KeyStone Architecture Phase-Locked Loop (PLL)

User's Guide

Literature Number: SPRUGV2I
November 2010–Revised July 2017

Contents

List of Figures

List of Tables

Preface

About This Manual

This document describes the operation of the KeyStone software-programmable phase-locked loop (PLL) Controller. The PLL Controller offers flexibility and convenience by way of software-configurable multipliers and dividers to modify the input signal internally. The resulting clock outputs are passed to the CorePacs, peripherals, and other modules inside the device.

Notational Conventions

This document uses the following conventions:

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

Notes use the following conventions:

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

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

CAUTION

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

WARNING

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

Related Documentation from Texas Instruments

C66x CorePac User Guide	SPRUGW0
C66x CPU and Instruction Set Reference Guide	SPRUGH7
Hardware Design Guide for KeyStone Devices	SPRABI2

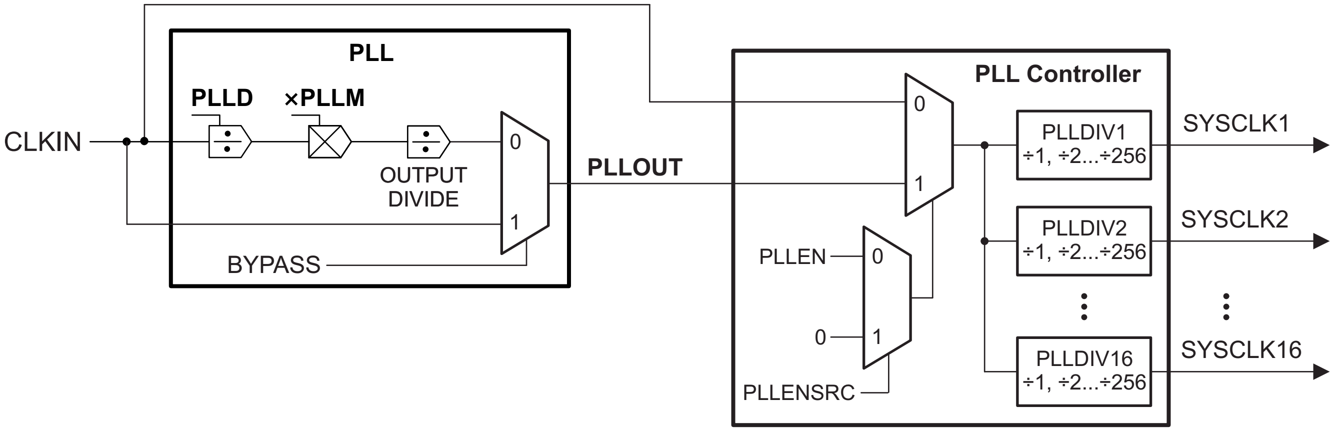
All trademarks are the property of their respective owners.

Overview

1.1 Overview

Figure 1-1 shows the logical implementation of the PLL and the PLL Controller. The PLL Controller offers flexibility and convenience with software-configurable dividers (PLLDIV1 to PLLDIV16) to modify the input clock signal internally. The PLL Controller also contains registers (PLLM and SECCTL) that are used to drive the PLLM, OUTPUT DIVIDE, and BYPASS logic of the PLL (see Figure 1-1). The resulting clock outputs from the PLL Controller are passed to the DSP core, peripherals, and other modules inside the device.

Figure 1-1. PLL and PLL Controller Block Diagram



The PLL Controller has the following input and output clocks:

- Input reference clock to the PLL Controller:
  - **PLLOUT:** Output signal from the PLL
  - **CLKIN:** Input reference clock to the PLL
- Resulting output clocks from the PLL Controller:
  - **SYSCLK1 to SYSCLK16:** System domain clocks, each output from its own divider (see Figure 1-1)

NOTE: All output clocks and dividers might not be used or be programmable on all devices. For more information about what PLL Controller outputs and registers are enabled, see the device-specific data manual.

For more details about the PLL, see the device-specific data manual.

Functional Description

The following sections describe the multiplier and dividers in the PLL Controller as well as the bypass mode and PLL mode operation.

Topic **Page**

2.1 Dividers

The clock dividers (PLLDIV1 to PLLDIV16) are programmable in a range from ÷1 to ÷256 and may be disabled. When a clock divider is disabled, no clock is output from that clock divider. A divider outputs a clock only when it is enabled in the corresponding PLLDIV *n* register.

PLLD is a divider inside the PLL. For more details, see the device-specific data manual.

2.2 Multiplier

The PLLM register in the PLL Controller is used to control the PLLM logic of the PLL. For more details see the device-specific data manual.

2.3 PLL Control Register and Secondary Control Register

After device reset, the value of the PLL enable bit (PLLEN) in the PLL control register (PLLCTL) can be changed, but it will not have any effect on the function of the PLL Controller. To enable the PLLEN bit, the PLLENSRC bit (also in the PLLCTL register) must first be cleared to 0. Once the PLLEN bit has been enabled, it can be used to select the bypass mode or PLL mode of the PLL Controller as discussed in the next two sections. The PLLRST bit in PLLCTL is used to reset the PLL Controller.

The Secondary Control register (SECCTL) is used to drive the OUTPUT DIVIDE and BYPASS logic of the PLL.

The PLL can be operated in Bypass or PLL mode based on the status of the BYPASS, PLLENSRC, and PLLEN bits in the PLLCTL and SECCTL registers and is discussed in the next two sections.

2.4 Bypass Mode

When BYPASS = 1 (bypass enabled in the PLL Mux) i.e. in bypass mode, the PLLM, PLLD, and OUTPUT DIVIDE logic of the PLL are bypassed and the input reference clock to the PLL (CLKIN) is input directly to the PLL Controller (see Figure 1-1). The PLL block is operating in Bypass Mode.

When PLLENSRC=0 and PLLEN=0 (bypass enabled in the PLL Controller mux), the entire PLL block is bypassed and the reference input from the PLL is fed as a direct input to the PLL Controller. The PLL Controller block is operating in Bypass Mode.

CAUTION

The PLL comes up in Bypass mode by default on powering up the device. Once the PLL is initialized in PLL mode, it should not be re-initialized back to Bypass mode unless the user intends to power down the device or re-program the PLL to a higher clock rate.

2.5 PLL Mode

When BYPASS = 0 (in PLL Mux) i.e. in PLL mode, the PLLM, PLLD, and OUTPUT DIVIDE logic of the PLL are used (see Figure 1-1). The output of the PLL (PLLOUT) is fed as an input to the PLL Controller. The PLL block is operating in PLL Mode.

When PLLENSRC=0 and PLLEN=1 (in the PLL Controller mux), the output of the PLL (PLLOUT) is fed as an input to the PLL Controller (see Figure 1-1). The PLL Controller block is operating in PLL Mode.

Further, when enabled (DnEN = 1), the system clock dividers (D1-D16) divide down by the RATIO value in PLLDIVn, the output clock of the PLL. The system clock dividers generate a 50% duty cycle output clock SYSCLKn.

Configuration

The following sections provide procedures for initialization, power down, and wake up of the PLL Controller.

Topic **Page**

3.1 Main PLL Initialization Sequence

The PLL and PLL Controller are initialized by software after reset. The PLL Controller registers should be modified only by the CPU or emulation. External masters, for example PCIe, should not be used to access the PLL Controller registers directly. The initialization of the PLL Controller should be performed as soon as possible at the beginning of the program, before initializing any peripherals. Upon device reset, the following software initialization procedures must be done to properly set up the PLL and PLL Controller.

PLL configuration registers (MAINPLLCTL0 and MAINPLLCTL1), located in the Bootcfg space, are write-protected at power-up. Software must first un-lock the KICK0 and KICK1 registers prior to writing to any chip-level registers. Software must lock the KICK0 and KICK1 registers after writing to any chip-level registers to prevent any unintentional changes. See the device-specific data manual for MAINPLLCTL0 and MAINPLLCTL1 register definitions and the KICK register un-locking procedure.

Users must ensure that applicable power domains are enabled before enabling specific PLLs. For example, before enabling the ARM PLL, the ARM power domains must be enabled. See the device-specific data manual for details on enabling/disabling power domains.

3.1.1 Initialization to PLL Mode

This section shows the initialization sequence if the user intends to use the PLL in PLL mode. The steps below also show when you should program the multipliers, divider, and system clock dividers, if required.

NOTE: Perform each step in sequence unless directed to jump directly to another step.

1. If executing this sequence immediately after device power-up, you must allow time for the PLL to become stable. PLL stabilization time is 100 µs.
2. Check the status of BYPASS bit in SECCTL register, execute following steps if BYPASS == 1 (if bypass enabled), if BYPASS == 0 then **Jump to Step 3**
   (a) In MAINPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
   (b) In PLLCTL, write PLLEN = 0 (bypass enabled in PLL Controller mux)
   (c) In PLLCTL, write PLLENSRC = 0 (enable PLLEN to control PLL Controller mux)
   (d) Wait 4 cycles of the reference clock CLKIN (to make sure the PLL Controller mux switches properly to the bypass)
   (e) In SECCTL, write BYPASS = 1 (bypass enabled in PLL mux)
   (f) In PLLCTL, write PLLPWRDN = 1 (power down the PLL)
   (g) Wait for at least 5 µs based on the reference clock CLKIN (PLL power down toggling time)
   (h) In PLLCTL, write PLLPWRDN = 0 (power up the PLL. **Jump to Step 4**
3. Enable BYPASS in the PLL Controller
   (a) In PLLCTL, write PLLEN = 0 (bypass enabled in PLL Controller mux)
   (b) In PLLCTL, write PLLENSRC = 0 (enable PLLEN to control PLL Controller mux)
   (c) Wait 4 cycles of the reference clock CLKIN (to make sure the PLL Controller mux switches properly to the bypass)
4. PLLM is split in two different registers. Program PLLM[5:0] in PLL multiplier control register (PLLM) and PLLM[12:6] in MAINPLLCTL0 register
5. BWADJ is split in two different registers. Program BWADJ[7:0] in MAINPLLCTL0 and BWADJ[11:8] in MAINPLLCTL1 register. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1) >> 1) - 1.
6. Program PLLD in MAINPLLCTL0 register
7. In SECCTL, write OD (Output Divide) = 1 (that is divide-by-2)
8. If necessary, program PLLDIVn. Do not re-program the PLLDIVn values if the PLL has previously been placed in PLL mode and is being re-programmed during this initialization. Note that you must apply the GO operation to change these dividers to a new ratios (see Section 3.2).
9. In PLLCTL, write PLLRST = 1 (PLL reset is asserted)
10. Wait for at least 7 µs based on the reference clock CLKIN (PLL reset time)
11. In PLLCTL, write PLLRST = 0 (PLL reset is de-asserted)
12. Wait for at least 500 × CLKIN cycles × (PLLD + 1) (PLL lock time)
13. In SECCTL, write BYPASS = 0 (enable PLL mux to switch to PLL mode)
14. In PLLCTL, write PLLEN = 1 (enable PLL Controller mux to switch to PLL mode)
15. The PLL and PLL Controller are now initialized in PLL mode.

CAUTION

Software must always perform read-modify-write to any register in the PLL. This is to ensure that only the relevant bits in the register are modified and the rest of the bits including the reserved bits are not affected.

3.1.2 Initialization to Bypass Mode

CAUTION

The PLL comes up in bypass mode by default on powering up the device. Once the PLL is initialized in PLL mode, it should not be re-initialized back to bypass mode unless the user intends to power down the device or re-program the PLL to a higher clock rate.

3.1.3 Reprogramming the Main PLL When Operating in PLL Mode

Some device use-cases (such as Power-Over Ethernet) have a power supply topology that places a limit on the transient load that can be placed on the main supply. These use-cases can be fulfilled by staging the PLL frequency bring-up so that transient power draw is reduced. As an example, a user could program the Main PLL to first initialize at half the final device operating frequency, and then after a short period of time reprogram the PLL to final device operating frequency.

PLL reprogramming should only occur during initial device bring-up immediately following the ROM Bootloader (RBL) Initialization. The RBL uses the boot mode pins to determine how to program the first program the PLL. If a user wishes to have the RBL program the PLL to a frequency other than those given as options in the device datasheet (e.g. half the operational frequency), the user can do so by misrepresenting the input clock frequency to the RBL using the boot mode pins. Each frequency setting in the RBL

The PLL reprogramming should follow the RBL initialization the in order to minimize system and/or application impact - the device should be in a low power state, where peripherals should be powered off, and no application-level code should be executing. The sequence for PLL reprogramming is identical to the one presented in Section 3.1.1 with the exception that the PLLDIVn values for SYSCLKn must not be re-programmed.

3.2 Divider n (D1 to Dn) and GO Operation

The GO operation is required to change the divider ratios of the PLLDIVn registers. Section 3.2.1 discusses the GO operation. Section 3.2.2 provides the software steps required to change the divider ratios.

3.2.1 GO Operation

The GO operation writes to the RATIO field in the PLLDIV *n.* Registers do not change the dividers' divide ratios immediately. The PLLDIV *n* dividers change to the new RATIO rates only during a GO operation. This section discusses the GO operation and alignment of the SYSCLKs.

The PLL Controller clock align control register (ALNCTL) determines which SYSCLKs must be aligned. Before a GO operation, program ALNCTL so that the appropriate clocks are aligned during the GO operation.

A GO operation is initiated by setting the GOSET bit in PLLCMD to 1. During a GO operation:

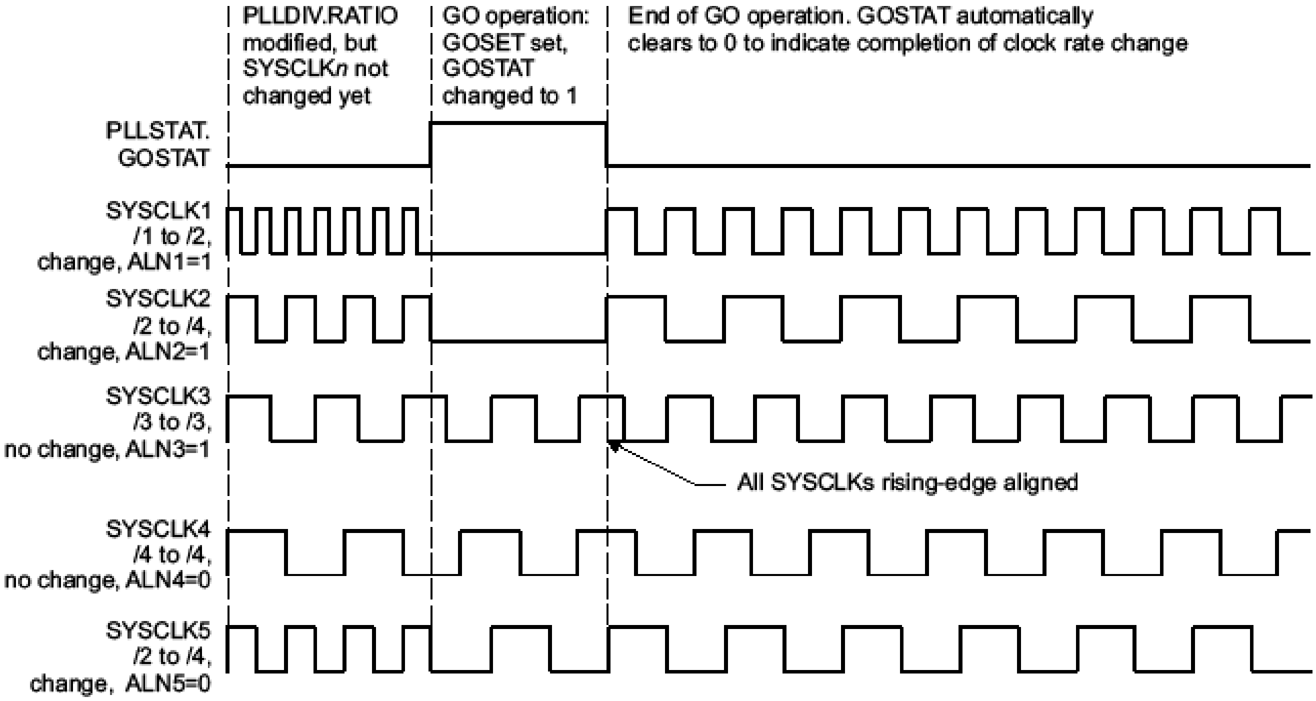
- Any SYSCLK *n* with the corresponding ALN *n* bit in ALNCTL and SYS *n* bit in DCHANGE set to 1 is paused at the low edge. Then the PLL Controller restarts all these SYSCLKs simultaneously, aligned at the rising edge. When the SYSCLKs are restarted, SYSCLK *n* toggles at the rate programmed in the RATIO field in PLLDIV *n*.
- Any SYSCLK *n* with the corresponding ALN *n* bit in ALNCTL cleared and the SYS *n* bit in DCHANGE set immediately changes to the new rate programmed in the RATIO field.
- The GOSTAT bit in PLLSTAT is set throughout the duration of a GO operation.

CAUTION

To help prevent errors, all device operation must be stopped before the GO operation.

Figure 3-1 shows an example of how the clocks are rising-edge aligned during a GO operation. Notice that because SYSCLK5 does not need to be aligned with the other clocks, it immediately switches to its new ratio during the GO operation.

Figure 3-1. Example Clock Ratio Change and Alignment with GO Operation


PLLDIV.RATIO modified, but SYSCLKn not changed yet
GO operation: GOSET set, GOSTAT changed to 1
End of GO operation. GOSTAT automatically clears to 0 to indicate completion of clock rate change
PLLSTAT. GOSTAT
SYSCLK1 /1 to /2, change, ALN1=1
SYSCLK2 /2 to /4, change, ALN2=1
SYSCLK3 /3 to /3, no change, ALN3=1
All SYSCLKs rising-edge aligned
SYSCLK4 /4 to /4, no change, ALN4=0
SYSCLK5 /2 to /4, change, ALN5=0

3.2.2 *Software Steps to Modify PLLDIVn Ratios*

Perform the following steps to modify PLLDIV *n*.

1. Check that the GOSTAT bit in PLLSTAT is cleared to show that no GO operation is currently in progress.
2. Program the RATIO field in PLLDIV *n* to the desired new divide-down rate. If the RATIO field changed, the PLL Controller will flag the change in the corresponding bit of DCHANGE.
3. Set the respective ALN *n* bits in ALNCTL to align any SYSCLKs after the GO operation.
4. Set the GOSET bit in PLLCMD to initiate the GO operation to change the divide values and align the SYSCLKs as programmed.
5. Read the GOSTAT bit in PLLSTAT to make sure the bit returns to 0 to indicate that the GO operation has completed.

3.3 Main PLL Power Down

The PLL may be powered down, in which case the PLL is in bypass mode and the DSP runs from a divided-down version of the input reference clock. The DSP is able to respond to events because it is still being clocked by the bypass clock (directly from CLKIN), although at a lower frequency.

Perform the following procedure to power down the PLL:

1. In SECCTL, write BYPASS = 1 (bypass mode).
2. Wait 4 cycles of the slower of PLLOUT or CLKIN.
3. In PLLCTL, write PLLPWRDN = 1 to power down the PLL.

The above sequence assumes that the device has been powered up long enough that the PLL stabilization time has been met. If executing this sequence immediately after device power-up, you must allow time for the PLL to become stable before performing these steps. For PLL stabilization time, see the device-specific data manual.

3.4 Main PLL Wake Up

Perform the following procedure to wake up the PLL from its power-down mode.

1. In SECCTL, write BYPASS = 1 (bypass mode).
2. In PLLCTL, write PLLPWRDN = 0 to wake up the PLL.
3. Follow the PLL reset sequence in Section 3.1.1 (steps 3 to 9) to reset the PLL. Wait for the PLL to lock and to switch from bypass to PLL mode.

3.5 DDR3 PLL Initialization Sequence

The Main PLL and PLL Controller must always be initialized prior to initializing the DDR3 PLL. The sequence shown below must be followed to initialize the DDR3 PLL.

1. In DDR3PLLCTL1, write ENSAT = 1 (for optimal PLL operation)
2. In DDR3PLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
3. Program PLLM and PLLD in the DDR3PLLCTL0 register
4. Program BWADJ[7:0] in DDR3PLLCTL0 and BWADJ[11:8] in the DDR3PLLCTL1 register. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1) >> 1) - 1.
5. In DDR3PLLCTL1, write PLLRST = 1 (PLL is asserted)
6. Wait for at least 5 µs based on the reference clock (PLL reset time)
7. In DDR3PLLCTL1, write PLLRST = 0 (PLL reset is de-asserted)
8. Wait for at least 500 *REFCLK cycles * (PLLD + 1) (PLL lock time)
9. In DDR3PLLCTL0, write BYPASS = 0 (switch to PLL mode)
10. DDR PLL is now initialized

CAUTION

The DDR interface needs to reset every time the DDR PLL is re-programmed.

3.6 PASS PLL Initialization Sequence

The Main PLL and PLL Controller must always be initialized prior to initializing the PASS PLL. The sequence shown below must be followed to initialize the PASS PLL.

1. In PASSPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
2. In PASSPLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
3. Program PLLM and PLLD in the PASSPLLCTL0 register
4. Program BWADJ[7:0] in PASSPLLCTL0 and BWADJ[11:8] in the PASSPLLCTL1 register. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1) >> 1) - 1.
5. In PASSPLLCTL1, write PLLRST = 1 (PLL is asserted)
6. Wait for at least 5 µs based on the reference clock (PLL reset time)
7. In PASSPLLCTL1, write PLLSELECT = 1 (for selecting the output of PASS PLL as the input to PASS)
8. In PASSPLLCTL1, write PLLRST = 0 (PLL reset is de-asserted)
9. Wait for at least 500 * REFCLK cycles * (PLLD + 1) (PLL lock time)
10. In PASSPLLCTL0, write BYPASS = 0 (switch to PLL mode)
11. PASS PLL is now initialized

3.7 ARM PLL Initialization Sequence

The Main PLL and PLL Controller must always be initialized prior to initializing the ARM PLL. The sequence shown below must be followed to initialize the ARM PLL.

1. Enable bypass glitchfree mux
   - For K2HK: In CHIP_MISC_CTL1, write TETRIS_PLL_EN= 0
   - For K2L: in ARMPLLCTL1 write BIT[15] = 0
2. Set CHIPMISCCTL1[13]=0 (enable glitchfree bypass)
3. In ARMPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
4. In ARMPLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
5. Program PLLM, PLLD, and PLLOD in the ARMPLLCTL0 register
6. Program BWADJ[7:0] in ARMPLLCTL0 and BWADJ[11:8] in the ARMPLLCTL1 register. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation:
   BWADJ = ((PLLM+1) >> 1) - 1.
7. In ARMPLLCTL1, write PLLRST = 1 (PLL is asserted)
8. Wait for at least 5 µs based on the reference clock (PLL reset time)
9. In ARMPLLCTL1, write PLLRST = 0 (PLL reset is deasserted)
10. Wait for at least 500 * REFCLK cycles * (PLLD + 1) (PLL lock time)
11. In ARMPLLCTL0, write BYPASS = 0 (switch to PLL mode)
12. Select the output of ARM PLL as the input to ARM
    - For K2HK: In CHIP_MISC_CTL1, write TETRIS_PLL_EN= 1
    - For K2L: in ARMPLLCTL1 write BIT[15] = 1
13. ARM PLL is now initialized

3.8 DFE PLL Initialization Sequence

The Main PLL and PLL Controller must always be initialized prior to initializing the DFE PLL. The sequence shown below must be followed to initialize the DFE PLL.

1. In DFEPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
2. In DFEPLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
3. Program PLLM and PLLD in the DFEPLLCTL0 register
4. Program BWADJ[7:0] in DFEPLLCTL0 and BWADJ[11:8] in the DFEPLLCTL1 register. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1) >> 1) - 1.
5. In DFEPLLCTL1, write PLLRST = 1 (PLL is asserted)
6. Wait for at least 5 µs based on the reference clock (PLL reset time)
7. In DFEPLLCTL1, write DFEPLL= 1 (for selecting the output of DFE PLL as the input to DFE)
8. In DFE_CLKDIV_CTL , set the DIVMODE according to desired operating frequency for DFE.
9. In DFEPLLCTL1, write PLLRST = 0 (PLL reset is de-asserted)
10. Wait for at least 500 * REFCLK cycles * (PLLD + 1) (PLL lock time)
11. In DFEPLLCTL0, write BYPASS = 0 (switch to PLL mode)
12.
13. (Only perform this step if re-initializing the DFE PLL outside of chip reset such as when re-synchronizing the DFE) In DFE_CLKSYNC_CTL, write SYNC_EN = 0 to disable the sync logic. Wait 1000 PLL output cycles.
14. In DFE_CLKSYNC_CTL, write SYNC_EN=1 to enable the sync logic.
15. DFE PLL is now initialized

CAUTION

Software must always perform Read-modify-write to any register in the Main PLL, PLL Controller, DDR PLL, PASS PLL, DFE PLL, or ARM PLL. This is to ensure that only the relevant bits in the register are modified and the rest of the bits, including the reserved bits, are not affected.

PLL Controller Registers

Topic Page

4.1 PLL Controller Registers

The PLL Controller registers are listed in Table 4-1. For the memory address of these registers, see the device-specific data manual. All other register offset addresses not listed in Table 4-1 should be considered as reserved locations and the register contents should not be modified.

NOTE: This section includes a list of all the PLL Controller registers. However, depending on the PLL being programmed, not all the registers may be used. Furthermore, the reset values for the fields within the registers may be different. For more details, see the device-specific data manual.

Table 4-1. PLL Controller Registers

Offset	Acronym	Register Description	Section
E4h	RSTYPE	Reset Type Status Register	Section 4.11
E8h	RSCTRL	Reset Control Register	Section 4.12
ECh	RSCFG	Reset Configuration Register	Section 4.13
F0h	RSISO	Reset Isolation Register	Section 4.14
100h	PLLCTL	PLL Control Register	Section 4.2
108h	SECCTL	PLL Secondary Control Register	Section 4.3
110h	PLLM	PLL multiplier Control Register	Section 4.4
118h	PLLDIV1	PLL Controller Divider 1 Register	Section 4.5
11Ch	PLLDIV2	PLL Controller Divider 2 Register	Section 4.5
120h	PLLDIV3	PLL Controller Divider 3 Register	Section 4.5
138h	PLLCMD	PLL Controller Command Register	Section 4.6
13Ch	PLLSTAT	PLL Controller Status Register	Section 4.7
140h	ALNCTL	PLL Controller Clock Align Control Register	Section 4.8
144h	DCHANGE	PLLDIV Divider Ratio Change Status Register	Section 4.9
150h	SYSTAT	Sysclk Status Register	Section 4.10
160h	PLLDIV4	PLL Controller Divider 4 Register	Section 4.5
164h	PLLDIV5	PLL Controller Divider 5 Register	Section 4.5
168h	PLLDIV6	PLL Controller Divider 6 Register	Section 4.5
16Ch	PLLDIV7	PLL Controller Divider 7 Register	Section 4.5
170h	PLLDIV8	PLL Controller Divider 8 Register	Section 4.5
174h-193h	PLLDIV9-PLLDIV16	PLL Controller Divider 9 To 16 Registers	Section 4.5

4.2 PLL Control Register (PLLCTL)

The PLL control register (PLLCTL) is shown in Figure 4-1 and described in Table 4-2.

Figure 4-1. PLL Control Register (PLLCTL)

31 6	5	4	3	2	1	0
Reserved	PLLENSRC	Reserved	PLLRST	Reserved	PLLPWRDN	PLLEN
R/W-n	R-n	R-n	R/W-n	R-n	R/W-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-2. PLL Control Register (PLLCTL) Field Descriptions

Bit	Field	Description
31-6	Reserved	Reserved.
5	PLLENSRC	PLL enable source bit. • 0 = PLLEN bit is enabled. The value of the PLLEN bit will affect the operation of the PLL Controller. • 1 = PLLEN bit is disabled. The value of the PLLEN bit has no effect on the operation of the PLL Controller.
4	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
3	PLLRST	PLL reset bit. • 0 = PLL reset is released. • 1 = PLL reset is asserted.
2	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	PLLPWRDN	PLL power-down mode select bit. • 0 = PLL is operational. • 1 = PLL is placed in a power-down state; i.e., all analog circuitry in the PLL is turned off.
0	PLLEN.	PLL enable bit. • 0 = Bypass mode. • 1 = PLL mode.

4.3 PLL Secondary Control Register (SECCTL)

The PLL Secondary Control Register is shown in Figure 4-2 and described in Table 4-3.

Figure 4-2. PLL Secondary Control Register (SECCTL)

31 24	23	22 19	18 0
Reserved	BYPASS	OUTPUT DIVIDE	Reserved
R-n	RW-n	RW-n	RW-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-3. PLL Secondary Control Register (SECCTL) Field Descriptions

Bit	Field	Description
31-24	Reserved	Reserved.
23	BYPASS	PLL Bypass Enable. • 0 = PLL Bypass disabled. • 1 = PLL Bypass enabled.
22-19	OUTPUT DIVIDE	Output Divider ratio bits. For Keystone I devices, 1h = Divide frequency by 2 (default) for Keystone II devices, 0h = Divide frequency by 1 1h-Fh = Divide frequency by 2, to divide frequency by 16 (only even values)
18-0	Reserved	Reserved.

4.4 PLL Multiplier Control Register (PLLM)

The PLL multiplier control register (PLLM) is shown in Figure 4-3 and described in Table 4-4. The MSB bits PLLM[12:6] come from the chip-level register (MAINPLLCTL0), see the device-specific data manual for more details.

NOTE: Table 4-4 lists all the possible values for the PLL multiplier bits (PLLM). However, some of these values may not be valid for your particular device. For a list of valid values for PLLM, see the device-specific data manual.

Figure 4-3. PLL Multiplier Control Register (PLLM)

31 6	5 0
Reserved	PLLM
R-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-4. PLL Multiplier Control Register (PLLM) Field Descriptions

Bit	Field	Description
31-6	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
5-0	PLLM	PLL multiplier bits. Defines the frequency multiplier of the input reference clock. • 0h = ×1 multiplier rate. • 1h = ×2 multiplier rate. • 2h = ×3 multiplier rate. • 3h = ×4 multiplier rate. • 4h - 3Fh = ×5 multiplier rate to ×64 multiplier rate.

4.5 PLL Controller Divider Register (PLLDIV1-PLLDIV16)

The PLL Controller divider registers (PLLDIV1-PLLDIV16) are shown in Figure 4-4 and described in Table 4-5. The PLLDIV *n* dividers generate a 50% duty cycle output clock SYSCLK *n* when enabled. The PLLDIVn registers contain the default divider values on power up. The user will need to re-program a new value only if they desire to change the default values. Otherwise, this register can be left unchanged. Also, not all PLLDIVn registers are programmable. See the device-specific data manual for more information on programmable PLLDIVn registers and their default values.

Figure 4-4. PLL Controller Divider Register (PLLDIVn)

31 — 16	15	14 — 8	7 — 0
Reserved	D*n*EN	Reserved	RATIO
R-n	R/W-n	R-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-5. PLL Controller Divider Register (PLLDIVn) Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15	D*n*EN	Divider D *n* enable bit. • 0 = Divider *n* is disabled. • 1 = Divider *n* is enabled.
14-8	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7-0	RATIO	Divider ratio bits. • 0h = ÷1. Divide frequency by 1. • 1h = ÷2. Divide frequency by 2. • 2h = ÷3. Divide frequency by 3. • 3h = ÷4. Divide frequency by 4. • 4h - FFh = ÷5 to ÷256 Divide frequency by 5 to divide frequency by 256.

4.6 PLL Controller Command Register (PLLCMD)

The PLL Controller command register (PLLCMD) contains the command bit for the GO operation. PLLCMD is shown in Figure 4-5 and described in Table 4-6.

Figure 4-5. PLL Controller Command Register (PLLCMD)

31 2	1	0
Reserved	Rsvd	GOSET
R-n	R/W-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-6. PLL Controller Command Register (PLLCMD) Field Descriptions

Bit	Field	Description
31-2	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GOSET	GO operation command for SYSCLK rate change and phase alignment. Before setting this bit to initiate a GO operation, check the GOSTAT bit in the PLLSTAT register to ensure all previous GO operations have completed. • 0 = No effect. Write of 0 clears bit. • 1 = Initiates GO operation. Write of 1 initiates GO operation. Once set, GOSET remains set but further writes of 1 can reinitiate the GO operation.

4.7 PLL Controller Status Register (PLLSTAT)

The PLL Controller status register (PLLSTAT) shows the PLL Controller status. PLLSTAT is shown in Figure 4-6 and described in Table 4-7.

Figure 4-6. PLL Controller Status Register (PLLSTAT)

31 1	0
Reserved	GOSTAT
R-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-7. PLL Controller Status Register (PLLSTAT) Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GOSTAT	GO operation status. • 0 = GO operation is not in progress. SYSCLK divide ratios are not being changed. • 1 = GO operation is in progress. SYSCLK divide ratios are being changed.

4.8 PLL Controller Clock Align Control Register (ALNCTL)

The PLL Controller clock align control register (ALNCTL) is shown in Figure 4-7 and described in Table 4-8.

NOTE: The default value of this register should not be changed.

Figure 4-7. PLL Controller Clock Align Control Register (ALNCTL)

31–16
Reserved
R-n

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALN16	ALN15	ALN14	ALN13	ALN12	ALN11	ALN10	ALN9	ALN8	ALN7	ALN6	ALN5	ALN4	ALN3	ALN2	ALN1
R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-8. PLL Controller Clock Align Control Register (ALNCTL) Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved. This reserved bit location is always read as 0. A value written to this field has no effect.
15-0	ALN *n*	SYSCLK *n* alignment. Do not change the default values of these fields. • 0 = Do not align SYSCLK *n* to other SYSCLKs during GO operation. If SYS *n* in DCHANGE is set, SYSCLK *n* switches to the new ratio immediately after the GOSET bit in PLLCMD is set. • 1 = Align SYSCLK *n* to other SYSCLKs selected in ALNCTL when the GOSET bit in PLLCMD is set and SYS *n* in DCHANGE is 1. The SYSCLK *n* rate is set to the ratio programmed in the RATIO bit in PLLDIV *n*.

4.9 PLLDIV Divider Ratio Change Status Register (DCHANGE)

When a different ratio is written to the PLLDIV *n* registers, the PLLCTL flags the change in the DCHANGE status register. During the GO operation, the PLL

Controller changes only the divide ratio of the SYSCLKs with the bit set in DCHANGE. Note that the ALNCTL register determines if that clock also needs to be aligned to other clocks. The PLLDIV divider ratio change status register is shown in Figure 4-8 and described in Table 4-9.

Figure 4-8. PLLDIV Divider Ratio Change Status Register (DCHANGE)

31–16
Reserved
R-n

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS16	SYS15	SYS14	SYS13	SYS12	SYS11	SYS10	SYS9	SYS8	SYS7	SYS6	SYS5	SYS4	SYS3	SYS2	SYS1
R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n	R/W-n

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-9. PLLDIV Divider Ratio Change Status Register (DCHANGE) Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved. This reserved bit location is always read as 0. A value written to this field has no effect.
15-0	SYS *n*	Identifies when the SYSCLK *n* divide ratio has been modified. • 0 = SYSCLK *n* ratio has not been modified. When GOSET is set, SYSCLK *n* will not be affected. • 1 = SYSCLK *n* ratio has been modified. When GOSET is set, SYSCLK *n* will change to the new ratio.

4.10 SYSCLK Status Register (SYSTAT)

The SYSCLK status register (SYSTAT) shows the status of SYSCLKn. SYSTAT is shown in Figure 4-9 and described in Table 4-10.

Figure 4-9. SYSCLK Status Register (SYSTAT)

<table>
<tr><td colspan="2">31</td><td colspan="2">16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td></tr>
<tr><td colspan="4">Reserved</td><td>SYS16ON</td><td>SYS15ON</td><td>SYS14ON</td><td>SYS13ON</td><td>SYS12ON</td><td>SYS11ON</td></tr>
<tr><td colspan="4">R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td></tr>
<tr><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>SYS10ON</td><td>SYS9ON</td><td>SYS8ON</td><td>SYS7ON</td><td>SYS6ON</td><td>SYS5ON</td><td>SYS4ON</td><td>SYS3ON</td><td>SYS2ON</td><td>SYS1ON</td></tr>
<tr><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td><td>R-n</td></tr>
</table>

Legend: R/W = Read/Write; R = Read only; - *n* = value after reset; for reset value, see the device-specific data manual

Table 4-10. SYSCLK Status Register (SYSTAT) Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-0	SYS[N[1]]ON	SYSCLK[N] on status. • 0 = SYSCLK[N] is gated. • 1 = SYSCLK[N] is on.

[1] Where N = 1, 2, 3,....N (Not all these output clocks may be used on a specific device. For more information, see the device-specific data manual)

4.11 Reset Type Status Register (RSTYPE)

The reset type status (RSTYPE) register latches the cause of the last reset. If multiple reset sources occur simultaneously, this register latches the highest priority reset source. The Reset Type Status register is shown in Figure 4-10 and described in Table 4-11.

Figure 4-10. Reset Type Status Register (RSTYPE)

31 29	28	27 16	15 8	7 3	2	1	0
Reserved	EMU-RST	Reserved	WDRST[N[1]]	Reserved	PLLCTRLRST	$\overline{\text{RESET}}$	POR
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

Legend: R = Read only; - *n* = value after reset

[1] Where N = 1, 2, 3,....N (Not all these output may be used on a specific device. For more information, see the device-specific data manual)

Table 4-11. Reset Type Status Register (RSTYPE) Field Descriptions

Bit	Field	Description
31-29	Reserved	Reserved. Read only. Always reads as 0. Writes have no effect.
28	EMU-RST	Reset initiated by emulation. • 0 = Not the last reset to occur. • 1 = The last reset to occur.
27-16	Reserved	Reserved. Read only. Always reads as 0. Writes have no effect.
15-8	WDRST[N]	Reset initiated by Watchdog Timer[N]. • 0 = Not the last reset to occur. • 1 = The last reset to occur.
7-3	Reserved	Reserved. Read only. Always reads as 0. Writes have no effect.
2	PLLCTLRST	Reset initiated by PLLCTL. • 0 = Not the last reset to occur. • 1 = The last reset to occur.
1	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$ reset. • 0 = $\overline{\text{RESET}}$ was not the last reset to occur. • 1 = $\overline{\text{RESET}}$ was the last reset to occur.
0	POR	Power-on reset. • 0 = Power-on reset was not the last reset to occur. • 1 = Power-on reset was the last reset to occur.

4.12 Reset Control Register (RSTCTRL)

This register contains a key that enables writes to the MSB of this register and the RSTCFG register. The key value is 0x5A69. A valid key will be stored as 0x000C, any other key value is invalid. When the RSTCTRL or the RSTCFG is written, the key is invalidated. Every write must be set up with a valid key. The Software Reset Control register (RSTCTRL) is shown in Figure 4-11 and described in Table 4-12.

Figure 4-11. Reset Control Register (RSTCTRL)

31 17	16	15 0
Reserved	SWRST	KEY
R-0x0000	R/W-0x[1]	R/W-0x0003

Legend: R = Read only; - *n* = value after reset;

[1] Writes are conditional based on valid key.

Table 4-12. Reset Control Register (RSTCTRL) Field Descriptions

Bit	Field	Description
31-17	Reserved	Reserved.
16	SWRST	Software reset. • 0 = Reset. • 1 = Not reset.
15-0	KEY	Key used to enable writes to RSTCTRL and RSTCFG.

4.13 Reset Configuration Register (RSTCFG)

This register is used to configure the type of reset initiated by $\overline{\text{RESET}}$, the watchdog timer, and the PLL Controller's RSTCTRL register: a hard reset or a soft reset. By default, these resets are hard resets. The Reset Configuration Register (RSTCFG) is shown in Figure 4-12 and described in Table 4-13.

Figure 4-12. Reset Configuration Register (RSTCFG)

31 16	15 14	13	12	11 8	7 0
Reserved	Reserved	PLLCTLRST TYPE	$\overline{\text{RESET}}$TYPE	Reserved	WDTYPE[N[1]]
R-0x0000	R-00	R/W-0[1]	R/W-0[1]	R-0x0	R/W-0x00[1]

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

[1] Where N = 1, 2, 3,....N (Not all these output may be used on a specific device. For more information, see the device-specific data manual)

[1] Writes are conditional based on valid key. For details, see Section Section 4.12.

Table 4-13. Reset Configuration Register (RSTCFG) Field Descriptions

Bit	Field	Description
31-14	Reserved	Reserved.
13	PLLCTLRSTTYPE	PLL Controller initiates a software driven reset of type: • 0 = Hard reset (default) • 1 = Soft reset
12	$\overline{\text{RESET}}$TYPE	$\overline{\text{RESET}}$ initiates a reset of type: • 0 = Hard reset (default) • 1 = Soft reset
11-8	Reserved	Reserved.
7-0	WDTYPE[N]	Watchdog Timer [N] initiates a reset of type: • 0 = Hard reset (default) • 1 = Soft reset

4.14 Reset Isolation Register (RSISO)

This register is used to select the module clocks that must maintain their clocking without pausing through non power-on reset. Setting any of these bits effectively blocks reset to all PLLCTL registers in order to maintain current values of PLL multiplier ratios, divide ratios, and other settings. The Reset Isolation register (RSTCTRL) is shown in Figure 4-13 and described in Table 4-14.

NOTE: For a list of modules that can be reset-isolated, see the device-specific data manual.

Figure 4-13. Reset Isolation Register (RSISO)

31 16	15 0
Reserved	MOD_ISO[n]
R-0	R/W-0

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

Table 4-14. Reset Isolation Register (RSISO) Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved.
15-0	MOD_ISO[n]	Isolate MOD_ISO[n]. • 0 = Not reset isolated. • 1 = Reset Isolated.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from December 1, 2015 to October 1, 2016 (from G Revision (November 2015) to H Revision)	Page
• Updated ARM PLL Initialization Sequence.	14

Revision History

Changes from May 1, 2013 to November 30, 2015 (from F Revision (May 2013) to G Revision)	Page
• Updated image.	5
• Added Caution to Bypass Mode section.	7
• Updated Main PLL Initialization Sequence section.	9
• Added Step 8 to PLL Initialization Sequence.	10
• Added Reprogramming the Main PLL When Operating in PLL Mode section.	11
• Removed PLL Controller Post-Divide Register.	17
• Updated Bits [22:10] of PLL Secondary Control Register (SECCTL)	19

Revision I History

Changes from H Revision (September 2016) to I Revision	Page
• Added Set CHIPMISCCTL1[13]=0 (enable glitchfree bypass) to new step 2.	14

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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