

TMS320DM646x DMSoC DSP Subsystem

Reference Guide



Literature Number: SPRUEP8

December 2007

Preface	5
1 Introduction	7
2 TMS320C64x+ Megamodule	7
2.1 TMS320C64x+ CPU	9
2.2 Memory Controllers	10
2.3 Internal Peripherals	13
3 Memory Map	15
3.1 DSP Internal Memory (L1P, L1D, L2)	15
3.2 External Memory	15
3.3 ARM Internal Memory	16
3.4 Internal Peripherals	16
3.5 Device Peripherals	16
4 ARM–DSP Integration	16
5 DSP Subsystem Clock	18
6 Power Management	18
7 Boot and Reset	18

List of Figures

1	DSP Subsystem Block Diagram	8
2	C64x+ Cache Memory Architecture.....	11
3	ARM-DSP Integration	17

List of Tables

1	DSP Interrupt Map	13
2	External Memory Table	15
3	DSP Boot Configuration	18

Read This First

About This Manual

Describes the digital signal processor (DSP) subsystem in the TMS320DM646x Digital Media System-on-Chip (DMSoC). The DM646x DSP subsystem includes TI's standard TMS320C64x+™ Megamodule and several blocks of internal memory (L1P, L1D, and L2). This document provides an overview of the DSP subsystem and the following considerations associated with it:

- Memory mapping
- Interrupts
- ARM–DSP integration
- DSP subsystem clocking
- Boot and reset
- Power management

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
 - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
 - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

The following documents describe the TMS320DM646x Digital Media System-on-Chip (DMSoC). Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

The current documentation that describes the DM646x DMSoC, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

[SPRUEP9](#) — *TMS320DM646x DMSoC ARM Subsystem Reference Guide*. Describes the ARM subsystem in the TMS320DM646x Digital Media System-on-Chip (DMSoC). The ARM subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the DSP subsystem and a majority of the peripherals and external memories.

[SPRUEQ0](#) — *TMS320DM646x DMSoC Peripherals Overview Reference Guide*. Provides an overview and briefly describes the peripherals available on the TMS320DM646x Digital Media System-on-Chip (DMSoC).

[SPRAA84](#) — *TMS320C64x to TMS320C64x+ CPU Migration Guide*. Describes migrating from the Texas Instruments TMS320C64x digital signal processor (DSP) to the TMS320C64x+ DSP. The objective of this document is to indicate differences between the two cores. Functionality in the devices that is identical is not included.

[SPRU732](#) — ***TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide***. Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C64x and TMS320C64x+ digital signal processors (DSPs) of the TMS320C6000 DSP family. The C64x/C64x+ DSP generation comprises fixed-point devices in the C6000 DSP platform. The C64x+ DSP is an enhancement of the C64x DSP with added functionality and an expanded instruction set.

[SPRU871](#) — ***TMS320C64x+ DSP Megamodule Reference Guide***. Describes the TMS320C64x+ digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

[SPRU656](#) — ***TMS320C6000 DSP Cache User's Guide***. Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C621x/C671x/C64x digital signal processors (DSPs) of the TMS320C6000 DSP family can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C621x/C671x/C64x DSPs is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.

[SPRU862](#) — ***TMS320C64x+ DSP Cache User's Guide***. Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C64x+ digital signal processor (DSP) of the TMS320C6000 DSP family can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C64x+ DSP is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.

TMS320DM646x DMSoC DSP Subsystem

1 Introduction

The TMS320DM646x DSP subsystem ([Figure 1](#)) includes TI's standard TMS320C64x+ Megamodule and several blocks of internal memory (L1P, L1D, and L2). This document provides an overview of the DSP subsystem and the following considerations associated with it:

- Memory mapping
- Interrupts
- ARM-DSP integration
- DSP subsystem clocking
- Boot and reset
- Power management

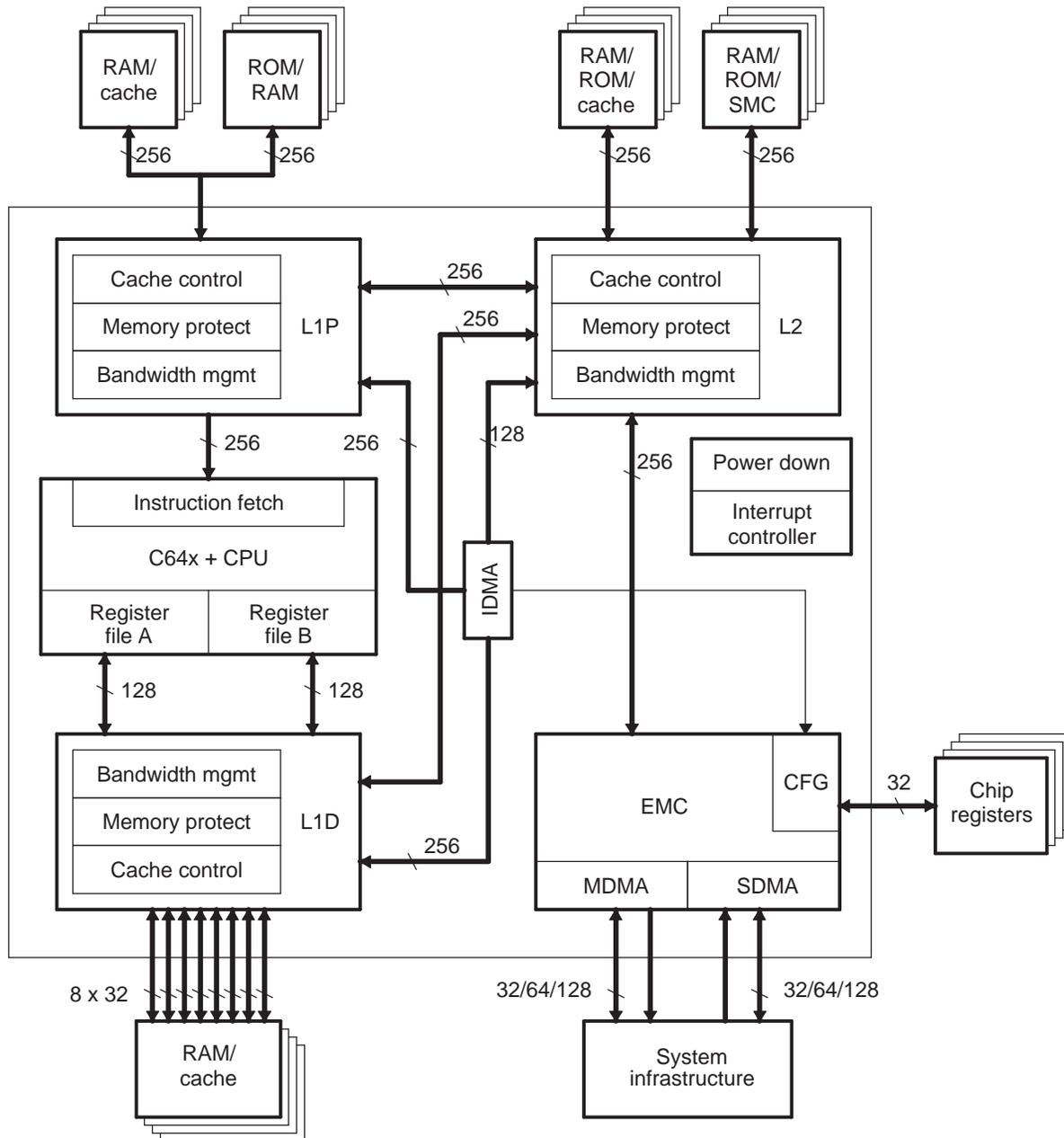
For more information, see the *TMS320C64x+ Megamodule Peripherals Reference Guide* ([SPRU871](#)), the *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* ([SPRU732](#)), and the *TMS320C64x+ DSP Cache User's Guide* ([SPRU862](#)).

2 TMS320C64x+ Megamodule

The C64x+ Megamodule consists of the following components:

- TMS320C64x+ CPU
- Internal memory controllers:
 - Program memory controller (PMC)
 - Data memory controller (DMC)
 - Unified memory controller (UMC)
 - External memory controller (EMC)
 - Internal direct memory access (IDMA) controller
- Internal peripherals
 - Interrupt controller
 - Power-down controller (PDC)

Figure 1. DSP Subsystem Block Diagram



2.1 TMS320C64x+ CPU

The C64x+ Megamodule includes the C64x+ CPU. The C64x+ CPU is a member of the TMS320C6000™ generation of devices. The C6000™ devices execute up to eight 32-bit instructions per cycle. The CPU consists of 64 general-purpose 32-bit registers and eight functional units. The eight functional units contain two multipliers and six ALUs. For more information on the CPU, see the *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* ([SPRU732](#)).

Features of the C6000 devices include:

- Advanced VLIW CPU with eight functional units, including two multipliers and six arithmetic units
 - 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 rapid development time
- Instruction packing
 - Gives code-size equivalence for eight instructions that execute 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 rapid development and improved parallelization
- 8/16/32-bit data support, providing efficient memory support for a variety of applications
- 40-bit arithmetic options add extra precision for vocoders and other computationally intensive applications
- Saturation and normalization provide support for key arithmetic operations
- Field manipulation and instruction extract, set, clear, and bit counting support a common operation found in control and data manipulation applications

The C64x+ devices include 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 nonaligned 32-bit (word) and 64-bit (double word) memory accesses
- Special communication-specific instructions 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
- Industry's first assembly optimizer for rapid development and improved parallelization

2.2 Memory Controllers

The C64x+ Megamodule implements a two-level internal cache-based memory architecture with external memory support. Level 1 memory is split into separate program memory (L1P memory) and data memory (L1D memory). [Figure 2](#) shows a diagram of the memory architecture. L1P and L1D are configurable as part L1 RAM (normal addressable on-chip memory) and part L1 cache. L1 memory is accessible to the CPU without stalls. Level 2 memory (L2) can also be split into L2 RAM (normal addressable on-chip memory) and L2 cache for caching external memory locations.

The following controllers manage RAM/cache configuration and cache data paths:

- Program memory controller (PMC)
- Data memory controller (DMC)
- Unified memory controller (UMC)
- External memory controller (EMC)

The Internal Direct Memory Access Controller (IDMA) manages DMA among the L1P, L1D, and L2 memories.

This section briefly describes the cache and DMA controllers. For detailed information about each of these controllers, see the *TMS320C6000 DSP Cache User's Guide* ([SPRU656](#)) and the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)).

NOTE: The C64x+ Megamodule includes the memory controllers; however, the physical L1P, L1D, and L2 memories are not part of the megamodule. Thus, they are described separately because the C64x+ Megamodule supports a variety of memory configurations. Refer to [Section 3](#) for more information on the L1P, L1D, and L2 memory configuration specific to the DM646x DMSoC.

2.2.1 Program Memory Controller (PMC)

The program memory controller (PMC) is the hardware interface between level 1 program memory (L1P memory) and the other components in the C64x+ Megamodule (for example, C64x+ CPU, UMC, and EMC). The PMC responds to instruction fetch requests from the C64x+ CPU and manages transfer operations between L1P memory and the UMC and between L1P memory and the EMC.

The DM646x DMSoC does not support the PMC memory protection feature of the standard C64x+ Megamodule.

Refer to the *TMS320C6000 DSP Cache User's Guide* ([SPRU656](#)) and to the program memory controller section of the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)) for more information on the PMC and for a description of its control registers.

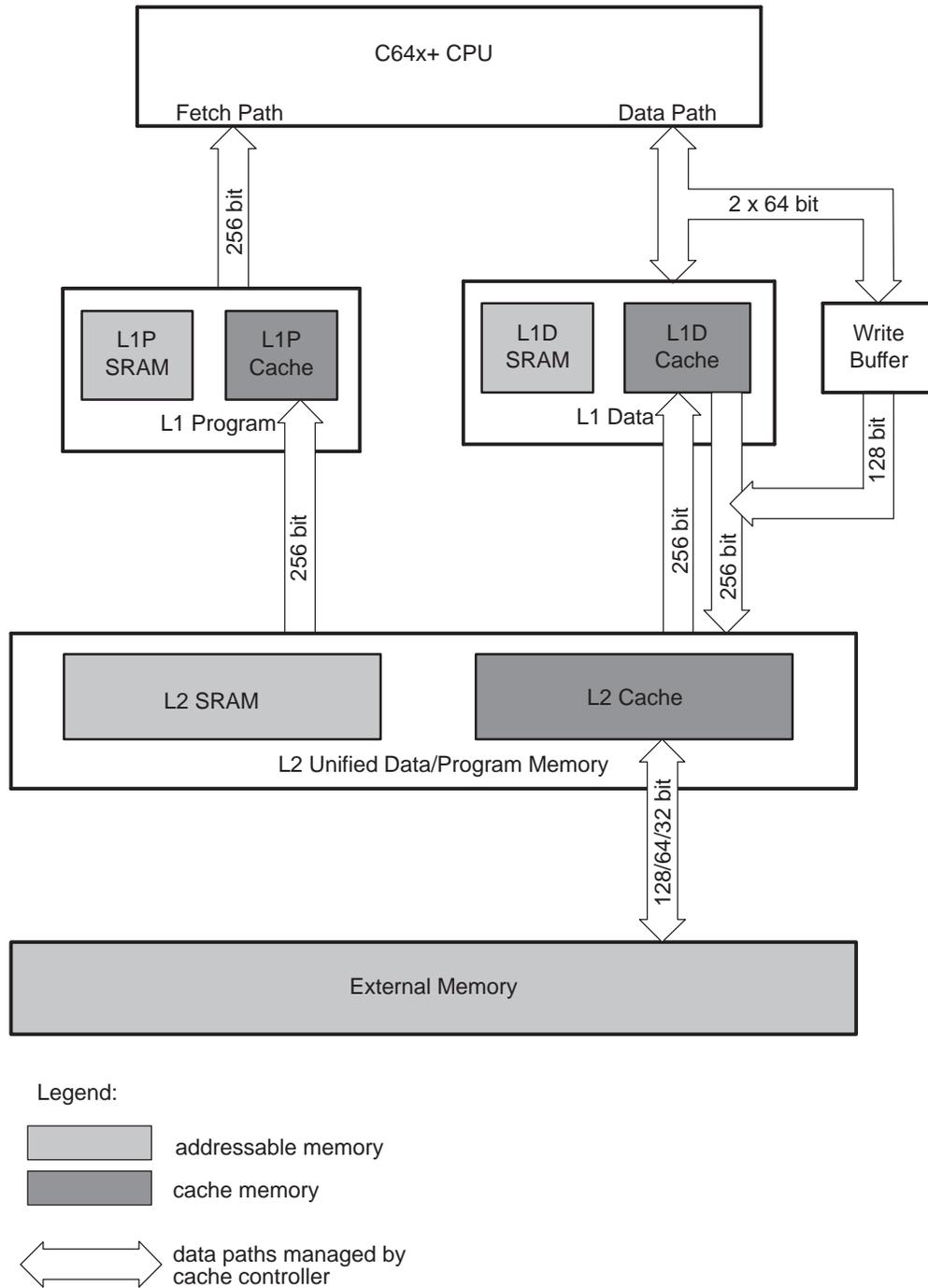
2.2.2 Data Memory Controller (DMC)

The data memory controller (DMC) is the hardware interface between level 1 data memory (L1D memory) and the other components in the C64x+ Megamodule (for example, C64x+ CPU, UMC, and EMC). The DMC responds to data requests from the C64x+ CPU and manages transfer operations between L1D memory and the UMC and between L1D memory and the EMC.

L1D memory includes 32 KB of RAM in the DM646x DMSoC. The DMC has a register interface that allows you to configure part of the L1D RAM as normal data RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, or 16 KB of the 32 KB of RAM. The DM646x DMSoC does not support the DMC memory protection features of the standard C64x+ Megamodule.

Refer to the *TMS320C6000 DSP Cache User's Guide* ([SPRU656](#)) and to the data memory controller (DMC) section of the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)) for more information on the DMC and for a description of its control registers.

Figure 2. C64x+ Cache Memory Architecture



2.2.3 Unified Controller (UMC)

The unified memory controller (UMC) is the hardware interface between level 2 memory (L2 memory) and the other components in the C64x+ Megamodule (for example, PMC, DMC, and EMC). The UMC manages transfer operations between L2 memory and the other memory controllers (PMC, DMC, and EMC).

L2 memory includes 128 KB of RAM in the DM646x DMSoC. The UMC has a register interface that allows you to configure part or all of the L2 RAM as normal RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, or 128 KB of the 128 KB of RAM.

The DM646x DMSoC does not support the UMC memory protection feature of the standard C64x+ Megamodule.

Refer to the *TMS320C6000 DSP Cache User's Guide* ([SPRU656](#)) and to the program unified controller section of the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)) for more information on the UMC and for a description of its control registers.

2.2.4 External Memory Controller (EMC)

The external memory controller (EMC) is the hardware interface between the external memory map (external memory and external registers) and the other controllers in the C64x+ Megamodule (for example, PMC, DMC, and UMC). The EMC manages transfer operations between external memory and registers and the other memory controllers (PMC, DMC, and EMC).

EMC does not support the UMC memory protection feature of the standard C64x+ Megamodule.

Refer to the *TMS320C6000 DSP Cache User's Guide* ([SPRU656](#)) and to the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)) for more information on the EMC and for a description of its control registers.

2.2.5 Internal DMA (IDMA)

The internal DMA (IDMA) controller facilitates DMA transfers between any two internal memory-mapped locations. Internal memory-mapped locations include L1P, L1D, L2, and internal peripheral configuration registers.

NOTE: The IDMA cannot facilitate DMA to or from external memory-mapped locations. The EDMA facilitates external DMA transfers. Refer to [Section 3](#) and to the *TMS320DM646x DMSoC Enhanced Direct Memory Access (EDMA) Controller User's Guide* ([SPRUEQ5](#)) for information on EDMA.

The IDMA controller enables the rapid paging of data sections to any local memory-mapped RAM. A key advantage of the IDMA is that it allows paging between slower L2 and faster L1D data memory. These transfers take place without CPU intervention and without cache stalls.

Another key advantage is that you can use the IDMA controller to program internal peripheral configuration registers without CPU intervention.

Refer to the internal DMA (IDMA) controller section in the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)) for more information on the IDMA controller and for a description of its control registers.

2.3 Internal Peripherals

This C64x+ Megamodule includes the following internal peripherals:

- Interrupt controller (INTC)
- Power-down controller (PDC)

This section briefly describes the INTC and PDC. For more information on these peripherals, see the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)).

2.3.1 Interrupt Controller (INTC)

The C64x+ Megamodule includes an interrupt controller (INTC) to manage DSP CPU interrupts. The INTC maps DSP device events into 12 prioritized interrupts. The source for each of the 12 CPU interrupts is user-programmable and is listed in [Table 1](#). All other interrupt numbers not listed in [Table 1](#) should be considered as reserved. The interrupt controller section of the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)) fully describes the INTC.

Table 1. DSP Interrupt Map

DSP Interrupt Number	Acronym	Source	DSP Interrupt Number	Acronym	Source
0	EVT0	C64x+ Interrupt controller 0	69	GPIO5	GPIO
1	EVT1	C64x+ Interrupt controller 1	70	GPIO6	GPIO
2	EVT2	C64x+ Interrupt controller 2	71	GPIO7	GPIO
3	EVT3	C64x+ Interrupt controller 3	84	CCINT1	EDMA CC Region 1
4	TINTLO	Timer 0 lower – TINT12	85	CCERRINT	EDMA CC Error
5	TINTH0	Timer 0 upper – TINT34	86	TCERRINT0	EDMA TC0 Error
6	TINTL1	Timer 1 lower – TINT12	87	TCERRINT1	EDMA TC1 Error
7	TINTH1	Timer 1 upper – TINT34	88	TCERRINT2	EDMA TC2 Error
9	EMU_DTDMA	C64x+ EMC	89	TCERRINT3	EDMA TC3 Error
11	EMU_RTDXRX	C64x+ RTDX	90	IDEINT	ATA
12	EMU_RTDXTX	C64x+ RTDX	96	INTERR	C64x+ Interrupt Controller Dropped CPU Interrupt Event
13	IDMAINT0	C64x+ EMC 0	97	EMC_IDMAERR	C64x+ EMC Invalid IDMA Parameters
14	IDMAINT1	C64x+ EMC 1	113	PMC_ED	C64x+ PMC
16	ARM2DSP0	ARM to DSP Controller 0	116	UMCED1	C64x+ UMC 1
17	ARM2DSP1	ARM to DSP Controller 1	117	UMCED2	C64x+ UMC 2
18	ARM2DSP2	ARM to DSP Controller 2	118	PDCINT	C64x+ PDC
19	ARM2DSP3	ARM to DSP Controller 3	119	SYSCMPA	C64x+ SYS
54	AXINT0	McASP 0 Transmit	120	PMCCMPA	C64x+ PMC
55	ARINT0	McASP 0 Receive	121	PMCDMPA	C64x+ PMC
56	AXINT1	McASP 1 Transmit	122	DMCCMPA	C64x+ DMC
64	GPIO0	GPIO	123	DMCDMPA	C64x+ DMC
65	GPIO1	GPIO	124	UMCCMPA	C64x+ UMC
66	GPIO2	GPIO	125	UMCDMPA	C64x+ UMC
67	GPIO3	GPIO	126	EMCCMPA	C64x+ EMC
68	GPIO4	GPIO	127	EMCBUSERR	C64x+ EMC

2.3.2 Power-Down Controller (PDC)

The C64x+ Megamodule includes a power-down controller (PDC). The PDC can power-down all of the following components of the C64x+ Megamodule and internal memories of the DSP subsystem:

- C64x+ CPU
- Program memory controller (PMC)
- Data memory controller (DMC)
- Unified memory controller (UMC)
- Extended memory controller (EMC)
- Internal Direct Memory Access controller (IDMA)
- L1P memory
- L1D memory
- L2 memory

The C64x Megamodule is capable of providing both dynamic and static power-down; however, only static power-down is supported on the DM646x DMSoC. The *TMS320C64x+ Megamodule Reference Guide (SPRU871)* describes the power-down control in more detail.

- Static power-down: The PDC initiates power down of the entire C64x+ Megamodule and all internal memories immediately upon command from software.

Static power-down affects all components of the C64x+ Megamodule and all internal memories. Software can initiate static power-down via a register bit in the PDC register. For more information on the PDC, see the *TMS320C64x+ Megamodule Reference Guide (SPRU871)*.

NOTE: The DM646x DMSoC does not support dynamic power-down.

2.3.3 Bandwidth Manager

The bandwidth manager provides a programmable interface for optimizing bandwidth among the requesters for resources, which include the following:

- EDMA-initiated DMA transfers (and resulting coherency operations)
- IDMA-initiated transfers (and resulting coherency operations)
- Programmable cache coherency operations
 - Block based coherency operations
 - Global coherency operations
- CPU direct-initiated transfers
 - Data access (load/store)
 - Program access

The resources include the following:

- L1P memory
- L1D memory
- L2 memory
- Configuration bus

Since any given requestor could potentially block a resource for extended periods of time, the bandwidth manager is implemented to assure fairness for all requesters.

The bandwidth manager implements a weighted-priority-driven bandwidth allocation. Each requestor (EDMA, IDMA, CPU, etc.) is assigned a priority level on a per-transfer basis. The programmable priority level has a single meaning throughout the system. There are a total of nine priority levels, where priority zero is the highest priority and priority eight is the lowest priority. When requests for a single resource contend, access is granted to the highest-priority requestor. When the contention occurs for multiple successive cycles, a contention counter assures that the lower-priority requestor gets access to the resource every 1 out of n arbitration cycles, where n is programmable. A priority level of -1 represents a transfer whose priority has been increased due to expiration of the contention counter or a transfer that is fixed as the highest-priority transfer to a given resource.

3 Memory Map

Refer to your device-specific data manual for memory-map information.

3.1 DSP Internal Memory (L1P, L1D, L2)

This section describes the configuration of the DSP internal memory in the DM646x DMSoC that consists of L1P, L1D, and L2. In the DM646x DMSoC:

- L1P memory includes 32 KB of RAM. The PMC allows you to configure part or all of the L1P RAM as normal program RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, 16 KB, or 32 KB of the 32 KB of RAM.
- L1D memory includes 32 KB of RAM. The DMC allows you to configure part of the L1D RAM as normal data RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, or 16 KB of the 32 KB of RAM.
- L2 memory includes 128 KB of RAM. The UMC allows you to configure part or all of the L2 RAM as normal RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, or 128 KB of the 128 KB of RAM.

3.2 External Memory

Devices connected to the DDR2 port and to the asynchronous EMIF are accessible to the DSP. Making the devices accessible to the DSP allows the DSP to access program memory and data memory from DDR2 on the DDR2 port and from devices attached to the asynchronous EMIF, such as NOR FLASH or SRAM. The EMC facilitates DSP access to these memories in the C64x+ Megamodule. The following external memories are accessible to the DSP:

- DDR2 port (8000:0000h–8FFF:FFFFh)
- Asynchronous EMIF (for example, NOR and NAND FLASH in 4 EM_CS regions)

NOTE: The DSP has access to the data space of the DDR2 port and asynchronous EMIF, but the DSP does not have access to the control registers of these EMIF controllers. The ARM is responsible for configuring the control registers of the DDR2 port and the asynchronous EMIF in the DM646x system. Refer to the *TMS320DM646x DMSoC ARM Subsystem Reference Guide* ([SPRUEP9](#)) for more information on how the ARM configures the DDR2 port and the asynchronous EMIF.

Table 2. External Memory Table

Region	Address			Accessibility		
	Start	End	Size	ARM	DSP	EDMA
ASYNC EMIF Data (EM_CS2)	4200 0000h	43FF FFFFh	32M	√	√	√
ASYNC EMIF Data (EM_CS3)	4400 0000h	45FF FFFFh	32M	√	√	√
ASYNC EMIF Data (EM_CS4)	4600 0000h	47FF FFFFh	32M	√	√	√
ASYNC EMIF Data (EM_CS5)	4800 0000h	49FF FFFFh	32M	√	√	√

3.3 ARM Internal Memory

ARM internal memory is accessible to the DSP and includes 32 KB of ARM internal RAM and 8 KB of ARM internal ROM.

NOTE: ARM internal memory is external to the C64x+ Megamodule. The EMC facilitates DSP access to ARM internal memory in C64x+ Megamodule for the DDR2 port and for asynchronous EMIF. Refer to the memory mapping section in the *TMS320DM646x DMSoC ARM Subsystem Reference Guide* ([SPRUJEP9](#)) for more information on the ARM's internal memory.

3.4 Internal Peripherals

The following internal peripherals are accessible to the DSP:

- Power-down controller (PDC)
- Interrupt controller

For more information on the internal peripherals, see the *TMS320C64x+ Megamodule Reference Guide* ([SPRU871](#)).

3.5 Device Peripherals

The following device peripherals (external to the C64x+ DSP Megamodule) are accessible to the DSP:

- McASP (multichannel audio serial port)
- EDMA (enhanced direct memory access)
- Timer0 and Timer1

NOTE: The McASP and EDMA peripherals are not in the DSP internal memory-map because these peripherals are not part of the standard C64x+ Megamodule, by design. Hence, such peripherals are occasionally described as external peripherals, with respect to the C64x+ Megamodule.

Refer to the *TMS320DM646x DMSoC Peripherals Overview Reference Guide* ([SPRUEQ0](#)) for more information on these peripherals.

4 ARM–DSP Integration

The DM646x DMSoC integrates an ARM core for overall system control functions and a DSP subsystem for complex data and image/video processing functions. [Figure 3](#) shows the interconnections between the ARM and the DSP cores and the shared resources. Both the ARM and the DSP have access to the EDMA, McASP, Timer0, and Timer1 peripherals. Both the ARM and DSP have access to several blocks of shared memory, including ARM internal memory, DSP internal memory, and external memory of the DDR2 memory controller and asynchronous EMIF (EMIF). The system control module includes registers that allow the ARM to interrupt the DSP and conversely allow the DSP to interrupt the ARM. The power and sleep controller (PSC) and the system control module (SYS) provide the ARM with a set of registers to boot the DSP, enable/disable the DSP clock, and reset the DSP.

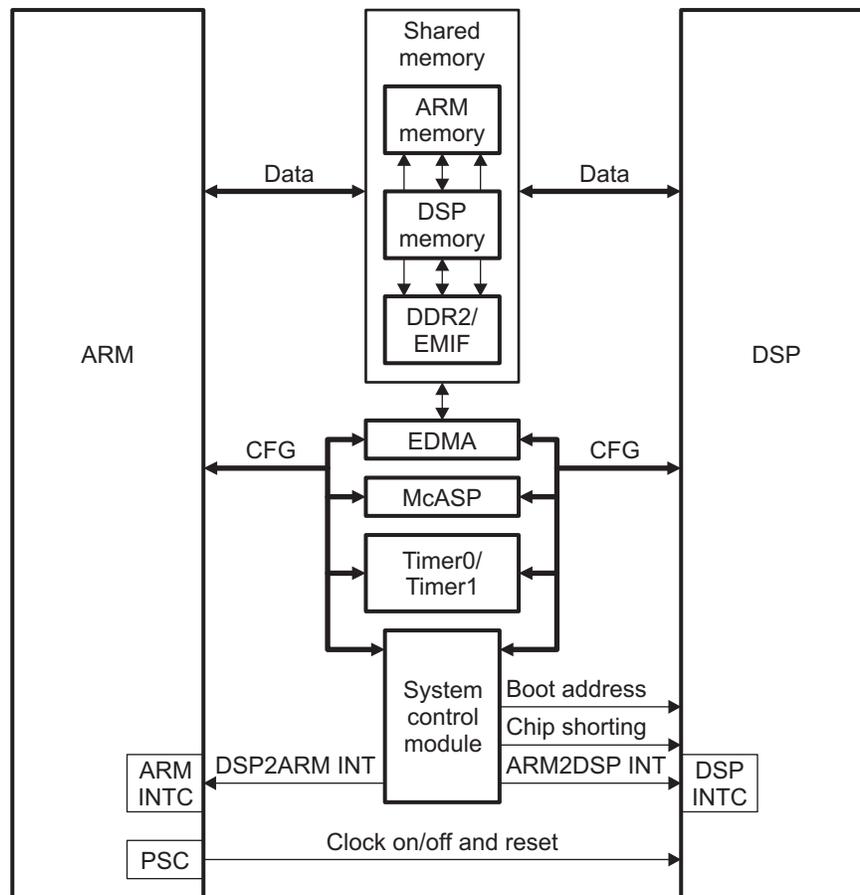
See the *TMS320DM646x DMSoC ARM Subsystem Reference Guide* ([SPRUJEP9](#)) for complete information on ARM–DSP integration.

The DM646x DMSoC includes the following features associated with ARM–DSP integration:

- Shared peripherals:
 - ARM and DSP have access to EDMA
 - ARM and DSP have access to McASP
 - ARM and DSP have access to Timer0 and Timer1

- Shared memory:
 - ARM has access to DSP internal memory (L1P, L1D, L2)
 - DSP has access to ARM internal memory
 - ARM and DSP have access to DDR2 memory controller and asynchronous EMIF
- ARM–DSP interrupts:
 - ARM can interrupt the DSP (via 4 general interrupts and 1 NMI)
 - DSP can interrupt the ARM (via 1 general interrupt)
- As system master, the ARM may manage the following DSP functions:
 - Boot the DSP
 - Enable/disable the DSP clock
 - Reset the DSP

Figure 3. ARM-DSP Integration



5 DSP Subsystem Clock

See the *TMS320DM646x DMSoC ARM Subsystem Reference Guide* ([SPRUEP9](#)) for complete information on DSP subsystem clocking.

PLL1 drives the DSP subsystem. The clock at device pin DEV_MXI drives the DSP in PLL bypass mode. Refer to the device-specific data manual for more information on supported clock speeds. The ARM controls turning the DSP clock on/off.

6 Power Management

See the *TMS320DM646x DMSoC ARM Subsystem Reference Guide* ([SPRUEP9](#)) for complete information on DSP subsystem power management.

- DSP clock can be completely shut off
- C64x Megamodule can be put in sleep mode (refer to [Section 2.3.2](#))
 - C64x+ CPU can be put in sleep mode
 - DSP internal memories (L1P, L1D, L2) can be statically put in sleep mode

7 Boot and Reset

The DSP can boot in either of two modes: ARM boots DSP mode or DSP self-boot mode.

- In the ARM boots DSP mode, the ARM is responsible for managing the DSP boot after power-on/reset.
- In the DSP self-boot mode, the DSP boots without ARM intervention immediately upon power-on/reset.

The boot mode is determined by sampling the DSPBOOT pin at power-on/reset ([Table 3](#)). Refer to the ARM–DSP Integration section of the *TMS320DM646x DMSoC ARM Subsystem Reference Guide* ([SPRUEP9](#)) for more information on boot and reset modes.

Table 3. DSP Boot Configuration

Device Configuration	Function	Sampled Pin	Default Setting
DSP boot	0 = ARM boots DSP 1 = DSP self-boots	DSPBOOT	ARM boots DSP