0x40D0 0010
DSP_SYS_BUS_CONFIG	RW	32	0x0000 0014	0x01D0 0014	0x40D0 0014
DSP_SYS_MMU_CONFIG	RW	32	0x0000 0018	0x01D0 0018	0x40D0 0018
DSP_SYS_IRQWAKEEN0	RW	32	0x0000 0020	0x01D0 0020	0x40D0 0020
DSP_SYS_IRQWAKEEN1	RW	32	0x0000 0024	0x01D0 0024	0x40D0 0024
DSP_SYS_DMAWAKEEN0	RW	32	0x0000 0030	0x01D0 0030	0x40D0 0030
DSP_SYS_DMAWAKEEN1	RW	32	0x0000 0034	0x01D0 0034	0x40D0 0034
DSP_SYS_EVTOUT_SET	RW	32	0x0000 0040	0x01D0 0040	0x40D0 0040
DSP_SYS_EVTOUT_CLR	RW	32	0x0000 0044	0x01D0 0044	0x40D0 0044
RESERVED	R	32	0x0000 0048	0x01D0 0048	0x40D0 0048
DSP_SYS_ERRINT_IRQSTATUS_RAW	RW	32	0x0000 0050	0x01D0 0050	0x40D0 0050
DSP_SYS_ERRINT_IRQSTATUS	RW	32	0x0000 0054	0x01D0 0054	0x40D0 0054
DSP_SYS_ERRINT_IRQENABLE_SET	RW	32	0x0000 0058	0x01D0 0058	0x40D0 0058
DSP_SYS_ERRINT_IRQENABLE_CLR	RW	32	0x0000 005C	0x01D0 005C	0x40D0 005C
DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW	RW	32	0x0000 0060	0x01D0 0060	0x40D0 0060
DSP_SYS_EDMAWAKE0_IRQSTATUS	RW	32	0x0000 0064	0x01D0 0064	0x40D0 0064

Table 5-14. DSP_SYSTEM and DSP1_SYSTEM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_SYSTEM Physical Address DSP1 and DSP2 private	DSP1_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_EDMAWAKE0_IRQENABLE_SET	RW	32	0x0000 0068	0x01D0 0068	0x40D0 0068
DSP_SYS_EDMAWAKE0_IRQENABLE_CLR	RW	32	0x0000 006C	0x01D0 006C	0x40D0 006C
DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW	RW	32	0x0000 0070	0x01D0 0070	0x40D0 0070
DSP_SYS_EDMAWAKE1_IRQSTATUS	RW	32	0x0000 0074	0x01D0 0074	0x40D0 0074
DSP_SYS_EDMAWAKE1_IRQENABLE_SET	RW	32	0x0000 0078	0x01D0 0078	0x40D0 0078
DSP_SYS_EDMAWAKE1_IRQENABLE_CLR	RW	32	0x0000 007C	0x01D0 007C	0x40D0 007C
RESERVED	R	32	0x0000 00E0	0x01D0 00E0	0x40D0 00E0
RESERVED	R	32	0x0000 00E4	0x01D0 00E4	0x40D0 00E4
DSP_SYS_HW_DBGOUT_SEL	RW	32	0x0000 00F8	0x01D0 00F8	0x40D0 00F8
DSP_SYS_HW_DBGOUT_VAL	R	32	0x0000 00FC	0x01D0 00FC	0x40D0 00FC

Table 5-15. DSP2_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_REVISION	R	32	0x0000 0000	0x4150 0000
DSP_SYS_HWINFO	R	32	0x0000 0004	0x4150 0004
DSP_SYS_SYSCONFIG	RW	32	0x0000 0008	0x4150 0008
DSP_SYS_STAT	R	32	0x0000 000C	0x4150 000C
DSP_SYS_DISC_CONFIG	RW	32	0x0000 0010	0x4150 0010
DSP_SYS_BUS_CONFIG	RW	32	0x0000 0014	0x4150 0014
DSP_SYS_MMU_CONFIG	RW	32	0x0000 0018	0x4150 0018
DSP_SYS_IRQWAKEEN0	RW	32	0x0000 0020	0x4150 0020
DSP_SYS_IRQWAKEEN1	RW	32	0x0000 0024	0x4150 0024
DSP_SYS_DMAWAKEEN0	RW	32	0x0000 0030	0x4150 0030
DSP_SYS_DMAWAKEEN1	RW	32	0x0000 0034	0x4150 0034
DSP_SYS_EVTOUT_SET	RW	32	0x0000 0040	0x4150 0040
DSP_SYS_EVTOUT_CLR	RW	32	0x0000 0044	0x4150 0044
RESERVED	R	32	0x0000 0048	0x4150 0048
DSP_SYS_ERRINT_IRQSTATUS_RAW	RW	32	0x0000 0050	0x4150 0050
DSP_SYS_ERRINT_IRQSTATUS	RW	32	0x0000 0054	0x4150 0054
DSP_SYS_ERRINT_IRQENABLE_SET	RW	32	0x0000 0058	0x4150 0058
DSP_SYS_ERRINT_IRQENABLE_CLR	RW	32	0x0000 005C	0x4150 005C
DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW	RW	32	0x0000 0060	0x4150 0060
DSP_SYS_EDMAWAKE0_IRQSTATUS	RW	32	0x0000 0064	0x4150 0064
DSP_SYS_EDMAWAKE0_IRQENABLE_SET	RW	32	0x0000 0068	0x4150 0068
DSP_SYS_EDMAWAKE0_IRQENABLE_CLR	RW	32	0x0000 006C	0x4150 006C
DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW	RW	32	0x0000 0070	0x4150 0070
DSP_SYS_EDMAWAKE1_IRQSTATUS	RW	32	0x0000 0074	0x4150 0074
DSP_SYS_EDMAWAKE1_IRQENABLE_SET	RW	32	0x0000 0078	0x4150 0078
DSP_SYS_EDMAWAKE1_IRQENABLE_CLR	RW	32	0x0000 007C	0x4150 007C
RESERVED	R	32	0x0000 00E0	0x4150 00E0
RESERVED	R	32	0x0000 00E4	0x4150 00E4
DSP_SYS_HW_DBGOUT_SEL	RW	32	0x0000 00F8	0x4150 00F8

Table 5-15. DSP2_SYSTEM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_HW_DBGOUT_VAL	R	32	0x0000 00FC	0x4150 00FC

5.4.3.2 DSP_SYSTEM Register Description

Table 5-16. DSP_SYS_REVISION

Address Offset	0x0000 0000		
Physical Address	0x01D0 0000 0x40D0 0000 0x4150 0000	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
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	0x-(1)

(1) TI Internal Data

Table 5-17. Register Call Summary for Register DSP_SYS_REVISION

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0] [1]

Table 5-18. DSP_SYS_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x01D0 0004 0x40D0 0004 0x4150 0004	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
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
INFO																												NUM			

Bits	Field Name	Description	Type	Reset
31:4	INFO	0x0: No configurable options in subsystem.	R	0x0
3:0	NUM	Instance Number Set by subsystem input. In a multi-DSP system, provides a unique/incrementing values for each DSP.	R	0x0

Table 5-19. Register Call Summary for Register DSP_SYS_HWINFO

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0] [1]

Table 5-20. DSP_SYS_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x01D0 0008 0x40D0 0008 0x4150 0008	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
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																							RESERVED	RESERVED		STANDBYMODE		IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Read returns 0.	R	0x00 0000
8	RESERVED	Reserved. User must write 0.	RW	0x0
7:6	RESERVED	Reserved. Read returns 0.	R	0x0
5:4	STANDBYMODE	0x0: FORCE_STANDBY This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the SAF asserts with minimal hardware condition the "STANDBY" status. It is the responsibility of the software to ensure that the SAF is in a correct quiet state before programming this mode. Additionally when in this mode, the SAF is not allowed to generate wakeup request. 0x1: NO_STANDBY This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the SAF in C66xOSS asserts the "STANDBY" status. 0x2: SMART_STANDBY default. C66xOSS generates the standby status based upon all hardware internal status, namely after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode, the SAF is not allowed to generate wakeup request. 0x3: SMART_STANDBY_WKUP Same as Smart-Standby. (C66xOSS generates the standby status based upon all hardware internal status, namely after having performed all hardware operations necessary to be in a correct quiet state). . Additionally when in this mode, the SAF is allowed to generate wakeup request	RW	0x2

Bits	Field Name	Description	Type	Reset
3:2	IDLEMODE	0x0: FORCE_IDLE This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the IAF acknowledges a request to go idle from the power manager with minimal hardware condition. It is the responsibility of the software to ensure that the IAF are in a correct quiet state before requesting a force-idle transition. Additionally when in this mode, the IAF is not allowed to generate any wakeup request. 0x1: NO_IDLE When in this mode, the IAF disregards any request to go idle from the power manager. 0x2: SMART_IDLE default. When in this mode, the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode, the IAF is not allowed to generate any wakeup request 0x3: SMARTIDLEWKUP When in this mode, the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state. Additionally when in this mode, the IAF is allowed to generate wakeup request.	RW	0x2
1:0	RESERVED	Reserved	R	0x0

Table 5-21. Register Call Summary for Register DSP_SYS_SYSCONFIG

DSP Subsystems Functional Description • DSP Power Management: [0] [1] [2] • DSP Input Interrupts: [3]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [4] [5]

Table 5-22. DSP_SYS_STAT

Address Offset	0x0000 000C		
Physical Address	0x01D0 000C 0x40D0 000C 0x4150 000C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is intended to provide indication to software (including a remote host) as to whether the DSP is able to enter a low power mode.		
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																										OCPI_DISC_STAT		RESERVED	TC1_STAT	TC0_STAT	C66X_STAT

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5:4	OCPI_DISC_STAT	L3_MAIN (OCP) Initiator(s) Disconnect Status Read 0x0 : OCP inititiator ports are disconnected Read 0x1 : OCP initiator ports are attempting to disconnect. Read 0x2 : OCP initiator ports are active, no request to disconnect is pending.	R	0x2
3	RESERVED		R	0
2	TC1_STAT	EDMA TC1 Status 0x0: IDLE 0x1: ACTIVE - Active, based on inverse of tptc1_mstandby	R	1
1	TC0_STAT	EDMA TC0 Status 0x0: IDLE 0x1: ACTIVE - Active, based on inverse of tptc0_mstandby	R	1
0	C66X_STAT	C66x Status 0x0: IDLE C66x core is idle 0x1: ACTIVE C66x core is active.	R	1

Table 5-23. Register Call Summary for Register DSP_SYS_STAT

DSP Subsystems Functional Description • DSP System and Device Level Settings of DSP EDMA: [0] [1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2] [3]

Table 5-24. DSP_SYS_DISC_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x01D0 0010 0x40D0 0010 0x4150 0010	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to manually disconnect the OCP busses.		
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																															OCPI_DISC

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	OCPI_DISC	OCP Initiator (on L3_MAIN) Disconnect request Read 0: Disconnect not in progress, or has completed. Write 0: No effect. Read 1: Disconnect request is in progress. Write 1: Request for OCP Initiator to disconnect and mask write byte enable signals.	RW	0

Table 5-25. Register Call Summary for Register DSP_SYS_DISC_CONFIG

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0] [1]

Table 5-26. DSP_SYS_BUS_CONFIG

Address Offset	0x0000 0014		
Physical Address	0x01D0 0014 0x40D0 0014 0x4150 0014	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register controls the burst and priority settings for the internal initiators.		
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	SDMA_PRI			RESERVED			NOPOSTOVERRIDE	RESERVED		SDMA_L2PRES		RESERVED		CFG_L2PRES		RESERVED		TC1_L2PRES		RESERVED		TC0_L2PRES		RESERVED		TC1_DBS		RESERVED		TC0_DBS	

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:28	SDMA_PRI	Sets the CBA/VBusM Priority for the DSP C66x CorePac SDMA port. Can typically be left at default value. 0x0 is highest, ..., 0x7 is lowest priority.	RW	0x4
27:25	RESERVED		R	0x0
24	NOPOSTOVERRIDE	OCP Posted Write vs Non-Posted Write override 0x0: MIX Posted writes are used for cacheable write transactions. Non-posted writes are used for non-cacheable write transactions. 0x1: NOPOST Non-posted writes are used exclusively.	RW	1
23:22	RESERVED		R	0x0
21:20	SDMA_L2PRES	OCP Target port L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect. 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
19:18	RESERVED		R	0x0
17:16	CFG_L2PRES	DSP CFG L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13:12	TC1_L2PRES	TC1 L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
11:10	RESERVED		R	0x0
9:8	TC0_L2PRES	TC0 L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect. 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
7:6	RESERVED		R	0x0
5:4	TC1_DBS	TC1 Default Burst size. 0x0: BYTE_16 - "16-Byte" bursts 0x1: BYTE_32 - "32-Byte" bursts 0x2: BYTE_64 - "64-Byte" bursts 0x3: BYTE_128 - "128-Byte" bursts	RW	0x3
3:2	RESERVED		R	0x0
1:0	TC0_DBS	TC0 Default Burst size 0x0: BYTE_16 - "16-Byte" bursts 0x1: BYTE_32 - "32-Byte" bursts 0x2: BYTE_64 - "64-Byte" bursts 0x3: BYTE_128 - "128-Byte" bursts	RW	0x3

Table 5-27. Register Call Summary for Register DSP_SYS_BUS_CONFIG

DSP Subsystems Functional Description
- DSP TMS320C66x CorePac: [0]
- DSP System and Device Level Settings of DSP EDMA: [1] [2]
- DSP NoC Arbitration: [3]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [4] [5]

Table 5-28. DSP_SYS_MMU_CONFIG

Address Offset	0x0000 0018		
Physical Address	0x01D0 0018 0x40D0 0018 0x4150 0018	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to enable the subsystem MMUs.		
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																			MMU1_ABORT	RESERVED			MMU0_ABORT	RESERVED			MMU1_EN	RESERVED			MMU0_EN

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	MMU1_ABORT	MMU1 Abort 0x0: NOABORT = Abort not requested. 0x1: ABORT = MMU abort requested. Can be used in case of page translation failure or lockup.	RW	0
11:9	RESERVED		R	0x0
8	MMU0_ABORT	MMU0 Abort 0x0: NOABORT = Abort not requested. 0x1: ABORT = MMU abort requested. Can be used in case of page translation failure or lockup.	RW	0
7:5	RESERVED		R	0x0
4	MMU1_EN	MMU1 Enable 0x0: DISABLED = MMU is disabled and the MMU IP is bypassed. 0x1: ENABLED = MMU is enabled. (The MMU mmrs (including Enable bit) need to be set in addition to this bit.)	RW	1
3:1	RESERVED		R	0x0
0	MMU0_EN	MMU0 Enable 0x0: DISABLED = MMU is disabled and the MMU IP is bypassed. 0x1: ENABLED = MMU is enabled. (The MMU mmrs (including Enable bit) need to be set in addition to this bit.)	RW	1

Table 5-29. Register Call Summary for Register DSP_SYS_MMU_CONFIG

DSP Subsystems Functional Description
- Routing MDMA Traffic through DSP MMU0: [0] [1]
- Routing EDMA Traffic thorugh DSP MMU1: [2] [3]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [4] [5]

Table 5-30. DSP_SYS_IRQWAKEEN0

Address Offset	0x0000 0020		
Physical Address	0x01D0 0020 0x40D0 0020 0x4150 0020	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global interrupt wakeup enable bit vector that defines which input interrupts are used to cause a wake from powerdown state (via the slave idle protocol). IRQWAKEEN0 is for interrupt inputs 63 thru 32, and IRQWAKEEN1 is for interrupt inputs 95 thru 64. Internal interrupts cannot cause a wake condition since in this state there is no guaranteed clock and all sub-modules should be in idle state. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding interrupt for wakeup). Reads of this register return the actual state of the enable 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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable bit vector for interrupt #n+32 0x0: DISABLE = Interrupt #n+32 disabled for wakeup 0x1: ENABLE = Interrupt #n+32 enabled for wakeup	RW	0x0

Table 5-31. Register Call Summary for Register DSP_SYS_IRQWAKEEN0

DSP Subsystems Functional Description • DSP Power Management: [0] [1] [2] [3] • DSP Input Interrupts: [4] [5]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [6] [7]

Table 5-32. DSP_SYS_IRQWAKEEN1

Address Offset	0x0000 0024		
Physical Address	0x01D0 0024 0x40D0 0024 0x4150 0024	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global interrupt wakeup enable bit vector that defines which input interrupts are used to cause a wake from powerdown state (via the slave idle protocol). IRQWAKEEN0 is for interrupt inputs 63 thru 32, and IRQWAKEEN1 is for interrupt inputs 95 thru 64. Internal interrupts cannot cause a wake condition since in this state there is no guaranteed clock and all sub-modules should be in idle state. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding interrupt for wakeup). Reads of this register return the actual state of the enable 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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable bit vector for interrupt #n+64 0x0: DISABLE = Interrupt #n+64 disabled for wakeup 0x1: ENABLE = Interrupt #n+64 enabled for wakeup	RW	0x0

Table 5-33. Register Call Summary for Register DSP_SYS_IRQWAKEEN1

DSP Subsystems Functional Description • DSP Power Management: [0] [1] [2] • DSP Input Interrupts: [3] [4]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [5] [6]

Table 5-34. DSP_SYS_DMAWAKEEN0

Address Offset	0x0000 0030		
Physical Address	0x01D0 0030 0x40D0 0030 0x4150 0030	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global dma event wakeup enable bit vector that defines which input dma events are used to cause a wake from powerdown state (via the slave idle protocol). DMAWAKEEN0 is for dma event inputs 31 thru 0, and DMAWAKEEN1 is for dma event inputs 63 thru 32. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding dma event for wakeup). Reads of this register return the actual state of the enable 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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable for event #n 0x0: DISABLE = Interrupt #n disabled for wakeup 0x1: ENABLE = Interrupt #n enabled for wakeup	RW	0x0

Table 5-35. Register Call Summary for Register DSP_SYS_DMAWAKEEN0

DSP Subsystems Functional Description
- DSP Power Management: [0] [1]
- DSP DMA Requests: [2] [3]
- DSP EDMA Wakeup Interrupt: [4]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [5] [6]

Table 5-36. DSP_SYS_DMAWAKEEN1

Address Offset	0x0000 0034		
Physical Address	0x01D0 0034 0x40D0 0034 0x4150 0034	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global dma event wakeup enable bit vector that defines which input dma events are used to cause a wake from powerdown state (via the slave idle protocol). DMAWAKEEN0 is for dma event inputs 31 thru 0, and DMAWAKEEN1 is for dma event inputs 63 thru 32. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding dma event for wakeup). Reads of this register return the actual state of the enable 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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable for event #n+32 0x0: DISABLE = Interrupt #n+32 disabled for wakeup 0x1: ENABLE = Interrupt #n+32 enabled for wakeup	RW	0x0

Table 5-37. Register Call Summary for Register DSP_SYS_DMAWAKEEN1

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP DMA Requests: [1]
- DSP EDMA Wakeup Interrupt: [2]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [3] [4]

Table 5-38. DSP_SYS_EVTOUT_SET

Address Offset	0x0000 0040		
Physical Address	0x01D0 0040 0x40D0 0040 0x4150 0040	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	These registers can be used to drive event outputs from the DSP subsystem to a desired state.		
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
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Output Event for event #n Write 0x00 0001: Drive output event #n high/1. Read 0x00 0000: Event #n is low/0. Read 0x00 0001 : Event #n is high/1. Write 0x00 0000 : No action.	RW	0x0

Table 5-39. Register Call Summary for Register DSP_SYS_EVTOUT_SET

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0] [1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2] [3]

Table 5-40. DSP_SYS_EVTOUT_CLR

Address Offset	0x0000 0044		
Physical Address	0x01D0 0044 0x40D0 0044 0x4150 0044	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	These registers can be used to drive event outputs from the DSP subsystem to a desired state.		
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
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Output Event for event #n Read 0x00 0000: Event #n is low/0. Write 0x00 0001: Drive output event #n low/0. Read 0x00 0001 : Event #n is high/1. Write 0x00 0000 : No action.	RW	0x0

Table 5-41. Register Call Summary for Register DSP_SYS_EVTOUT_CLR

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0] [1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2] [3]

Table 5-42. DSP_SYS_ERRINT_IRQSTATUS_RAW

Address Offset	0x0000 0050		
Physical Address	0x01D0 0050 0x40D0 0050 0x4150 0050	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
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																EVENT															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	EVENT	Settable raw status for event #n Read 0x00 0000 : No event pending Write 0x00 0000 : No action Read 0x00 0001: Event pending Write 0x00 0001: Set event (for debug)	RW	0x0

Table 5-43. Register Call Summary for Register DSP_SYS_ERRINT_IRQSTATUS_RAW

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0] [1] [2]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [3] [4]

Table 5-44. DSP_SYS_ERRINT_IRQSTATUS

Address Offset	0x0000 0054		
Physical Address	0x01D0 0054 0x40D0 0054 0x4150 0054	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		
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																EVENT															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	EVENT	Clearable, enabled status for event #n Read 0x00 0000 : No enabled event pending Write 0x00 0000 : No action Read 0x00 0001 : Enabled Event pending Write 0x00 0001 : Clear raw event	RW	0x0

Table 5-45. Register Call Summary for Register DSP_SYS_ERRINT_IRQSTATUS

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0] [1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2] [3]

Table 5-46. DSP_SYS_ERRINT_IRQENABLE_SET

Address Offset	0x0000 0058		
Physical Address	0x01D0 0058 0x40D0 0058 0x4150 0058	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
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																ENABLE															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	ENABLE	Enable for event #n Read 0x00 0000 : Interrupt disabled Write 0x00 0000 : No action Read 0x00 0001 : Interrupt enabled Write 0x00 0001 : Enable interrupt	RW	0x0

Table 5-47. Register Call Summary for Register DSP_SYS_ERRINT_IRQENABLE_SET

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0] [1] [2]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [3] [4]

Table 5-48. DSP_SYS_ERRINT_IRQENABLE_CLR

Address Offset	0x0000 005C		
Physical Address	0x01D0 005C 0x40D0 005C 0x4150 005C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
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																ENABLE															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	ENABLE	Enable for event #n Read 0x00 0000 : Interrupt disabled Write 0x00 0000 : No action Read 0x00 0001 : Interrupt enabled Write 0x00 0001 : Disable interrupt (i.e., clear ENABLEn bit)	RW	0x0

Table 5-49. Register Call Summary for Register DSP_SYS_ERRINT_IRQENABLE_CLR

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0] [1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2] [3]

Table 5-50. DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW

Address Offset	0x0000 0060		
Physical Address	0x01D0 0060 0x40D0 0060 0x4150 0060	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		

Table 5-50. DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW (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
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Settable raw status for event #n Read 0x0000 0000 : No event pending Write 0x0000 0001 : Set event (for debug) Read 0x0000 0001 : Event pending Write 0x0000 0000 : No action	RW	0x0

Table 5-51. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP EDMA Wakeup Interrupt: [1] [2]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [3] [4]

Table 5-52. DSP_SYS_EDMAWAKE0_IRQSTATUS

Address Offset	0x0000 0064		
Physical Address	0x01D0 0064 0x40D0 0064 0x4150 0064	Instance	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		
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
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable, enabled status for event #n Read 0x0000 0000 : No enabled event pending Write 0x0000 0001 : Clear raw event Read 0x0000 0001 : Enabled Event pending Write 0x0000 0000 : No action	RW	0x0

Table 5-53. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQSTATUS

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP EDMA Wakeup Interrupt: [1]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [2] [3]

Table 5-54. DSP_SYS_EDMAWAKE0_IRQENABLE_SET

Address Offset	0x0000 0068		
Physical Address	0x01D0 0068 0x40D0 0068 0x4150 0068	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Enable interrupt Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0000 0000

Table 5-55. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQENABLE_SET

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP EDMA Wakeup Interrupt: [1] [2]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [3] [4]

Table 5-56. DSP_SYS_EDMAWAKE0_IRQENABLE_CLR

Address Offset	0x0000 006C		
Physical Address	0x01D0 006C 0x40D0 006C 0x4150 006C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Disable interrupt (i.e., clear ENABLEn bit) Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0

Table 5-57. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQENABLE_CLR

DSP Subsystems Functional Description
- DSP EDMA Wakeup Interrupt: [0]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [1] [2]

Table 5-58. DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW

Address Offset	0x0000 0070		
Physical Address	0x01D0 0070 0x40D0 0070 0x4150 0070	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
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
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Settable raw status for event #n+32 Read 0x0000 0000: No event pending Write 0x0000 0001: Set event (for debug) Read 0x0000 0001: Event pending Write 0x0000 0000 : No action	RW	0x0

Table 5-59. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0] [1]

Table 5-60. DSP_SYS_EDMAWAKE1_IRQSTATUS

Address Offset	0x0000 0074		
Physical Address	0x01D0 0074 0x40D0 0074 0x4150 0074	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		
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
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable, enabled status for event #n+32 Read 0x0000 0000: No enabled event pending Write 0x0000 0001: Clear raw event Read 0x0000 0001: Enabled Event pending Write 0x0000 0000: No action	RW	0x0

Table 5-61. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQSTATUS

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0] [1]

Table 5-62. DSP_SYS_EDMAWAKE1_IRQENABLE_SET

Address Offset	0x0000 0078		
Physical Address	0x01D0 0078 0x40D0 0078 0x4150 0078	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n+32 Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Enable interrupt Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0

Table 5-63. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQENABLE_SET

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0] [1]

Table 5-64. DSP_SYS_EDMAWAKE1_IRQENABLE_CLR

Address Offset	0x0000 007C		
Physical Address	0x01D0 007C 0x40D0 007C 0x4150 007C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
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
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n+32 Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Disable interrupt (i.e., clear ENABLEn bit) Read 0x0000 0001: Interrupt enabled Write 0x0000 0000 : No action	RW	0x0

Table 5-65. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQENABLE_CLR

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0] [1]

Table 5-66. DSP_SYS_HW_DBGOUT_SEL

Address Offset	0x0000 00F8		
Physical Address	0x01D0 00F8 0x40D0 00F8 0x4150 00F8	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	THis register is used to select which group of internal signals are mapped to the hw_dbgout output bus.		
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																												GROUP			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	GROUP	Debug Group output control mux select 0x0 : Disabled, debug outputs driven to 0x0. 0x1 : G1 = select output group 1 0x2 : G2 = select output group 2 N: GN = select output group N	RW	0x0

Table 5-67. Register Call Summary for Register DSP_SYS_HW_DBGOUT_SEL

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0] [1]

Table 5-68. DSP_SYS_HW_DBGOUT_VAL

Address Offset	0x0000 00FC		
Physical Address	0x01D0 00FC 0x40D0 00FC 0x4150 00FC	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to read the value of the currently selected debug output group.		
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
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Read returns state of hw_dbgout bus	R	0x0

Table 5-69. Register Call Summary for Register DSP_SYS_HW_DBGOUT_VAL

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0] [1]

5.4.4 DSP_FW_L2_NOC_CFG Registers

This section covers the DSPSS level defined public firewall (DSP_MMU0, DSP_MMU1) and L2 interconnect (DSP_NoC) functional registers.

5.4.4.1 DSP_FW_L2_NOC_CFG Register Summary

NOTE: While the DSP1_FW_L2_NOC_CFG and DSP2_FW_L2_NOC_CFG addresses are part of the device L3_MAIN memory space, **the DSP_FW_L2_NOC_CFG addresses are visible only within the DSP_ICFG internal configuration space (visible only to C66x CPU and debug logic).**

Table 5-70. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_FW_L2_NOC_CFG DSP1 and DSP2 private	DSP1_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0	RW	32	0x0000 0000	0x01D0 3000	0x40D0 3000
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 0004	0x01D0 3004	0x40D0 3004
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL	RW	32	0x0000 0040	0x01D0 3040	0x40D0 3040
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 0088	0x01D0 3088	0x40D0 3088
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 008C	0x01D0 308C	0x40D0 308C
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1	RW	32	0x0000 0090	0x01D0 3090	0x40D0 3090
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1	RW	32	0x0000 0094	0x01D0 3094	0x40D0 3094
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1	RW	32	0x0000 0098	0x01D0 3098	0x40D0 3098
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1	RW	32	0x0000 009C	0x01D0 309C	0x40D0 309C
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0	RW	32	0x0000 1000	0x01D0 4000	0x40D0 4000
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 1004	0x01D0 4004	0x40D0 4004
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL	RW	32	0x0000 1040	0x01D0 4040	0x40D0 4040
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 1088	0x01D0 4088	0x40D0 4088
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 108C	0x01D0 408C	0x40D0 408C
DSPNOC_FLAGMUX_ID_COREID	R	32	0x0000 4000	0x01D0 7000	0x40D0 7000
DSPNOC_FLAGMUX_ID_REVISIONID	R	32	0x0000 4004	0x01D0 7004	0x40D0 7004
DSPNOC_FLAGMUX_FAULTEN	RW	32	0x0000 4008	0x01D0 7008	0x40D0 7008
DSPNOC_FLAGMUX_FAULTSTATUS	R	32	0x0000 400C	0x01D0 700C	0x40D0 700C
DSPNOC_FLAGMUX_FLAGINEN0	RW	32	0x0000 4010	0x01D0 7010	0x40D0 7010
DSPNOC_FLAGMUX_FLAGINSTATUS0	R	32	0x0000 4014	0x01D0 7014	0x40D0 7014
DSPNOC_ERRORLOG_ID_COREID	R	32	0x0000 4200	0x01D0 7200	0x40D0 7200
DSPNOC_ERRORLOG_ID_REVISIONID	R	32	0x0000 4204	0x01D0 7204	0x40D0 7204
DSPNOC_ERRORLOG_FAULTEN	RW	32	0x0000 4208	0x01D0 7208	0x40D0 7208
DSPNOC_ERRORLOG_ERRVLD	R	32	0x0000 420C	0x01D0 720C	0x40D0 720C
DSPNOC_ERRORLOG_ERRCLR	RW	32	0x0000 4210	0x01D0 7210	0x40D0 7210
DSPNOC_ERRORLOG_ERRLOG0	R	32	0x0000 4214	0x01D0 7214	0x40D0 7214
DSPNOC_ERRORLOG_ERRLOG1	R	32	0x0000 4218	0x01D0 7218	0x40D0 7218
DSPNOC_ERRORLOG_ERRLOG3	R	32	0x0000 4220	0x01D0 7220	0x40D0 7220

Table 5-70. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_FW_L2_NOC_CFG DSP1 and DSP2 private	DSP1_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
DSPNOC_ERRORLOG_ERRLOG5	R	32	0x0000 4228	0x01D0 7228	0x40D0 7228

Table 5-71. DSP2_FW_L2_NOC_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0	RW	32	0x0000 0000	0x4150 3000
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 0004	0x4150 3004
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4150 3040
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 0088	0x4150 3088
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 008C	0x4150 308C
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1	RW	32	0x0000 0090	0x4150 3090
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1	RW	32	0x0000 0094	0x4150 3094
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1	RW	32	0x0000 0098	0x4150 3098
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1	RW	32	0x0000 009C	0x4150 309C
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0	RW	32	0x0000 1000	0x4150 4000
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 1004	0x4150 4004
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL	RW	32	0x0000 1040	0x4150 4040
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 1088	0x4150 4088
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 108C	0x4150 408C
DSPNOC_FLAGMUX_ID_COREID	R	32	0x0000 4000	0x4150 7000
DSPNOC_FLAGMUX_ID_REVISIONID	R	32	0x0000 4004	0x4150 7004
DSPNOC_FLAGMUX_FAULTEN	RW	32	0x0000 4008	0x4150 7008
DSPNOC_FLAGMUX_FAULTSTATUS	R	32	0x0000 400C	0x4150 700C
DSPNOC_FLAGMUX_FLAGINEN0	RW	32	0x0000 4010	0x4150 7010
DSPNOC_FLAGMUX_FLAGINSTATUS0	R	32	0x0000 4014	0x4150 7014
DSPNOC_ERRORLOG_ID_COREID	R	32	0x0000 4200	0x4150 7200
DSPNOC_ERRORLOG_ID_REVISIONID	R	32	0x0000 4204	0x4150 7204
DSPNOC_ERRORLOG_FAULTEN	RW	32	0x0000 4208	0x4150 7208
DSPNOC_ERRORLOG_ERRVLD	R	32	0x0000 420C	0x4150 720C
DSPNOC_ERRORLOG_ERRCLR	RW	32	0x0000 4210	0x4150 7210
DSPNOC_ERRORLOG_ERRLOG0	R	32	0x0000 4214	0x4150 7214
DSPNOC_ERRORLOG_ERRLOG1	R	32	0x0000 4218	0x4150 7218
DSPNOC_ERRORLOG_ERRLOG3	R	32	0x0000 4220	0x4150 7220
DSPNOC_ERRORLOG_ERRLOG5	R	32	0x0000 4228	0x4150 7228

5.4.4.2 DSP_FW_L2_NOC_CFG Register Description

Table 5-72. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0

Address Offset	0x0000 0000		
Physical Address	0x01D0 3000 0x40D0 3000 0x4150 3000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Error log 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				BLK_BURST_VIOLATION	RESERVED			REGION_START_ERRLOG				REGION_END_ERRLOG				REQINFO_ERRLOG																	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	BLK_BURST_VIOLATION	2D burst not allowed or exceeding allowed size	RW	0x0
26	RESERVED		R	0x0
25:21	REGION_START_ERRLOG	Wrong access hit this region number	RW	0x0
20:16	REGION_END_ERRLOG	Wrong access hit this region number	RW	0x0
15:0	REQINFO_ERRLOG	Error in reqinfo vector	RW	0x0

Table 5-73. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-74. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0

Address Offset	0x0000 0004		
Physical Address	0x01D0 3004 0x40D0 3004 0x4150 3004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Logical Physical Address Error log 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
RESERVED								SLVOFS_LOGICAL																							

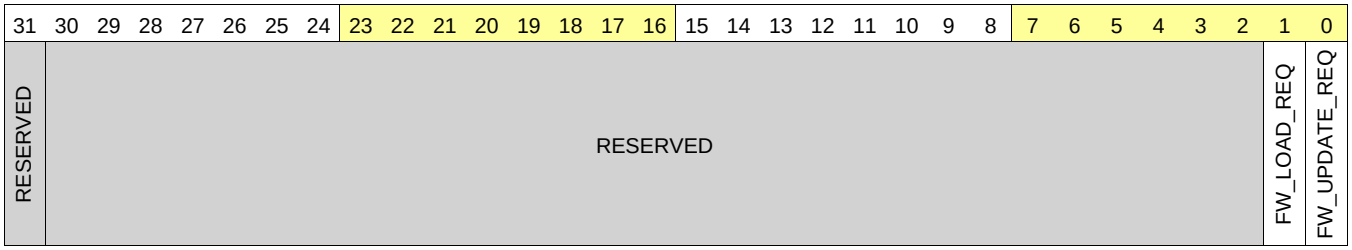
Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	SLVOFS_LOGICAL	Address generated by the ARM before being translated	R	0x0

Table 5-75. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0

DSP Subsystems Functional Description • DSP Public Firewall Settings: [0]
DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [1] [2]

Table 5-76. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL

Address Offset	0x0000 0040		
Physical Address	0x01D0 3040 0x40D0 3040 0x4150 3040	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Register update control register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:2	RESERVED	Reserved	R	0x0
1	FW_LOAD_REQ	HW set/SW clear	RW	0x1
0	FW_UPDATE_REQ	HW set/SW clear	RW	0x1

Table 5-77. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL

DSP Subsystems Functional Description • DSP Public Firewall Settings: [0]
DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [1] [2]

Table 5-78. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0

Address Offset	0x0000 0088		
Physical Address	0x01D0 3088 0x40D0 3088 0x4150 3088	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	MRM_PERMISSION_REGION_0_LOW 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																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 5-79. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0

DSP Subsystems Functional Description
- DSP Public Firewall Settings: [0]

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [1] [2]

Table 5-80. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0

Address Offset	0x0000 008C		
Physical Address	0x01D0 308C 0x40D0 308C 0x4150 308C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_HIGH 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
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1

Bits	Field Name	Description	Type	Reset
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 5-81. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0

DSP Subsystems Functional Description
- DSP Public Firewall Settings: [0]

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [1] [2]

Table 5-82. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1

Address Offset	0x0000 0090		
Physical Address	0x01D0 3090 0x40D0 3090 0x4150 3090	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Start physical address of region 1		
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																												START_REGION_1			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	START_REGION_1	Physical target start address of firewall region 1	RW	0x0

Table 5-83. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-84. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1

Address Offset	0x0000 0094		
Physical Address	0x01D0 3094 0x40D0 3094 0x4150 3094	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	End physical address of region 1		
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
END_REGION_1_ENABLE	RESERVED																							END_REGION_1							

Bits	Field Name	Description	Type	Reset
31	END_REGION_1_ENABLE	End Region 1 enable	RW	0x0
30:4	RESERVED		R	0x0
3:0	END_REGION_1	Physical target end address of firewall region 1	RW	0x0

Table 5-85. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-86. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1

Address Offset	0x0000 0098		
Physical Address	0x01D0 3098 0x40D0 3098 0x4150 3098	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_1_LOW 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																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 5-87. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-88. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1

Address Offset	0x0000 009C		
Physical Address	0x01D0 309C 0x40D0 309C 0x4150 309C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_1_HIGH 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
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1

Bits	Field Name	Description	Type	Reset
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 5-89. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-90. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0

Address Offset	0x0000 1000		
Physical Address	0x01D0 4000 0x40D0 4000 0x4150 4000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Error log 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				BLK_BURST_VIOLATION	RESERVED			REGION_START_ERRLOG								REGION_END_ERRLOG								REQINFO_ERRLOG								

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	BLK_BURST_VIOLATION	2D burst not allowed or exceeding allowed size	RW	0x0
26	RESERVED		R	0x0
25:21	REGION_START_ERRLOG	Wrong access hit this region number	RW	0x0
20:16	REGION_END_ERRLOG	Wrong access hit this region number	RW	0x0
15:0	REQINFO_ERRLOG	Error in reqinfo vector	RW	0x0

Table 5-91. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-92. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0

Address Offset	0x0000 1004		
Physical Address	0x01D0 4004 0x40D0 4004 0x4150 4004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Logical Physical Address Error log 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
RESERVED								SLVOFS_LOGICAL																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	SLVOFS_LOGICAL	Address generated by the ARM before being translated	R	0x0

Table 5-93. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-94. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL

Address Offset	0x0000 1040		
Physical Address	0x01D0 4040 0x40D0 4040 0x4150 4040	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Register update control 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								RESERVED																						FW_LOAD_REQ	FW_UPDATE_REQ

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:2	RESERVED	Reserved	R	0x0
1	FW_LOAD_REQ	HW set/SW clear	RW	0x1
0	FW_UPDATE_REQ	HW set/SW clear	RW	0x1

Table 5-95. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-96. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0

Address Offset	0x0000 1088		
Physical Address	0x01D0 4088 0x40D0 4088 0x4150 4088	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_LOW 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																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 5-97. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0

DSP Subsystems Functional Description
- DSP Public Firewall Settings: [0]

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [1] [2]

Table 5-98. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

Address Offset	0x0000 108C		
Physical Address	0x01D0 408C 0x40D0 408C 0x4150 408C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_HIGH 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
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 5-99. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

DSP Subsystems Functional Description • DSP Public Firewall Settings: [0]
DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [1] [2]

Table 5-100. DSPNOC_FLAGMUX_ID_COREID

Address Offset	0x0000 4000		
Physical Address	0x01D0 7000 0x40D0 7000 0x4150 7000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0xff71d7
7:0	CORETYPEID	Field identifying the type of IP.	R	0xb

Table 5-101. Register Call Summary for Register DSPNOC_FLAGMUX_ID_COREID

DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-102. DSPNOC_FLAGMUX_ID_REVISIONID

Address Offset	0x0000 4004		
Physical Address	0x01D0 7004 0x40D0 7004 0x4150 7004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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	0x-(1)

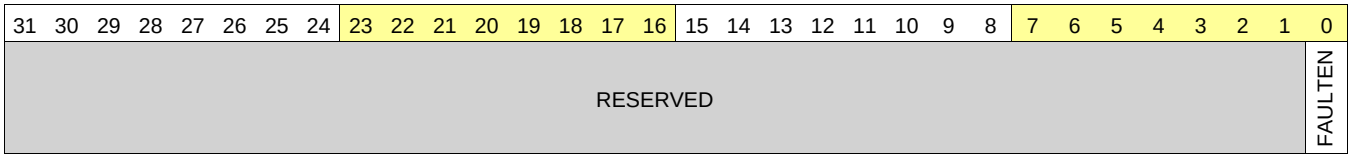
(1) TI internal data

Table 5-103. Register Call Summary for Register DSPNOC_FLAGMUX_ID_REVISIONID

DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-104. DSPNOC_FLAGMUX_FAULTEN

Address Offset	0x0000 4008		
Physical Address	0x01D0 7008 0x40D0 7008 0x4150 7008	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Global Fault Enable register	RW	0x1

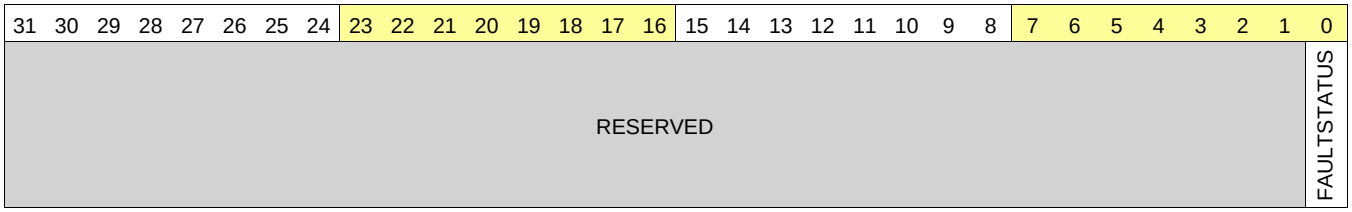
Table 5-105. Register Call Summary for Register DSPNOC_FLAGMUX_FAULTEN

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-106. DSPNOC_FLAGMUX_FAULTSTATUS

Address Offset	0x0000 400C		
Physical Address	0x01D0 700C 0x40D0 700C 0x4150 700C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTSTATUS	Global Fault Status register	R	0x0

Table 5-107. Register Call Summary for Register DSPNOC_FLAGMUX_FAULTSTATUS

DSP Subsystem Register Manual

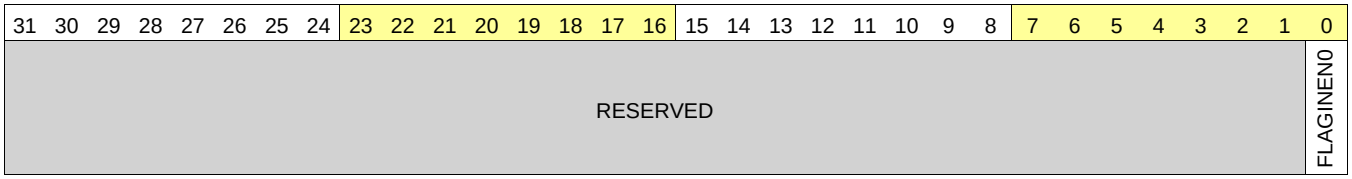
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-108. DSPNOC_FLAGMUX_FLAGINEN0

Address Offset	0x0000 4010		
Physical Address	0x01D0 7010 0x40D0 7010 0x4150 7010	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG

Table 5-108. DSPNOC_FLAGMUX_FLAGINEN0 (continued)

Description	
Type	RW



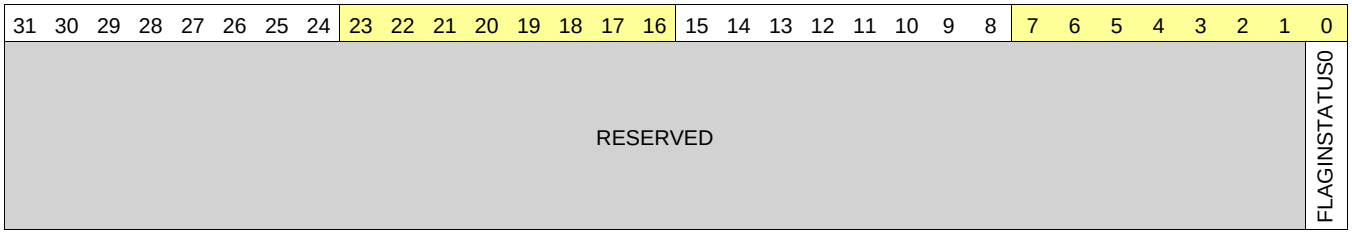
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINEN0	FlagIn Enable register #0	RW	0x1

Table 5-109. Register Call Summary for Register DSPNOC_FLAGMUX_FLAGINEN0

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-110. DSPNOC_FLAGMUX_FLAGINSTATUS0

Address Offset	0x0000 4014		
Physical Address	0x01D0 7014 0x40D0 7014 0x4150 7014	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINSTATUS0	FlagIn Status register #0	R	0x0

Table 5-111. Register Call Summary for Register DSPNOC_FLAGMUX_FLAGINSTATUS0

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-112. DSPNOC_ERRORLOG_ID_COREID

Address Offset	0x0000 4200		
Physical Address	0x01D0 7200 0x40D0 7200 0x4150 7200	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			

Table 5-112. DSPNOC_ERRORLOG_ID_COREID (continued)

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
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0xaf434
7:0	CORETYPEID	Field identifying the type of IP.	R	0xd

Table 5-113. Register Call Summary for Register DSPNOC_ERRORLOG_ID_COREID

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-114. DSPNOC_ERRORLOG_ID_REVISIONID

Address Offset	0x0000 4204		
Physical Address	0x01D0 7204 0x40D0 7204 0x4150 7204	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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	0x-(1)

(1) TI Internal data

Table 5-115. Register Call Summary for Register DSPNOC_ERRORLOG_ID_REVISIONID

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-116. DSPNOC_ERRORLOG_FAULTEN

Address Offset	0x0000 4208		
Physical Address	0x01D0 7208 0x40D0 7208 0x4150 7208	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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																															FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Enable Fault output	RW	0x1

Table 5-117. Register Call Summary for Register DSPNOC_ERRORLOG_FAULTEN

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-118. DSPNOC_ERRORLOG_ERRVLD

Address Offset	0x0000 420C		
Physical Address	0x01D0 720C 0x40D0 720C 0x4150 720C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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																															ERRVLD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRVLD	Error logged Valid	R	0x0

Table 5-119. Register Call Summary for Register DSPNOC_ERRORLOG_ERRVLD

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-120. DSPNOC_ERRORLOG_ERRCLR

Address Offset	0x0000 4210		
Physical Address	0x01D0 7210 0x40D0 7210 0x4150 7210	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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																															ERRCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRCLR	Clr ErrVld status	RW	0x0

Table 5-121. Register Call Summary for Register DSPNOC_ERRORLOG_ERRCLR

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-122. DSPNOC_ERRORLOG_ERRLOG0

Address Offset	0x0000 4214		
Physical Address	0x01D0 7214 0x40D0 7214 0x4150 7214	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Header: Lock, Opcode, Len1, ErrCode values		
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
FORMAT	RESERVED							LEN1								RESERVED					ERRCODE			RESERVED			OPC				LOCK

Bits	Field Name	Description	Type	Reset
31	FORMAT	Format of ErrLog0 register	R	0x1
30:28	RESERVED		R	0x0
27:16	LEN1	Header: Len1 value	R	0x0
15:11	RESERVED		R	0x0
10:8	ERRCODE	Header: Error Code value	R	0x0
7:5	RESERVED		R	0x0
4:1	OPC	Header: Opcode value	R	0x0
0	LOCK	Header: Lock bit value	R	0x0

Table 5-123. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG0

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-124. DSPNOC_ERRORLOG_ERRLOG1

Address Offset	0x0000 4218		
Physical Address	0x01D0 7218 0x40D0 7218 0x4150 7218	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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																ERRLOG1															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:0	ERRLOG1	Header: RouteId lsb value	R	0x0

Table 5-125. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG1

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-126. DSPNOC_ERRORLOG_ERRLOG3

Address Offset	0x0000 4220		
Physical Address	0x01D0 7220 0x40D0 7220 0x4150 7220	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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								ERRLOG3																							

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:0	ERRLOG3	Header: Addr lsb value	R	0x0

Table 5-127. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG3

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]

Table 5-128. DSPNOC_ERRORLOG_ERRLOG5

Address Offset	0x0000 4228		
Physical Address	0x01D0 7228 0x40D0 7228 0x4150 7228	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
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																ERRLOG5															

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:0	ERRLOG5	Header: User lsb value	R	0x0

Table 5-129. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG5

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0] [1]