TMS320C6472/TMS320TCI6486 DSP Power/Sleep Controller (PSC)

User's Guide

Literature Number: SPRUEG3B

March 2007–Revised October 2009

List of Figures

List of Tables

Read This First

About This Manual

This document describes the power/sleep controller (PSC) in the TMS320TCI6486/TMS320C6472 DSPs.

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 C6000™ devices and related support tools. Copies of these documents are available on the Internet. *Tip:* Enter the literature number in the search box provided at www.ti.com.

SPRU189 — ***TMS320C6000 DSP CPU and Instruction Set Reference Guide.*** Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C6000 digital signal processors (DSPs).

SPRU198 — ***TMS320C6000 Programmer's Guide.*** Describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

SPRU301 — ***TMS320C6000 Code Composer Studio Tutorial.*** Introduces the Code Composer Studio™ integrated development environment and software tools.

SPRU321 — ***Code Composer Studio Application Programming Interface Reference Guide.*** Describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.

SPRU871 — ***TMS320C64x+ 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.

SPRUEG5 — ***TMS320C6472/TMS320TCI6486 Shared Memory Controller User's Guide.*** Describes the shared memory controller (SMC) for the TMS320C6472/TMS320TCI6486 digital signal processors (DSPs).

C6000, TMS320C6000, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

User's Guide
SPRUEG3B–March 2007–Revised October 2009

C6472/TCI6486 PSC

1 Introduction/Feature Overview

The power/sleep controller (PSC) in the TMS320TCI6486/TMS320C6472 device controls reset, power, and local clocking to modules. The PSC module consists of two sub-modules: Global PSC (GPSC) and Local PSC (LPSC).

In PSC, power saving modes do the following:

- Set modules to sleep and supply only minimal power to retain sleep mode. This is useful for faster wake up times.
- Turn off module clocks but keep the power domain. This is useful when only selective modules on the domain need to be clock gated.

The PSC has power-saving modes to sleep down the modules and turn off module clocks. For the TMS320TCI6486/TMS320C6472 device, setting the sleep down mode is handled by the GPSC while turning off module clocks is handled by the LPSC.

1.1 Features

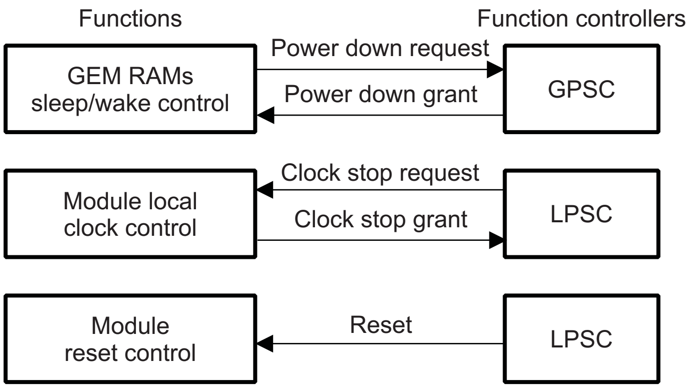
The PSC has these features:

- Provides 9 power domains (inclusive of the Always ON power domain) and allows sequencing when powering ON power domains to avoid current surge concerns when all domains turn ON at the same time.
- Provides 14 software controlled clock domains.
- Along with PLL controller, controls module clock/reset upon device power up.
- Intelligently handles soft and hard reset commands from PLL controller and propagates it to modules.
- Supports IcePick clock/reset control.

1.2 Functional Block Diagram

An overview of the PSC components and their functions is shown in Figure 1.

Figure 1. PSC Module Control



2 PSC Architecture

The PSC is in charge of power control through the power domain(s) sleep down and/or clock gating of modules. Power domains are configured at the chip level.

The Always ON power domain for a chip is powered on constantly while the chip is powered on.

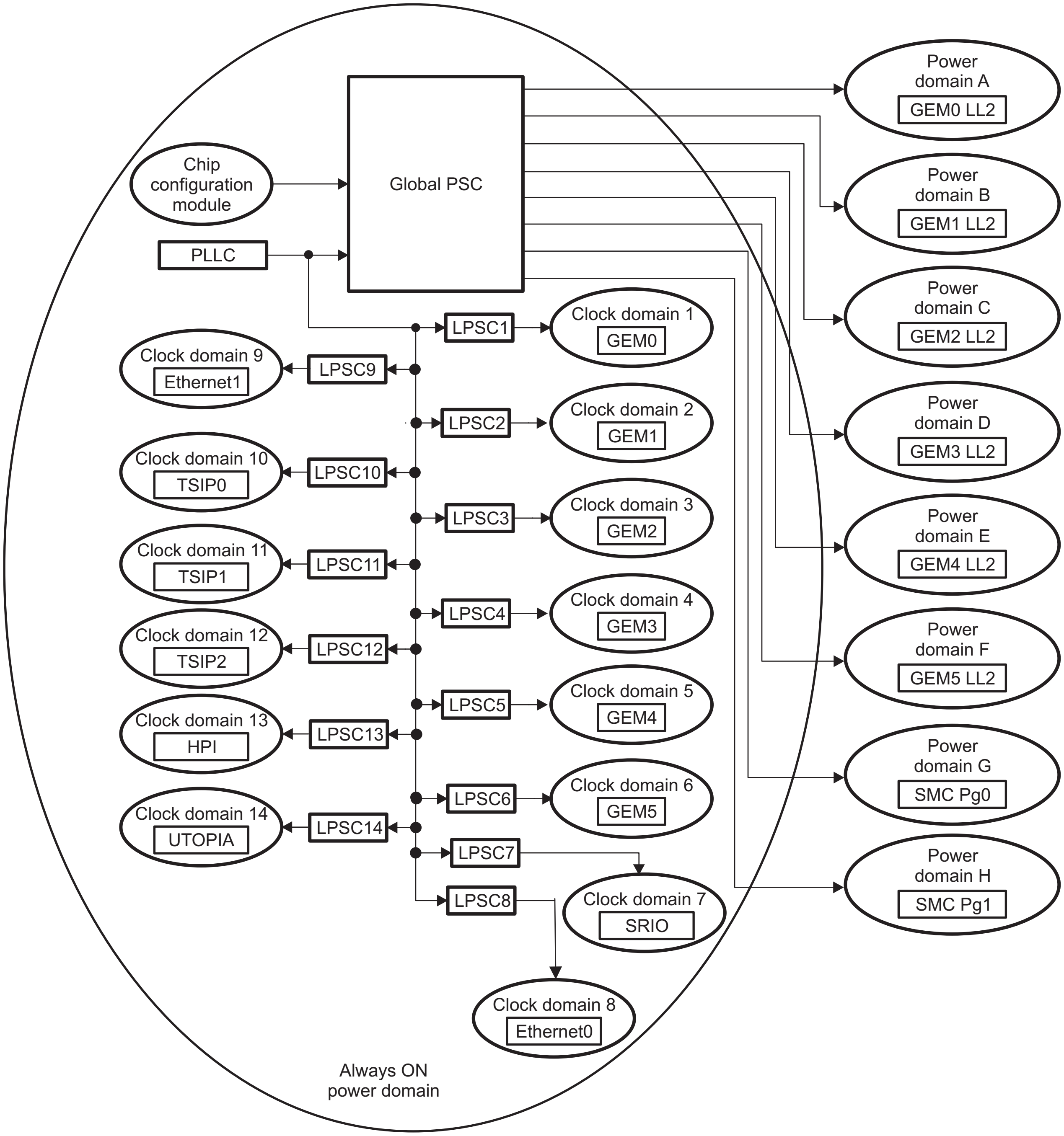
For normal operation, the PSC takes the modules out of reset and enables clocks to the modules. For reduced power consumption, the PSC gates clocks to the modules through the LPSC. For maximum power savings, the PSC sets power domains to sleep down through the GPSC.

2.1 Interface and Block Diagram

The PSC is interconnected with the PLL (phase-locked loop) controller, C64x+ Megamodules, and other modules. Its two main components are the GPSC and LPSC (local power/sleep controller).

Figure 2 shows the PSC components and the power and clock domains they control.

Figure 2. TMS320TCI6486/TMS320C6472 Power and Clock Domains



2.1.1 GPSC

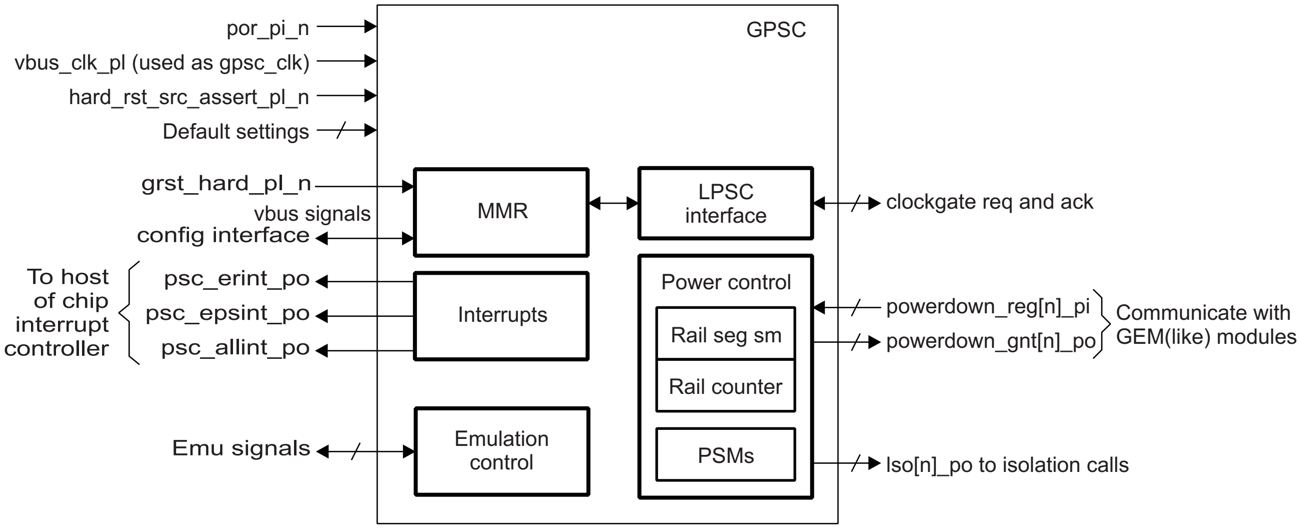
The TMS320TCI6486/TMS320C6472 device comprises several power domains to control minimizing power dissipation for unused modules/RAMs in the device. The GPSC manages each of the power domains, allowing any of them to be independently switched off from the power grid, removing all of the clocking and leakage power contribution by that block. The GPSC is located in the Always ON power domain. GPSC resolves module power down requests based on their priority and grants requests accordingly. GPSC also provides control signals to the local PSC (LPSC) to bring the module to its appropriate initial state.

The GPSC module contains the following:

- All memory-mapped registers (MMRs) for PSC
- A unique power state machine (PSM) for each power domain
- Control signals for power and isolation cells
- Power-up request and grant handshake
- Interrupt control

Figure 3 shows the functional block diagram of GPSC.

Figure 3. GPSC Block Diagram



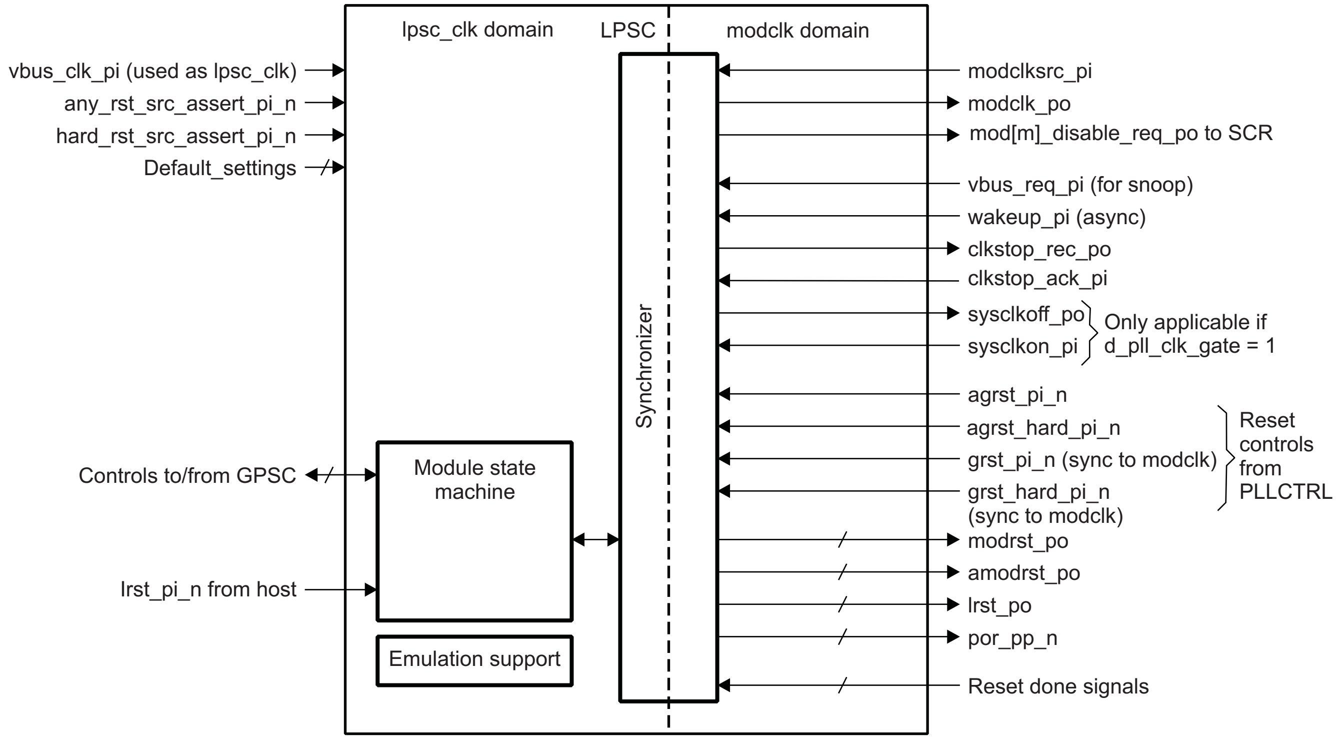
2.1.2 LPSC

The LPSC controls the reset and initial power-on states of the module, and gating and ungating of module clocks. For power up and reset sequencing, the LPSC receives instructions from the GPSC to assert/deassert and turn on/off clocks appropriately. For modules with multiple clock domains, only one LPSC is required for the module's clock domain that interfaces to a VBUS master. All module LPSCs are located in the Always ON power domain. Each LPSC communicates with the PLL controller(s) to enable or disable their module clock at the source.

The LPSC module contains the following:

- Clock control to modules on the VBUS interface side
- Reset control to modules
- Module state machine (MSM)
- Necessary synchronization to present the reset and clocks to the modules

Figure 4. LPSC Block Diagram



2.2 Methods of Power Control

The TMS320TCI6486/TMS320C6472 device includes the power/sleep controller (PSC) module to manage power and clocks to the various modules. The PSC allows software to gracefully gate clocks to any unused modules and to reassert clocks (and reset) these modules during device operation.

2.2.1 Hardware Control Method

The PSC does not actually control the L2 and SL2 power domain; instead the UMC and SMC are acting in a master role to power control the L2 and SL2 power domains, respectively. The PSC acts as a proxy counter to facilitate other modules in taking components into low power mode. In this case, the PSC does not actually control the component's power. The UMC or SMC is the master in controlling the component's power. The UMC or SMC simply makes use of the PSC's counter to grant the permission.

Memory retention is a consideration. Turning off the memory power achieves better power savings, but the benefit of putting the RAM in sleep state is that it can retain memory, and it gives a faster wakeup time. The C64x+ Megamodule directly controls the sleep/wake state of its RAM. The C64x+ Megamodule does not inform the PSC when it wants to put any RAM page to sleep. The C64x+ Megamodule only informs the PSC when the C64x+ Megamodule wants to wake up a RAM page, because it wants to rely on the PSC to perform arbitration to prevent power surge from waking up multiple pages or power domains on chip at the same time.

2.2.2 Software Control Method

LPSC can clock gate individual C64x+ Megamodules and other peripherals too. The clock gating can be done by software. The LPSC handshakes with C64x+ Megamodules or peripherals and will clock gate only if all internal components have ceased their operation before indicating to the PLL controller that the C64x+ Megamodule's clocks or peripheral's clocks may be shut off.

You can disable/enable the clock of the LPSC modules by programming the corresponding MDCTL register's NEXT field as a desired state and then setting the corresponding PTCMD register's GO field. The hardware goes through the required transition and initiates the process for clock disable/enable of the module and indicates to the PLL controller that the C64x+ Megamodule/peripherals clock has been disabled/enabled.

3 Memory Power Down Operational Details

The TMS320TCI6486/TMS320C6472 device is composed of six C64x+ Megamodule L2 power domains and two SL2 power domains.

L2 Memory

The TMS320TCI6486/TMS320C6472 device has 608K bytes of local L2 RAM that is divided into two pages: page 0 and page 1. The start and end addresses of page 0 and 1 are described in Table 1.

Table 1. L2 Memory Map Details

	Address		
Region	Start	End	Size
L2 Page 0	00800000	0087FFFF	512K
L2 Page 1	00880000	00897FFF	96K

By default L2 memories are in wake up state. C64x+ Megamodules can sleep down/wake up their own L2 memories. At the same time, C64x+ Megamodules can also wake up other C64x+ Megamodules L2 memories by accessing their memory pages, but cannot sleep another down.

SL2 Memory

The TMS320TCI6486/TMS320C6472 device has 768 Kbytes of shared L2 RAM (SL2 RAM) that is divided into two pages: page 0 and page 1. The start and end addresses of page 0 and 1 are described in Table 2.

Table 2. SL2 Memory Map Details

	Address		
Region	Start	End	Size
SL2 Page 0	00200000	0027FFFF	512K
SL2 Page 1	00280000	002BFFFF	256K

By default SL2 memories are in wakeup state. SL2 memory sleep/wakeup is controlled by the Shared Memory Controller (SMC). The SL2 memories are comprised of two power down pages. Since there are two pages, there is one memory power down controller per page.

All C64x+ Megamodules have to raise sleep down requests to the SMC to sleep down SL2. It can be woken up by any one of the C64x+ Megamodules. Programs can give permission to the SMC to put the physical page(s) to sleep, or to wake. When the SMC has the permission from all the C64x+ Megamodules connected to it then the SMC puts the page(s) to sleep. This allows the program to manually control what portions of SL2 memory are powered up or sleeping. C64x+ Megamodules can write to SL2PDWAKE/SL2PDSLEEP registers only in secure supervisor mode or supervisor mode and a read returns 0. If C64x+ Megamodules try to write these MMRs in secure user or user mode, then the SMC generates an exception back to that C64x+ Megamodule as well as to other C64x+ Megamodules connected to it. No exceptions are generated for the reads in any mode to these registers.

3.1 L2 Memory Sleep Down

To convert the logical pages of L2 memory into sleep mode, the software must set the appropriate L2PDSLEEP register bits. For instance, to sleep down the entire L2 memory, your program writes the value 0x0000 0003 (bits 1-0 set to 1) to L2PDSLEEP. Clearing bits in L2PDSLEEP has no effect on the corresponding logical pages. A page put to sleep can be woken up by accesses to addresses within that page. The page does not go back to sleep unless another L2PDSLEEP command is issued for it. UMC may drop sleep requests for a page if it is currently servicing accesses for that page, or if other requests for that page arrive within a short time window of receiving the sleep request. This prevents the UMC from storing the sleep request until all outstanding requests complete. Instead, programs should check L2PDSTAT to determine whether a given sleep request was honored.

See Section 9.9 for details on the L2PDSLEEP registers. See Section 9.10 for details on the L2PDSTAT registers.

3.2 L2 Memory Wakeup

You can wake up L2 pages individually by using either of these methods:

- Wake up register method
- Access L2 pages method

3.2.1 Wake Up Register (L2PDWAKE)

To wake logical pages in L2 memory, the software must set the appropriate L2PDWAKE register bits. Waking pages may incur a performance penalty (stall) for other requests to the UMC if the pages were actually asleep. For instance, to wake up the entire L2 memory, the program writes the value 0x0000 0003 (bits 1-0 set to 1) to L2PDWAKE. To wake an individual page of L2 memory, for instance page0, the program writes the value 0x0000 0001 (bit 0 set to 1) to L2PDWAKE. Similarly for page1, the program writes the value 0x0000 0002 (bit 1 set to 1) to L2PDWAKE. Clearing bits in the L2PDWAKE and L2PDSLEEP registers has no effect on the corresponding logical pages. Check L2PDSTAT to determine whether a given wake request was honored.

See Section 9.11 for details on the L2PDWAKE registers and Section 9.10 for details on the L2PDSTAT registers.

3.2.2 Accessing L2 Memory

By accessing L2 pages within their own boundary level, the corresponding page wakes up. On the C64x+ Megamodule accessing L2 page 0, page 0 wakes up and if access exceeds page 0 boundary and enters into page1, then page 1 also gets woken up. Pages do not enter sleep mode until the software sets them to sleep down. An access can be either read or write operation.

3.3 SL2 Memory Sleep Down

To give the permission to put physical pages of SL2 memory to sleep, the C64x+ Megamodule has to set the page-sleep bits in the appropriate SL2PDSLEEP register in the SMC. For instance, C64x+ Megamodule0 and C64x+ Megamodule1 decides to put SL2 page 1 to sleep then C64x+ Megamodule0 writes 0x00000008 (bit 3 set to 1) in its corresponding SL2PDSLEEP0 register and C64x+ Megamodule1 writes 0x00000008 (bit 3 set to 1) in its corresponding SL2PDSLEEP1 register. If all the C64x+ Megamodules gives permission to set SL2 page(s) to sleep, then the SMC sets the corresponding SL2 pages to sleep. That is, the SMC ANDs all the sleep down permissions and decides whether the page(s) are to sleep or not.

Clearing bits into SL2PDSLEEP has no effect on the corresponding physical pages. The sleep permission from that C64x+ Megamodule may or may not latch into the corresponding SL2PDSLEEP register. The C64x+ Megamodules that want memory to sleep checks the powerdown status (by reading the SL2PDSTAT register) periodically to make sure the page is asleep. If the page is not sleep, the C64x+ Megamodule has to set the appropriate bits in its SL2PDSLEEP register in the SMC again.

For details on the SL2PDSLEEP and SL2PDSTAT registers, see the *TMS320C6472/TMS320TCI6486 Shared Memory Controller User's Guide* (SPRUEG5).

3.4 SL2 Memory Wake Up

You can wake up SL2 pages individually by using either of these methods:

- Wake up register method
- Access SL2 pages method

3.4.1 Wake Up Register (SL2PDWAKE)

To give permission to put physical pages of SL2 memory to wake, the C64x+ Megamodule sets the page-sleep bits in the appropriate SL2PDWAKE register in the SMC. For instance, C64x+ Megamodule0 decides to wake SL2 page 1 then C64x+ Megamodule0 writes 0x00000008 (bit 3 set to 1) in its corresponding SL2PDWAKE0 register. If any of the C64x+ Megamodules gives permission to wake up SL2 page(s), then the SMC wakes the corresponding SL2 page(s). That is, the SMC ORs the wake up permissions and decides whether to wake the page(s) or not.

Clearing the SL2PDWAKE register bits has no effect on the corresponding physical pages. The wake permission from that C64x+ Megamodule may latch into the corresponding SL2PDWAKE register. The C64x+ Megamodule checks the power down status (by reading the SL2PDSTAT register) periodically to make sure the page is awake. If the page is not awake, the C64x+ Megamodule has to set the appropriate bits in its SL2PDWAKE register in the SMC again.

For details on the SL2PDWAKE registers see the *TMS320TCI6486/TMS320C6472 Shared Memory Controller User's Guide* (SPRUEG5).

3.4.2 Accessing SL2 Memory

By accessing SL2 pages within their own boundary level, the corresponding page is woken up. On the C64x+ Megamodule accessing SL2 page 0, page 0 wakes up and if access exceeds the page 0 boundary level and enters into the page1 boundary, then page 1 also wakes up. Pages do not return to sleep mode until the software or hardware sets them to sleep mode. An access can be a read or write operation.

4 Clock Gating Operational Details

Clocks to the C64x+ Megamodule and peripherals are controlled by corresponding LPSCs. Modules and their LPSCs are listed in Table 3. On your request, LPSC has to raise a clock stop request to its peripherals/C64x+ Megamodules. They should respond with the acknowledgement whether the peripheral/C64x+ Megamodule is in BUSY or IDLE state. Clock gating the C64x+ Megamodule and other peripherals is described in Section 4.1.

Table 3. Modules and Corresponding LPSCs

Module	Clock Domain	Module	Clock Domain
C64x+ Megamodule0	LPSC1	EMAC0	LPSC8
C64x+ Megamodule1	LPSC2	EMAC1	LPSC9
C64x+ Megamodule2	LPSC3	TSIP0	LPSC10
C64x+ Megamodule3	LPSC4	TSIP1	LPSC11
C64x+ Megamodule4	LPSC5	TSIP2	LPSC12
C64x+ Megamodule5	LPSC6	HPI	LPSC13
SRIO	LPSC7	UTOPIA	LPSC14

4.1 C64x+ Megamodule Clock Gating

LPSC does clock gate individual C64x+ Megamodules. The LPSC handshakes with the C64x+ Megamodule and checks whether all its internal components have ceased their operation before indicating to the PLL controller that the C64x+ Megamodule's clocks can be shut off. All C64x+ Megamodules cannot be clock gated at any point of time as at least one C64x+ Megamodule should be alive to enable the clock to other C64x+ Megamodules. C64x+ Megamodule local timers are clock gated as well when the corresponding C64x+ Megamodule is clock gated.

4.1.1 C64x+ Megamodule Idle

For C64x+ Megamodule to be clock gated, it has to execute IDLE instruction to get into the IDLE state. During IDLE only, the C64x+ Megamodule can respond to its LPSC's clock stop request. LPSC can raise a clock stop request to the C64x+ Megamodule module even if it is in busy, but it is acknowledged by the C64x+ Megamodule only when the C64x+ Megamodule is in IDLE. Before making the C64x+ Megamodule idle, we need to set PMC, DMC, EMC, UMC and GEMPD into sleep mode. To make this happen write 1555h into the PDCCMD register which is shown in Figure 5 and described in Table 4.

Figure 5. PDCCMD Register

31 17	16
Reserved	GEMPD
	R/W

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
EMCMEM	EMCLOG	UMCMEM	UMCLOG	DMCMEM	DMCLOG	PMCMEM	PMCLOG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 4. PDCCMD Register Field Descriptions

Bit	Field	Value	Description
31-17			Reserved
16	GEMPD		Power-down during idle
		0	Normal operation
		1	Sleep mode

Table 4. PDCCMD Register Field Descriptions (continued)

Bit	Field	Value	Description
15-14	EMCMEM		SRAM sleep modes
		00	No sleep mode is supported.
		01	Sleep mode
13-12	EMCLOG		Logic clock gating mode
		00	No clock gating is supported.
		01	Sleep mode
11-10	UMCMEM		SRAM sleep mode
		00	No sleep mode is supported.
		01	Sleep mode
9-8	UMCLOG		Logic clock gating mode
		00	No clock gating is supported.
		01	Static clock gating of unused module regions when the C64x+ Megamodule is active
7-6	DMCMEM		SRAM sleep mode
		00	No sleep mode is supported.
		01	Sleep mode
5-4	DMCLOG		Logic clock gating mode
		00	No clock gating is supported.
		01	Static clock gating of unused module regions when the C64x+ Megamodule is active
3-2	PMCMEM		SRAM sleep mode
		00	No sleep mode is supported.
		01	Sleep mode
1-0	PMCLOG		Logic clock gating mode
		00	No clock gating is supported.
		01	Static clock gating of unused module regions when the C64x+ Megamodule is active

4.1.2 Clock Disable to C64x+ Megamodule

To raise a clock stop request to C64x+ Megamodule0-5, you set the NEXT field to Disable in the LPSC1-6 module control registers (MDCTL1-6). LRSTZ in MDCTLx has to be set to have the reset de-asserted. You need to set the GO[0] bit of the PTCMD register which sets the bit PTSTAT.GOSTAT=1. When GOSTAT=1, the LPSC starts the transition. When transitions for all state machines on that domain are complete, PTSTAT.GOSTAT reverts to 0. For this, you write 102h into the corresponding MDCTL1-6 register and also write 1h into the PTCMD register to clock-gate C64x+ Megamodule0-5 respectively, but the clock is gated only if the C64x+ Megamodule is in IDLE at this time. You can read the status of all the C64x+ Megamodules from the corresponding LPSC MDSTAT register's STATUS field. The GO bit status can be read from the PTSTAT register.)

For further details, see Section 9.8 for the MDCTL1-6 registers, Table 5 for the MDSTAT register, Section 9.5 for the PTCMD register, and Section 9.6 for the PTSTAT register.

The LPSC1-6 module control registers (MDCTL1-6) NEXT fields have these possible values:

00000 — If a module doesn't need to have its clock running, chip team can default the module to SwRstDisable, where clocks are gated to the module and module reset is asserted. Before powering off a domain, PSC ensures that all modules are in SwRstDisable

00001 — The SyncReset State is not a useful run-time state. But it is a useful state for users wanting to reset a module. In this state, the module clocks are running but module is held in reset. When PSC is powering up a domain from Off to On, it ensures that all modules go through SyncRst to get properly reset.

00010 — A disabled module has its reset de-asserted (out of reset state) and its functional clocks gated. When a module gets disabled, its LPSC informs the infrastructure that it is disabled so new commands get routed to NULL end points.

00011 — An Enabled module has its reset de-asserted (out of reset state) and its functional clocks ticking. When a module gets enabled, its LPSC informs the infrastructure that it is enabled so commands get routed to the module.

The MDSTAT register STATE bit indicates the current module state. States with MSB (bit5) = 0 are user-defined states. States with MSB (bit 5) =1 are transitional states.

Table 5. State Definitions

User-Defined States	
Bit Value	**State**
000000	SwRstDisable
000001	SyncRst
000010	Disable
000011	Enable
000100 to 011111	Reserved
Transitional States	
100000	Disable In-Prog (waiting for clkstop_ack = 1)
100001	Disable In-Prog (Remove clkstop request)
100010	Disable In-Prog (Retry-give clkstop_req = 1 again)
100011	ClkOff (from DisableClkOn to Disable)
100100	Enable In-Prog
100101	Disable Clk On
100110	ClkOn (from Disable to DisableClkOn)
100111	ClkOn (from SwRstDisable to SyncRst)
101000	ClkOff (from SyncRst to SwRstDisable)
101001 to 111111	Reserved

4.1.3 Clock Enable to C64x+ Megamodule

Configuring the NEXT field to Enable in the MDCTLx Register along with the GO[0] bit in the PTCMD register enables the clock to the corresponding C64x+ Megamodule. C64x+ Megamodule0-5 does not start execution immediately when the corresponding clock returns, but the C64x+ Megamodule remains IDLE until any one of the interrupts takes out of IDLE. You can read the status of the C64x+ Megamodule clock from the corresponding LPSC MDSTAT register as described as in Section 4.1.2.

4.2 Peripheral Clock Gating

Peripheral clock gating is similar to C64x+ Megamodule clock gating. The only difference is LPSC does not wait for the peripheral to be in IDLE, except for TSIPs. The software has to give the proper guideline to LPSC to disable the clock to the peripherals. Peripherals respond immediately to its LPSC module whenever it asserts clock stop request, except for TSIPs. TSIP allows clock-gating only when it is disabled. In the TMS320TCI6486/TMS320C6472 device, all peripherals are controlled by separate LPSCs which are listed in Table 3. To clock-gate the peripherals, you have to write 102h into the MDCTLx register and also set the GO[0] bit in the PTCMD register. Similarly, to enable the clock to the peripherals, you have to write 103h into the MDCTL register and also set the GO[0] bit in PTCMD register. You can read the status of the peripherals state from the corresponding LPSC MDSTATx register's STATUS field and the GO bit status in the PTSTAT register.

5 Reset Strategy

The TMS320TCI6486/TMS320C6472 device can be taken to the default state partly or fully by hard and soft resets.

5.1 Hard Reset

The PSC does not have any control over hard reset assertion.

On hard reset or POR reset assertion the following occur:

- Power domains are reset to the off state (except for the always on domain).
- LPSC module domains are reset to the SwRstDisable state.

5.2 Soft Reset

LPSC can assert soft reset only to C64x+ Megamodules. To generate a local reset to C64x+ Megamodule0-5, you clear the LRSTz bit in the LPSC1-5 MDCTLx register. There is no soft reset assertion to peripherals.

On soft reset the following occur:

- Power domains are not reset.
- Module domains are reset to SwrstDisable.

6 Ice Pick Emulation Support

In the TMS320TCI6486/TMS320C6472 device, microprocessor cores implement emulation through JTAG interfaces that conform to the IEEE 1149.1 Test Access Port (TAP) protocols and requirements. The TMS320TCI6486/TMS320C6472 device has six cores with each associated TAP embedded in the core.

6.1 Operational Details

The IcePick power and clock controls affect the module clock domains in these ways:

- On assertion of InhibitSleep to the PSC from ICEPICK, prevents clock domains from leaving the enable state.
- On assertion of ForceActive, forces clock domains to enable state.

On assertion of the ForceActive signal from ICEPICK, LPSC sets PTSTAT.GOSTAT=1 and causes the internal module NEXT state to enable (without changing the actual MDCTLx.NEXT fields). As soon as the module domain reaches enable, PTSTAT.GOSTAT goes back to 0, regardless of the MDCTL.NEXT field you programmed. When ForceActive is de-asserted, PTSTAT.GOSTAT is set to 1 again and the state machine transitions to the desired location as indicated by the NEXT fields. If NEXT is already on/enable, then GOSTAT goes back low immediately.

See Section 9.6 for details on the PTSTAT register.

7 Error Interrupts

The PSC generates the following interrupts: PSC_ERRINT and PSC_ALLINT

7.1 Interrupt Conditions and Generation

An error interrupt is generated when either the MDCTLx register bit EMURSTIE is set along with MDSTATx register's EMURST bit, or the corresponding MDCTLx register's EMUIHBIE bit is set along with MDSTAT register, EMUIHB bit. The MDSTAT register is a read-only register and its EMURST/EMUIHB bit gets set only when the emulation reset/inhibit signal gets asserted to that particular module.

Once the pending register MERRPR is set, the interrupt remains set until it is explicitly cleared by setting the corresponding bit in the clear register MERRCR. The pending registers are read-only status registers that default to 0, while clearing registers are read/write status registers that default to 0.

As long as a bit is set in the pending register, the corresponding interrupt is generated. The internal interrupt pending signal remains high until all enabled interrupt conditions are cleared in all the relevant pending registers.

For further details, see Section 9.8 for the MDCTL1-6 registers, Table 5 for the MDSTAT register, Section 9.3 for the MERRPR register, and Section 9.4 for the MERRCR register.

7.2 Clearing Interrupts

Setting the clear register MERRCR bits clears the corresponding bits in the pending register MERRPR, but clearing the clear register bits have no effects. Reads from the clear registers always return zero.

If you issue MMR commands, where the first command is a write of 1 to the clear register and the second command is a read from the pending register, the corresponding bit in the pending register must be returned as cleared. If a new condition gets captured in pending register on the same time that it is being cleared via software (write to same bit in clear register), the incoming condition is given higher priority than the clear operation.

For further details, see Section 9.4 for the MERRCR register and Section 9.3 for the MERRPR register.

7.3 ALLINT

The all interrupt, ALLINT, is set along with the error interrupt, ERRINT. ALLINT retains its value until ERRINT is cleared.

7.4 Interrupt Evaluation and Set

The interrupt evaluation register (INTEVAL) is a command pseudo register. It has no actual storage space. Reads from INTEVAL always return zero. Write of 0 has no effect. Writes of 1 to the INTEVAL.EV bit cause the corresponding interrupt logic to re-evaluate all pending interrupts, and re-pulse the interrupt output (psc_errint_po, psc_allint_po) if any relevant interrupt conditions are pending.

For further details on INTEVAL, see Section 9.2.

8 Software Programming

These software sequences perform state transitions for L2 and SL2 memory power down and clock gating the C64x+ Megamodule and peripherals.

Example 1 illustrates psuedo code to power down and wake up the L2 memory.

To sleep down/wake up the L2 memory pages, C64x+ Megamodules can write into its Unified Memory Controller (UMC) L2 power domain sleep/wake registers, respectively. Each C64x+ Megamodule has UMC control and status registers. C64x+ Megamodules can wake up L2 pages separately (either page0 or page1) but C64x+ Megamodules cannot sleep down their L2 pages individually. Instead, C64x+ Megamodules need to set the entire L2 memory to sleep down by writing 3 (11b) into the L2PDSLEEP register, which doesn't have any effect while writing either 1(01b) or 2(10b). Be careful that there are no ongoing accesses with L2 memory from any other master/peripheral at this time.

Example 1. L2 Memory Power Down and Wake Up

```
/* By writing 3 (11b) into the UMC_L2PDSLEEP register, L2 memory is set to sleep down. */

     *UMC_L2PDSLEEP=0x00000003;

/* By writing 1 (01b) into the UMC_L2PDWAKE register, L2 memory page0 only is woken up. */

     *UMC_L2PDWAKE=0x00000001;

/* By writing 3 (11b) into the UMC_L2PDSLEEP register, L2 memory is set to sleep down. */

     *UMC_L2PDSLEEP=0x00000003;

/* By writing 2 (10b) into the UMC_L2PDWAKE register, L2 memory page1 only is woken up. */

     *UMC_L2PDWAKE=0x00000002;

/* By writing 3 (11b) into the UMC_L2PDSLEEP register, L2 memory is set to sleep down. */

     *UMC_L2PDSLEEP=0x00000003;

/* By writing 3 (11b) into the UMC_L2PDWAKE register, L2 memory page0 & page1 are wakened.*/

     *UMC_L2PDWAKE=0x00000003;

/* By writing 3 (11b) into the UMC_L2PDSLEEP register, L2 memory is set to sleep down. */

     *UMC_L2PDSLEEP=0x00000003;
```

Example 2 illustrates psuedo code to power down and wake up the SL2 memory.

To sleep down the SL2 memory pages, all C64x+ Megamodules have to write 0xC (11xxb) into the Shared Memory Controller (SMC) SL2 power domain sleep registers(SMC_GxSL2PDSLEEP). However, any one of the C64x+ Megamodules can wake up the SL2 memory pages by writing (0x4/0x8/0xC (page0/page1/both page0 and page1, respectively)) into the SMC SL2 power domain wake registers (SMC_GxSL2PDWAKE). Each C64x+ Megamodule has separate SMC control registers but only one common status register. C64x+ Megamodules can wake up/sleep down SL2 pages separately (either page0 or page 1). Be careful that there are no ongoing accesses with SL2 memory from any other masters/peripherals at this time.

Example 2. SL2 Memory Power Down and Wake Up

```
*/ C64x+ Megamodule0 writes its SMC wake register to wake SL2 page0 */

     *SMC_G0SL2PDWAKE=0x00000004;  // C64x+ Megamodule0 wakes up SL2 page0 only

*/ C64x+ Megamodule5 writes its SMC wake register to wake SL2 page1 */

     *SMC_G5SL2PDWAKE=0x00000008;  // C64x+ Megamodule5 wakes up SL2 page1 only


/* Now, SL2 pages should be woken up. SL2 page0/page1 were woken by C64x+ Megamodule0/C64x+
Megamodule5, respectively.
Check whether the SMC status register (SL2PDSTAT) was updated with the value of 0xC (11xxb). */

     while (*SMC_SL2PDSTAT != 0xC);

/* To sleep down shared memory pages, all C64x+ Megamodules have to update their own SL2PDSLEEP
register
with 0xC (11xxb) */

     *SMC_G0SL2PDSLEEP = (*SMC_G0SL2PDSLEEP | 0x00000004); // Because C64x+ Megamodule0 woke up only
page0
     *SMC_G5SL2PDSLEEP = (*SMC_G5SL2PDSLEEP | 0x00000008); // Because C64x+ Megamodule5 woke up only
page 1

/* C64x+ Megamodule can wake up both SL2 pages by writing into its SL2 wake register */

     *SMC_G3SL2PDWAKE=0x0000000C;

/* C64x+ Megamodule can sleep down both SL2 pages by writing into its SL2 sleep down register*/

     *SMC_G3SL2PDSLEEP=0x0000000C;
```

Example 3 illustrates psuedo code used to clock gate the C64x+ Megamodule and to enable it.

Example 3. C64x+ Megamodule Clock Gating

```
//To Clock gate C64x+ Megamodule, it should be in IDLE condition

//Taking C64x+ Megamodule to IDLE
asm ("IDLE");

//To disable clock to C64x+ Megamodule
//Next state is disable
CSL_FINS(hPsc->regs->MDCTL0, PSC_MDCTL0_NEXT, CSL_PSC_MDCTL0_NEXT_DISABLE);

//Enable bit in the appropriate power domain module
CSL_FINS(hPsc->regs->PTCMD, PSC_PTCMD_G0,(0x00000001 << 0));

//To enable clock to C64x+ Megamodule
//Next state is enable
CSL_FINS(hPsc->regs->MDCTL0, PSC_MDCTL0_NEXT, CSL_PSC_MDCTL0_NEXT_ENABLE);

//Enable bit in appropriate power domain module
CSL_FINS(hPsc->regs->PTCMD, PSC_PTCMD_G0,(0x00000001 << 0));
```

Example 4 illustrates pseudo code for the TSIPs clock gate and to enable it back.

Example 4. Peripheral Clock Gating--TSIP as an Example

```
//To disable clocks to TSIP0 assuming TSIP0 is in disabled state
//Next state is disable
CSL_FINS(hPsc->regs->MDCTL9, PSC_MDCTL9_NEXT, CSL_PSC_MDCTL9_NEXT_DISABLE);

//Enable bit in appropriate power domain module
CSL_FINS(hPsc->regs->PTCMD, PSC_PTCMD_G0,(0x00000001 << 0));

//TSIP0 gets clockgated

//To enable clocks to TSIP0

//Next state is enable
CSL_FINS(hPsc->regs->MDCTL9, PSC_MDCTL9_NEXT, CSL_PSC_MDCTL9_NEXT_ENABLE);

//De-Assert local reset
CSL_FINS(hPsc->regs->MDCTL9, PSC_MDCTL9_LRSTZ, CSL_PSC_MDCTL9_LRSTZ_DEASSERTLOCALRESET);

//Enable bit in appropriate power domain module
CSL_FINS(hPsc->regs->PTCMD, PSC_PTCMD_G0,(0x00000001 << 0));
```

9 Registers

The Global PSC (GPSC) includes a configuration slave interface providing access to its memory mapped registers (MMRs). This section lists all the registers and their memory map of the PSC.

Table 6. PSC Global Registers

Offset	Acronym	Register Description	Section
000h	PID	Peripheral ID Register	Section 9.1
018h	INTEVAL	Interrupt Evaluation Register	Section 9.2
040h	MERRPR	Module Error Pending Register	Section 9.3
050h	MERRCR	Module Error Clear Register	Section 9.4
120h	PTCMD	Power Transition Command Register	Section 9.5
128h	PTSTAT	Power Domain Transition Status Register	Section 9.6
800h	MDSTAT0	Module Status Register	Section 9.7
804h	MDSTAT1	Module Status Register	Section 9.7
808h	MDSTAT2	Module Status Register	Section 9.7
80Ch	MDSTAT3	Module Status Register	Section 9.7
810h	MDSTAT4	Module Status Register	Section 9.7
814h	MDSTAT5	Module Status Register	Section 9.7
818h	MDSTAT6	Module Status Register	Section 9.7
81Ch	MDSTAT7	Module Status Register	Section 9.7
820h	MDSTAT8	Module Status Register	Section 9.7
824h	MDSTAT9	Module Status Register	Section 9.7
828h	MDSTAT10	Module Status Register	Section 9.7
82Ch	MDSTAT11	Module Status Register	Section 9.7
830h	MDSTAT12	Module Status Register	Section 9.7
834h	MDSTAT13	Module Status Register	Section 9.7
A00h	MDCTL0	Module Control Register	Section 9.8
A04h	MDCTL1	Module Control Register	Section 9.8
A08h	MDCTL2	Module Control Register	Section 9.8
A0Ch	MDCTL3	Module Control Register	Section 9.8
A10h	MDCTL4	Module Control Register	Section 9.8
A14h	MDCTL5	Module Control Register	Section 9.8
A18h	MDCTL6	Module Control Register	Section 9.8
A1Ch	MDCTL7	Module Control Register	Section 9.8
A20h	MDCTL8	Module Control Register	Section 9.8
A24h	MDCTL9	Module Control Register	Section 9.8
A28h	MDCTL10	Module Control Register	Section 9.8
A2Ch	MDCTL11	Module Control Register	Section 9.8
A30h	MDCTL12	Module Control Register	Section 9.8
A34h	MDCTL13	Module Control Register	Section 9.8

9.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) is a constant register that contains the ID and ID revision number for that module. The PID stores version information used to identify the module. All bits within this register are read-only (writes have no effect) meaning that the values within this register should be hard-coded with the appropriate values and must not change from their hard-coded values.

The PID register is shown in Figure 6 and described in Table 7.

Figure 6. Peripheral Identification Register (PID)

31 30	29 28	27 26 16
SCHEME	Reserved	FUNC
R	R	R

15 11	10 8	7 6	5 0
RTL	MAJOR_REVISION	CUSTOM	MINOR_REVISION
R	R	R	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 7. Peripheral Identification Register (PID) Field Descriptions

Bit	Field	Value	Description
31-30	SCHEME	01	Distinguish between old scheme and current
29-28	Reserved	00	Reserved for future expansion
27-16	FUNC	0X2808	Indicating a software compatible module family
15-11	RTL	0	RTL Version
10-8	MAJOR_REVISION	001b	Major Revision
7-6	CUSTOM	00b	Indicating a Special Version for a particular device
5-0	MINOR_REVISION	000000b	Minor Revision

9.2 Interrupt Evaluation Register (INTEVAL)

The interrupt evaluation (INTEVAL) register has no storage. Reads from this register return 0. The INTEVAL register is shown in Figure 7 and described in Table 8.

Figure 7. Interrupt Evaluation Register (INTEVAL)

31 18	17	16
Reserved	ERRSET	ALLSET
R	R/W	R/W

15 2	1	0
Reserved	ERREV	ALLEV
R	R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 8. Interrupt Evaluation Register (INTEVAL) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved		
17	ERRSET		Error interrupt set
		0	No effect
		1	Causes the PSC_ERRINT to be pulsed
16	ALLSET		Combined interrupt set
		0	No effect
		1	Causes the PSC_ALLINT to be pulsed
15-2	Reserved		Reserved
1	ERREV		Re-evaluate error interrupt
		0	No effect
		1	Re-evaluate error interrupt PSC_ERRINT
0	ALLEV		Re-evaluate the combined PSC interrupt
		0	No effect
		1	Re-evaluate the combined PSC interrupt PSC_ALLINT

9.3 Module Error Pending Register (MERRPR)

The module error pending (MERRPR) register records pending error conditions for all modules. Each bit represents one module. The MERRPR register is shown in Figure 8 and described in Table 9.

Figure 8. Module Error Pending Register (MERRPR)

31	16
Reserved	
R	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
R		R	R	R	R	R	R	R	R	R	R	R	R	R	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9. Module Error Pending Register (MERRPR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved		Reserved
13	M13		Module 13 error condition
		0	Does not have the error condition
		1	Error condition exists
12	M12	[1]	Module 12 error condition
11	M11	[1]	Module 11 error condition
10	M10	[1]	Module 10 error condition
9	M9	[1]	Module 9 error condition
8	M8	[1]	Module 8 error condition
7	M7	[1]	Module 7 error condition
6	M6	[1]	Module 6 error condition
5	M5	[1]	Module 5 error condition
4	M4	[1]	Module 4 error condition
3	M3	[1]	Module 3 error condition
2	M2	[1]	Module 2 error condition
1	M1	[1]	Module 1 error condition
0	M0	[1]	Module 0 error condition

[1] See the M13 bit for the value descriptions.

9.4 Module Error Clear Register (MERRCR)

The module error clear (MERRCR) register has no storage. Reads from this register return 0. The MERRCR register is shown in Figure 9 and described in Table 10.

Figure 9. Module Error Clear Register (MERRCR)

31	16
Reserved	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10. Module Error Clear Register (MERRCR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved		Reserved
13	M13		Module 13 clear condition
		0	No effect
		1	Clears corresponding MERRPR bit
12	M12	[1]	Module 12 clear condition
11	M11	[1]	Module 11 clear condition
10	M10	[1]	Module 10 clear condition
9	M9	[1]	Module 9 clear condition
8	M8	[1]	Module 8 clear condition
7	M7	[1]	Module 7 clear condition
6	M6	[1]	Module 6 clear condition
5	M5	[1]	Module 5 clear condition
4	M4	[1]	Module 4 clear condition
3	M3	[1]	Module 3 clear condition
2	M2	[1]	Module 2 clear condition
1	M1	[1]	Module 1 clear condition
0	M0	[1]	Module 0 clear condition

[1] See the M13 bit for the values descriptions.

9.5 Power Transition Command Register (PTCMD)

The power transition command register (PTCMD) is a pseudo-command register with no actual storage. Reads return 0. There is one bit for each power domain.

The PTCMD register is shown in Figure 10 and described in Table 11.

Figure 10. Power Transition Command Register (PTCMD)

31	16
Reserved	
R	

15	9	8	7	6	5	4	3	2	1	0
Reserved		GO8	GO7	GO6	GO5	GO4	GO3	GO2	GO1	GO0
R		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 11. Power Transition Command Register (PTCMD) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved		Reserved
8	GO8		Power domain 8 GO transistion
		0	Not enabled
		1	Sets corresponding PTSTAT.GOSTAT field
7	GO7	[1]	Power domain 7 GO transition
6	GO6	[1]	Power domain 6 GO transition
5	GO5	[1]	Power domain 5 GO transition
4	GO4	[1]	Power domain 4 GO transition
3	GO3	[1]	Power domain 3 GO transition
2	GO2	[1]	Power domain 2 GO transition
1	GO1	[1]	Power domain 1 GO transition
0	GO0	[1]	Power domain 0 GO transition

[1] See the GO8 bit for the value descriptions.

9.6 Power Domain Transition Status (PTSTAT)

The power domain transition status (PTSTAT) register is a status register (read only). If a GOSTAT field is 1, you can determine in a module in the corresponding domain in undergoing a transistion by comparing MDSTAT.STATE field (see Section 9.7) to the MDCTL.NEXT field (see Section 9.8). If the values are not the same, the module is in transition.

The PTSTAT register is shown in Figure 11 and described in Table 12.

Figure 11. Power Domain Transition Status (PTSTAT)

31	16
Reserved	
R	

15 9	8	7	6	5	4	3	2	1	0
Reserved	GOSTAT8	GOSTAT7	GOSTAT6	GOSTAT5	GOSTAT4	GOSTAT3	GOSTAT2	GOSTAT1	GOSTAT0
R	R	R	R	R	R	R	R	R	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 12. Power Domain Transition Status (PTSTAT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved		Reserved
8	GOSTAT8		Power domain 8 transition command status
		0	No transitions in progress
		1	Transistion in progress
7	GOSTAT7	[1]	Power domain 7 transistion command status
6	GOSTAT6	[1]	Power domain 6 transistion command status
5	GOSTAT5	[1]	Power domain 5 transistion command status
4	GOSTAT4	[1]	Power domain 4 transistion command status
3	GOSTAT3	[1]	Power domain 3 transistion command status
2	GOSTAT2	[1]	Power domain 2 transistion command status
1	GOSTAT1	[1]	Power domain 1 transistion command status
0	GOSTAT0	[1]	Power domain 0 transistion command status

[1] See the GOSTAT8 bit for value descriptions.

9.7 Module Status Register (MDSTAT)

The module status MDSTAT(1-13) registers bit-fields show the status of each module. The MDSTAT register is shown in Figure 12 and described in Table 13.

Figure 12. Module Status Register (MDSTAT)

31 18	17	16
Reserved	EMUHIB	EMURST
R	R/W	R/W

15 13	12	11	10	9	8	7 6	5 0
Reserved	MCKOUT	MRSTDONE	MRSTZ	LRSTDONE	LRSTZ	Reserved	STATE
R	R/W	R/W	R/W	R/W	R/W	R	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 13. Module Status Register (MDSTAT) Field Descriptions

Bit	Field	Value	Description
31-18			Reserved
17	EMUHIB		Emulation alters module state
		0	No emulation
		1	Emulation alters your defined module state
16	EMURST		Emulation alters reset to module
		0	No emulation
		1	Emulation has altered module reset
15-13	Reserved		Reserved
12	MCKOUT		Actual module output to module
		0	Modclk gated
		1	Modclk on
11	MRSTDONE		Module reset initialization done status
		0	Reset initialization not done
		1	Reset initialization done
10	MRSTZ		Module reset actual status
		0	Asserted
		1	De-asserted
9	LRSTDONE		Module local reset initialization done status
		0	Local reset initialization not done
		1	Local reset initialization done
8	LRSTZ		Module local reset actual size
		0	Local reset asserted
		1	Local reset de-asserted
7-6	Reserved		Reserved

Table 13. Module Status Register (MDSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
5-0	STATE		Current module state
		00	SwRstDisable
		01	SyncRst
		10	Disable
		11	Enable
		000100-011111	Reserved
		100000	DisableIn-Prog
		100001	DisableIn-Prog
		100010	DisableIn-Prog
		100011	ClkOff
		100100	EnableIn-Prog
		100101	DisableClkOn
		100110	ClkOn (from Disable to DisableClkOn)
		100111	ClkOn (from SwRstDisable to SyncRst)
		101000	ClkOff (from SynRst to SwRstDisable)
		101001-111111	Reserved

9.8 Module Control Register (MDCTL)

The module control MDCTL register controls the clock, reset, and emulation behavior of the module. The MDCTL register is shown in Figure 13 and described in Table 14.

Figure 13. Module Control Register (MDCTL)

31	16
Reserved	
R	

15 11	10	9	8	7 6	4 0
Reserved	EMUIHBIE	EMURSTIE	LRSTZ	Reserved	NEXT
R	R/W	R/W	R/W	R	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14. Module Control Register (MDCTL) Field Descriptions

Bit	Field	Value	Description
31-11	Reserved		Reserved
10	EMUIHBIE		Emulation alters module state
		0	Not enabled
		1	Interrupt enabled
9	EMURSTIE		Emulation alters reset interrupt enable
		0	Not enabled
		1	Interrupt enabled
8	LRSTZ		Module local reset control
		00	Assert local reset
		01	De-assert reset
7-5	Reserved		Reserved
4-0	NEXT		Module next state
		0	SwRstDisable
		1	SyncRst
		2	Disable
		3	Enable
		4-15	Reserved

9.9 L2 Memory Sleep Down Register (L2PDSLEEP)

The L2 memory sleep down (L2PDSLEEP) register converts the logical pages of L2 memory into sleep mode. The L2PDSLEEP register is shown in Figure 14 and described in Table 15.

See Section 3.1 for an explanation of how the L2 memory status and sleep registers interact.

Figure 14. L2 Memory Sleep Down Register (L2PDSLEEP)

31	16
Reserved	
R	

15	2	1	0
Reserved		P1	P0
R		R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15. L2 Memory Sleep Down Register (L2PDSLEEP) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved		Reserved
1	P1		Value of 1 puts L2 page 1 memory in sleep mode
0	P0		Value of 1 puts L2 page 0 memory in sleep mode

9.10 L2 Memory Status Register (L2PDSTAT)

The L2 memory status (L2PDSTAT) register provides the status on the logical pages of L2 memory. The register is shown in Figure 15 and described in Table 16.

See Section 3.2 for an explanation of how the L2 memory status and wake registers interact.

Figure 15. L2 Memory Status Register (L2PDSTAT)

31	16
Reserved	
R	

15	2	1	0
Reserved		P1	P0
R		R	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16. L2 Memory Status Register (L2PDSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved		Reserved
1	P1		Status of L2 page 1 memory sleep mode
0	P0		Status of L2 page 0 memory sleep mode

9.11 L2 Memory Wakeup Register (L2PDWAKE)

The L2 memory wake (L2PDWAKE) register wakes up the logical pages of L2 memory. The register is shown in Figure 16 and described in Table 17.

See Section 3.1 for an explanation of how the L2 memory status and sleep registers interact.

Figure 16. L2 Memory Wakeup Register (L2PDWAKE)

31 … 16
Reserved
R

15 … 2	1	0
Reserved	P1	P0
R	R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 17. L2 Memory Wakeup Register (L2PDWAKE) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved		Reserved
1	P1		Value of 1 wakes L2 page 1 memory
0	P0		Value of 1 wakes L2 page 0 memory

Appendix A Revision History

This revision history highlights the technical changes made to the document in this revision.

Table 18. TCI6486/C6472 PSC Revision History

See	Additions/Modifications/Deletions
Global	Added C6472 device

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

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

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

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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