TMS320C55x DSP Peripherals Overview

User's Guide

Literature Number: SPRU317K
April 2006–Revised December 2011

Contents

List of Figures

List of Tables

Read This First

About This Manual

This manual is for the peripherals that are on the digital signal processors (DSPs) in the TMS320C55x™ (C55x™) DSP generation.

This manual is in transition. Most peripheral information has been revised and moved from chapters in this manual to separate documents. These separate documents are referenced in Chapter 1. The peripheral information that is still in this manual is being revised and will be in separate documents in the future. When the revisions are complete, this manual is to remain as an overview that points to all of the separate documents. In some cases, information has been moved from this manual to the device-specific data manuals.

User's Guide

SPRU317K–April 2006–Revised December 2011

TMS320C55x DSP Peripherals Overview

1 Introduction

Table 1 lists the peripherals of the TMS320C55x (C55x) DSP generation and indicates how many copies of each peripheral are on the specific C55x devices. On a given device, some peripherals may share pins, making the peripherals’ use mutually exclusive; see the device-specific data manual for details.

For a detailed description of a peripheral, see the chapter or document listed in the last column of the table. If a peripheral has its own reference guide, the table shows the literature number (SPRUxxx) for that reference guide. If you are viewing the table online, you can click the literature number to view or download a portable document format (PDF) file. Otherwise, you can find the PDF files on the Internet at http://www.ti.com. Tip: Enter the literature number in the search box provided at http://www.ti.com.

All information about general-purpose I/O pins and system control registers has been moved from this document to the device-specific data manuals.

Table 1. TMS320C55x DSP Peripherals (5501–5515)

Peripheral	5501	5502	5503	5505	5504	5506	5507	5509/ 5509A	5510A	5515	5514	For Details, See...
Analog-to-digital converter (ADC)							1	1				SPRU586
				1						1		SPRUFP1
Clock generator with PLL			1			1	1	1	1			Section 3 and the device-specific data manual.
				1	1					1	1	SPRUFP0
	1	1										Device-specific data manual.
Direct memory access (DMA) controller			1			1	1	1	1			SPRU587
				4	4					4	4	SPRUFO9
	1	1										SPRU613
External memory interface (EMIF)									1			SPRU590
			1			1	1	1				SPRU670
				1	1					1	1	SPRUFO8
	1	1										SPRU621
Host port interface (HPI)									1			SPRU588
			1				1	1				SPRU619
	1	1										SPRU620
Instruction cache									1			SPRU576
	1	1										SPRU630
Inter-integrated circuit (I2C) module	1	1	1			1	1	1				SPRU146
				1	1					1	1	SPRUFO1
Multichannel buffered serial port (McBSP)	2	3	3		1	3	3	3	3		1	SPRU592
MultiMediaCard / SD card controller					2							SPRU593
				2				2		2	2	SPRUFO6
Power management / Idle configurations	1	1	1	1	1	1	1	1	1	1	1	Section 8 and device-specific data manual.
				2	2					2	2	SPRUFP0
Real-time clock (RTC)			1			1	1	1				SPRU594
				1	1					1	1	SPRUFO7
Timer, general-purpose			2			2	2	2	2			SPRU595
				3	3					3	3	SPRUFO2
	2	2										SPRU618
Timer, watchdog			1			1	1	1				SPRU595
				1	1					1	1	SPRUFO2
	1	1										SPRU618
Universal Asynchronous Receiver/Transmitter (UART)	1	1										SPRU597
				1	1					1	1	SPRUFO5

Table 1. TMS320C55x DSP Peripherals (5501–5515) (continued)

Peripheral	5501	5502	5503	5505	5504	5506	5507	5509/ 5509A	5510A	5515	5514	For Details, See...
Universal Serial Bus (USB) module			1			1	1	1				SPRU596
				1	1					1	1	SPRUFO0
Serial Peripheral Interface (SPI)				1	1					1	1	SPRUFO3
Liquid Crystal Display Controller (LCDC)				1						1		SPRUFP3
Inter-IC Sound (I2S) Peripheral				4	4					4	4	SPRUFP4

Table 2. TMS320C55x DSP Peripherals (5532–5535)

Peripheral	5532	5533	5534	5535	For Details, See...
Analog-to-digital converter (ADC)				1	SPRUH87
Clock generator with PLL	1	1	1	1	
Direct memory access (DMA) controller	4	4	4	4	
External memory interface (EMIF)					
Host port interface (HPI)					
Instruction cache					
Inter-integrated circuit (I2C) module	1	1	1	1	
Multichannel buffered serial port (McBSP)					
MultiMediaCard / SD card controller	2	2	2	2	
Power management / Idle configurations	1	1	1	1	
	2	2	2	2	
Real-time clock (RTC)					
Timer, general-purpose					
Timer, watchdog	1	1	1	1	
Universal Asynchronous Receiver/Transmitter (UART)	1	1	1	1	
Universal Serial Bus (USB) module		1	1	1	
Serial Peripheral Interface (SPI)	1	1	1	1	
Liquid Crystal Display Controller (LCDC)				1	
Inter-IC Sound (I2S) Peripheral	4	4	4	4	

2 Analog-to-Digital Converter (ADC)

The ADC is described in the *TMS320VC5507/5509 DSP Analog-to-Digital Converter (ADC) Reference Guide* (SPRU586), and the *TMS320C5515/05/VC05 Successive Approximation (SAR) Analog/Digital Converter (ADC) User's Guide* (SPRUFP1).

3 Clock Generator

This chapter describes the clock generator that is in TMS320VC5509, TMS320VC5509A, and TMS320VC5510 DSPs. The clock generator for the C5515 is described in the *TMS320C5515 DSP System User's Guide* (SPRUFX5). The clock generator for the C5514 is described in the *TMS320C5514 DSP System User's Guide* (SPRUFX6). The clock generator for the C5505 is described in the *TMS320C5505 DSP System User's Guide* (SPRUGH5). The clock generator for the C5504 is described in the *TMS320C5504 DSP System User's Guide* (SPRUGH6). The clock generator described here accepts an input clock at the CLKIN pin and enables you to modify that signal internally to produce an output clock with the desired frequency. The clock generator passes this output clock (the CPU clock) to the CPU, to peripherals, and to other modules inside the C55x. DSP. The CPU clock is also passed through a programmable clock divider to the CLKOUT pin. Check the device-specific data manual for additional clock-generation information.

NOTE: For information about clock generation in other devices, see Table 1.

3.1 Introduction to the DSP Clock Generator

The DSP clock generator supplies the DSP with a clock signal that is based on an input clock signal connected at the CLKIN pin. Included in the clock generator is a digital phase-lock loop (PLL), which can be enabled or disabled. You can configure the clock generator to create a CPU clock signal that has the desired frequency.

The clock generator has a clock mode register, CLKMD (see Section 3.8), for controlling and monitoring the activity of the clock generator. For example, you can write to the PLL ENABLE bit in CLKMD to toggle between the two main modes of operation:

- In the bypass mode (see Section 3.3), the PLL is bypassed, and the frequency of the output clock signal is equal to the frequency of the input clock signal divided by 1, 2, or 4. Because the PLL is disabled, this mode can be used to save power.
- In the lock mode (see Section 3.4), the input frequency can be both multiplied and divided to produce the desired output frequency, and the output clock signal is phase-locked to the input clock signal. The lock mode is entered if the PLL ENABLE bit of the clock mode register is set and the phase-locking sequence is complete. (During the phase-locking sequence, the clock generator is kept in the bypass mode.)

The clock generator also has an idle mode (see Section 3.5) for power conservation. You place the clock generator into its idle mode by turning off the CLKGEN idle domain. For information on turning on and off idle domains, see Section 8, Idle Configurations.

The output of the clock generator or a divided down version of that output can be seen on the CLKOUT pin. For details, see Section 3.6, The CLKOUT Pin and the Associated Clock Divider.

3.2 Operational Flow of the DSP Clock Generator

Figure 1 and Table 3 describe the operational states (A-F) of the DSP clock generator. The clock mode register (CLKMD) is loaded by software or by a DSP reset. If the write to CLKMD enables the PLL, the PLL begins its phase-locking sequence (state A). If the write disables the PLL, the clock generator enters its bypass mode (state D).

Figure 1. Operational Flow of the DSP Clock Generator

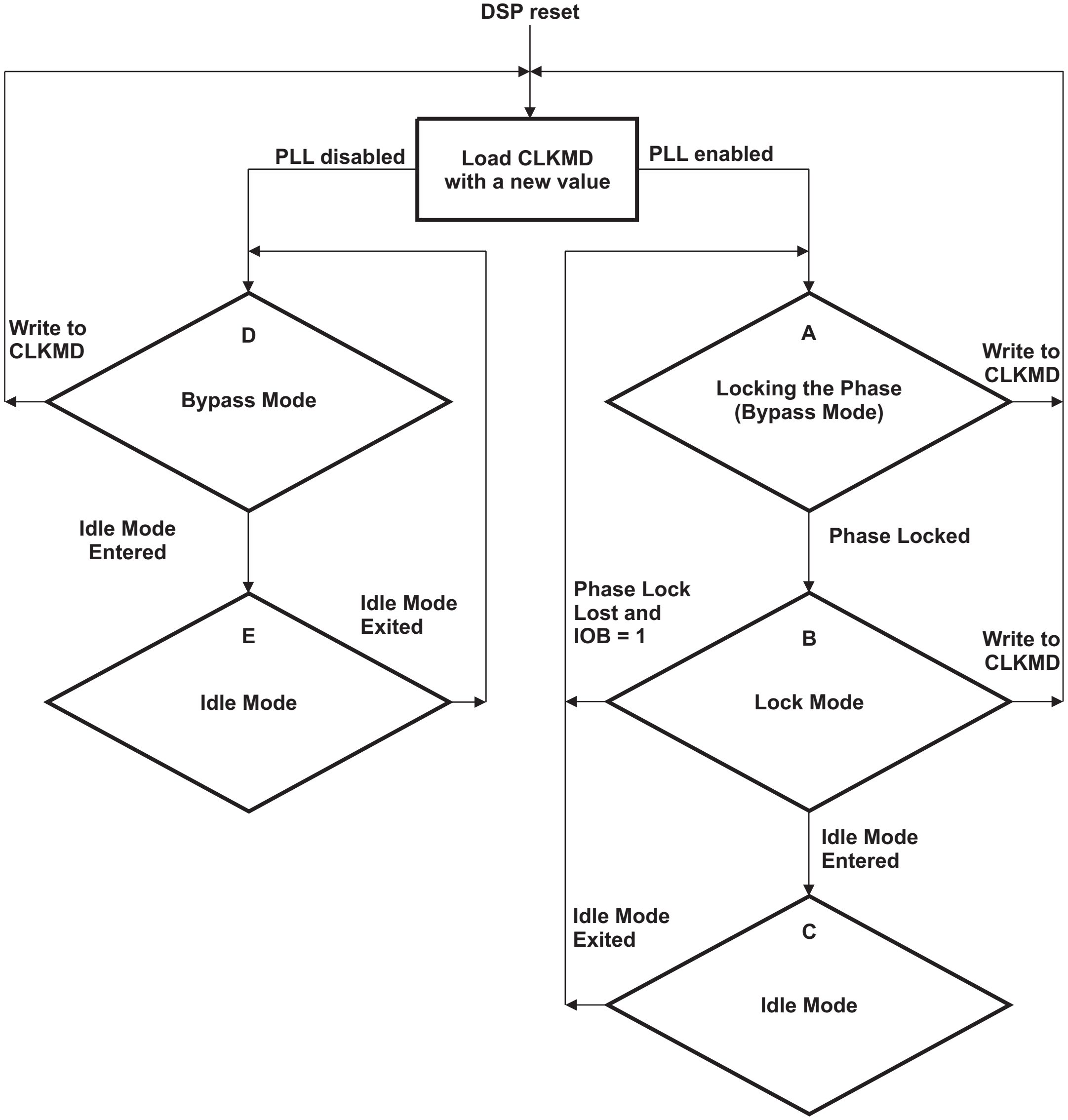


Table 3. Operational States

State	Description
A	**Locking the phase.** The clock generator enters the bypass mode, and the PLL locks the phase of the output clock signal to that of the input clock signal. Once the phase is locked and the output signal is at the frequency defined by the PLL MULT bits and the PLL DIV bits of CLKMD, the clock generator enters its lock mode (state B). You can reconfigure the clock generator by writing to CLKMD.
B	**Lock mode.** In the lock mode, the PLL is generating an output signal with the selected frequency. The output signal is phase-locked to the input signal. If the PLL loses the lock and the IOB bit of CLKMD is 1, the clock generator returns to the bypass mode and reacquires the lock (state A); if the IOB bit is 0, the clock generator does not reacquire the lock. An idle instruction can place the clock generator into its idle mode (state C). To change to the bypass mode or to reconfigure the clock generator in other ways, you can write to CLKMD.
C	**Idle mode (entered from the lock mode).** An idle instruction has placed the clock generator into its idle mode. If the idle mode is properly exited, the clock generator starts again and reacquires the phase lock (state A). The method used to reacquire the lock depends on the IAI bit of CLKMD.
D	**Bypass mode.** The PLL is disabled, and the clock generator is in the bypass mode. The divider within the clock generator produces an output clock signal at the frequency defined by the BYPASS DIV bits of CLKMD. An idle instruction can place the clock generator into its idle mode (state E). To change to the lock mode or to reconfigure the clock generator in other ways, you can write to CLKMD.
E	**Idle mode (entered from the bypass mode).** An idle instruction has placed the clock generator into its idle mode. If the idle mode is properly exited, the clock generator starts again in the bypass mode.

3.3 Bypass Mode

When the DSP clock generator is in the bypass mode and the phase-lock loop (PLL) is disabled, the frequency of the output clock signal is equal to the frequency of the input clock signal divided by 1, 2, or 4.

3.3.1 Entering and Exiting the Bypass Mode

To enter the bypass mode, write a 0 to the PLL ENABLE bit in the clock mode register (CLKMD). The PLL will be disabled.

To exit the bypass mode, write a 1 to the PLL ENABLE bit. The PLL will start up and enter its phase-locking sequence. After the PLL is generating the configured output frequency and the phase of the output clock signal is locked to the phase of the input clock signal, the clock generator enters the lock mode. Until then, the clock generator stays in the bypass mode.

If the clock generator is in the lock mode and the PLL must reacquire its phase lock (IOB = 1), the clock generator enters the bypass mode until the phase is locked again.

3.3.2 CLKMD Bits Used in the Bypass Mode

Table 4 describes the bits of the clock mode register (CLKMD) that are used in the bypass mode. The reserved bits in CLKMD (Rsvd and TEST) should not be used in either the bypass mode or the lock mode. For a detailed description of CLKMD, see Section 3.8.

Table 4. CLKMD Bits Used in the Bypass Mode

CLKMD Bit Field	Role in the Bypass Mode
PLL ENABLE	Allows you to switch to the lock mode.
BYPASS DIV	Determines how the input clock frequency is divided (if at all) to produce the output clock frequency.
LOCK	Is 0 in the bypass mode.

3.3.3 Setting the Output Frequency for the Bypass Mode

The output frequency is determined by the input frequency and the value in the BYPASS DIV bits. Load BYPASS DIV as required to divide the input frequency by 1, 2, or 4.

3.4 Lock Mode

In the lock mode, the input frequency can be both multiplied and divided to produce the desired output frequency, and the output clock signal is phase locked to the input clock signal.

3.4.1 Entering and Exiting the Lock Mode

To enter the lock mode, write a 1 to the PLL ENABLE bit in the clock mode register (CLKMD). The PLL will start up and will enter its phase-locking sequence. After the PLL is generating the configured output frequency and the phase of the output clock signal is locked to the phase of the input clock signal, the clock generator enters the lock mode. Until then, the clock generator stays in the bypass mode.

If the clock generator is in the lock mode and the PLL must reacquire its phase lock (IOB = 1 in CLKMD), the clock generator will enter the bypass mode until the phase is locked again.

To exit the lock mode (enter the bypass mode), write a 0 to the PLL ENABLE bit. The PLL will be disabled.

3.4.2 CLKMD Bits Used in the Lock Mode

Table 5 describes the bits of the clock mode register (CLKMD) that are used in the lock mode. The reserved bits (Rsvd and TEST) in CLKMD should not be used in either the lock mode or the bypass mode. For a detailed description of CLKMD, see Section 3.8.

Table 5. CLKMD Bits Used in the Lock Mode

CLKMD Bit Field(s)	Role in the Lock Mode
PLL ENABLE	Allows you to switch to the lock mode.
PLL MULT and PLL DIV	Determine how the input clock frequency is modified (if at all) to produce the output clock frequency.
IAI	Determines whether the PLL restarts the phase-locking sequence when the clock generator exits its idle mode.
BREAKLN	Indicates when the phase lock has been broken.
IOB	Determines whether the PLL will reacquire a lost phase lock.
LOCK	Is 1 in the lock mode.

3.4.3 Setting the Output Frequency for the Lock Mode

The input frequency is multiplied by the PLL MULT value of CLKMD and is divided according to the PLL DIV value of CLKMD. PLL MULT can be a value from 2 to 31. PLL DIV can be a value from 0 (divide by 1) to 3 (divide by 4). The output frequency can be calculated with the following equation:

$$\textbf{Output frequency} = \frac{\textbf{PLL MULT}}{(\textbf{PLL DIV+1})} \times \textbf{Input frequency}$$

Table 6 shows some examples of using PLL MULT and PLL DIV to select an output frequency.

Table 6. Examples of Selecting a Lock Mode Frequency

PLL MULT	PLL DIV	Output Frequency
31	0 (divide by1)	31 x Input frequency (maximum frequency)
10	1 (divide by 2)	5 x Input frequency
2	2 (divide by 3)	2/3 x Input frequency
2	3 (divide by 4)	1/2 x Input frequency (minimum frequency)

3.4.4 Lock Time

The lock time is dependent on the multiply and divide factors chosen as well as the operating frequency. A good rule of thumb is to estimate 100μs for the lock time.

3.5 Idle (Low-Power) Mode

To save power, you can put the DSP clock generator into its idle mode by loading an idle configuration that turns off the CLKGEN idle domain. When the clock generator is idle, the output clock is stopped and held high. For more details, see Section 8, Idle Configurations.

When the clock generator exits its idle mode, the reaction of the clock generator depends on several factors. If the clock generator was in its bypass mode before the idle instruction was executed, the PLL returns to the bypass mode. If the clock generator was in its lock mode before the idle instruction was executed, the clock generator switches to its bypass mode, reacquires the phase lock, and then returns to the lock mode. The method used for reacquiring the phase lock depends on the IAI bit of CLKMD:

Table 7. Method Used for Reacquiring the Phase Lock

IAI	Description
0	The PLL does not restart the phase-locking sequence. Instead, the PLL reacquires the phase lock using the same lock settings that were in use just before the idle mode was entered.
1	The PLL restarts the phase-locking sequence. This option is recommended if the input clock has or may have changed while the clock generator was idle.

3.6 The CLKOUT Pin and the Associated Clock Divider

The DSP clock generator generates the CPU clock that is supplied to the CPU, to peripherals, and to other modules inside the DSP. As shown in Figure 2, the CPU clock is also passed to a clock divider that supplies a signal (CLKOUT) to the CLKOUT pin. The frequency of CLKOUT depends on the CLKDIV bits of the system register, SYSR (see Table 8). Consult the device-specific data manual to determine whether this clock divider feature is available on a particular C55x DSP.

Figure 2. Dividing the CPU Clock for the CLKOUT Pin

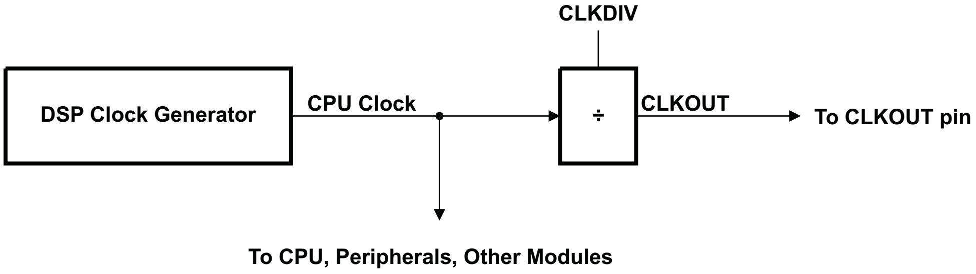


Table 8. Effect of CLKDIV Bits on CLKOUT Frequency

CLKDIV	Frequency of CLKOUT
000b	1/1 x CPU clock frequency
001b	1/2 x CPU clock frequency
010b	1/4 x CPU clock frequency
011b	1/6 x CPU clock frequency
100b	1/8 x CPU clock frequency
101b	1/10 x CPU clock frequency
110b	1/12 x CPU clock frequency
111b	1/14 x CPU clock frequency

3.7 DSP Reset Conditions of the DSP Clock Generator

The following sections describe the operation of the DSP clock generator when the DSP is held in its reset state and when the DSP is removed from its reset state.

3.7.1 Clock Generator During Reset

The DSP can make use of the output clock signal during reset. While the DSP reset signal is held low:

- The clock generator is in the bypass mode.
- The output clock frequency is determined by the level of the signal on the CLKMD input pin:

CLKMD Signal	Output Frequency
Low	Input frequency
High	1/2 x Input frequency

3.7.2 Clock Generator After Reset

On the rising edge of the DSP reset signal (when reset is deasserted), the clock mode register is loaded with a value determined by the level on the CLKMD pin:

CLKMD Signal	Clock Mode Register Value
Low	2002h
High	2006h

Table 9 summarizes the effects of this load to the clock mode register.

Table 9. Reset Values of CLKMD Bits and The Effects

Reset Value	Effect
IAI = 0	Only applicable in the lock mode. Initialize-after-idle is not selected. After the idle mode is exited, the PLL reacquires the phase lock using the same lock settings that were in use just before the idle mode was entered (the phase-locking sequence is not restarted).
IOB= 1	Only applicable in the lock mode. Initialize-on-break is selected. Any time the PLL loses its phase lock, the clock generator switches to its bypass mode and starts a new phase-locking sequence.
PLL MULT= 00000b PLL DIV= 00b	Only applicable in the lock mode. The output frequency is equal to the input frequency.
PLL ENABLE = 0	The PLL is disabled. The clock generator is in its bypass mode.
If CLKMD signal is low	If CLKMD signal is low
BYPASS DIV= 00b	Output frequency = Input frequency
If CLKMD signal is high	If CLKMD signal is high
BYPASS DIV= 01b	Output frequency = 1/2 x Input frequency
BREAKLN = 1	The break-lock indicator is reset.
LOCK = 0	The lock-mode indicator reflects the fact that the clock generator is in the bypass mode.

3.8 Clock Mode Register (CLKMD)

You control the DSP clock generator with the clock mode register, CLKMD. Figure 3 and Table 10 describe the contents of CLKMD, which is accessible in I/O space. After the DSP reset signal becomes inactive, the CLKMD register is initialized with a predetermined value dependent only upon the state of the CLKMD input pin (the difference is in the BYPASS DIV bits):

CLKMD Signal Level at Reset	CLKMD Register Reset Value
Low	2002h
High	2006h

Figure 3. Clock Mode Register (CLKMD)

15	14	13	12	11	10	9	8
Reserved	IAI	IOB	TEST	PLLMULT			
R-0	R/W-0	R/W-1	(Keep 0)	R/W-00000			

7	6	5	4	3	2	1	0
PLLMULT	PLLDIV		PLLENABLE	BYPASSDIV		BREAKLN	LOCK
R/W-00000	R/W-00		R/W-0	R/W-PIN		R-1	R-0

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

Table 10. Clock Mode Register (CLKMD) Field Descriptions

Bit	Field	Value	Description
15	Reserved	0	Reserved. This bit is not available for use. This bit is always 0.
14	IAI		Initialize after idle bit. IAI determines how the PLL reacquires the phase lock after the clock generator exits its idle mode (when the CLKGEN idle domain is reactivated):
		0	The PLL does not restart the phase-locking sequence. Instead the PLL reacquires the lock using the same lock settings that were in use just before the idle mode was entered.
		1	The PLL restarts the phase-locking sequence. This option is recommended if the input clock has or may have changed while the clock generator was idle.
13	IOB		Initialize on break bit. IOB determines whether the clock generator initializes the PLL phase-locking sequence whenever the phase lock is broken. If the PLL indicates a break in the phase lock:
		0	The clock generator does not interrupt the PLL. The clock generator stays in the lock mode, and the PLL continues to output the current clock signal.
		1	The clock generator switches to its bypass mode and restarts the PLL phase-locking sequence.
12	TEST	0-1	This reserved test bit is cleared during a DSP reset and your program must be keep it 0 for proper operation of the clock generator. Make sure that whenever your program modifies CLKMD, it writes a 0 to bit 12.
11-7	PLLMULT	0-1Fh	PLL multiply value. When the PLL is enabled (PLL ENABLE = 1), the frequency of the input clock signal is multiplied according to the value in PLL MULT. PLL MULT can be a value from 2 to 31. The input clock is multiplied by the unsigned integer in PLL MULT and is divided according to the value in the PLL DIV bits. The maximum frequency for the PLL output clock signal is 31 times the frequency of the input clock signal. To obtain this maximum frequency, load PLL MULT with 31 (multiply by 31), and load PLL DIV with 0 (divide by 1). The minimum frequency for the output clock signal is 1/2 the frequency of the input clock signal. To obtain this minimum frequency, load PLL MULT with 2 (multiply by 2) and load PLL DIV with 3 (divide by 4).

Table 10. Clock Mode Register (CLKMD) Field Descriptions (continued)

Bit	Field	Value	Description
6-5	PLLDIV		PLL divide value. When the PLL is enabled (PLL ENABLE = 1), the two PLL DIV bits select one of four divide options listed in the following table. The PLL also uses the multiply value supplied by the PLL MULT bits. To program the minimum or maximum frequency, see the description for the PLL MULT bits (bits 11-7).
		0	No division/divide by 1 The input frequency is not divided.
		1h	Divide by 2 The input frequency is divided by 2.
		2h	Divide by 3 The input frequency is divided by 3.
		3h	Divide by 4 The input frequency is divided by 4.
4	PLLENABLE		PLL enable bit. Write to PLL ENABLE to enable or disable the PLL. When you set PLL ENABLE, you request the clock generator to enter the lock mode. The clock generator does not enter the lock mode until the PLL is creating a phase-locked signal with the frequency selected by the PLL MULT bits and the PLL DIV bits.
		0	Disable the PLL (enter the bypass mode).
		1	Enable the PLL and, when the correct output clock signal is generated, enter the lock mode.
3-2	BYPASSDIV		Bypass-mode divide value. In the bypass mode, BYPASS DIV determines the frequency of the output clock signal. During a DSP reset, if the level on the CLKMD pin is low, BYPASS DIV is reset to 00b (no division). If the level on CLKMD is high, BYPASS DIV is reset to 01b (divide by 2). **Note:** 00b if CLKMD signal is low. 01b if CLKMD signal is high.
		0	No division/divide by 1 The frequency of the output clock signal is the same as the frequency of the input clock signal.
		1h	Divide by 2 The frequency of the output clock signal is 1/2 the frequency of the input clock signal.
		2h	Divide by 4 The frequency of the output clock signal is 1/4 the frequency of the input clock signal.
		3h	Divide by 4 The frequency of the output clock signal is 1/4 the frequency of the input clock signal.
1	BREAKLN		Break-lock indicator. BREAKLN indicates whether the PLL has broken the phase lock. In addition, if you write to CLKMD, BREAKLN is forced to 1.
		0	The PLL has broken the phase lock.
		1	The phase lock is restored, or a write to CLKMD has occurred.
0	LOCK		Lock-mode indicator. LOCK indicates whether the clock generator is in its lock mode:
		0	The clock generator is in the bypass mode. The output clock signal has the frequency determined by the BYPASS DIV bits, or the PLL is in the process of getting a phase lock.
		1	The clock generator is in the lock mode. The PLL has a phase lock, and the output clock has the frequency determined by the PLL MULT bits and the PLL DIV bits.

4 Direct Memory Access (DMA) Controller

The DMA controller is described in the *TMS320VC5501/5502 DSP Direct Memory Access (DMA) Controller Reference Guide* (SPRU613), the *TMS320VC5503/5507/5509/5510 DSP Direct Memory Access (DMA) Controller Reference Guide* (SPRU587); and the *TMS320VC5505/5504 DSP Direct Memory Access (DMA) Controller User's Guide* (SPRUFO9).

5 External memory Interface (EMIF)

For the EMIF information that applies to your C55x device, see the corresponding device-specific reference guide:

- *TMS320VC5501/5502 DSP External Memory Interface (EMIF) Reference Guide* (SPRU621),
- *TMS320VC5503/5507/5509 DSP External Memory Interface (EMIF) Reference Guide* (SPRU670), or
- *TMS320VC5510 DSP External Memory Interface (EMIF) Reference Guide* (SPRU590),
- *TMS320C5515/14/05/04 DSP External Memory Interface (EMIF) User's Guide* (SPRUGU6).

6 General-Purpose I/O Port

For information on the C5515/14/05/04/VC05/VC04 general-purpose I/O (GPIO) ports, see the *TMS320C5515/14/05/04/VC05/VC04 DSP General-Purpose Input/Output User's Guide* (SPRUFO4). For information on the GPIO ports on other devices, see the device-specific data manuals.

7 Host Port Interface (HPI)

For the HPI information that applies to your C55x device, see the corresponding device-specific reference guide:

- *TMS320VC5501/5502 DSP Host Port Interface (HPI) Reference Guide* (SPRU620),
- *TMS320VC5503/5507/5509 DSP Host Port Interface (HPI) Reference Guide* (SPRU619), or
- *TMS320VC5510 DSP Host Port Interface (HPI) Reference Guide* (SPRU588).

The name “enhanced host port interface (EHPI)” has been reduced to “host port interface (HPI)” to align with a new peripheral naming convention.

8 Idle Configurations

The TMS320C55x DSP is divided into the idle domains described in this chapter. To minimize power consumption, you can choose which domains are active and which domains are idle at any given time. The current state of all domains is collectively called the idle configuration.

NOTE:

1. Check the device-specific data manual for additional information about deactivating and reactivating idle domains of the DSP.
2. For idle configurations of the C5515 DSP, see the *TMS320C5515 DSP System User's Guide* (SPRUFX5). For idle configurations of the C5514 DSP, see the *TMS320C5514 DSP System User's Guide* (SPRUFX6). For idle configurations of the C5505 DSP, see the *TMS320C5505 DSP System User's Guide* (SPRUGH5). For idle configurations of the C5504 DSP, see the *TMS320C5504 DSP System User's Guide* (SPRUGH6).

8.1 Idle Domains

The DSP is divided into the idle domains described in Table 11. You can control which of these idle domains are active and which are idle at any given time, as described in Section 8.2.

NOTE:

1. The peripheral bus controller and the host port interface (HPI) on the DSP are not part of any idle domain. The only way to turn these modules off is to put the clock generator into its idle mode (make the CLKGEN domain idle).
2. The internal memory blocks (SARAM and DARAM) and the external memory are shared by two domains (CPU and DMA). When both domains are idle, memory accesses are disabled.

Table 11. Idle Domains in the DSP

DOMAIN	Contents of the Domain	Configurability
CPU	CPU and CPU buses	When the idle instruction is executed, the CPU remains active or becomes idle, depending on the chosen idle configuration. Regardless of this domain's state before a DSP reset, it is active after a DSP reset.
DMA	DMA controller and DMA buses	When the idle instruction is executed, the DMA (direct memory access) controller remains active or becomes idle, depending on the chosen idle configuration. Regardless of this domain's state before a DSP reset, it is active after a DSP reset.
CASHE	Instruction cache	When the idle instruction is executed, the instruction cache remains active or becomes idle, depending on the chosen idle configuration. Regardless of this domain's state before a DSP reset, it is active after a DSP reset.
PERIPH	Timers, serial ports, and other peripherals	Each of the peripherals in this domain has an idle enable bit that determines whether the peripheral can be placed in its idle mode when the idle instruction is executed. If the PERIPH domain is configured to be idle and an idle enable bit is 1, the corresponding peripheral is placed in its idle mode. Regardless of this domain's state before a DSP reset, it is active after a DSP reset.
CLKGEN	Clock generator, including the phase-lock loop (PLL) circuitry	When the idle instruction is executed, the clock generator remains active or becomes idle, depending on the chosen idle configuration. When the clock generator is in its idle mode, no clocking is available for the CPU or the DMA controller. If the clock generator is configured to be idle and the CPU, the DMA controller, or the cache is configured to be active, a bus error interrupt request is sent to the CPU. If properly enabled, the interrupt will force the CLKGEN domain to be reactivated. Peripherals that do not depend on the DSP clock signal are not affected by the state of the CLKGEN domain. Regardless of this domain's state before a DSP reset, it is active after a DSP reset.
EMIF	External memory interface (EMIF)	When the idle instruction is executed, the EMIF is disabled or enabled, depending on the chosen idle configuration. Regardless of this domain's state before a DSP reset, it is active after a DSP reset.

8.2 Idle Configuration Process

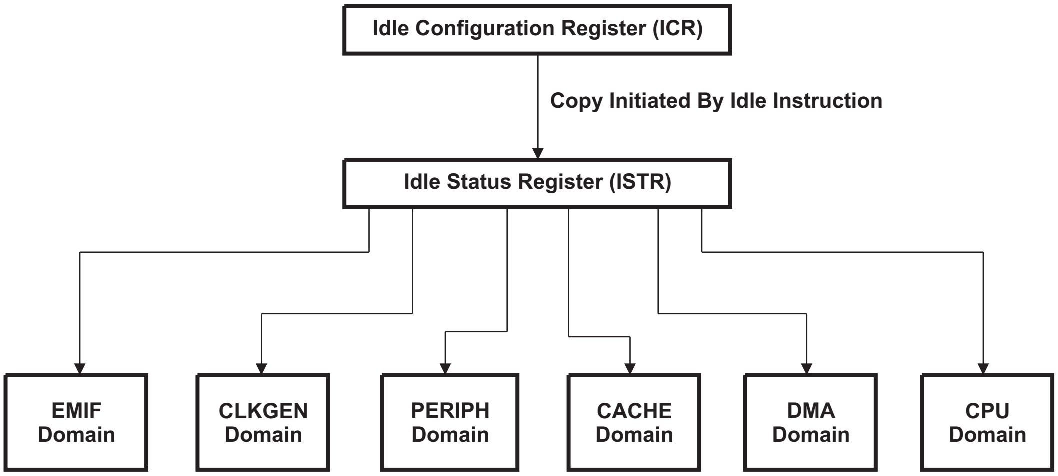
The idle configuration indicates which idle domains will be idle, and which idle domains will be active, the next time the idle instruction is executed. The basic steps to the idle configuration process are:

1. Define a new idle configuration by writing to the bits in the idle configuration register (ICR). Make sure that you use a valid idle configuration (see Section 8.3).
2. Apply the new idle configuration by executing the idle instruction. The effects are shown in Figure 4. The content of ICR is copied to the idle status register (ISTR). The bits of ISTR are then propagated through the system to enable or disable each of the chosen domains.

The idle instruction cannot be executed in parallel with another instruction.

NOTE: If you intend to switch among multiple idle configurations, make sure that your system has the means to change from one idle configuration to the next. For important considerations, see Section 8.4.

Figure 4. Idle Configuration Process


Idle Configuration Register (ICR)
Copy Initiated By Idle Instruction
Idle Status Register (ISTR)
EMIF Domain
CLKGEN Domain
PERIPH Domain
CACHE Domain
DMA Domain
CPU Domain

8.3 Valid Idle Configurations

Not all of the values that you can write to the idle configuration register (ICR) provide valid idle configurations. The valid configurations are limited by dependencies within the system. For example, when the CLKGEN domain is idle (the DSP clock generator is disabled), the DMA controller, the CPU, and any peripherals that do not have their own external clocks cannot operate.

8.4 To Change Idle Configurations (Key Conditions)

Before you use the idle instruction, make sure that there is a method for the DSP to change the idle configuration afterward. Table 12 summarizes the methods available under three key conditions. The table also describes the effects on the idle registers: the idle status register (ISTR) and the idle configuration register (ICR). For more details about these idle registers, see Section 8.7.

Table 12. Changing Idle Configurations

Condition	Available Methods for Changing Idle Configuration	ISTR After Change	ICR After Change
1. CLKGEN domain is active CPU domain is active (see Section 8.4.1)	A. Write a new configuration to the idle configuration register (ICR), and then execute the idle instruction.	A. Modified by the idle instruction; contains a copy of the new ICR value.	A. Contains the new value that was loaded by the program.
	B. Initiate a DSP hardware reset.	B. Cleared (all 0s).	B. Cleared (all 0s)
2. CLKGEN domain is active CPU domain is idle (see Section 8.4.2)	A. Use an unmasked hardware interrupt or the nonmaskable hardware interrupt called NMI_.	A. CLKGENIS and CPUIS bits are 0. No other bits were modified.	A. Not modified.
	B. Initiate a DSP hardware reset.	B. Cleared (all 0s).	B. Cleared (all 0s).
3. CLKGEN domain is idle (see Section 8.4.3)	A. Use an unmasked hardware interrupt or the nonmaskable hardware interrupt called NMI_.	A. CLKGENIS and CPUIS bits are 0. No other bits were modified.	A. Not modified.
	B. Initiate a DSP hardware reset.	B. Cleared (all 0s).	B. Cleared (all 0s).

8.4.1 Condition 1: CLKGEN and CPU Domains Active

When the CLKGEN domain is active (the DSP clock generator is enabled) and the CPU domain is active (the DSP CPU is running), program flow continues. In this case, there are two methods of changing idle configurations:

- Write a new idle configuration to the idle configuration register (ICR), and then execute the idle instruction. The idle instruction copies the content of the ICR to the idle status register (ISTR), and the ISTR bit values are propagated to the idle domains. After the domains change states, the value in ISTR matches the value in ICR.
- Initiate a DSP hardware reset at the DSP reset pin. When the DSP resets, all domains are made active.

8.4.2 Condition 2: CLKGEN Domain Active, CPU Domain Idle

When the CPU domain is idle, program flow is halted. It is not possible to write a new value to the idle configuration register (ICR) or to execute the idle instruction. Two other methods are available for changing the idle configuration:

- Use an unmasked interrupt or the nonmaskable interrupt called NMI_. The interrupt clears the CLKGENIS and CPUIS bits of the idle status register (ISTR). The change to CPUIS reactivates the CPU domain, and the change to CLKGENIS ensures that the CLKGEN domain is also active. The content of the idle configuration register (ICR) is not modified. To learn how the CPU responds to the interrupt, see Section 8.5.
- Initiate a DSP hardware reset at the DSP reset pin. When the DSP resets, all domains are made active.

Once program flow has begun again, you can reactivate or deactivate other domains by writing a new idle configuration to ICR and then executing the idle instruction.

8.4.3 Condition 3: CLKGEN Domain Idle

When the CLKGEN domain is idle (the DSP clock generator is disabled), no internal clocks are active, including the CPU clock. With the CPU halted, it is not possible to write a new value to the idle configuration register (ICR) or to execute the idle instruction. Two other methods are available for waking the DSP:

- Use an unmasked interrupt or the nonmaskable interrupt called NMI_. The interrupt clears the CLKGENIS and CPUIS bits of the idle status register (ISTR). The change to CPUIS reactivates the CPU domain, and the change to CLKGENIS ensures that the CLKGEN domain is also active. The content of the idle configuration register (ICR) is not modified. To learn how the CPU responds to the interrupt, see Section 8.5.
- Initiate a DSP hardware reset at the DSP reset pin. When the DSP resets, all domains are made active.

Once program flow has begun again, you can reactivate or deactivate other domains by writing a new idle configuration to ICR and then executing the idle instruction.

8.5 Interrupt Handling When the CPU Is Reactivated

If the CPU has been halted by an idle configuration, it can be reactivated by a nonmaskable interrupt (NMI_ or RESET_) or by a maskable interrupt that is enabled in an interrupt enable register (IER0 or IER1). A maskable interrupt request will also set the corresponding interrupt flag bit in an interrupt flag register (IFR0 or IFR1). Table 13 summarizes how the CPU responds after being reactivated by maskable and nonmaskable interrupts. INTM is the global interrupt mask bit in status register ST1_55.

Table 13. CPU Response After Reactivation

Interrupt	CPU Response After Reactivation
A maskable interrupt	If INTM = 0: The CPU executes the interrupt service routine, executes the instruction that follows the idle instruction, and continues from there. If INTM = 1: The CPU executes the instruction that follows the idle instruction and then continues from there. The interrupt service routine cannot be executed until the interrupt has been enabled by INTM.
NMI_ (nonmaskable)	The CPU executes the interrupt service routine, executes the instruction that follows the idle instruction, and continues from there.
RESET_ (nonmaskable)	The DSP is reset. (During a DSP reset, all idle domains are made active.)

8.6 Effect of a DSP Reset on the Idle Domains

Driving the DSP reset signal low starts a DSP reset. During a DSP reset, all idle domains are made active.

8.7 Idle Registers

Two registers provide the means for you to individually configure and monitor each of the idle domains: the idle configuration register (ICR) and the idle status register (ISTR). These registers (see Figure 5 and Figure 6) are part of the peripheral bus controller and are accessible to the CPU. Table 14 describes the read/write bits of ICR, and Table 15 describes the read-only bits of ISTR.

ICR lets you configure how each idle domain will respond the next time the idle instruction is executed. When you execute the idle instruction, the content of ICR is copied to ISTR. Then the ISTR values are propagated to the idle domains. Making the clock generator idle (CLKGENI = 1) provides the lowest level of power consumption in the DSP by stopping all the system clocks.

Figure 5. Idle Configuration Register (ICR)

15 6	5	4	3	2	1	0
Reserved	EMIFI	CLKGENI	PERI	CACHEI	DMAI	CPUI
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 14. Idle Configuration Register (ICR) Field Descriptions

Bit	Field	Value	Description
16-6	Reserved	0	Reserved. These bits are read-only bits and return 0s when read.
5	EMIFI		EMIF-domain idle configuration bit. EMIFI determines whether the external memory interface (EMIF) will be idle after the next execution of the idle instruction:
		0	EMIF will be active.
		1	EMIF will be idle.
4	CLKGENI		CLKGEN-domain idle configuration bit. CLKGENI determines whether the DSP clock generator will be idle after the next execution of the idle instruction:
		0	Clock generator will be active.
		1	Clock generator will be idle
			Note: For a proper power-down, when you set CLKGEN = 1, make sure you also set CPUI = 1, DMAI = 1, and CACHEI = 1. If you do not, a bus error interrupt (BERRINT) request is sent to the CPU.
3	PERI		PERIPH-domain idle configuration bit. Peripherals that are in the PERIPH domain each have an idle enable bit. PERI, in conjunction with the idle enable bits, determines which of the peripherals in the domain will be idle after the next execution of the idle instruction:
		0	All peripherals in the domain will be active.
		1	For each peripheral in the domain: If the idle enable bit is 1, the peripheral will be in its idle mode. If the idle enable bit is 0, the peripheral will be active.
2	CACHEI		CACHE-domain idle configuration bit. CACHEI determines whether the cache will be idle after the next execution of the idle instruction:
		0	Cache will be active.
		1	Cache will be idle.
1	DMAI		DMA-domain idle configuration bit. DMAI determines whether the DMA controller will be idle after the next execution of the idle instruction:
		0	DMA controller will be active.
		1	DMA controller will be idle.
0	CPUI		CPU-domain idle configuration bit. CPUI determines whether the DSP CPU will be idle after the next execution of the idle instruction:
		0	CPU will be active.
		1	CPU will be active.

Figure 6. Idle Status Register (ISTR)

15 6	5	4	3	2	1	0
Reserved	EMIFIS	CLKGENIS	PERIS	CACHEIS	DMAIS	CPUIS
R-0	R-0	R-0	R-0	R-0	R-0	R-0

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

Table 15. Idle Status Register (ISTR) Field Descriptions

Bit	Field	Value	Description
15-6	Reserved	0	Reserved. These bits are read-only bits and return 0s when read.
5	EMIFIS		EMIF-domain idle status bit. EMIFIS is a copy of EMIFI made during the execution of the idle instruction. EMIFIS reflects the current idle status of the external memory interface (EMIF):
		0	EMIF is active.
		1	EMIF is idle.
4	CLKGENIS		CLKGEN-domain idle status bit. CLKGENIS is a copy of CLKGENI made during the execution of the idle instruction. CLKGENIS reflects the current idle status of the DSP clock generator:
		0	Clock generator is active.
		1	Clock generator is idle.
3	PERIS		PERIPH-domain idle status bit. PERIS is a copy of PERI made during the execution of the idle instruction. PERIS reflects the current idle status of the peripherals in the PERIPH domain:
		0	All peripherals in the domain are active.
		1	For each peripheral in the domain: If the idle enable bit is 1, the peripheral is idle. If the idle enable bit is 0, the peripheral is active.
2	CACHEIS		CACHE-domain idle status bit. CACHEIS is a copy of CACHEI made during the execution of the idle instruction. CACHEIS reflects the current idle status of the cache:
		0	Cache is active.
		1	Cache is idle.
1	DMAIS		DMA-domain idle status bit. DMAIS is a copy of DMAI made during the execution of the idle instruction. DMAIS reflects the current idle status of the DMA controller:
		0	DMA controller is active.
		1	DMA controller is idle.
0	CPUIS		CPU-domain idle status bit. CPUIS is a copy of CPUI made during the execution of the idle instruction. CPUIS reflects the current idle status of the DSP CPU:
		0	CPU is active.
		1	CPU is idle.

9 Instruction Cache

The instruction cache that is in TMS320VC5501 and TMS320VC5502 DSPs is described in the *TMS320VC5501/5502 DSP Instruction Cache Reference Guide* (SPRU630). The instruction cache that is in TMS320VC5510 DSPs is described in the *TMS320VC5510 DSP Instruction Cache Reference Guide* (SPRU576).

10 Inter-Integrated Circuit (I2C) Module

For more information about the I2C module, see the *TMS320VC5501/5502/5503/5507/5509 DSP Inter-Integrated Circuit (I2C) Module Reference Guide* (SPRU146) and the *TMS320C5515/14/05/04/VC05/VC04 DSP Inter-Integrated Circuit (I2C) Peripheral User's Guide* (SPRUFO1).

11 Multichannel Buffered Serial Port (McBSP)

For more information about the McBSP, see the *TMS320VC5501/5502/5503/5507/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRU592).

12 Real-Time Clock (RTC)

For more information about the RTC, see the *TMS320VC5503/5507/5509 DSP Real-Time Clock (RTC) Reference Guide* (SPRU594) and the *TMS320C5515/14/05/04 DSP Real-Time Clock (RTC) User's Guide* (SPRUFX2).

13 System Control Registers

For more information on the C5515 system control registers, see the *TMS320C5515 DSP System User's Guide* (SPRUFX5). For more information on the C5514 system control registers, see the *TMS320C5514 DSP System User's Guide* (SPRUFX6). For more information on the C5505 system control registers, see the *TMS320C5505 DSP System User's Guide* (SPRUGH5). For more information on the C5504 system control registers, see the *TMS320C5504 DSP System User's Guide* (SPRUGH6). System control register information for other devices has been moved to the device-specific data manuals.

14 Timer (General-Purpose)

The general-purpose timer is described in the *TMS320VC5501/5502 DSP Timers Reference Guide* (SPRU618), the *TMS320VC5503/5507/5509/5510 DSP Timers Reference Guide* (SPRU595), and the *TMS320C5515/14/05/04/VC05/VC04 DSP Timer/Watchdog Timer User's Guide* (SPRUFO2).

15 Universal Asynchronous Receiver/Transmitter (UART)

UART information is in the *TMS320VC5501/5502 DSP Universal Asynchronous Receiver/Transmitter (UART) Reference Guide* (SPRU597) and the *TMS320C5515/14/05/04/VC05/VC04 DSP Universal Asynchronous Receiver/Transmitter (UART) User's Guide* (SPRUFO5).

16 Universal Serial Bus (USB) Module

For more information about the USB module, see the *TMS320VC5507/5509 DSP Universal Serial Bus (USB) Module Reference Guide* (SPRU596), and the *TMS320C5515/14/05/04 DSP Universal Serial Bus 2.0 (USB) Controller User's Guide* (SPRUGH9).

17 Watchdog Timer

The watchdog timer is described in the *TMS320VC5501/5502 DSP Timers Reference Guide* (SPRU618), the *TMS320VC5503/5507/5509/5510 DSP Timers Reference Guide* (SPRU595), and the *TMS320C5515/14/05/04/VC05/VC04 DSP Timer/Watchdog Timer User's Guide* (SPRUFO2).

18 Serial Peripheral Interface (SPI)

For more information about the serial peripheral interface (SPI), see the *TMS320C5515/14/05/04/VC05/VC04 DSP Serial Peripheral Interface (SPI) User's Guide* (SPRUFO3).

19 Liquid Crystal Display Controller (LCDC)

For more information about the liquid crystal display controller (LCDC), see the *TMS320C5515/05/VC05 DSP Liquid Crystal Display Controller (LCDC) User's Guide* (SPRUFP3).

20 Inter-IC Sound (I2S) Peripheral

For more information about the inter-IC sound (I2S) peripheral, see the *TMS320C5515/14/05/04 DSP Inter-IC Sound (I2S) User's Guide* (SPRUFX4).

21 MultiMedia Card (MMC) / SD Card Controller

For more information about the multimedia card/SD card controller, see the *TMS320VC5509 DSP MultiMediaCard / SD Card Controller Reference Guide* (SPRU593) and *TMS320C5515/14/05/04/VC05/VC04 DSP Multimedia Card (MMC)/Secure Digital (SD) Card Controller Reference Guide* (SPRUFO6).

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

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