TMS320VC5503/5507/5509/5510 DSP Timers Reference Guide

Literature Number: SPRU595C
April 2006

Texas Instruments

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 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.

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.

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Copyright © 2006, Texas Instruments Incorporated

Preface

Read This First

About This Manual

This manual describes the features and operation of the timers that are on the TMS320VC5503/5507/5509 and TMS320VC5510 digital signal processors (DSPs) in the TMS320C55x™ (C55x™) DSP generation. The TMS320VC5503/5507/5509 device has two general-purpose timers and one watchdog timer. The TMS320VC5510 device has two general-purpose timers.

Chapter 1 describes the TMS320VC5503/5507/5509/5510 general-purpose timer. Chapter 2 describes the TMS320VC5503/5507/5509 watchdog timer.

Notational Conventions

This document uses the following convention.

- In most cases, hexadecimal numbers are shown with the suffix h. For example, the following number is a hexadecimal 40 (decimal 64):

  40h

  Similarly, binary numbers often are shown with the suffix b. For example, the following number is the decimal number 4 shown in binary form:

  0100b

Related Documentation From Texas Instruments

The following documents describe the C55x 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.

TMS320VC5503 Fixed-Point Digital Signal Processor Data Manual (literature number SPRS245) describes the features of the TMS320VC5503 fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

TMS320VC5507 Fixed-Point Digital Signal Processor Data Manual (literature number SPRS244) describes the features of the TMS320VC5507 fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

TMS320VC5509 Fixed-Point Digital Signal Processor Data Manual (literature number SPRS163) describes the features of the TMS320VC5509 fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

TMS320VC5509A Fixed-Point Digital Signal Processor Data Manual (literature number SPRS205) describes the features of the TMS320VC5509A fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual (literature number SPRS076) describes the features of the TMS320VC5510 fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

TMS320C55x Technical Overview (literature number SPRU393). This overview is an introduction to the TMS320C55x DSPs, the latest generation of fixed-point DSPs in the TMS320C5000™ DSP platform. Like the previous generations, this processor is optimized for high performance and low-power operation. This book describes the CPU architecture, low-power enhancements, and embedded emulation features.

TMS320C55x DSP CPU Reference Guide (literature number SPRU371) describes the architecture, registers, and operation of the CPU for the TMS320C55x DSPs.

TMS320C55x DSP Peripherals Overview Reference Guide (literature number SPRU317) introduces the peripherals, interfaces, and related hardware that are available on TMS320C55x DSPs.

TMS320C55x DSP Algebraic Instruction Set Reference Guide (literature number SPRU375) describes the TMS320C55x DSP algebraic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the mnemonic instruction set.

TMS320C55x DSP Mnemonic Instruction Set Reference Guide (literature number SPRU374) describes the TMS320C55x DSP mnemonic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the algebraic instruction set.

TMS320C55x Optimizing C/C++ Compiler User's Guide (literature number SPRU281) describes the TMS320C55x C/C++ Compiler. This C/C++ compiler accepts ISO standard C and C++ source code and produces assembly language source code for TMS320C55x devices.

TMS320C55x Assembly Language Tools User's Guide (literature number SPRU280) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for TMS320C55x devices.

TMS320C55x DSP Programmer's Guide (literature number SPRU376) describes ways to optimize C and assembly code for the TMS320C55x DSPs and explains how to write code that uses special features and instructions of the DSPs.

Trademarks

TMS320, TMS320C5000, TMS320C55x, and C55x are trademarks of Texas Instruments.

All trademarks are the property of their respective owners.

This page is intentionally left blank.

Contents

Figures

Tables

Examples

Chapter 1

General-Purpose Timer

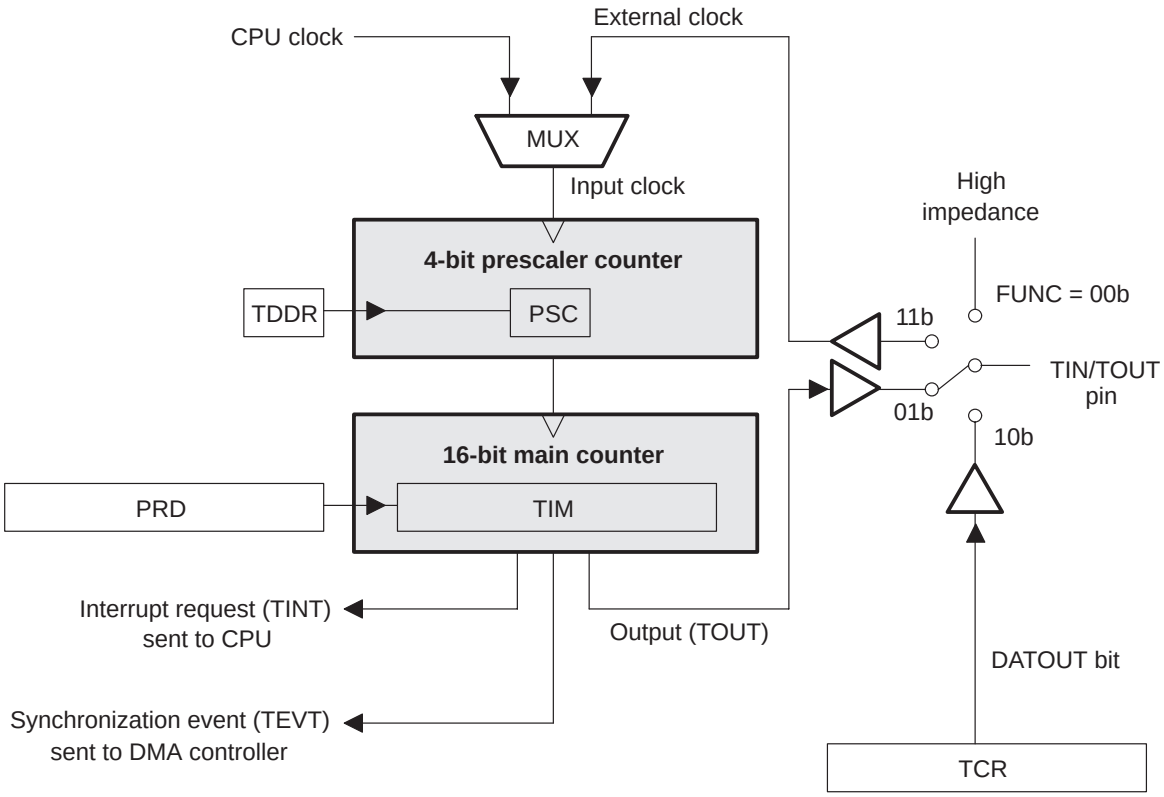
This document describes the type of general-purpose (GP) timer available on the TMS320VC5503/5507/5509 and TMS320VC5510 DSPs. These DSPs each have two identical but independent copies of this 20-bit, software-programmable timer. Two uses for the timer are to generate periodic interrupts and to provide periodic signals to devices outside the C55x DSP.

Topic **Page**

1.1 Introduction to the General-Purpose Timer

The general-purpose timer has up to a 20-bit dynamic range provided by two counters: a 4-bit prescaler counter and a 16-bit main counter. Figure 1−1 shows a high-level diagram of the timer.

Figure 1−1. Conceptual Block Diagram of the General-Purpose Timer



The timer has two count registers (PSC and TIM) and two period registers (TDDR and PRD). During timer initialization or during timer reloading, the contents of the period registers are copied into the count registers. A timer control register (TCR) controls and monitors the operation of the timer and the timer pin (TIN/TOUT). Depending on the value of the FUNC bits in TCR, the pin can be a general-purpose output (connected to the DATOUT bit of TCR), a timer output, or a clock input—or it can be in the high impedance state. (The details of the timer registers are in section 1.11, which begins on page 1-21. For more information about the timer pin, see section 1.2 on page 1-4.)

The prescaler counter is driven by an input clock, which can be the CPU clock or an external clock. PSC is decremented by 1 every input clock cycle. One cycle after PSC reaches 0, the TIM is decremented by 1. One cycle after TIM reaches 0, the timer sends an interrupt request (TINT) to the CPU, a synchronization event (TEVT) to the DMA controller, and (if applicable) an output to the timer pin. The rate at which the timer sends these signals is|

$$TINT\ rate = \frac{Input\ clock\ rate}{(TDDR + 1) \times (PRD + 1)}$$

The timer can be configured in auto-reload mode by setting the auto-reload bit (ARB) in the TCR. In this mode, the prescaler and the timer counter are reloaded and the timer continues counting each time the timer count reaches zero. For proper operation of the timer output pin in auto-reload mode, the timer period [(TDDR+1) x (PRD+1)] must be 4 cycles or greater.

1.2 Timer Pin

The general-purpose timer has one pin. The pin can be configured in the ways described in Table 1−1. The two FUNC bits in the timer control register (TCR) define the function of the timer pin and determine the required clock source for the timer. As described in the table, in some cases, the signal on the pin is affected by other TCR bits (POLAR, CP, and PWID, or DATOUT).

Sections 1.2.1 through 1.2.4 show different uses for the timer pin (different settings for the FUNC bits).

Note:

There are limitations on changing the FUNC bits. For details, see section 1.6 on page 1-11.

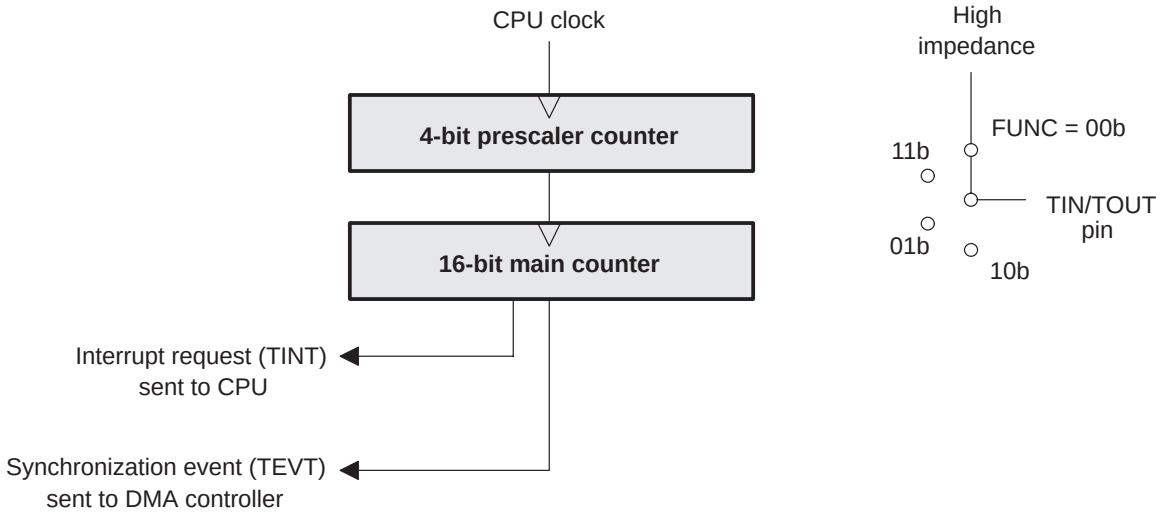
Table 1−1. Configuring the General-Purpose Timer Pin With the FUNC Bits

FUNC Bits	Timer Pin Function	Clock Source
00b	None The pin is in the high impedance state.	Internal (from DSP clock generator)
01b	Timer output The signal on the pin changes each time the main counter decrements to 0. The signal polarity is selected by the POLAR bit, and the signal toggles or pulses, depending on the CP bit. If pulsing is selected, the pulse width is defined by the PWID bits.	Internal (from DSP clock generator)
10b	General-purpose output The signal level on the pin reflects the value in the DATOUT bit.	Internal (from DSP clock generator)
11b	External clock input The pin receives a clock signal from a source outside the DSP.	External

1.2.1 Timer Pin in the High-Impedance State

When FUNC = 00b (see Figure 1−2), it is independent of the general-purpose timer, and it neither receives nor drives a signal. The timer input clock must be the CPU clock.

Figure 1−2. Timer Pin in High-Impedance State

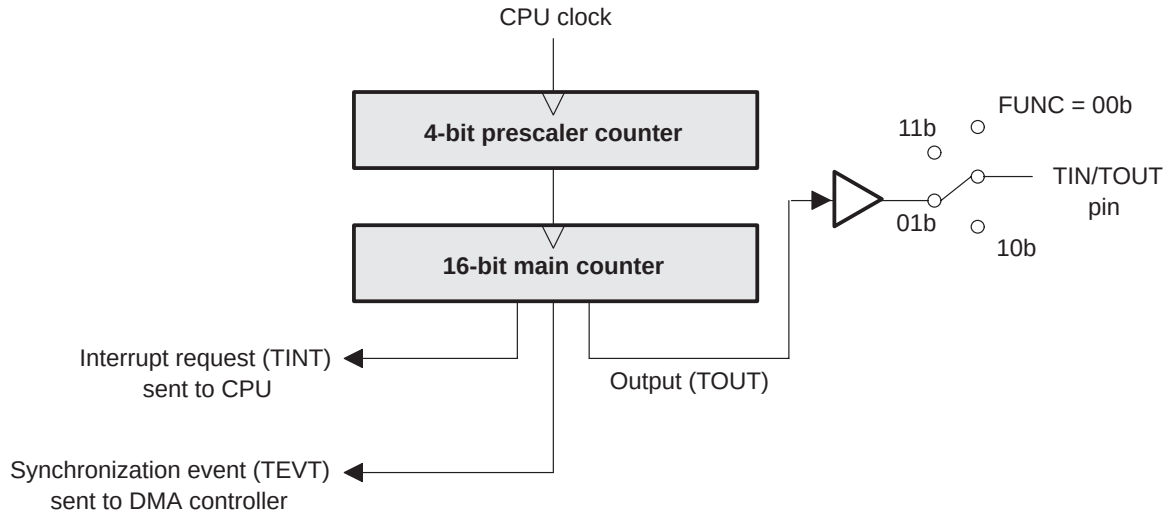


1.2.2 Timer Pin Configured to Reflect the Timer Output (TOUT)

Figure 1−3 shows the case when FUNC = 01b. The timer pin is used for the timer output and, therefore, cannot be used for an external clock source. The input clock must be the CPU clock.

When the TIN/TOUT pin is configured as an output, the CP, POLAR, and PWID bits in TCR control the behavior of the output signal. CP controls whether the pin operates in pulse mode (CP = 0) or in clock mode (CP = 1). In pulse mode, the timer pin produces an output pulse each time TIM counts down to zero. The width of the pulse can be 1, 2, 4, or 8 CPU clocks as controlled by the PWID bits. The polarity of the pulse is controlled by the POLAR bit. When POLAR = 0, the pulse is active high. When POLAR = 1, the pulse is active low. When the timer pin is configured for clock mode (CP = 1), the timer output signal toggles state each time TIM counts down to zero. When POLAR = 0, the timer output signal starts low and then toggles each time TIM reaches zero. When POLAR = 1, the timer output signal starts high and then toggles each time TIM reaches zero. For detailed information on the operation of these control bits, see the TCR description in Table 1−8 (page 1-24).

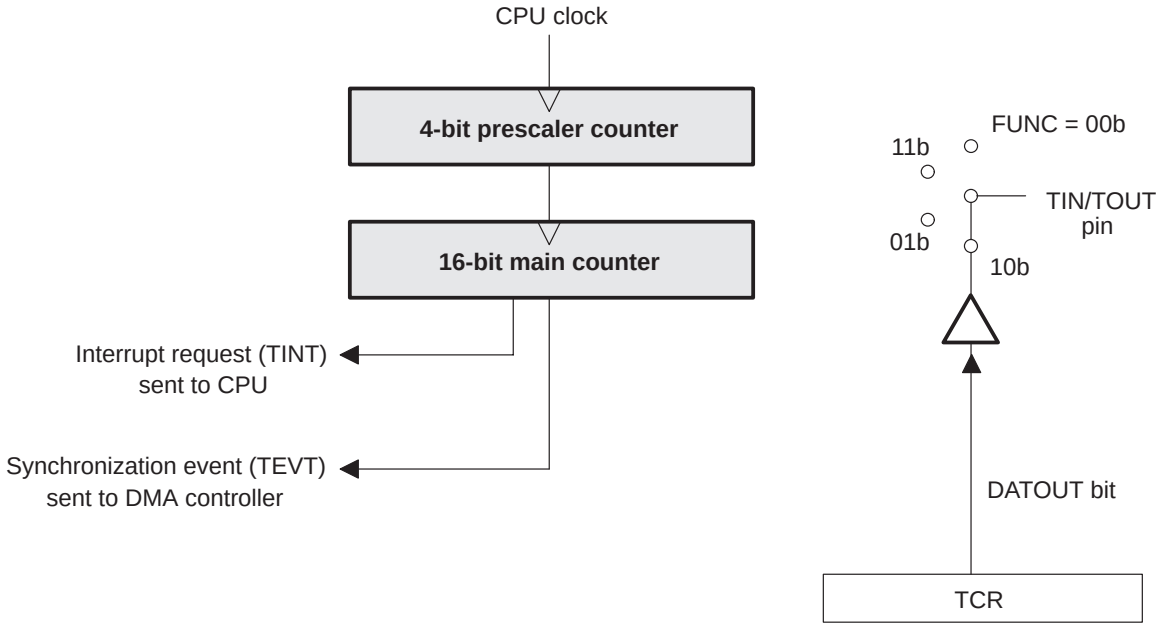
Figure 1−3. Timer Pin Configured to Reflect Timer Output (TOUT)



1.2.3 Timer Pin Used as a General-Purpose Output

Figure 1−4 shows the case when FUNC = 10b. The timer pin is configured as a general-purpose output; it is independent of the general-purpose timer. The pin reflects the value of the DATOUT bit. To bring the output signal low, write 0 to DATOUT. To bring the output signal high, write 1 to DATOUT. The timer input clock must be the CPU clock.

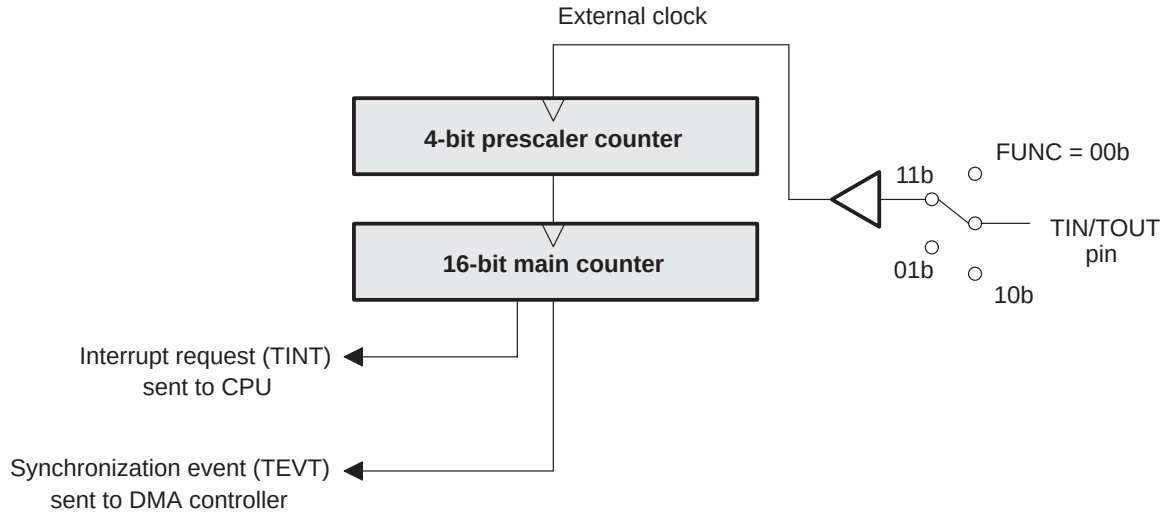
Figure 1−4. Timer Pin Used as a General-Purpose Output



1.2.4 Timer Pin Used for an External Input Clock

Figure 1−5 shows the case when FUNC = 11b. The timer pin is used by an external clock source and, therefore, cannot be used for the timer output. The only outputs of the general-purpose timer are the interrupt request and the DMA synchronization event.

Figure 1−5. Timer Pin Used for External Input Clock



1.3 Timer Interrupt

The general-purpose timer has a timer interrupt signal (TINT). A timer interrupt request is sent to the CPU when the main count register (TIM) counts down to 0. The timer interrupt rate is:

$$TINT\ rate = \frac{Input\ clock\ rate}{(TDDR + 1) \times (PRD + 1)}$$

TINT automatically sets a flag in one of the interrupt flag registers (IFR0 and IFR1). You can enable or disable the interrupt in one of the interrupt enable registers (IER0 and IER1) and in one of the debug interrupt enable registers (DBIER0 and DBIER1). When you are not using the general-purpose timer, disable the timer interrupt so that it does not cause unexpected interrupts.

The timer can be configured in auto-reload mode by setting the auto-reload bit (ARB) in the TCR. In this mode, the prescaler and the timer counter are reloaded and the timer continues counting each time the timer count reaches zero. For proper operation of the timer output pin in auto-reload mode, the timer period [(TDDR+1) x (PRD+1)] must be 4 cycles or greater.

1.4 Initializing the Timer

Use the following procedure to initialize the general-purpose timer:

1) Make sure the timer is stopped (TSS = 1), timer loading is enabled (TLB = 1), and the other control bits in TCR are set properly. While TLB = 1, the count registers (TIM and PSC) are loaded from the period registers (PRD and TDDR).

2) Load the desired prescaler counter period (in input clock cycles) by writing to TDDR in PRSC.

3) Load the desired main counter period (in input clock cycles) into PRD.

4) Turn off timer loading (TLB = 0) and start the timer (TSS = 0). When the timer starts, TIM holds the PDR value, and PSC holds the TDDR value.

Several initialization examples are provided in section 1.10 (page 1-15).

1.5 Stopping/Starting the Timer

Use the TSS bit of the timer control register (TCR) to stop the general-purpose timer (TSS = 1) or start the general-purpose timer (TSS = 0).

1.6 Changing the Timer Pin Function/Clock Source

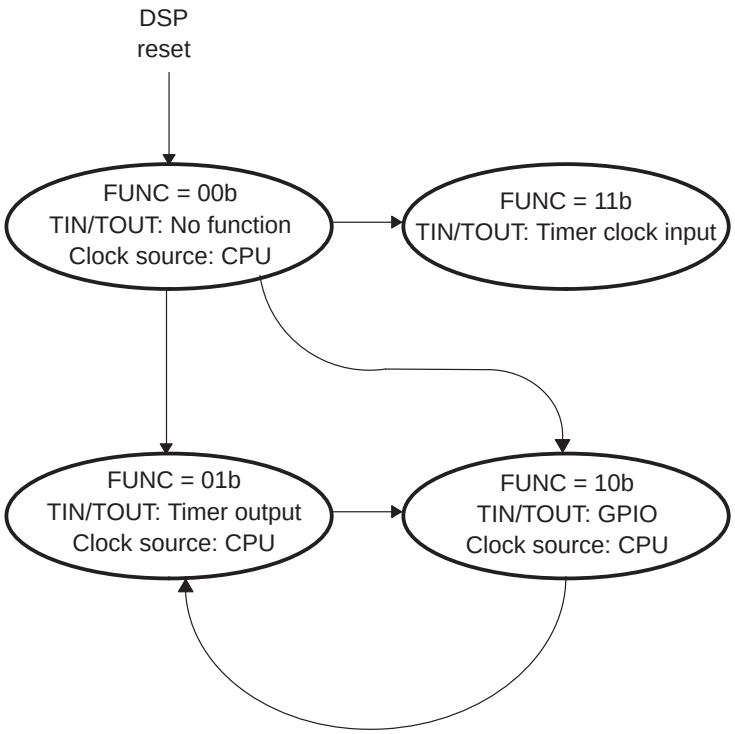
Table 1−2 explains when and how you can select each of the four pin function/clock source configurations. The legal transitions between modes is graphically shown in Figure 1−6. Following the table and figure is a procedure for using an external clock source. All of the bits mentioned in this section are in the timer control register (see page 1-24).

A DSP reset forces FUNC = 00b. You can keep it at 00b or choose another configuration listed in Table 1−2. If FUNC = 01b or 10b, you can only toggle between these two values until you reset the DSP.

Table 1−2. Selecting and Switching Pin Function/Clock Source Configurations

Current Configuration	**Can Be Directly Changed To …**	**Comments**
FUNC=00b (default) Pin: In high impedance state Clock Source: Internal	FUNC=01b, 10b, or 11b	If you want to switch to an external clock (FUNC = 11b), see the procedure that follows this table.
FUNC = 01b Pin: Timer output Clock Source: Internal	FUNC=10b Pin: General-purpose output Clock Source: Internal	To change to a value other than FUNC = 10b, you must reset the DSP. Otherwise, the change creates an error condition that sets the ERRTIM error flag in the timer control register (TCR). To recover from the error condition, reset the DSP.
FUNC = 10b Pin: General-purpose output Clock Source: Internal	FUNC = 01b Pin: Timer output Clock Source: Internal	To change to a value other than FUNC = 01b, you must reset the DSP. Otherwise, the change creates an error condition that sets the ERRTIM error flag in TCR. To recover from the error condition, reset the DSP.
FUNC = 11b Pin: Timer clock input Clock Source: External	No other configuration	To change from FUNC = 11b to another configuration, you must reset the DSP. Otherwise, the change creates an error condition that sets the ERRTIM error flag in TCR. To recover from the error condition, reset the DSP.

Figure 1−6. Legal FUNC Mode Transitions on the General-Purpose Timer



Using an external clock source. FUNC = 11b is the only configuration that supports an external clock source. The following procedure explains how to use FUNC = 11b.

1) Reset the DSP. (This forces FUNC = 00b.)

2) The external clock source on TIN/TOUT must provide at least 4 clock cycle transitions before the FUNC mode is changed.

3) Write 11b to the FUNC bits.

4) Poll the INTEXT bit in TCR until it is 1. When INTEXT = 1, the general-purpose timer is ready to run using an external clock source from the timer pin. (Reading INTEXT clears the bit automatically.)

5) Complete the other steps in timer initialization (see page 1-10).

An initialization example with the timer using an external clock source is provided in section 1.10.3 (page 1-19).

1.7 Reloading the Timer Count Registers

The timer count registers can be reloaded manually or automatically. To reload them manually, write to the I/O addresses of the registers.

The TLB and ARB bits in the timer control register (TCR) provide two different methods to have the registers reloaded automatically. While TLB = 1, PSC is loaded from TDDR, and TIM is loaded from PRD. This method helps with timer initialization (see section 1.4 on page 1-10).

If ARB = 1, the count registers are reloaded from the period registers each time TIM reaches 0. This method allows the general-purpose timer to count continually without input from your program. If ARB = 0, the general-purpose timer stops counting the next time TIM reaches 0.

Initialization examples using auto-reload are provided in section 1.10 (page 1-15).

1.8 Timer Emulation Modes

You can program the general-purpose timer to respond in one of three ways to a software breakpoint that occurs during emulation:

- ❏ Stop when the main counter decrements to 0.
- ❏ Stop immediately.
- ❏ Do not stop.

To program a response, use the SOFT and FREE bits of the timer control register. These bits are described in Table 1−8 on page 1-25.

1.9 Timer State After a DSP Reset

A DSP reset forces the following conditions for the general-purpose timer by resetting the timer registers. For details on the registers, see section 1.11 on page 1-21.

- The timer is stopped (TSS = 1).
- The count for the prescaler counter is 0.
- The count for the main counter is FFFFh.
- The timer is set to count down once (ARB = 0) rather than repeatedly.
- The timer cannot be forced into its idle mode by the idle instruction (IDLEEN = 0).
- A software breakpoint during emulation causes the timer to stop immediately (SOFT = FREE = 0).
- The timer pin is in the high impedance state, and the clock source is internal (FUNC = 00b).

1.10 Timer Initialization Examples

1.10.1 Example 1: Timer Generating a Clock on the TIN/TOUT Pin

Configure the timer to produce a 2 MHz clock source on the TIN/TOUT pin given a DSP CPU clock speed of 200 MHz. In the event of a software breakpoint or emulation halt, the timer should continue to run. If the DSP peripheral domain is set into IDLE mode, this timer should continue to run.

The following timer configuration causes the timer to generate a clock on the TIN/TOUT pin:

- To configure the TIN/TOUT pin as a timer output, set the FUNC field in TCR to 01b. To operate the pin in clock mode, set the CP field in TCR to 1b. This causes the pin to toggle state each time the timer count runs out. For this example, the polarity of the TIN/TOUT pin is the default value of 0b.
- Since the TIN/TOUT pin toggles each time the timer count runs out, the total period of the output clock is twice the count programmed into the timer. To divide the CPU clock frequency by 100, a total timer count of 50 is required for each high and low cycle. In this example, a divide-down of 50 is achieved by setting the timer period to 10 [PRD=9] with a prescaler value of 5 [TDDR=4]. Auto-reload [ARB=1] causes the timer to reload the count and restart after the count runs out each time.
- To make the timer continue running even after an emulation breakpoint has occurred, set the FREE bit in TCR to 1. In free-running mode, the state of the SOFT bit in TCR has no effect.
- To make the timer remain active, even when the peripheral clock domain is idle, clear the IDLEEN bit in TCR to 0b.

See Example 1−2 for sample initialization code.

Example 1−1. Timer Generating a Clock Initialization Code

```
;***************************************************************************
;       TIMER Register Addresses
;***************************************************************************
TIM0  .set 0x1000  ; TIMER 0, Timer Count Register
PRD0  .set 0x1001  ; TIMER 0, Timer Period Register
TCR0  .set 0x1002  ; TIMER 0, Timer Control Register
PRSC0 .set 0x1003  ; TIMER 0, Timer Prescaler Register

;***************************************************************************
;       TIMER Configuration
;***************************************************************************
TIMER_PERIOD    .set   9   ;for timer period of 10
TIMER_PRESCALE  .set   4   ;for prescale value of 5

   .text
INIT:
  mov #TIMER_PERIOD, port(#PRD0)          ;configure the timer period register

  mov #TIMER_PRESCALE, port(#PRSC0)       ;configure the timer prescaler register

  mov #0000110100111000b, port(#TCR0)
      ;0~~~~~~~~~~~~~~   IDLEEN          0 = do not idle with Peripheral Domain
      ;~0~~~~~~~~~~~~~   INTEXT          n/a
      ;~~0~~~~~~~~~~~~   ERR_TIM         1 = if illegal function change occurs
      ;~~~01~~~~~~~~~~   FUNC            01 = TIN/TOUT pin is a timer output
      ;~~~~~1~~~~~~~~~   TLB             1 = loading from period registers
      ;~~~~~~0~~~~~~~~   SOFT            n/a
      ;~~~~~~~1~~~~~~~   FREE            1 = Timer doesn't stop on emulation halt
      ;~~~~~~~~00~~~~~   PWID            n/a
      ;~~~~~~~~~~1~~~~   ARB             1 = auto-reload enabled
      ;~~~~~~~~~~~1~~~   TSS             1 = stop timer
      ;~~~~~~~~~~~~1~~   CP              0 = pulse mode, 1=clock (toggle) mode
      ;~~~~~~~~~~~~~0~   POLAR           0 = normal polarity
      ;~~~~~~~~~~~~~~0   DATOUT          n/a
      ;~~~~~~~~~~~~~~~   Reserved
 and #1111101111101111b, port(#TCR0)
      ;~~~~~0~~~~~~~~~~   TLB             0 = stop loading from period registers
      ;~~~~~~~~~~~0~~~~   TSS             0 = start timer
```

1.10.2 Example 2: Timer Generating a Periodic Pulse on the TIN/TOUT Pin

Configure the timer to produce a logic-low output pulse every 125 μs (8 kHz rate) on the TIN/TOUT pin given a DSP CPU clock speed of 200 MHz. The output pulse should be 4 clock cycles wide. In the event of a software breakpoint, the timer should stop immediately. If the DSP peripheral domain is set into IDLE mode, this timer should idle also.

The following timer configuration causes the timer to generate a periodic pulse on the TIN/TOUT pin:

- To configure the TIN/TOUT pin as a timer output, set the FUNC field in TCR to 01b. To operate the pin in pulse mode, set the CP field in TCR to 0b. This causes the pin to produce a single pulse each time the timer count runs out. To generate a logic-low pulse, the polarity of the TIN/TOUT pin is set to 1b (reversed polarity). To generate a 4-cycle wide output pulse, the PWID field in TCR is set to 10b.
- Since one pulse is generated on the TIN/TOUT pin each time the timer count runs out, the total period of the output clock is equal to the count programmed into the timer. To generate an 8 kHz period from a 200 MHz CPU clock, the timer must count (200,000,000 / 8,000) = 25,000 CPU cycles between output pulses. In this example, a divide-down of 25,000 is achieved by setting the timer period to 25,000 [PRD=24,999] with a prescaler value of 1 [TDDR=0]. Auto-reload [ARB=1] causes the timer to reload the count and restart after the count runs out each time.
- To make the timer stop immediately when an emulation breakpoint has occurred, set the FREE bit in TCR to 0b and the SOFT bit in TCR to 0b.
- To make the timer respond to an idle request from the peripheral clock domain, set the IDLEEN bit in TCR to 1b.

See Example 1−2 for sample initialization code:

Example 1−2. Timer Generating a Periodic Pulse Initialization Code

```
;***************************************************************************
;       TIMER Register Addresses
;***************************************************************************
TIM0  .set 0x1000  ; TIMER 0, Timer Count Register
PRD0  .set 0x1001  ; TIMER 0, Timer Period Register
TCR0  .set 0x1002  ; TIMER 0, Timer Control Register
PRSC0 .set 0x1003  ; TIMER 0, Timer Prescaler Register

;***************************************************************************
;       TIMER Configuration
;***************************************************************************
TIMER_PERIOD    .set 24999   ;for timer period of 25,000
TIMER_PRESCALE  .set     0   ;for prescale value of 1

      .text
INIT:
     mov #TIMER_PERIOD, port(#PRD0)     ;configure the timer period register

     mov #TIMER_PRESCALE, port(#PRSC0)  ;configure the timer prescaler register

     mov #1000110010110100b, port(#TCR0)
         ;1~~~~~~~~~~~~~~   IDLEEN   1 = idle timer with Peripheral Domain
         ;~0~~~~~~~~~~~~~   INTEXT  n/a
         ;~~0~~~~~~~~~~~~   ERR_TIM  1 = if illegal function change occurs
         ;~~~01~~~~~~~~~~   FUNC    01 = TIN/TOUT pin is a timer output
         ;~~~~~1~~~~~~~~~   TLB      1 = loading from period registers
         ;~~~~~~0~~~~~~~~   SOFT     0 = On emulation halt, stop immediately
         ;~~~~~~~0~~~~~~~   FREE     0 = Behavior controlled by SOFT
         ;~~~~~~~~10~~~~~~  PWID    10 = output pulse width is 4 cycles
         ;~~~~~~~~~~1~~~~~  ARB      1 = auto-reload enabled
         ;~~~~~~~~~~~1~~~~  TSS      1 = stop timer
         ;~~~~~~~~~~~~0~~~  CP       0 = pulse mode, 1=clock (toggle) mode
         ;~~~~~~~~~~~~~1~~  POLAR    1 = reverse polarity
         ;~~~~~~~~~~~~~~0~  DATOUT    n/a
         ;~~~~~~~~~~~~~~~0  Reserved

      and #1111101111101111b, port(#TCR0)
          ;~~~~~0~~~~~~~~~~   TLB    0 = stop loading from period registers
          ;~~~~~~~~~~~0~~~~   TSS    0 = start timer
```

1.10.3 Example 3: Timer Running From an External Clock

Configure the timer to generate a DMA sync event after every 50 cycles of an external clock provided at the TIN/TOUT pin. In the event of a software breakpoint, the timer should continue to free run. If the DSP peripheral domain is set into IDLE mode, this timer should continue to run.

The following timer configuration causes the timer to run from an external clock:

- To configure the TIN/TOUT pin as a timer input, set the FUNC field in TCR to 11b. This causes the timer to count down based on the clock received externally from the TIN/TOUT pin. When used in this mode, at least 4 external clock cycles should occur on the TIN/TOUT pin between reset and when the FUNC field is changed to 11b to ensure that the timer is properly configured.
- The timer automatically generates a DMA sync event every time the count reaches zero. No additional configuration of the timer is required. The DMA should be configured to appropriately respond to the timer sync events.
- To make the timer continue running even after an emulation breakpoint has occurred, set the FREE bit in TCR to 1. In free-running mode, the state of the SOFT bit in TCR has no effect.
- To make the timer remain active, even in the peripheral clock domain is idled, clear the IDLEEN bit in TCR to 0b.

See Example 1−3 for sample initialization code.

Example 1−3. Timer Running From an External Clock Initialization Code

```
;***************************************************************************
;       TIMER Register Addresses
;***************************************************************************
TIM0  .set 0x1000  ; TIMER 0, Timer Count Register
PRD0  .set 0x1001  ; TIMER 0, Timer Period Register
TCR0  .set 0x1002  ; TIMER 0, Timer Control Register
PRSC0 .set 0x1003  ; TIMER 0, Timer Prescaler Register

;***************************************************************************
;       TIMER Configuration
;***************************************************************************
TIMER_PERIOD    .set    49   ;for timer period of 50
TIMER_PRESCALE  .set     0   ;for prescale value of 1

      .text
INIT:
   mov #TIMER_PERIOD, port(#PRD0)     ;configure the timer period register

   mov #TIMER_PRESCALE, port(#PRSC0)  ;configure the timer prescaler register

   mov #0001110100110100b, port(#TCR0)
       ;0~~~~~~~~~~~~~~   IDLEEN    0 = do not idle timer with Periph Domain
       ;~0~~~~~~~~~~~~~   INTEXT   Ready to use external clock when INTEXT=1
       ;~~0~~~~~~~~~~~~   ERR_TIM   1 = if illegal function change occurs
       ;~~~11~~~~~~~~~~   FUNC     11 = TIN/TOUT pin is a timer input
       ;~~~~~1~~~~~~~~~   TLB       1 = loading from period registers
       ;~~~~~~0~~~~~~~~   SOFT          N/A when FREE=1
       ;~~~~~~~1~~~~~~~   FREE      1 = Continue running on emulation halt
       ;~~~~~~~~00~~~~~   PWID          N/A when TIN/TOUT pin is an input
       ;~~~~~~~~~~1~~~~   ARB       1 = auto-reload enabled
       ;~~~~~~~~~~1~~~~   TSS       1 = stop timer
       ;~~~~~~~~~~~0~~~   CP           N/A when TIN/TOUT pin is an input
       ;~~~~~~~~~~~~1~~   POLAR     0 = normal polarity
       ;~~~~~~~~~~~~~0~   DATOUT        N/A when TIN/TOUT pin is an input
       ;~~~~~~~~~~~~~~0   Reserved

wait_for_FUNC_change:
    btst #14, port(#TCR0), TC1          ;poll the INTEXT bit to determine
    bcc wait_for_FUNC_change, !TC1      ;when the clock source has changed.

   and #1111101111101111b, port(#TCR0)
       ;~~~~~0~~~~~~~~~~   TLB          0 = stop loading from period registers
       ;~~~~~~~~~~~0~~~~   TSS           0 = start timer
```

1.11 Timer Registers

For the general-purpose timer, the DSP contains the registers listed in Table 1−3. To obtain the I/O address of each register, see the data sheet for your TMS320C55x DSP.

Table 1−3. Registers of the General-Purpose Timer

Register	Description	For Details, See ...
TIM	Main count register	Section 1.11.1
PRD	Main period register	Section 1.11.1
PRSC	Timer prescaler register	Section 1.11.1
TCR	Timer control register	Section 1.11.2 (page 1-24)

1.11.1 Period and Count Registers (TDDR, PSC, PRD, TIM)

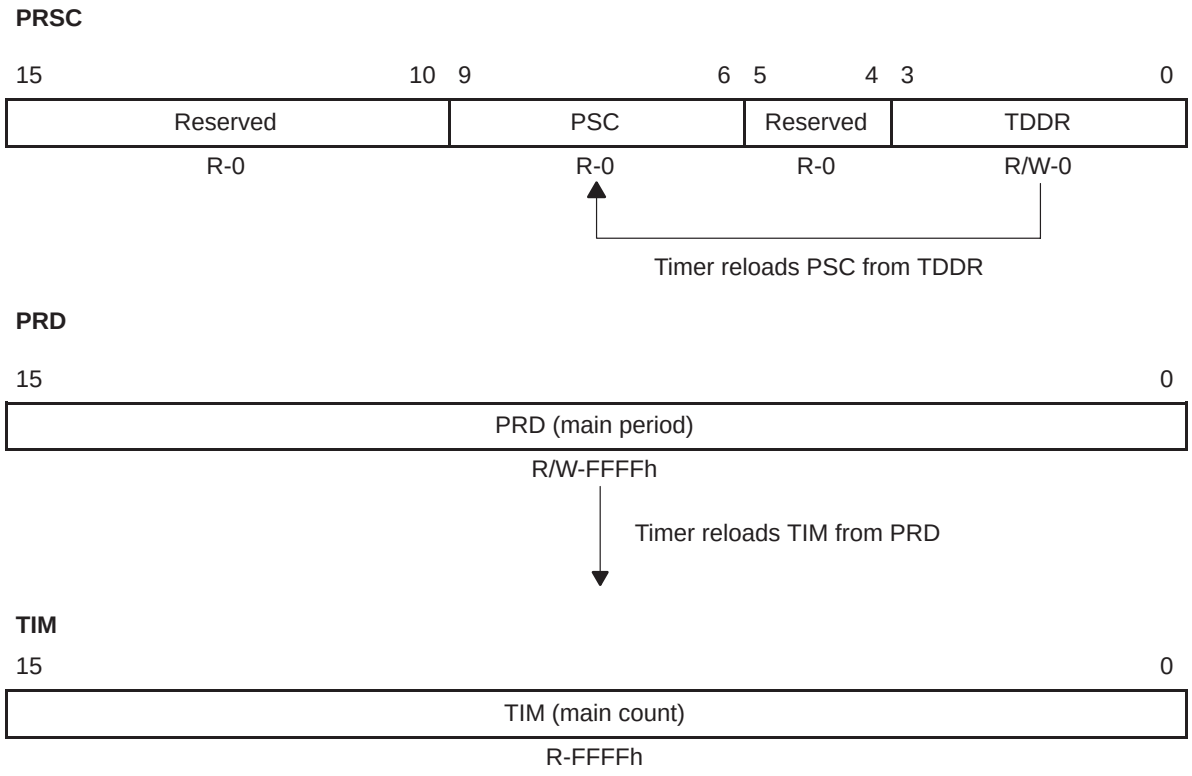
Table 1−4 summarizes the period and count registers available for the general-purpose timer. This timer has two counters: a 4-bit prescaler counter and a 16-bit main counter. Each of the two counters has a count register and a period register. During timer operation, the count registers are decremented. The timer can automatically reload each count register by copying the content of the associated period register. The bits of the registers are shown in Figure 1−7 and described in Table 1−5 through Table 1−7.

For proper operation of the timer output pin in auto-reload mode, the timer period [(TDDR+1) x (PRD+1)] must be 4 cycles or greater.

Table 1−4. Period and Count Registers Listed by Counter

Counter	Register	Description
Prescaler counter	PSC	Prescaler count register. Bits 9-6 of the timer prescaler register (PRSC).
	TDDR	Timer divide-down register (prescaler period register). Bits 3-0 of PRSC.
Main counter	TIM	Main count register
	PRD	Main period register

Figure 1−7. Period and Count Registers of the General-Purpose Timer



Legend: R = Read; W = Write; -*n* = Value after reset

Table 1−5. PRSC Bit Field Descriptions

Bit	Field	Value	Description
15-10	Reserved		These reserved bits are always read as 0. A value written to this field has no effect.
9-6	PSC	0h-Fh	Prescaler count register. This register contains the current count of the prescaler counter. PSC is decremented by 1 every input clock cycle. One cycle after PSC reaches 0, the TIM is decremented by 1.
5-4	Reserved		These reserved bits are always read as 0. A value written to this field has no effect.
3-0	TDDR	0h-Fh	Timer divide-down register (prescaler period register). When PSC must be loaded/reloaded, the content of TDDR is copied to PSC.

Table 1−6. PRD Bit Field Descriptions

Bit	Field	Value	Description
15-0	PRD	0000h-FFFFh	Main period register. When the main count register (TIM) must be loaded/reloaded, the content of PRD is copied to TIM.

Table 1−7. TIM Bit Field Descriptions

Bit	Field	Value	Description
15-0	TIM	0000h-FFFFh	Main count register. This register contains the current count of the main counter. PSC is decremented by 1 every input clock cycle. One cycle after PSC reaches 0, the TIM is decremented by 1.

1.11.2 Timer Control Register (TCR)

The timer control register (TCR) has the form shown in Figure 1−8. Table 1−8 describes the bits of TCR. Using specific bits in TCR, you can configure, start, stop, load, and reload the general-purpose timer. Other bits in the TCR control the functionality of the associated timer output pin.

Figure 1−8. Timer Control Register (TCR)

15	14	13	12	11	10	9	8
IDLEEN	INTEXT	ERRTIM	FUNC		TLB	SOFT	FREE
R/W-0	R-0	R-0	R/W-00		R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
PWID		ARB	TSS	CP	POLAR	DATOUT	Reserved
R/W-00		R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read; W = Write; -*n* = Value after reset

Table 1−8. TCR Bit Field Descriptions

Bit	Field	Value	Description
15	IDLEEN		Idle enable bit for the timer. If the PERIPH idle domain is configured to be idle and IDLEEN = 1, the timer stops and enters a low-power (idle) state.
		0	The timer cannot be placed in an idle state.
		1	If the PERIPH domain is idle (PERIS = 1 in the idle status register), the timer is stopped in a low-power state.
14	INTEXT		Internal-to-external clock change indicator. When changing the timer clock source from internal to external, a program can check this bit to determine when the timer is ready to use an external clock source.
		0	Timer not ready to use an external clock source.
		1	Timer ready to use an external clock source.
13	ERRTIM		Timer-pin error flag. Some changes to the FUNC bits create an error condition that is reflected in ERRTIM. When ERRTIM = 1, reset the DSP and reinitialize the timer.
		0	No error detected or ERRTIM has been read.
		1	Error detected in a write to the FUNC bits. One of the following changes was attempted: Change from FUNC = 01b to FUNC = 00b or 11b Change from FUNC = 10b to FUNC = 00b or 11b Change from FUNC = 11b to any other value

Table 1−8. TCR Bit Field Descriptions (Continued)

Bit	Field	Value	Description
12-11	FUNC		Function bits for the timer pin. The two FUNC bits define the function of the timer pin and determine the required clock source for the timer. If the internal clock source is chosen, the CPU clock drives the counting of the timer. If an external clock source is chosen, the timer is driven by a clock signal coming in on the timer pin.
		00b	Pin Function**:** None. The pin is in the high impedance state.
			Clock Source**:** Internal (from DSP clock generator).
		01b	Pin Function**:** Timer output. The signal on pin changes each time the main counter decrements to 0. The signal polarity is selected by the POLAR bit. The signal toggles or pulses, depending on the CP bit. If pulsing is selected, the pulse width is defined by the PWID bits.
			Clock Source**:** Internal (from DSP clock generator).
		10b	Pin Function**:** General-purpose output. The signal level on the pin reflects the value in the DATOUT bit.
			Clock Source**:** Internal (from DSP clock generator).
		11b	Pin Function**:** External clock input. The pin receives a clock signal from a source outside the DSP.
			Clock Source**:** External.
10	TLB		Timer load bit. While TLB = 1, the count registers are loaded from the period registers.
		0	When TLB = 0, TIM and PSC are not being loaded.
		1	Until TLB = 0, load TIM from PRD, and load PSC from TDDR.
9	SOFT		Soft stop bit. When the FREE bit is 0, SOFT determines the response of the timer when a breakpoint or emulation halt is encountered during debugging. If FREE = 1, the timer does not stop, regardless of the value of SOFT.
		0	Hard stop. The timer stops immediately.
		1	Soft stop. The timer stops when the main count register (TIM) decrements to 0.

Table 1−8. TCR Bit Field Descriptions (Continued)

Bit	Field	Value	Description
8	FREE		Free run bit. When a breakpoint or emulation halt is encountered during debugging, FREE determines whether the timer continues to run or the timer is affected as determined by the SOFT bit.
		0	The timer is affected as determined by the SOFT bit.
		1	Free run. The timer continues to run.
7-6	PWID		Timer-output pulse width bit. PWID determines the width of each pulse on the timer pin under the following conditions: ❑ The timer pin is configured to show the timer output (FUNC = 01b). ❑ The pulse mode is selected (CP = 0). The pulse width is defined in CPU clock periods:
		00b	1 CPU clock period
		01b	2 CPU clock periods
		10b	4 CPU clock periods
		11b	8 CPU clock periods
5	ARB		Auto-reload bit. When ARB = 1, the count registers are automatically reloaded from the period registers whenever the main count register (TIM) reaches 0.
		0	ARB has been cleared.
		1	Each time TIM reaches 0, reload TIM from PRD, and reload PSC from TDDR.
4	TSS		Timer stop status bit. Use TSS to stop the timer, start the timer, or determine whether the timer is stopped.
		0	Start the timer.
		1	Stop the timer. (Reset condition)

Table 1−8. TCR Bit Field Descriptions (Continued)

Bit	Field	Value	Description
3	CP		Clock mode/pulse mode bit. When the timer pin is configured to show the timer output (FUNC = 01b), CP determines whether the signal on the pin is pulsed or toggled.
		0	Pulse mode. A pulse is driven on the timer pin each time the main count register (TIM) reaches 0. The PWID bits define the width of the pulse. The POLAR bit defines the polarity of the pulse.
		1	Clock mode. The signal on the timer pin has a 50% duty cycle. The signal toggles (from high to low or from low to high) each time TIM reaches 0.
2	POLAR		Timer-output polarity bit. When the timer pin is configured to show the timer output (FUNC = 01b), POLAR determines the polarity of the signal on the pin. The specific effect of this bit depends on whether pulse mode (CP = 0) or clock mode (CP = 1) is selected.
		0	The signal on the timer pin starts low. Then the following events occur:
			Pulse mode: Each time the main count register (TIM) reaches 0, a high pulse is driven on the timer pin. The PWID bits define the width of the pulse. Between pulses the signal is low.
			Clock mode: The first time TIM reaches 0, the signal on the timer pin toggles high. During subsequent countdowns: If the signal is high, it toggles low; if the signal is low, it toggles high.
		1	The signal on the timer pin starts high. Then the following events occur:
			Pulse mode: Each time TIM reaches 0, a low pulse is driven on the timer pin. The PWID bits define the width of the pulse. Between pulses, the signal is high.
			Clock mode: The first time TIM reaches 0, the signal on the timer pin toggles low. During subsequent countdowns: if the signal is low, it is toggled high; if the signal is high, it is toggled low.
1	DATOUT		Data output bit. When the timer pin is configured as a general-purpose output pin (FUNC = 10b), use DATOUT to control the signal level on the pin.
		0	Drive the signal on the timer pin low.
		1	Drive the signal on the timer pin high.
0	Reserved		This reserved bit is always read as 0. A value written to this bit has no effect.

This page is intentionally left blank.

Chapter 2

Watchdog Timer

Each TMS320VC5503/5507/5509 DSP includes one watchdog timer of the type described in this chapter. The watchdog timer is available to prevent a system from locking up if the software becomes trapped in loops with no controlled exit.

Topic **Page**

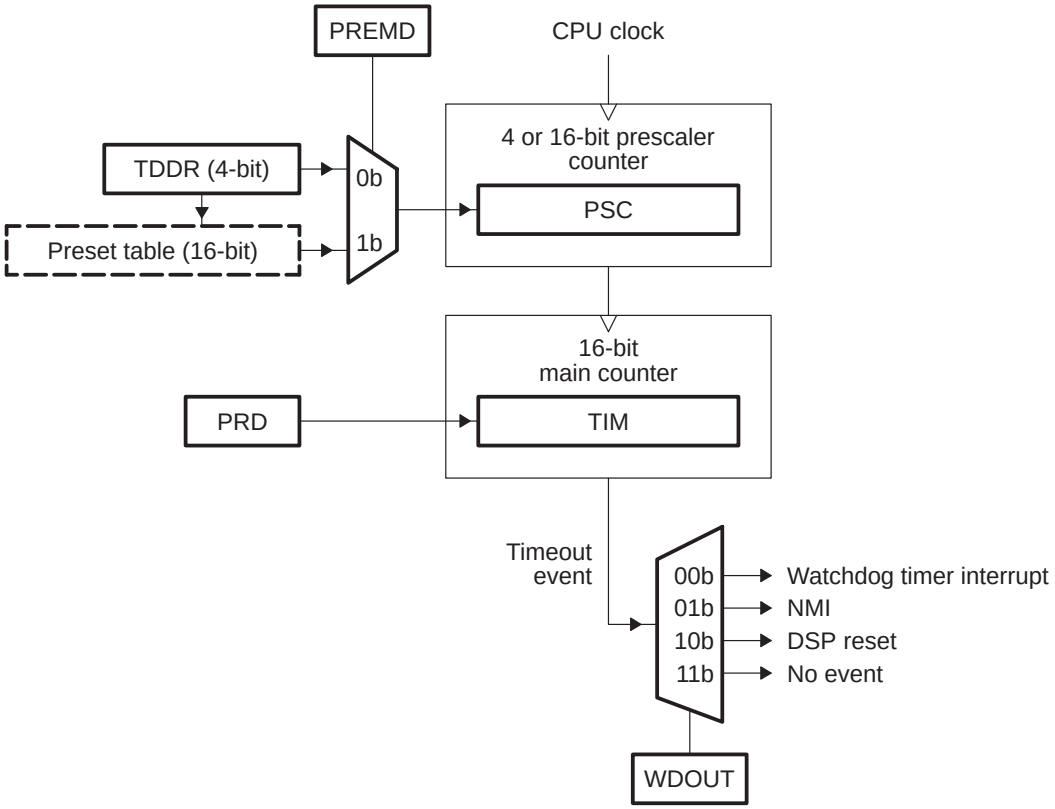
2.1 Introduction to the Watchdog Timer

The watchdog timer provides an automatic mechanism for recovery from application software error conditions by counting down for a pre-defined number of cycles. If the watchdog timer is serviced by the application software before the count reaches zero, the watchdog timer resets and begins the count again. In the event that the application enters a non-recoverable condition (such as a loop with no exit), the watchdog timer counts down to zero triggering a timeout condition. This timeout condition can be used to trigger an interrupt or a DSP reset.

The watchdog timer consists of a 16-bit main counter preceded by a prescaler, also with up to 16-bit resolution, providing a counter with up to 32-bit dynamic range. The watchdog timer is disabled after reset allowing the application software to configure the watchdog timer before it is enabled. After the watchdog timer is enabled, it cannot be disabled without a DSP reset or timeout condition, preventing a software error from inadvertently disabling its function. The watchdog timer is serviced by writing a specific key value to a register. Each time this key value is written, the watchdog timer reloads its counter registers and begins counting again. A timeout condition occurs if the key value does not arrive before the counter reaches zero or if an incorrect key value is written.

Figure 2−1 is a block diagram of the watchdog timer. The CPU clock provides the time reference for the watchdog timer. The prescaler counts down by 1 on each occurrence of the CPU clock. Each time the prescaler reaches 0 it clocks the main counter, which also decrements by 1. The prescaler automatically reloads each time it reaches 0 and begins to count again. When the main counter reaches 0, a timeout event occurs that causes one of the following programmable events to occur: a watchdog timer interrupt, a DSP reset, a non-maskable interrupt (NMI), or no event. The timeout event generated is configured by programming the WDOUT field of the watchdog timer control register (WDTCR). The watchdog timer registers are described in section 2.6 (page 2-10).

Figure 2−1. Watchdog Timer Block Diagram



Each time the prescaler reaches 0 it is reloaded and starts counting down again. The value loaded in the prescaler is determined by the TDDR bits in WDTCR and the prescaler mode bit (PREMD) in watchdog timer control register 2 (WDTCR2). In prescaler direct mode (PREMD = 0) , the prescaler is always loaded with the 4-bit contents of the TDDR directly. This mode provides prescaler values from 0 to 15 and provides an overall 20-bit dynamic range for the watchdog timer. In prescaler indirect mode (PREMD = 1) , the prescaler is indirectly loaded with the 16-bit preset number associated with the value in TDDR. This mode provides 16 predefined prescale values from 1 to 65535 and a 32-bit dynamic range for the watchdog timer. The prescaler values available are shown in Table 2−4 on page 2-13.

When the watchdog timer is initially enabled, the main count (TIM) is loaded with the value programmed in the watchdog timer period register (WDPRD). The main counter counts down until the watchdog timer is serviced by the application software by writing a sequence of key values to the WDKEY. Each time the watchdog timer is serviced, the prescaler and the main counter are reloaded and start counting down again. If the main counter ever reaches 0, a timeout condition occurs. In this state the watchdog timer is disabled and requires a DSP reset to be used again.

The clock to the watchdog timer is under the influence of the clock generation (CLKGEN) idle domain. If the CLKGEN domain enters the idle state, the clock to the watchdog timer stops and does not resume until the CLKGEN domain is removed from idle. So the watchdog timer operates as long as the CLKGENIS bit of the idle status register (ISTR) is cleared.

2.2 Configuring the Watchdog Timer

After reset, the watchdog timer is disabled and in the Initial State shown in Figure 2−2. During this time, the counters do not run and the watchdog timer output is disconnected from the watchdog time-out event.

Once the watchdog timer is enabled, the watchdog timer output is connected to the watchdog time-out event (as configured in WDOUT) and the counters and prescalers are reloaded and begin counting down. After the watchdog timer is enabled, it cannot be disabled by software but can be disabled by a watchdog time-out event or hardware reset. A special key sequence is provided to prevent the watchdog timer from being accidentally serviced while the software is trapped in a loop with no exit or some other software failure.

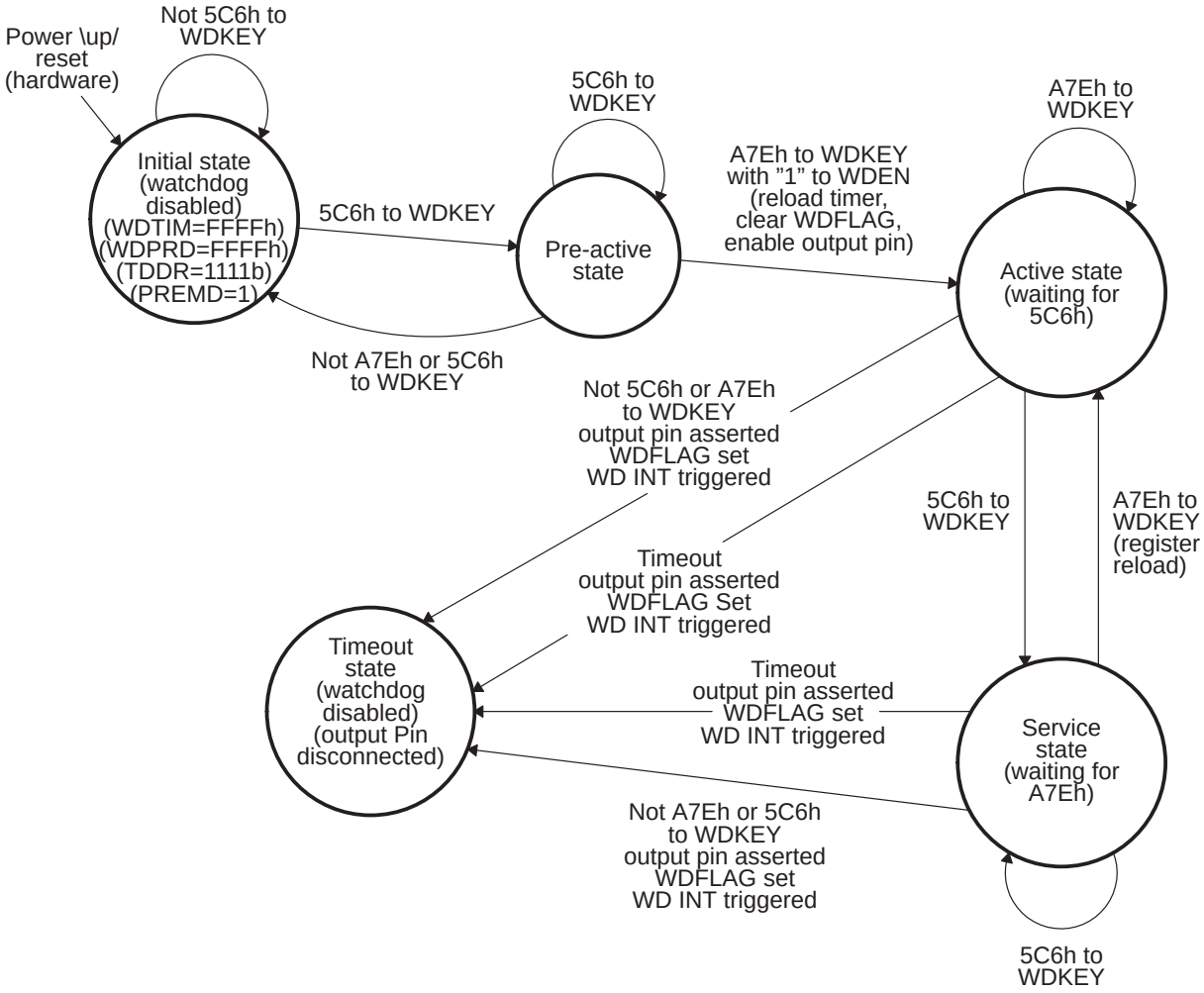
Before it is enabled, the watchdog timer should be initialized with the desired count values and mode selection. See section 2.6 on page 2-10 for a full description of all of the watchdog timer registers and their functions.

Follow these steps to initialize and enable the watchdog timer:

- ❑ Program the desired period (PRD) for the main counter into the watchdog timer period register (WDPRD).
- ❑ Program the watchdog timer control register (WDTCR) with the desired mode bits (WDOUT, SOFT, and FREE) and the desired prescaler control value in TDDR.
- ❑ Write the key value 5C6h to WDKEY in the watchdog timer control register 2 (WDTCR2). This causes the watchdog timer to enter the Pre-Active State.
- ❑ Write the key value A7Eh to WDKEY, set WDEN to 1, and write the desired value of the prescaler mode bit (PREMD) to WDTCR2. This second WDKEY value completes the key sequence indicating a valid service sequence and WDEN = 1 enables the watchdog timer. These writes can be performed as a single write to WDTCR2.

At this point, the watchdog timer is active, and the prescaler and main counter values are loaded and begin counting down. The selected timeout event (WDOUT) is also now connected and will occur if a watchdog timer timeout occurs.

Figure 2−2. Watchdog Timer Operation State Diagram



2.3 Watchdog Timer Servicing

The watchdog timer has to be serviced periodically such that a 5C6h is written followed by an A7Eh to the WDKEY bits, in WDTCR2, before a watchdog timer time-out event occurs. Both 5C6h and A7Eh can be written to WDKEY. Only the write sequence of a 5C6h followed by an A7Eh to WDKEY services the watchdog timer. Any other writes to WDKEY trigger the watchdog timer time-out event immediately, and consequently:

- The WDFLAG bit, in WDTCR2, is set to 1
- The internal maskable watchdog timer interrupt, nonmaskable interrupt (NMI), or RESET is triggered

However, reads from WDTCR2 do not cause a time-out event.

When the watchdog timer is in a time-out state, the watchdog timer is disabled and the WDEN bit, in WDTCR2, is cleared. The watchdog timer output event is disconnected.

Following reset, the watchdog timer is disabled and reads or writes to the watchdog timer registers are allowed. However, once a 5C6h is written to the WDKEY bits and the watchdog timer enters the pre-active state, a write to WDTCR2 is allowed only when the write comes with the correct key (5C6h or A7Eh) to WDKEY. In the Pre-Active state, an A7Eh written to WDKEY with the WDEN bit set to 1 enables the watchdog timer (Active state). Once the watchdog timer is enabled, it cannot be disabled by software but can be disabled by a watchdog time-out event or hardware reset. The watchdog timer service is secured by WDKEY.

The registers WDTIM, WDPRD, and WDTCR; and the PREMD bit in WDTCR2 should be configured as desired before the watchdog timer enters the active state. You can set the WDEN bit to 1 and configure the PREMD bit at the same time an A7Eh is written to the WDKEY bits, during the pre-active state. By default, WDTIM = FFFFh, WDPRD = FFFFh, WDTCR2(PREMD) = 1, and the WDTCR(TDDR) = 1111b. Every time the watchdog timer is serviced, the watchdog timer counters and prescalers are automatically reloaded accordingly.

Note:

On the TMS320VC5503/5507/5509/5509A, the RTC interrupt is shared with External Interrupt 4 (INT4) in the Interrupt Flag Register 1 (bit 3 in IFR1). The presence of an interrupt flag in this bit indicates that either INT4 or the RTC interrupt has occurred. The source of an RTC interrupt can be identified by reading RTCINTFL. If no RTC interrupt flags are present, the interrupt source was INT4.

This interrupt should not be used for both INT4 and RTC at the same time because some INT4 events may be missed if both interrupt sources occur near the same time. In this case, when the interrupt service is entered, the CPU clears the interrupt flag in the CPU. After the return from the interrupt service routine, the second interrupt source is not serviced since the interrupt flag in the CPU was cleared.

2.4 Recovery From the Time-Out Condition

If the timeout event selected in WDOUT is a DSP reset, the DSP automatically resets when a time-out occurs and all the DSP registers will be reset.

If the selected timeout event is a watchdog timer interrupt, it is possible to have active interrupts simultaneously from both the external INT3 source and the watchdog timer. When an interrupt is detected in this bit, the watchdog timer status register should be polled to determine if the watchdog timer is the interrupt source. However, it is recommended that either the INT3 or WDINT interrupt be used. If both INT3 and WDINT interrupts are used, one interrupt event can potentially hold off the other interrupt. For example, if INT3 is asserted first and held low, the WDINT interrupt will not be recognized until the INT3 pin is back to high-logic state again. The INT3 pin must be pulled high if only the WDINT interrupt is used.

If the selected timeout event is a NMI, it means that when the timeout happens the CPU gets a Non-Maskable Interrupt.

The Watchdog Timer can not be re-initialized in the interrupt service routine for either of the interrupts. In all cases after a timeout event if the watchdog needs to be re-used, a hardware reset needs to be provided to the DSP. The only way to prevent a WD Timeout is to service it periodically with the defined key words.

2.5 Power, Emulation, and Reset Considerations

The following sections describe how the watchdog timer is affected by DSP low power modes (idle configurations), emulation breakpoints/halts, and a DSP reset.

2.5.1 Watchdog Timer Response to Low Power Modes (Idle Configurations)

The DSP is divided into idle domains that can be programmed to be idle or active. The state of all domains is called the idle configuration. Any idle configuration that disables the clock generator (CLKGEN) domain stops the clock to the watchdog timer and, therefore, stops activity in the watchdog timer. Otherwise, the watchdog timer is not affected by any other idle domain on the DSP.

2.5.2 Emulation Modes of the Watchdog Timer

The FREE and SOFT bits of WDTCR control the behavior of the watchdog timer when a breakpoint or emulation halt is encountered during debugging. If FREE = 0, a breakpoint or emulation halt suspends watchdog timer activity. If FREE = 0 and SOFT = 0, the watchdog timer stops immediately. If FREE = 0 and SOFT = 1, the watchdog timer stops when the timer counter reaches 0. If FREE = 1, the watchdog timer continues to run on a breakpoint or emulation halt. The FREE and SOFT bits are described in section 2.6.3 on page 2-12.

2.5.3 Watchdog Timer after DSP Reset

A DSP reset resets the watchdog timer and the watchdog timer configuration registers. The register figures in section 2.5 that follow indicate the effects of reset on the register contents.

2.6 Watchdog Timer Registers

Once the watchdog timer is enabled, the registers are under write protection. Writes to WDTIM, WDPRD, and WDTCR have no effect. Writes to the WDFLAG, WDEN, and PREMD bits in WDTCR2 have no effect. However, incorrect key (not 5C6h or A7Eh) sequence to the WDKEY bits results in an watchdog timer time-out event.

The watchdog timer memory-mapped registers are listed in Table 2−1.

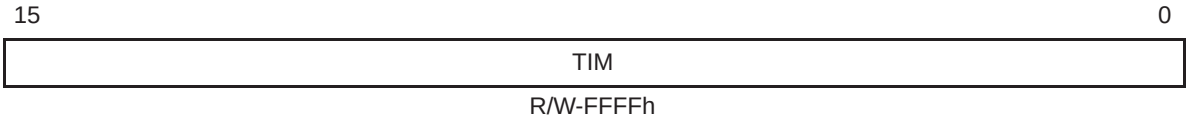
Table 2−1. Watchdog Timer Memory-Mapped Registers

I/O Address (Hex)	Name	Description
4000	WDTIM	Watchdog Timer Counter Register
4001	WDPRD	Watchdog Timer Period Register
4002	WDTCR	Watchdog Timer Control Register
4003	WDTCR2	Watchdog Timer Control Register 2

2.6.1 Watchdog Timer Counter Register (WDTIM)

WDTIM is loaded with the 16-bit value in the watchdog period register (WDPRD). When the prescaler value is decremented to 0, the WDTIM value is decremented.

Figure 2−3. Watchdog Timer Counter Register (WDTIM)



Legend: R = Read; W = Write

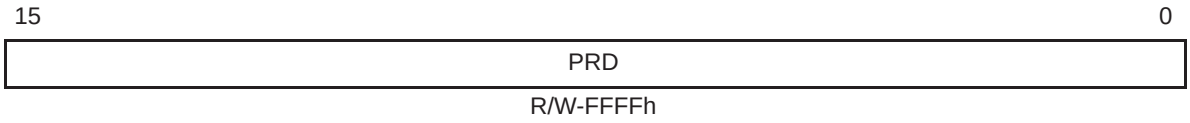
Table 2−2. Watchdog Timer Counter Register (WDTIM) Field Values

Bit	Field	Value	Description
15-0	TIM	0000h-FFFFh	This 16-bit value is loaded with the watchdog timer period register (WDPRD) value and decremented.

2.6.2 Watchdog Timer Period Register (WDPRD)

WDPRD is used to reload the watchdog timer counter register (WDTIM).

Figure 2−4. Watchdog Timer Period Register (WDPRD)



Legend: R = Read; W = Write

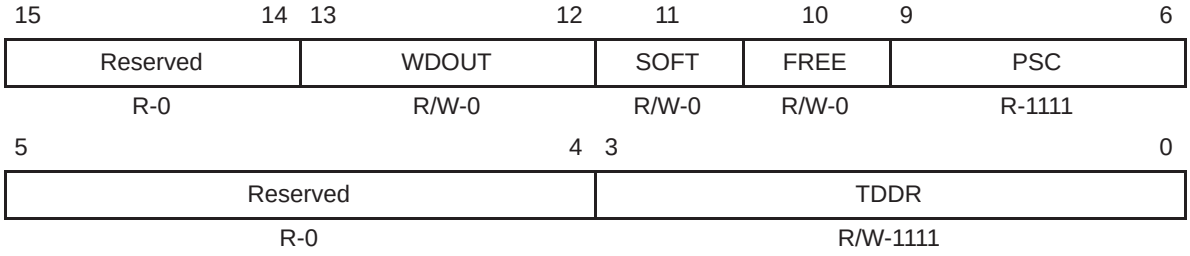
Table 2−3. Watchdog Timer Period Register (WDPRD) Field Values

Bit	Field	Value	Description
15-0	PRD	0000h-FFFFh	This 16-bit value is used to reload the watchdog timer counter register (WDTIM).

2.6.3 Watchdog Timer Control Register (WDTCR)

WDTCR provides the control and status information for the watchdog timer.

Figure 2−5. Watchdog Timer Control Register (WDTCR)



Legend: R = Read; W = Write; -*n* = Value after reset

Table 2−4. Watchdog Timer Control Register (WDTCR) Field Values

Bit	Field	Value	Description
15-14	Reserved		These reserved bits are always read as 0s. A value written to this field has no effect.
13-12	WDOUT		Watchdog timer output bits. These bits control the 4-output multiplexer to determine where the watchdog timer output is connected.
		00b	Output is internally connected to the watchdog timer interrupt.
		01b	Output is internally connected to the nonmaskable interrupt NMI.
		10b	Output is internally connected to RESET.
		11b	Output is not connected.

Table 2−4. Watchdog Timer Control Register (WDTCR) Field Values (Continued)

Bit	Field	Value	Description
11	SOFT		SOFT field bit is used in conjunction with FREE bit to determine the state of the watchdog timer when a breakpoint or emulation halt is encountered during debugging. When the FREE bit is cleared, the SOFT bit selects the watchdog timer mode.
		0	The watchdog timer stops immediately.
		1	The watchdog timer stops when the watchdog timer counter register (WDTIM) decrements to 0.
10	FREE		FREE field bit is used in conjunction with SOFT bit to determine the state of the watchdog timer when a breakpoint or emulation halt is encountered during debugging. When the FREE bit is cleared, the SOFT bit selects the watchdog timer mode.
		0	SOFT bit selects the watchdog timer mode.
		1	The watchdog timer runs free regardless of SOFT bit status.
9-6	PSC	xxxxb	Watchdog timer prescaler counter bits This read-only value specifies the count for the watchdog timer when in prescaler direct mode (PREMD = 0 in the WDTCR2). When the prescaler counter decrements to 0 or the watchdog timer is reset, the prescaler is reloaded (based on the TDDR bits) and the WDTIM value is decremented. In prescaler indirect mode (PREMD = 1), the prescaler is an internal 16-bit register that cannot be read and the PSC bits have no function.
5-4	Reserved		These reserved bits are always read as 0s. A value written to this field has no effect.

Table 2−4. Watchdog Timer Control Register (WDTCR) Field Values (Continued)

Bit	Field	Value	Description
3-0	TDDR	0-15	The watchdog timer divide-down register (prescaler period) bits control the value used to initialize the prescaler.
			In prescaler direct mode (PREMD = 0 in WDTCR2):
		0-15	This value specifies a direct prescaler count, up to 15, for the watchdog timer. When PSC is decremented to 0, PSC is loaded with the contents of TDDR.
			In prescaler indirect mode (PREMD = 1 in WDTCR2):
			In this mode, the prescaler is an internal 16-bit counter and cannot be read. This value specifies an indirect prescaler count, up to 65535, for the watchdog timer. When the prescaler is decremented to 0, the prescaler is reloaded with the prescaler value selected with TDDR. The valid prescaler values are:
		0000b	Prescaler value: 0001h
		0001b	Prescaler value: 0003h
		0010b	Prescaler value: 0007h
		0011b	Prescaler value: 000Fh
		0100b	Prescaler value: 001Fh
		0101b	Prescaler value: 003Fh
		0110b	Prescaler value: 007Fh
		0111b	Prescaler value: 00FFh
		1000b	Prescaler value: 01FFh
		1001b	Prescaler value: 03FFh
		1010b	Prescaler value: 07FFh
		1011b	Prescaler value: 0FFFh
		1100b	Prescaler value: 1FFFh
		1101b	Prescaler value: 3FFFh
		1110b	Prescaler value: 7FFFh
		1111b	Prescaler value: FFFFh

2.6.4 Watchdog Timer Control Register 2 (WDTCR2)

WDTCR2 contains bits to indicate the watchdog time-out flag, to enable the watchdog timer, and to set prescaler mode. WDTCR2 also provides the 12-bit WDKEY for watchdog timer service.

Figure 2−6. Watchdog Timer Control Register 2 (WDTCR2)

15	14	13	12	11 0
WDFLAG	WDEN	Reserved	PREMD	WDKEY
R/W-0	R/W-0	R-0	R/W-1	R/W-000000000000

Legend: R = Read; W = Write; -*n* = Value after reset

Table 2−5. Watchdog Timer Control Register 2 (WDTCR2) Field Values

Bit	Field	Value	Description
15	WDFLAG		Watchdog timer flag bit This bit can be cleared by enabling the watchdog timer, by a device reset, or by writing with a 1.
		0	No watchdog timer time-out event occurred.
		1	Watchdog timer time-out event occurred.
14	WDEN		Watchdog timer enable bit
		0	Watchdog timer is disabled. Watchdog timer output is disconnected from the watchdog timer time-out event and the counters do not run.
		1	Watchdog timer is enabled. Watchdog timer output is connected to the watchdog timer time-out event. Watchdog timer can be disabled by a watchdog timer time-out event or by a device reset.
13	Reserved	0	This reserved bit is always read as 0. A value written to this bit has no effect.

Table 2−5. Watchdog Timer Control Register 2 (WDTCR2) Field Values (Continued)

Bit	Field	Value	Description
12	PREMD		Prescaler mode select bit
		0	Direct mode A 4-bit prescaler is used and can be read in the PSC bits of WDTCR. When the prescaler is decremented to 0, it is loaded with the contents of TDDR in WDTCR.
		1	Indirect mode An internal 16-bit prescaler is used and cannot be read. When the prescaler is decremented to 0, it is loaded with the prescaler value associated with the TDDR value in WDTCR.
11-0	WDKEY	5C6h or A7Eh	Watchdog timer reset key bits A 12-bit value used to service the watchdog timer before a time-out event occurs. Only a write sequence of a 5C6h followed by an A7Eh services the watchdog timer. Any other write triggers a watchdog timer time-out event immediately.

This page is intentionally left blank.

Appendix A

Revision History

Table A−1 lists the changes made since the previous version of this document.

Table A−1. Document Revision History

Page	Additions/Modifications/Deletions
2-3	Corrected Figure 2−1.

Index

A

C

D

E

F

G

I

L

M

P

R

S

T

W