TMS320C6000 DSP Software-Programmable Phase-Locked Loop (PLL) Controller

Reference Guide

Literature Number: SPRU233C
November 2006

TEXAS INSTRUMENTS

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the operation of the software-programmable phase-locked loop (PLL) controller in the digital signal processors (DSPs) of some of the TMS320C6000™ DSP family. Refer to the device-specific data manual to determine if the PLL controller is used on a particular device.

Notational Conventions

This document uses the following conventions.

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

Related Documentation From Texas Instruments

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

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

SPRU733 — ***TMS320C67x/C67x+ DSP CPU and Instruction Set Reference Guide.*** Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C67x and TMS320C67x+ digital signal processors (DSPs) of the TMS320C6000 DSP platform. The C67x/C67x+ DSP generation comprises floating-point devices in the C6000 DSP platform. The C67x+ DSP is an enhancement of the C67x DSP with added functionality and an expanded instruction set.

SPRU190 — ***TMS320C6000 DSP Peripherals Overview Reference Guide.*** Provides an overview and briefly describes the peripherals available on the TMS320C6000 family of digital signal processors (DSPs).

SPRU197 — ***TMS320C6000 Technical Brief.*** Provides an introduction to the TMS320C62x and TMS320C67x digital signal processors (DSPs) of the TMS320C6000 DSP family. Describes the CPU architecture, peripherals, development tools and third-party support for the C62x and C67x DSPs.

SPRU198 — ***TMS320C6000 Programmer's Guide.*** Reference for programming the TMS320C6000 digital signal processors (DSPs). Before you use this manual, you should install your code generation and debugging tools. Includes a brief description of the C6000 DSP architecture and code development flow, includes C code examples and discusses optimization methods for the C code, describes the structure of assembly code and includes examples and discusses optimizations for the assembly code, and describes programming considerations for the C64x DSP.

SPRU301 —***TMS320C6000 Code Composer Studio Tutorial.*** This tutorial introduces you to some of the key features of Code Composer Studio. Code Composer Studio extends the capabilities of the Code Composer Integrated Development Environment (IDE) to include full awareness of the DSP target by the host and real-time analysis tools. This tutorial assumes that you have Code Composer Studio, which includes the TMS320C6000 code generation tools along with the APIs and plug-ins for both DSP/BIOS and RTDX. This manual also assumes that you have installed a target board in your PC containing the DSP device.

SPRU273 —***TMS320C6x Peripheral Support Library Programmer's Reference.*** Describes the TMS320C6000 digital signal processor (DSP) peripheral support library of functions and macros. The C6000 DSP peripheral support library is a collection of macros and functions for programming the C6000 DSP registers and peripherals using the C programming language. This document serves as a reference for the C programmer in creating code for the C6000 DSP.

SPRU401 —***TMS320C6000 Chip Support Library API Reference Guide.*** Describes the TMS320C6000 chip support library (CSL) that is a set of application programming interfaces (APIs) used to configure and control all on-chip peripherals. CSL is intended to make it easier for developers by eliminating much of the tedious work usually needed to get algorithms up and running in a real system.

Trademarks

Phase-Locked Loop (PLL) Controller

This document describes the operation of the software-programmable phase-locked loop (PLL) controller in the digital signal processors (DSPs) of some of the TMS320C6000™ DSP family. Refer to the device-specific data manual to determine if the PLL controller is used on a particular device.

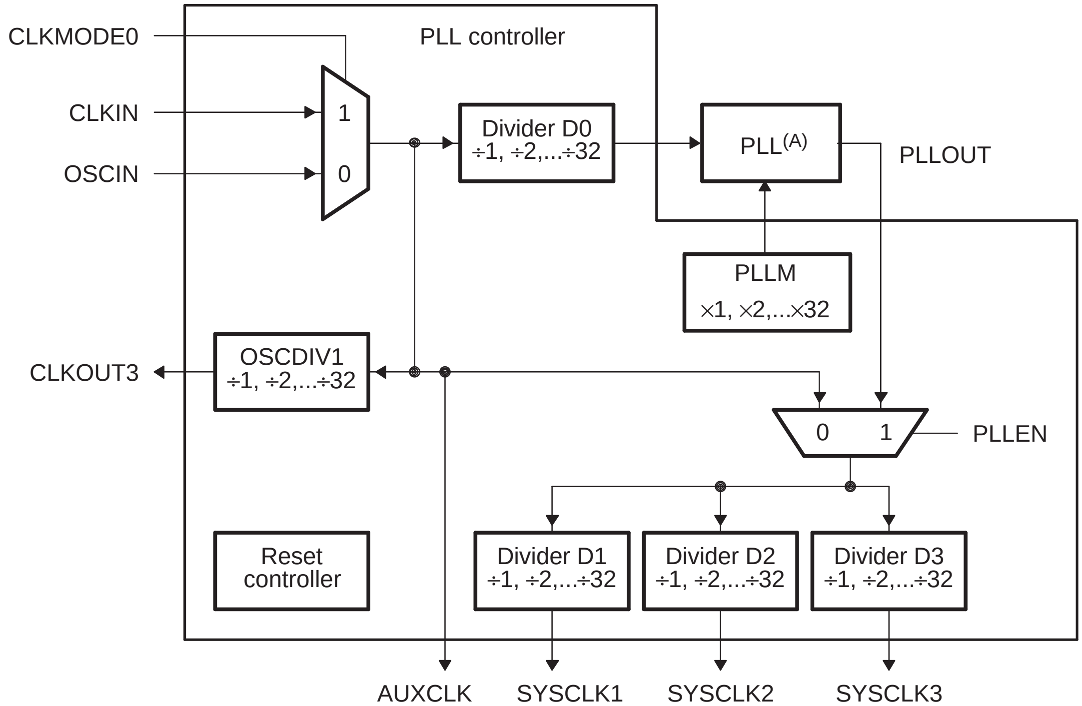
1 Overview

The PLL controller (Figure 1) features software-configurable PLL multiplier controller, dividers (OSCDIV1, D0, D1, D2, and D3), and reset controller. The PLL controller accepts an input clock, as determined by the logic state on the CLKMODE0 pin, from the CLKIN pin or from the on-chip oscillator output signal OSCIN. The PLL controller offers flexibility and convenience by way of software-configurable multiplier and dividers to modify the input signal internally. The resulting clock outputs are passed to the DSP core, peripherals, and other modules inside the C6000™ DSP.

- The input reference clocks to the PLL controller:
  - CLKIN: input signal from external oscillator (3.3V), CLKMODE0 = 1
  - OSCIN: output signal from on-chip oscillator (1.2V), CLKMODE0 = 0
- The resulting output clocks from the PLL controller:
  - AUXCLK: internal clock output signal directly from CLKIN or OSCIN.
  - CLKOUT3: output of divider OSCDIV1.
  - SYSCLK1: internal clock output of divider D1.
  - SYSCLK2: internal clock output of divider D2.
  - SYSCLK3: internal clock output of divider D3.

Refer to your device-specific data manual on how these inputs and outputs of the PLL controller are used. Some devices may not support all of the above clocks. See the device-specific data manual for the clocks supported.

Figure 1. PLL Controller Block Diagram



A See the device-specific data manual for more detail about the PLL.

2 Functional Description

The following sections describe the multiplier, dividers, and reset controller in the PLL controller.

2.1 Multiplier and Dividers

The PLL controller is capable of programming the PLL through the PLL multiplier control register (PLLM) from a ×1 to ×32 multiplier rate (see the device-specific data manual for the PLL multiplier rates supported on your device). The clock dividers (OSCDIV1, D0, D1, D2, and D3) are programmable from ÷1 to ÷32 divider ratio and may be disabled. When a clock divider is disabled, no clock is output from that clock divider. A divider only outputs a clock when it is enabled in the corresponding OSCDIV1 or PLLDIV*n* registers.

The input reference clock (either CLKIN or OSCIN) is directly output as the auxiliary clock (AUXCLK) for use by some peripherals. For example, the multichannel audio serial port (McASP) uses AUXCLK to generate audio clocks. In addition to AUXCLK, the input reference clock is directly input to the oscillator divider (OSCDIV1). If OSCDIV1 is enabled, OD1EN = 1, the input reference clock is divided down by the value in the oscillator divider ratio bits (RATIO) in OSCDIV1. The output from OSCDIV1 is the output clock CLKOUT3.

The divider D0 and the PLL may also be bypassed. The PLL enable bit (PLLEN) in the PLL control/status register (PLLCSR) determines the PLL controller mode. When PLLEN = 1, PLL mode, D0 and PLL are used; when PLLEN = 0, bypass mode, D0 and PLL are bypassed and the input reference clock is directly input to dividers D1, D2, and D3.

When in PLL mode (PLLEN = 1), the input reference clock is supplied to divider D0. If D0 is enabled, D0EN = 1, the input reference clock is divided down by the value in the PLL divider ratio bits (RATIO) in PLLDIV0. The output from divider D0 is input to the PLL. The PLL multiplies the clock by the value in the PLL multiplier bits (PLLM) in the PLL multiplier control register (PLLM). The output from the PLL (PLLOUT) is input to dividers D1, D2, and D3.

When enabled (D*n*EN = 1), the dividers D1, D2, and D3 divide down by the value in RATIO in PLLDIV*n* the output clock of the PLL. The output clocks of dividers D1, D2, and D3 have 50% duty cycle and are SYSCLK1, SYSCLK2, and SYSCLK3, respectively.

2.2 Reset Controller

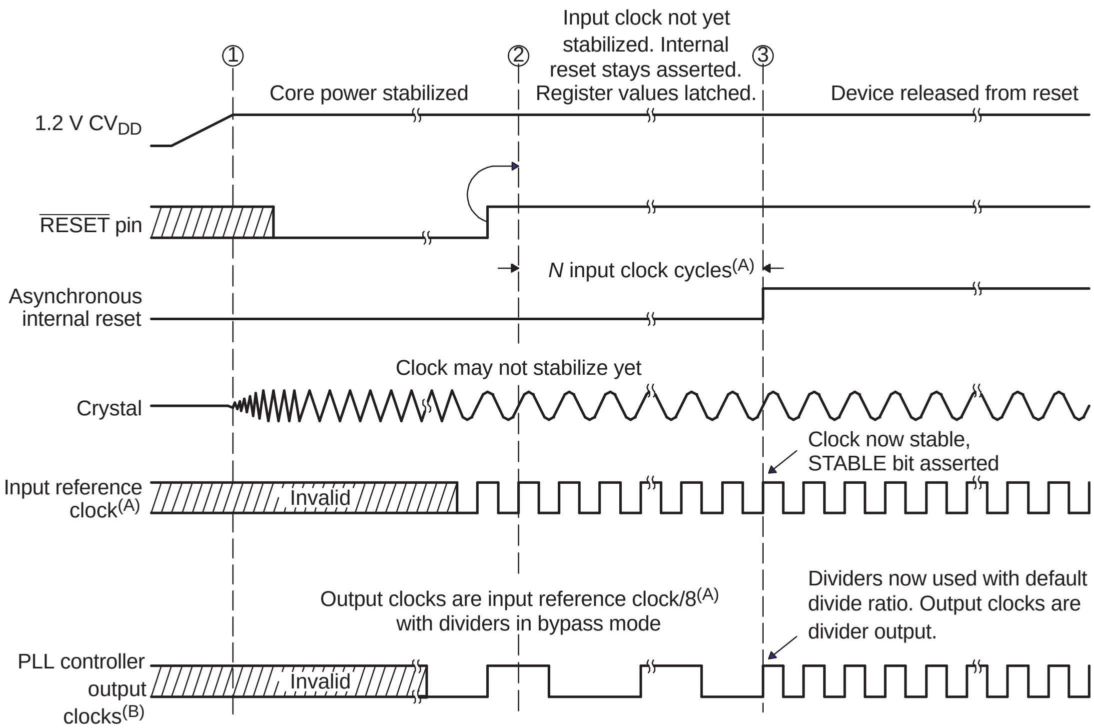
At power up, the device $\overline{\text{RESET}}$ signal may not be asserted long enough to wait for the on-chip or off-chip oscillator to stabilize. This means the input reference clock (either CLKIN or OSCIN) may be a bad clock when a device $\overline{\text{RESET}}$ signal is deasserted high. After $\overline{\text{RESET}}$ is deasserted, the reset controller lengthens the asynchronous internal reset signal to ensure that the input clock source is stable.

The reset controller resides within the PLL controller and the main function is to internally lengthen the reset signal from the $\overline{\text{RESET}}$ input pin until the input clock source is stable (after 512 CLKIN cycles or 4096 OSCIN cycles). This is to ensure that the rest of the device will see the internal reset deasserted only after the input clock is stabilized. Figure 2 shows the lengthening of the internal reset signal.

The PLL controller multiplier and dividers are bypassed when the internal reset signal is low. The frequency of all clock outputs of the PLL controller (AUXCLK, SYSCLK1, SYSCLK2, SYSCLK3, and CLKOUT3) are fixed to the input reference clock (CLKIN or OSCIN) divided by 8. After 512 CLKIN cycles or 4096 OSCIN cycles, the reset controller brings the device out of reset and sets the oscillator input stable bit (STABLE) in the PLL control/status register (PLLCSR). The dividers are used after this point and are set to their default divide ratio.

Values are latched into the registers at the rising edge of $\overline{\text{RESET}}$.

Figure 2. Reset Controller Lengthening the Internal Reset Signal



A N = 512 × CLKIN cycles when CLKMODE = 1 (input reference clock is CLKIN);
N = 4096 × OSCIN cycles when CLKMODE = 0 (input reference clock is OSCIN)

B Output clocks include AUXCLK, SYSCLK1, SYSCLK2, SYSCLK3, and CLKOUT3 at the PLL controller boundary. Refer to the device-specific data manual for clock behavior at the device pins.

3 Configuration

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

3.1 Initialization

The PLL and PLL controller are to be initialized by software after reset. The PLL controller registers should be modified only by the CPU or via emulation. The HPI should not be used to directly access the PLL controller registers. 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, one of the following two software initialization procedures must be done to properly set up the PLL and PLL controller.

3.1.1 Initialization to PLL Mode (PLLEN = 1)

If the system intends to use divider D0 and PLL, perform the following:

1. In PLLCSR, write PLLEN = 0 (bypass mode).
2. Wait 4 cycles of the slowest of PLLOUT, CLKIN (if CLKMODE = 1), and OSCIN (if CLKMODE = 0).
3. In PLLCSR, write PLLRST = 1 (PLL is reset).
4. Program PLLDIV0, PLLM, and OSCDIV1.
5. Program PLLDIV1-3. Note that there must be wait states between accesses to PLLDIV1-3. Each wait state is 8 cycles of the slowest of the old and new SYSCLK1-3 clock rates. Some devices may also require you to program the PLLDIV1-3 registers in a particular order, to ensure no violation of system clock ratios. See PLL Controller section of the device-specific data manual for other restrictions.
6. Wait for PLL to properly reset. See device-specific data manual for PLL reset time.
7. In PLLCSR, write PLLRST = 0 to bring PLL out of reset.
8. Wait for PLL to lock. See device-specific data manual for PLL lock time.
9. In PLLCSR, write PLLEN = 1 to enable PLL mode.

Steps 1, 2, and 3 are required when PLLEN and PLLRST bits are not already 0 and 1, respectively. These steps are not required when the device is coming out of reset (they are performed by hardware).

3.1.2 Initialization to Bypass Mode (PLLEN = 0)

If the system intends to bypass divider D0 and PLL, perform the following:

1. In PLLCSR, write PLLEN = 0 (bypass mode).
2. Wait 4 cycles of the slowest of PLLOUT, CLKIN (if CLKMODE = 1), and OSCIN (if CLKMODE = 0).
3. In PLLCSR, write PLLRST = 1 (PLL is reset).
4. Program OSCDIV1.
5. Program PLLDIV1-3. Note that there must be wait states between accesses to PLLDIV1-3. Each wait state is 8 cycles of the slowest of the old and new SYSCLK1-3 clock rates. See PLL Controller section of the device-specific data manual for other restrictions.

Steps 1, 2, and 3 are required when PLLEN and PLLRST bits are not already 0 and 1, respectively. These steps are not required when the device is coming out of reset (they are performed by hardware).

3.2 PLL Power Down

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

Perform the following procedure to power down the PLL:

1. In PLLCSR, write PLLEN = 0 (bypass mode).
2. Wait 4 cycles of the slowest of PLLOUT, CLKIN (if CLKMODE = 1), and OSCIN (if CLKMODE = 0).
3. In PLLCSR, write PLLPWRDN = 1 to power down the PLL.

3.3 PLL Power Wake Up

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

1. In PLLCSR, write PLLEN = 0 (bypass mode).
2. Wait 4 cycles of the slowest of PLLOUT, CLKIN (if CLKMODE = 1), and OSCIN (if CLKMODE = 0).
3. In PLLCSR, write PLLPWRDN = 0 to wake up the PLL.
4. Follow the reset sequence described in Section 3.1.

4 Registers

The PLL controller registers configure the operation of the PLL controller. The PLL controller registers are listed in Table 1. See the device-specific data manual for the memory address of these registers.

Table 1. PLL Controller Registers

Acronym	Register Description	Section
PLLPID	PLL controller peripheral identification register	Section 4.1
PLLCSR	PLL control/status register	Section 4.2
PLLM	PLL multiplier control register	Section 4.3
PLLDIV0-PLLDIV3	PLL controller divider registers	Section 4.4
OSCDIV1	Oscillator divider 1 register	Section 4.5

4.1 PLL Controller Peripheral Identification Register (PLLPID)

The PLL controller peripheral identification register (PLLPID) contains identification code for the PLL controller. PLLPID is shown in Figure 3 and described in Table 2.

Figure 3. PLL Controller Peripheral Identification Register (PLLPID)

31–24	23–16
Reserved	TYPE
R-0	R-01h

15–8	7–0
CLASS	REV
R-08h	R-x

LEGEND: R = Read only; -*n* = value after reset; -x = value is indeterminate after reset

Table 2. PLL Controller Peripheral Identification Register (PLLPID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TYPE		Identifies type of peripheral
		01h	PLL controller
15-8	CLASS		Identifies class of peripheral
		08h	Serial port
7-0	REV	x	Identifies revision of peripheral. See the device-specific data manual for the default value of this field.

4.2 PLL Control/Status Register (PLLCSR)

The PLL control/status register (PLLCSR) is shown in Figure 4 and described in Table 3.

Figure 4. PLL Control/Status Register (PLLCSR)

31–16
Reserved
R-0

15–7	6	5–4	3	2	1	0
Reserved	STABLE	Reserved	PLLRST	Rsvd	PLLPWRDN	PLLEN
R-0	R-1	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0

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

Table 3. PLL Control/Status Register (PLLCSR) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
6	STABLE		Oscillator input stable bit. Indicates if the OSCIN/CLKIN input has stabilized. The STABLE bit is set to 1 after the reset controller counts 4096 OSCIN or 512 CLKIN input clock cycles after the $\overline{\text{RESET}}$ signal is asserted high. It is assumed that the OSCIN/CLKIN input is stabilized after this number of cycles.
		0	OSCIN/CLKIN input is not yet stable. Oscillator counter is not finished counting.
		1	Oscillator counter is done counting and OSCIN/CLKIN input is assumed to be stable.
5-4	Reserved	0	Reserved. The reserved bit location is always read as 0. Always write a 0 to this location.
3	PLLRST		PLL reset bit.
		0	PLL reset is released.
		1	PLL reset is asserted.
2	Reserved	0	Reserved. The reserved bit location is always read as 0. Always write a 0 to this location.
1	PLLPWRDN		PLL power-down mode select bit.
		0	PLL is operational.
		1	PLL is placed in power-down state.
0	PLLEN		PLL enable bit.
		0	Bypass mode. Divider D0 and PLL are bypassed. SYSCLK1/SYSCLK2/SYSCLK3 are divided down directly from input reference clock.
		1	PLL mode. PLL output path is enabled. Divider D0 and PLL are not bypassed. SYSCLK1/SYSCLK2/SYSCLK3 are divided down from PLL output.

4.3 PLL Multiplier Control Register (PLLM)

The PLL multiplier control register (PLLM) is shown in Figure 5 and described in Table 4. The PLLM defines the input reference clock frequency multiplier in conjunction with the PLL divider ratio bits (RATIO) in the PLL controller divider 0 register (PLLDIV0).

Figure 5. PLL Multiplier Control Register (PLLM)

31–16
Reserved
R-0

15–5	4–0
Reserved	PLLM
R-0	R/W-x

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

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

Bit	Field	Value	Description
31-5	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
4-0	PLLM	0-1Fh	PLL multiplier bits. Defines the frequency multiplier of the input reference clock in conjunction with the PLL divider ratio bits (RATIO) in PLLDIV0. See the device-specific data manual for the PLL multiplier rates supported on your device.

4.4 PLL Controller Divider Registers (PLLDIV0-PLLDIV3)

The PLL controller divider register (PLLDIV0-PLLDIV3) is shown in Figure 6 and described in Table 5.

Figure 6. PLL Controller Divider Register (PLLDIV*n*)

31–16
Reserved
R-0

15	14–5	4–0
D*n*EN	Reserved	RATIO
R/W-1	R-0	R/W-x

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

Table 5. PLL Controller Divider Register (PLLDIV*n*) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	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. No clock output.
		1	Divider *n* is enabled.
14-5	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
4-0	RATIO	0-1Fh	Divider ratio bits.
		0	÷1. Divide frequency by 1.
		1h	÷2. Divide frequency by 2.
		2h-1Fh	÷3 to ÷32. Divide frequency by 3 to divide frequency by 32.

4.5 Oscillator Divider 1 Register (OSCDIV1)

The oscillator divider 1 register (OSCDIV1) is shown in Figure 7 and described in Table 6.

Figure 7. Oscillator Divider 1 Register (OSCDIV1)

31–16
Reserved
R-0

15	14–5	4–0
OD1EN	Reserved	RATIO
R/W-1	R-0	R/W-x

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

Table 6. Oscillator Divider 1 Register (OSCDIV1) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15	OD1EN		Oscillator divider 1 enable bit.
		0	Oscillator divider 1 is disabled. No clock output.
		1	Oscillator divider 1 is enabled.
14-5	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
4-0	RATIO	0-1Fh	Oscillator divider 1 ratio bits. Defines the frequency divider ratio for output clock CLKOUT3.
		0	÷1. Divide frequency by 1.
		1h	÷2. Divide frequency by 2.
		2h-1Fh	÷3 to ÷32. Divide frequency by 3 to divide frequency by 32.