Preface

About This Manual

This document describes the 32-bit timer in the digital signal processors (DSPs) of the TMS320C600™ DSP family.

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.

- **TMS320C600 CPU and Instruction Set Reference Guide** (literature number SPRU189) describes the TMS320C600™ CPU architecture, instruction set, pipeline, and interrupts for these digital signal processors.
- **TMS320C600 DSP Peripherals Overview Reference Guide** (literature number SPRU190) describes the peripherals available on the TMS320C600™ DSPs.
- **TMS320C600 Technical Brief** (literature number SPRU197) gives an introduction to the TMS320C62x™ and TMS320C67x™ DSPs, development tools, and third-party support.
- **TMS320C64x Technical Overview** (SPRU395) gives an introduction to the TMS320C64x™ DSP and discusses the application areas that are enhanced by the TMS320C64x VelociTI™.
- **TMS320C6000 Programmer's Guide** (literature number SPRU198) describes ways to optimize C and assembly code for the TMS320C600™ DSPs and includes application program examples.
- **TMS320C6000 Code Composer Studio Tutorial** (literature number SPRU301) introduces the Code Composer Studio™ integrated development environment and software tools.
- **Code Composer Studio Application Programming Interface Reference Guide** (literature number SPRU321) describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.
- **TMS320C6x Peripheral Support Library Programmer's Reference** (literature number SPRU273) describes the contents of the TMS320C600™ peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.
Related Documentation From Texas Instruments

*TMS320C6000 Chip Support Library API Reference Guide* (literature number SPRU401) describes a set of application programming interfaces (APIs) used to configure and control the on-chip peripherals.

**Trademarks**

C6000, TMS320C6000, TMS320C62x, TMS320C67x, TMS320C64x, VelociTI, Code Composer Studio are trademarks of Texas Instruments.
32-Bit Timer

This document describes the 32-bit timer in the digital signal processors (DSPs) of the TMS320C6000™ DSP family.

1 Overview

The C6000™ DSP device has 32-bit general-purpose timers that can be used to:

- Time events
- Count events
- Generate pulses
- Interrupt the CPU
- Send synchronization events to the DMA

The timers have two signaling modes and can be clocked by an internal or an external source. The timers have an input pin and an output pin. The input and output pins (TINP and TOUT) can function as timer clock input and clock output. They can also be respectively configured for general-purpose input and output.

With an internal clock, for example, the timer can signal an external A/D converter to start a conversion, or it can trigger the DMA controller to begin a data transfer. With an external clock, the timer can count external events and interrupt the CPU after a specified number of events. Table 1 summarizes the differences between the C6000 timers. Figure 1 shows a block diagram of the timers.

Table 1. Differences in TMS320C6000 DSP Timers

<table>
<thead>
<tr>
<th>Features</th>
<th>C620x/C670x DSP</th>
<th>C621x/C671x DSP</th>
<th>C64x DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emulation halt support</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Internal timer input clock source</td>
<td>CPU rate/4</td>
<td>CPU rate/4</td>
<td>CPU rate/8</td>
</tr>
</tbody>
</table>
Overview

Figure 1. Timer Block Diagram

Peripheral Bus to CPU and DMA

DATIN
TOUT pin
Synchronizer
Internal clock
source
(A)

INVINP
FUNC10
INVOUT
TSTAT, timer output TINT, timer interrupt to CPU and DMA

HLD
Count enable

Timer count register

Count zero

Timer period register

Equals comparator

CLKSRC

Pulse generator

CP
PWID

TSTAT, timer output TINT, timer interrupt to CPU and DMA

HLD

PULSOUT

DATOUT

0 1

TOUT pin

TINP pin

Internal clock source (A)

1 0

Edge detect

DATIN

INVINP

Synchronizer

A C62x/C67x DSP uses CPU/4 clock as the internal clock source to the timer. C64x DSP uses CPU/8 clock as the internal clock source to the timer.
2 Resetting the Timers and Enabling Counting

Table 2 describes how using the GO and HLD bits in the timer control register (CTL) enable basic features of timer operation.

<table>
<thead>
<tr>
<th>Operation</th>
<th>GO</th>
<th>HLD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holding the timer</td>
<td>0</td>
<td>0</td>
<td>Counting is disabled.</td>
</tr>
<tr>
<td>Restarting the timer</td>
<td>0</td>
<td>1</td>
<td>Timer continues from the value before hold. The timer counter is not reset.</td>
</tr>
<tr>
<td>Reserved</td>
<td>1</td>
<td>0</td>
<td>Undefined.</td>
</tr>
<tr>
<td>Starting the timer</td>
<td>1</td>
<td>1</td>
<td>Timer counter resets to 0 and starts counting whenever enabled. Once set, GO self-clears.</td>
</tr>
</tbody>
</table>

Configuring a timer requires four basic steps:

1. If the timer is not currently in the hold state, place the timer in hold (HLD = 0). Note that after device reset, the timer is already in the hold state.
2. Write the desired value to the timer period register (PRD).
3. Write the desired value to the timer control register (CTL). Do not change the GO and HLD bits in CTL.
4. Start the timer by setting the GO and HLD bits in CTL to 1.

3 Timer Counting
The timer counter runs at the CPU clock rate. However, counting is enabled on the low-to-high transition of the timer count enable source. This transition is detected by the edge-detect circuit shown in Figure 1. Each time an active transition is detected, one CPU-clock-wide clock enable pulse is generated. This makes the counter appear as if it were getting clocked by the count enable source. Thus, this count enable source is referred to as the timer input clock source.

Once the timer reaches a value equal to the value in the timer period register (PRD), the timer is reset to 0 on the next CPU clock. Thus, the counter counts from 0 to N. Consider the case where the period is 2 and the CPU clock/4 is selected as the timer clock source (CLKSRC = 1) for C62x/C67x DSP. Once started, the timer counts the following sequence: 0, 0, 0, 0, 1, 1, 1, 1, 2, 0, 0, 0, 1, 1, 1, 1, 2, 0, 0, 0,... Note that although the counter counts from 0 to 2, the period is 8 (2 × 4) CPU clock cycles rather than 12 (3 × 4) CPU clock cycles. Thus, the countdown period is the value of TIMER PERIOD, not TIMER PERIOD + 1.

4 Timer Clock Source Selection
Low-to-high transitions (or high-to-low transitions, if INVINP = 1) of the timer input clock allow the timer counter to increment. Two sources are available to drive the timer input clock:

- The input value on the TINP pin, selected by CLKSRC = 0. This signal is synchronized to prevent any metastability caused by asynchronous external inputs. The value present on the TINP pin is reflected by DATIN.
- Internal clock source, selected by CLKSRC = 1. The C62x/C67x DSPs use CPU clock/4 as an internal clock source. The C64x DSPs use CPU clock/8 as an internal clock source.

5 Timer Pulse Generation
The two basic pulse generation modes are pulse mode (Figure 2) and clock mode (Figure 3). You can select the mode with the CP bit in the timer control register (CTL). Note that in pulse mode, PWID in the CTL can set the pulse width to either one or two input clock periods. The purpose of this feature is to provide minimum pulse widths if TSTAT drives the TOUT output. TSTAT drives this pin when TOUT is used as a timer pin (FUNC = 1), and may be inverted by setting INVOUT = 1. Table 3 gives equations for various TSTAT timing parameters in pulse and clock modes.
Figure 2. Timer Operation in Pulse Mode (CP = 0)

1 × timer clock source period (PWID = 0)
2 × timer clock source period (PWID = 1)

Period Register \( f(\text{timer input clock}) \)

TSTAT, TINT

Timer counter = timer period, TINT timer interrupt period

Figure 3. Timer Operation in Clock Mode (CP = 1)

Timer clock source period

Period

TSTAT, TINT

2 \times \text{Period Register} \quad f(\text{timer input clock})

TINT timer interrupt period

Table 3. TSTAT Parameters in Pulse and Clock Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Frequency</th>
<th>Period</th>
<th>Width High</th>
<th>Width Low</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( f(\text{clock source}) )</td>
<td>timer period register</td>
<td>( f(\text{clock source}) )</td>
<td>( f(\text{clock source}) )</td>
</tr>
<tr>
<td>Pulse</td>
<td>timer period register</td>
<td>( f(\text{clock source}) )</td>
<td>( \text{timer period register} - (\text{PWID} + 1) )</td>
<td>( \text{timer period register} - (\text{PWID} + 1) )</td>
</tr>
<tr>
<td>Clock</td>
<td>( f(\text{clock source}) )</td>
<td>2 \times \text{timer period register}</td>
<td>( \text{timer period register} )</td>
<td>( \text{timer period register} )</td>
</tr>
<tr>
<td></td>
<td>( 2 \times \text{timer period register} )</td>
<td>( f(\text{clock source}) )</td>
<td>( f(\text{clock source}) )</td>
<td>( f(\text{clock source}) )</td>
</tr>
</tbody>
</table>
6 Boundary Conditions in the Control Registers

The following boundary conditions affect timer operation:

1. Timer period (PRD) and timer count (CNT) value is 0: After device reset and before the timer starts counting, TSTAT is held at 0. After the timer starts running by setting HLD = 1 and GO = 1 while the period and counter registers are 0, the operation of the timer depends on the CP mode selected. In pulse mode, TSTAT = 1 regardless of whether or not the timer is held. In clock mode, when the timer is held (HLD = 0), TSTAT keeps its previous value and when HLD = 1, TSTAT toggles with a frequency of 1/2 of the CPU clock frequency. Note that in clock mode interrupts occur at twice the PRD period, except for the first interrupt. The first interrupt generated occurs after one PRD clock period only, and subsequent interrupts are generated every two PRD clock periods.

2. Counter overflow: When the timer count register (CNT) is set to a value greater than the value of the timer period register (PRD), the counter reaches its maximum value (FFFF FFFFh), rolls over to 0, and continues.

3. Writing to registers of an active timer: Writes from the peripheral bus override register updates to CNT and new status updates to the timer control register (CTL).

4. Small timer period values in pulse mode: Note that small periods in pulse mode can cause TSTAT to remain high. This condition occurs when TIMER PERIOD ≤ PWID + 1.

7 Timer Interrupts

The TSTAT signal directly drives the CPU interrupt, as well as a DMA synchronization event. The frequency of the interrupt is the same as the frequency of TSTAT.

8 Timer Pins as General-Purpose Input/Output

Upon device reset, the timer pins TINP and TOUT are general-purpose input and output (I/O) pins, respectively. By configuring the timer control register (CTL), the TINP and TOUT pins can operate as general-purpose pins even when the timer is running.

The TINP pin is always a general-purpose input pin, if the timer is not running. If the timer is running, the TINP pin is a general-purpose input pin if CLKSRC = 1 in CTL, which indicates that an internal clock source is used instead of the TINP pin. When TINP is a general-purpose input pin, the input value is readable from the DATIN bit in CTL.

The TOUT pin is a general-purpose output pin if FUNC = 0 in CTL, independent of timer operation. The FUNC bit, as shown in Figure 1, selects either the DATOUT or the TSTAT value to be driven on the TOUT pin.

9 Emulation Operation

During debug using the emulator, the CPU may be halted on an execute packet boundary for single stepping, benchmarking, profiling, or other debug uses. For C620x/C670x DSP, during an emulation halt the timer halts when the CPU clock/4 is selected as the clock source (CLKSRC = 1). For C64x DSP, during an emulation halt the timer halts when the CPU clock/8 is selected as the clock source (CLKSRC = 1). Here, the counter is only enabled to count during those cycles when the CPU is not stalled due to the emulation halt. Thus, counting is reenabled during single-step operation. If CLKSRC = 0, the timer continues counting as programmed. For C621x/C671x DSP, the timer continues counting during emulation halt regardless of clock source.
10 Timer Registers

Table 4 describes the three registers that configure timer operation.

Table 4. Timer Registers

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Register Name</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>Timer Control Register</td>
<td>Section 10.1</td>
</tr>
<tr>
<td>PRD</td>
<td>Timer Period Register</td>
<td>Section 10.2</td>
</tr>
<tr>
<td>CNT</td>
<td>Timer Count Register</td>
<td>Section 10.3</td>
</tr>
</tbody>
</table>

10.1 Timer Control Register (CTL)

The timer control register (CTL) determines the operating mode of the timer, monitors the timer status, and controls the function of the TOUT pin. The timer control register is shown in Figure 4, for the C62x/C67x DSP, and in Figure 5, for the C64x DSP, and described in Table 5.

Figure 4. TMS320C62x/C67x Timer Control Register (CTL)

<table>
<thead>
<tr>
<th>31</th>
<th>16</th>
<th>15</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>16</td>
<td>15</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>R-0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>HLD</td>
<td>GO</td>
<td>Reserved</td>
<td>PWID</td>
<td>DATIN</td>
<td>DATOUT</td>
<td>INVOUT</td>
<td>FUNC</td>
</tr>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

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

Figure 5. TMS320C64x Timer Control Register (CTL)

<table>
<thead>
<tr>
<th>31</th>
<th>16</th>
<th>15</th>
<th>14</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>R-0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SPND</td>
<td>Reserved</td>
<td>TSTAT</td>
<td>INVINP</td>
<td>CLKSRC</td>
<td>CP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>R-0</td>
<td>R-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LEGEND: R = Read only; R/W = Read/Write; -n = value after reset; -x = value is indeterminate after reset
Table 5. Timer Control Register (CTL) Field Descriptions

<table>
<thead>
<tr>
<th>Bit</th>
<th><strong>field</strong> (1)</th>
<th><strong>symval</strong> (1)</th>
<th><strong>Value</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>31-16</td>
<td>Reserved</td>
<td>-</td>
<td>0</td>
<td>Reserved. The reserved bit location is always read as 0. A value written to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>this field has no effect.</td>
</tr>
<tr>
<td>15</td>
<td>SPND (2)</td>
<td></td>
<td>0</td>
<td>Reserved. Timer stops counting during an emulation halt. Only affects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>operation if the clock source is internal, CLKSRC = 1. Reads always</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>return a 0.</td>
</tr>
<tr>
<td>14-12</td>
<td>Reserved</td>
<td>-</td>
<td>0</td>
<td>Reserved. The reserved bit location is always read as 0. A value written to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>this field has no effect.</td>
</tr>
<tr>
<td>11</td>
<td>TSTAT</td>
<td></td>
<td>0</td>
<td>Timer status bit. Value of timer output.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>INVINP</td>
<td>NO</td>
<td>0</td>
<td>Noninverted TINP drives timer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>1</td>
<td>Inverted TINP drives timer.</td>
</tr>
<tr>
<td>9</td>
<td>CLKSRC</td>
<td>EXTERNAL</td>
<td>0</td>
<td>Timer input clock source bit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CUTOVR4</td>
<td>1</td>
<td>External clock source drives the TINP pin.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Internal clock source.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Only valid on C64x DSP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For C62x/C67x DSP: CPU clock/4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For C64x DSP: CPU clock/8</td>
</tr>
<tr>
<td>8</td>
<td>CP</td>
<td>PULSE</td>
<td>0</td>
<td>Clock/pulse mode enable bit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLOCK</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>HLD</td>
<td>YES</td>
<td>0</td>
<td>Counter is disabled and held in the current state.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td>1</td>
<td>Counter is allowed to count.</td>
</tr>
<tr>
<td>6</td>
<td>GO</td>
<td>NO</td>
<td>0</td>
<td>GO bit. Resets and starts the timer counter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td>-</td>
<td>0</td>
<td>Reserved. The reserved bit location is always read as 0. A value written to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>this field has no effect.</td>
</tr>
<tr>
<td>4</td>
<td>PWID</td>
<td>ONE</td>
<td>0</td>
<td>Pulse width bit. Only used in pulse mode (CP = 0).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TWO</td>
<td>1</td>
<td>TSTAT goes inactive after the timer counter value equals the timer period value.</td>
</tr>
<tr>
<td>3</td>
<td>DATIN</td>
<td>0</td>
<td>0</td>
<td>Data in bit. Value on TINP pin.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>Value on TINP pin is logic low.</td>
</tr>
<tr>
<td>2</td>
<td>DATOUT</td>
<td>0</td>
<td>0</td>
<td>Data output bit. DATOUT is driven on TOUT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>TSTAT is driven on TOUT after inversion by INVOUT.</td>
</tr>
</tbody>
</table>

(1) For CSL implementation, use the notation TIMER_CTL_field_symval
(2) For C620x/C670x/C64x DSPs; for C621x/C671x DSP, this bit is reserved.
### Timer Registers

#### Table 5. Timer Control Register (CTL) Field Descriptions (continued)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field (1)</th>
<th>symval (1)</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INVOUT</td>
<td></td>
<td></td>
<td>TOUT inverter control bit (used only if FUNC = 1).</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td>0</td>
<td>Noninverted TSTAT drives TOUT.</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td></td>
<td>1</td>
<td>Inverted TSTAT drives TOUT.</td>
</tr>
<tr>
<td>0</td>
<td>FUNC</td>
<td></td>
<td></td>
<td>Function of TOUT pin.</td>
</tr>
<tr>
<td></td>
<td>GPIO</td>
<td></td>
<td>0</td>
<td>TOUT is a general-purpose output pin.</td>
</tr>
<tr>
<td></td>
<td>TOUT</td>
<td></td>
<td>1</td>
<td>TOUT is a timer output pin.</td>
</tr>
</tbody>
</table>

#### 10.2 Timer Period Register (PRD)

The timer period register (PRD) contains the number of timer input clock cycles to count. This number controls the TSTAT signal frequency. The timer period register is shown in Figure 6 and described in Table 6.

![Figure 6. Timer Period Register (PRD)](image)

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

#### Table 6. Timer Period Register (PRD) Field Descriptions

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>symval (1)</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-0</td>
<td>PRD</td>
<td>OF(value)</td>
<td>0-FFFF FFFFh</td>
<td>Period bits. This 32-bit value is the number of timer input clock cycles to count and is used to reload the timer count register (CNT). This number controls the frequency of the timer output status bit (TSTAT).</td>
</tr>
</tbody>
</table>

(1) For CSL implementation, use the notation TIMER_PRD_PRD_symval
10.3 Timer Count Register (CNT)

The timer count register (CNT) contains the current value of the incrementing counter. The timer count register increments by 1 when it is enabled to count and resets to 0 on the next CPU clock after the value in the timer period register (PRD) is reached. The timer count register is shown in Figure 7 and described in Table 7.

Figure 7. Timer Count Register (CNT)

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer Count (CNT)</td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td></td>
</tr>
</tbody>
</table>

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

Table 7. Timer Count Register (CNT) Field Descriptions

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>symval (1)</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-0</td>
<td>CNT</td>
<td>OF(value)</td>
<td>0-FFFFFFFh</td>
<td>Main count bits. This 32-bit value is the current count of the main counter. This value is incremented by 1 every input clock cycle.</td>
</tr>
</tbody>
</table>

(1) For CSL implementation, use the notation TIMER_CNT_CNT_symval
Appendix A  Revision History

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

Table A-1. Document Revision History

<table>
<thead>
<tr>
<th>Page</th>
<th>Additions/Modifications/Deletions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Changed the emulation halted support feature from No to Yes under the C64x DSP column of Table 1.</td>
</tr>
<tr>
<td>Section 6</td>
<td>Changed number 1 of Section 6.</td>
</tr>
<tr>
<td>Figure 1</td>
<td>Changed Figure 1.</td>
</tr>
<tr>
<td>Section 9</td>
<td>Changed Section 9.</td>
</tr>
<tr>
<td>Table 5</td>
<td>Changed the second footnote of Table 5.</td>
</tr>
</tbody>
</table>
time events 3
count events 3
generate pulses 3
send synchronization events to the DMA 3
interrupt the CPU 3
overview 3
timer 3
  overview 3
timer 3
  resetting 3
timer 3
    enabling counting 3
    enabling counting 3
    resetting the timer 3
    counting 3
counting 3
clock source selection 3
timer 3
clock source selection 3
selection of clock sources 3
timer 3
  pulse generation 3
  pulse generation 3
timer 3
  register boundary conditions 3
control register boundary conditions 3
registers 3
  boundary conditions 3
timer 3
  interrupts 3
interrupts 3
timer 3
  general-purpose input/output 3
  general-purpose input/output 3
timer 3
  emulation operation 3
emulation operation 3
timer 3
registers 3
timer 3
registers 3
timer control register (CTL) 3
timer control register (CTL) 3
CTL 3
registers 3
timer period register (PRD) 3
timer period register (PRD) 3
PRD 3
registers 3
timer count register (CNT) 3
timer count register (CNT) 3
CNT 3
TSTAT bit 3
INVINP bit 3
CLKSRC bit 3
CP bit 3
HLD bit 3
GO bit 3
PWID bit 3
DATIN bit 3
DATOUT bit 3
INVOUt bit 3
FUNC bit 3
SPND bit 3
CNT bits 3
modes 3
  pulse and clock 3
TSTAT parameters 3
notational conventions 5
related documentation from Texas Instruments 5
trademarks 6
related documentation from Texas Instruments 6
time events 7
count events 7
generate pulses 7
send synchronization events to the DMA 7
interrupt the CPU 7
overview 7
timer 7
  overview 7
time events 8
count events 8
generate pulses 8
send synchronization events to the DMA 8
interrupt the CPU 8
overview 8
timer 8
  overview 8
timer 9
  resetting 9
timer 9
  enabling counting 9
  enabling counting 9
  resetting the timer 9
timer 9
  counting 9
counting 9
clock source selection 9
timer 9
clock source selection 9
selection of clock sources 9
timer 9
pulse generation 9
pulse generation 9
timer 9
resetting 9
timer 9
  enabling counting 9
  enabling counting 9
resetting the timer 9
modes 10
pulse generation 10
timer 10
default parameters 10
timer 10
default parameters 10
timer 11
register boundary conditions 11
control register boundary conditions 11
registers 11
  boundary conditions 11
timer 11
  interrupts 11
interrupts 11
timer 11
  general-purpose input/output 11
  general-purpose input/output 11
timer 11
  emulation operation 11
emulation operation 11
timer 11
  register boundary conditions 11
control register boundary conditions 11
registers 11
  boundary conditions 11
timer 12
  registers 12
registers 12
timer 12
registers 12
  timer control register (CTL) 12
timer control register (CTL) 12
CTL 12
TSTAT bit 12
INVINP bit 12
CLKSRC bit 12
CP bit 12
HLD bit 12
GO bit 12

Index
<table>
<thead>
<tr>
<th>Description</th>
<th>Bit Width</th>
<th>Register Size</th>
<th>Timers</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWID bit</td>
<td>12</td>
<td>registers 13</td>
<td>timer 13</td>
</tr>
<tr>
<td>DATIN bit</td>
<td>12</td>
<td>registers 14</td>
<td>timer 14</td>
</tr>
<tr>
<td>DATOUT bit</td>
<td>12</td>
<td>timer period register (PRD) 14</td>
<td></td>
</tr>
<tr>
<td>INVOUT bit</td>
<td>12</td>
<td>registers 15</td>
<td>timer 15</td>
</tr>
<tr>
<td>FUNC bit</td>
<td>12</td>
<td>timer period register (PRD) 14</td>
<td></td>
</tr>
<tr>
<td>SPND bit</td>
<td>12</td>
<td>PRD 14</td>
<td>timer 15</td>
</tr>
<tr>
<td>timer</td>
<td>12</td>
<td>registers 14</td>
<td>timer 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>Audio</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Automotive</td>
</tr>
<tr>
<td>DSP</td>
<td>Broadband</td>
</tr>
<tr>
<td>Interface</td>
<td>Digital Control</td>
</tr>
<tr>
<td>Logic</td>
<td>Military</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Optical Networking</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Telephony</td>
</tr>
<tr>
<td></td>
<td>Video &amp; Imaging</td>
</tr>
<tr>
<td></td>
<td>Wireless</td>
</tr>
</tbody>
</table>

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

Copyright © 2005, Texas Instruments Incorporated