

Introduction

This document describes the specifications for a Advanced Microcontroller Bus Architecture (AMBA®) specification's Advanced eXtensible Interface (AXI) Timer/Counter core.

The AXI Timer/Counter is a 32/64-bit timer module that attaches to the AXI4-Lite interface.

Features

- AXI interface based on the AXI4-Lite specification
- Two programmable interval timers with interrupt, event generation, and event capture capabilities
- Configurable counter width
- One Pulse Width Modulation (PWM) output
- Cascaded operation of timers in generate and capture modes
- Freeze input for halting counters during software debug

| LogiCORE IP Facts | | | | | | | | | | | | |
|---|--|-----|------------|--|--|--|--|--|--|--|--|--|
| Core Specifics | | | | | | | | | | | | |
| Supported Device Family ⁽¹⁾ | Zynq™-7000 ⁽²⁾ , Virtex®-7, Kintex™-7, Artix™-7, Virtex-6, Spartan®-6 | | | | | | | | | | | |
| Supported User Interfaces | AXI4-Lite | | | | | | | | | | | |
| Resources | | | Frequency | | | | | | | | | |
| Slices | LUTs | FFs | Block RAMs | Max Freq | | | | | | | | |
| See Table 12 and Table 13 | | | 0 | See Table 12 , Table 13 , Table 14 | | | | | | | | |
| Provided with Core | | | | | | | | | | | | |
| Documentation | Product Specification | | | | | | | | | | | |
| Design Files | ISE: VHDL Vivado: Encrypted RTL | | | | | | | | | | | |
| Example Design | Not Provided | | | | | | | | | | | |
| Test Bench | Not Provided | | | | | | | | | | | |
| Constraints File | None | | | | | | | | | | | |
| Simulation Model | None | | | | | | | | | | | |
| Supported S/W Driver ⁽³⁾ | Standalone and Linux | | | | | | | | | | | |
| Tested Design Tools ⁽⁴⁾ | | | | | | | | | | | | |
| Design Entry Tools | Vivado™ Design Suite v2012.2 ⁽⁵⁾ ISE™ Design Suite v14.2 | | | | | | | | | | | |
| Simulation | Mentor Graphics ModelSim | | | | | | | | | | | |
| Synthesis Tools | Xilinx Synthesis Technology (XST) Vivado High-Level Synthesis (HLS) | | | | | | | | | | | |
| Support | | | | | | | | | | | | |
| Provided by Xilinx, Inc. | | | | | | | | | | | | |

1. For a complete list of supported derivative devices, see [IDS Embedded Edition Derivative Device Support](#).
2. Supported in ISE Design Suite implementations only.
3. Standalone driver information can be found in the EDK or SDK installation directory.
See `xilinx_drivers.htm` in `<install_directory>/doc/usenglish`. Linux OS and driver support information is available from <http://wiki.xilinx.com>.
4. For the supported versions of the tools, see the [Xilinx Design Tools: Release Notes Guide](#).
5. Supports only 7 series devices.

Functional Description

The Timer/Counter is organized as two identical timer modules as shown in [Figure 2](#). Each timer module has an associated load register that is used to hold either the initial value for the counter for event generation or a capture value, depending on the mode of the timer.

The *generate value* is used to generate a single interrupt at the expiration of an interval or a continuous series of interrupts with a programmable interval. The *capture value* is the timer value that has been latched on detection of an external event. The clock rate of the timer modules is S_AXI_ACLK (no prescaling of the clock is performed). All of the Timer/Counter interrupts are ORed together to generate a single external interrupt signal. The interrupt service routine reads the control/status registers to determine the source of the interrupt.

Programming Model

Timer Modes

There are four modes that can be used with the two Timer/Counter modules:

- Generate mode
- Capture mode
- Pulse Width Modulation (PWM) mode
- Cascade mode

The modes and their characteristics are described in the following sections.

Generate Mode

In the Generate mode, the value in the load register is loaded into the counter. The counter, when enabled, begins to count up or down, depending on the selection of the Up/Down Count Timer (UDT) bit in the Timer Control Status Register (TCSR). See [Figure 6](#) and [Figure 7](#). On transition of the carry out of the counter, the counter stops or automatically reloads the generate value from the load register and continues counting as selected by the Auto Reload/Hold (ARHT) bit in the TCSR. The TINT bit is set in TCSR and, if enabled, the external GenerateOut signal is driven to 1 for one clock cycle. If enabled, the interrupt signal for the timer is driven to 1 for one clock cycle. This mode is useful for generating repetitive interrupts or external signals with a specified interval.

Characteristics

The generate mode has the following characteristics:

- The value loaded into the load register is called the generate value.
- On start-up, the generate value in the load register must be loaded into the counter by setting the Load bit in the Timer Control Status Register (TCSR). This applies whether the counter is set up to Auto Reload or Hold when the interval has expired. Setting the Load bit to '1' loads the counter with the value in the load register. For proper operation, the Load bit must be cleared before the counter is enabled or along with setting the enable bit. The timer/counter starts ticking when Enable is set (ENT).
- When the ARHT bit (Auto Reload/Hold) is set to '1' and the counter rolls over from all '1's to all '0's when counting up, or conversely from all '0's to all '1's when counting down, the generate value in the load register is automatically reloaded into the counter and the counter continues to count. If the GenerateOut signal is enabled (bit GENT in the TCSR), an output pulse is generated (one clock period in width). This is useful for generating a repetitive pulse train with a specified period.

- When the ARHT bit (Auto Reload/Hold) is set to '0' and the counter rolls over from all '1's to all '0's, when counting up, or conversely, from all '0's to all '1's, when counting down, the counter holds at the current value and does not reload the generate value. If the GenerateOut signal is enabled (bit GENT in the TCSR), an output pulse of one clock period in width is generated. This is useful for a one-shot pulse that is to be generated after a specified period of time.
- The counter can be set up to count either up or down as determined by the selection of the UDT bit in the TCSR. If the counter is set up as a down counter, the generate value is the number of clocks in the timing interval. The period of the GenerateOut signal is the generate value times the clock period.
- When the counter is set to count down,
$$\text{TIMING_INTERVAL} = (\text{TLRx} + 2) \times \text{AXI_CLOCK_PERIOD}$$
- When the counter is set to count up,
$$\text{TIMING_INTERVAL} = (\text{MAX_COUNT} - \text{TLRx} + 2) \times \text{AXI_CLOCK_PERIOD}$$
where MAX_COUNT is the maximum count value of the counter, such as 0xFFFFFFFF for a 32-bit counter.
- The GenerateOut signals can be configured as high-true or low-true.

Capture Mode

In Capture Mode, the value of the counter is stored in the load register when the external capture signal is asserted. The TINT bit is also set in the Timer Control Status Register (TCSR) on detection of the capture event. The counter can be configured as an up or down counter for this mode as determined by the selection of the UDT bit in TCSR. The ARHT bit controls whether the capture value is overwritten with a new capture value before the previous TINT flag is cleared. This mode is useful for time tagging external events while simultaneously generating an interrupt.

Characteristics

Capture Mode has the following characteristics:

- The capture signal can be configured to be low-true or high-true.
- The capture signal is sampled within the Timer/Counter with the S_AXI_ACLK. The capture event is defined as the transition on the sampled signal to the asserted state. For example, if the capture signal is defined to be high-true, then the capture event is when the sampled signal, synchronized to the S_AXI_ACLK, transitions from '0' to '1'.
- When the capture event occurs, the counter value is written to the load register. This value is called the capture value.
- When the ARHT bit (Auto Reload/Hold) is set to '0' and the capture event occurs, the capture value is written to the Load Register which holds the capture value until the load register is read. If the load register is not read, subsequent capture events do not update the load register and are lost.
- When the ARHT bit (Auto Reload /Hold) is set to '1' and the capture event occurs, the capture value is always written to the load register. Subsequent capture events update the load register and overwrite the previous value, whether it has been read or not.
- The counter can be set up to count either up or down as determined by the selection of the UDT bit in the Timer Control Status Register (TCSR).

Pulse Width Modulation (PWM) Mode

In PWM mode, two timer/counters are used as a pair to produce an output signal (`PWM0`) with a specified frequency and duty factor. Timer0 sets the period and Timer1 sets the high time for the `PWM0` output.

Characteristics

PWM Mode has the following characteristics:

- The mode for both Timer0 and Timer1 must be set to Generate Mode (bit MDT in the TCSR set to '0').
- The PWMA0 bit in TCSR0 and PWMB0 bit in TCSR1 must be set to '1' to enable PWM mode.
- The `GenerateOut` signals must be enabled in the TCSR (bit GENT set to '1'). The `PWM0` signal is generated from the `GenerateOut` signals of Timer0 and Timer1, so these signals must be enabled in both timer/counters.
- The assertion level of the `GenerateOut` signals for both timers in the pair must be set to '1'. This is done by setting `C_GEN0_ASSERT` and `C_GEN1_ASSERT` to '1'.
- The counter can be set to count up or down.

Setting the PWM Period and Duty Factor

The PWM period is determined by the generate value in the Timer0 load register (TLR0). The PWM high time is determined by the generate value in the Timer1 load register (TLR1). The period and duty factor are calculated as follows:

When counters are configured to count up (UDT = '0'):

```
PWM_PERIOD = (MAX_COUNT - TLR0 + 2) x AXI_CLOCK_PERIOD  
PWM_HIGH_TIME = (MAX_COUNT - TLR1 + 2) x AXI_CLOCK_PERIOD
```

When counters are configured to count down (UDT = '1'):

```
PWM_PERIOD = (TLR0 + 2) x AXI_CLOCK_PERIOD  
PWM_HIGH_TIME = (TLR1 + 2) x AXI_CLOCK_PERIOD
```

where MAX_COUNT is the maximum count value for the counter, such as 0xFFFFFFFF for a 32-bit counter.

Cascade Mode

In the Cascade mode, the two timer/counters are cascaded to operate as a single 64-bit counter/timer. The cascaded counter can work in both generate and capture modes. Timer Control Status Register for timer/counter 0 (TCSR0) acts as the control and status register for the cascaded counter. TCSR1 is ignored in this mode.

This mode is used when there is a requirement for a timer/counter of more than 32-bits wide. Cascaded operation requires using Timer0 and Timer1 together as a pair. The counting event for the Timer1 is when the Timer0 rolls over from all '1's to all '0's or vice-versa when counting down.

Characteristics

The cascade mode has the following characteristics:

- Parameter `C_ONE_TIMER_ONLY` should be set to '0' as both the timers are required for a cascaded operation.
- Parameter `C_COUNT_WIDTH` should be 32 as it represents the width of each timer/counter in the core.
- Load Registers of both timer/counters are used (TLR0 and TLR1 - TLR1 for higher 32-bit and TLR0 for lower 32-bit). The value loaded into the load registers is called the generate value in generate mode. And the capture value is captured in these load registers in capture mode.

- Timer/counter 0 control register TCSR0, GenerateOut0, Capture event 0 are valid in this mode. Timer 1 related signals are invalid, that is, TCSR1, GenerateOut1 and Capture event 1 are not used. TCSR1 is used only for loading the TLR1 register.
- CASC bit in Timer Control Status Register 0 (TCSR0) must be set for the counters to be in cascade mode. This bit must be set before enabling the timer/counter.
- The sequence of accesses for generate and capture modes are as mentioned in previous sections.
- In generate mode, when the counter is set to count down,

`TIMING_INTERVAL = (TLR + 4) × AXI_CLOCK_PERIOD`

where TLR is the concatenated value of TLR1 and TLR0 ($TLR = \{TLR1, TLR0\}$).

- In generate mode, when the counter is set to count up,

`TIMING_INTERVAL = (MAX_COUNT - TLR + 4) × AXI_CLOCK_PERIOD`

where MAX_COUNT is the maximum count value of the counter, such as 0xFFFFFFFFFFFFFF for a 64-bit counter and TLR is the concatenated value of TLR1 and TLR0 ($TLR = \{TLR1, TLR0\}$).

The following are the steps for running the 64-bit counter/timer in generate mode:

1. Clear the timer enable bits in control registers (TCSR0 and TCSR1).
2. Write the lower 32-bit timer/counter load register (TLR0).
3. Write the higher 32-bit timer/counter load register (TLR1).
4. Set the CASC bit in Control register TCSR0.
5. Set other mode control bits in control register (TCSR0) as needed.
6. Enable the timer in Control register (TCSR0).

The following are the steps for reading the 64-bit counter/timer:

1. Read the upper 32-bit timer/counter register (TCR1).
2. Read the lower 32-bit timer/counter register (TCR0).
3. Read the upper 32-bit timer/counter register (TCR1) again. If the value is different from the 32-bit upper value read previously, go back to previous step (reading TCR0). Otherwise 64-bit timer counter value is correct.

Interrupts

The TC interrupt signals can be enabled or disabled with the ENIT bit in the TCSR. The interrupt status bit (TINT) in the TCSR cannot be disabled and always reflects the current state of the timer interrupt. In Generate Mode, a timer interrupt is caused by the counter rolling over (the same condition used to reload the counter when ARHT is set to '1'). In Capture Mode, the interrupt event is the capture event. Characteristics of the interrupts are:

- Interrupt events can only occur when the timer is enabled. In Capture Mode, this prevents interrupts from occurring before the timer is enabled.
- The interrupt signal goes high when the interrupt condition is met and the interrupt is enabled in the TCSR. The interrupt is asserted when the interrupt signal is high.
- A single interrupt signal is provided. The interrupt signal is the OR of the interrupts from the two counters. The interrupt service routine must poll the TCSRs to determine the source or sources of the interrupt.
- The interrupt status bit (TINT in the TCSR) can only be cleared by writing a '1' to it. Writing a '0' to it has no effect on the bit. Because the interrupt condition is an edge (the counter rollover or the capture event), it can be cleared at any time and does not indicate an interrupt condition until the next interrupt event.
- In cascade mode, only Timer 0 interrupt events occur. There will not be any interrupts from Timer 1.

The top level block diagram of the AXI Timer/Counter is shown in [Figure 1](#).

The top level modules of the AXI Timer/Counter are:

- AXI4-Lite IPIF Module
- Timer/Counter

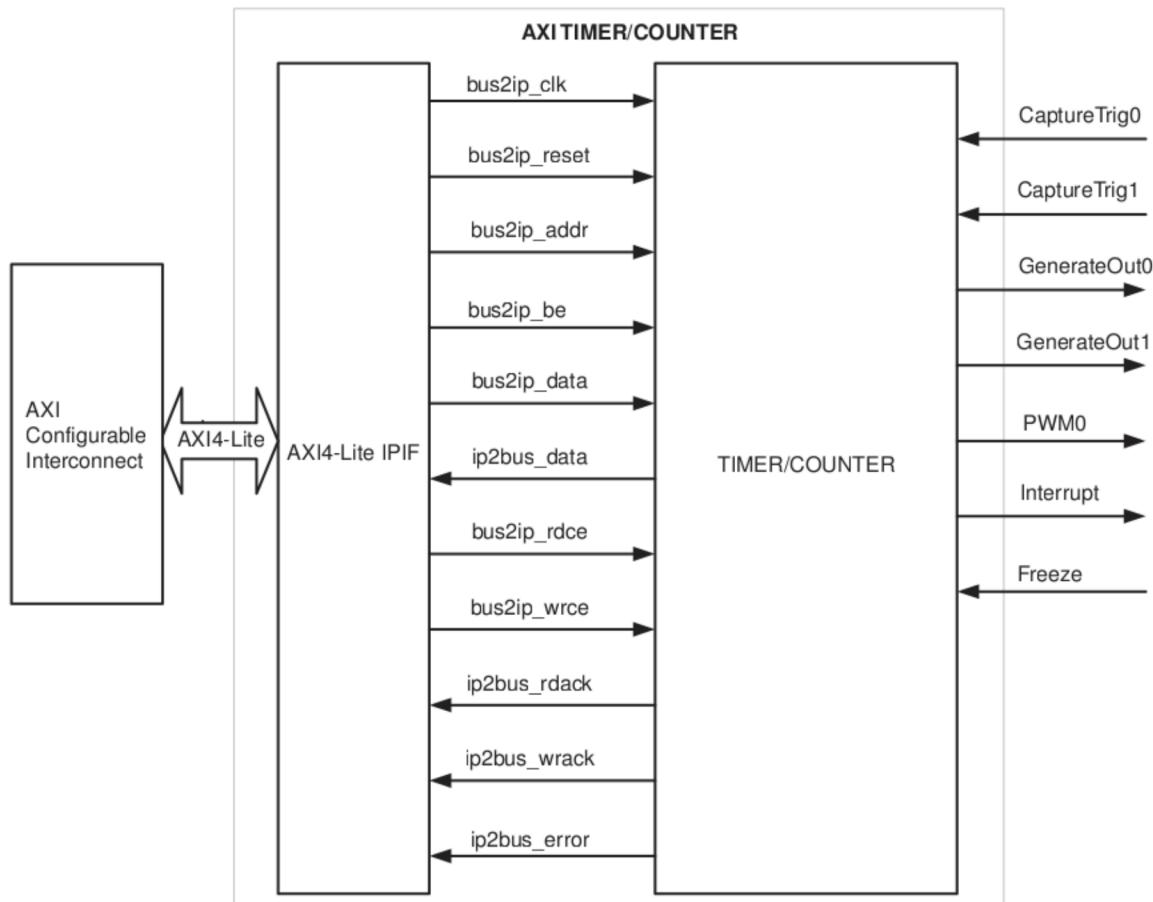


Figure 1: Block Diagram of AXI Timer

The detailed block diagram of the AXI Timer/Counter is shown in [Figure 2](#).

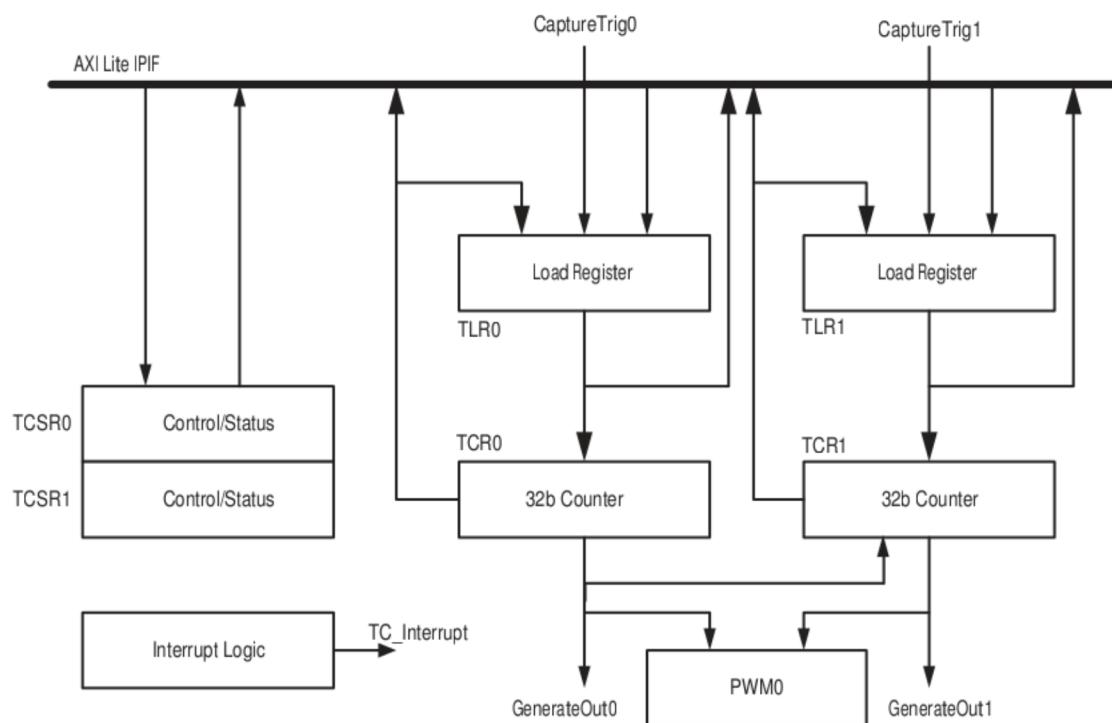


Figure 2: Detailed Block Diagram of AXI Timer

Input/Output Signals

The AXI Timer Input/Output (I/O) signals are listed and described in [Table 1](#).

Table 1: I/O Signal Description

| Port | Signal Name | Interface | I/O | Initial State | Description |
|-----------------------------------|--------------------------------------|-----------|-----|---------------|--|
| System Signals | | | | | |
| P1 | S_AXI_ACLK | System | I | - | AXI Clock |
| P2 | S_AXI_ARESETn | System | I | - | AXI Reset, active-Low |
| P3 | Interrupt | System | O | 0x0 | AXI Timer Interrupt; Level sensitive, rising |
| AXI Write Address Channel Signals | | | | | |
| P4 | S_AXI_AWADDR[C_S_AXI_ADDR_WIDTH-1:0] | AXI | I | - | AXI Write address. The write address bus gives the address of the write transaction. |
| P5 | S_AXI_AWVALID | AXI | I | - | Write address valid. This signal indicates that valid write address is available. |
| P6 | S_AXI_AWREADY | AXI | O | 0x0 | Write address ready. This signal indicates that the slave is ready to accept an address. |

Table 1: I/O Signal Description (Cont.)

| Port | Signal Name | Interface | I/O | Initial State | Description |
|------------------------------------|--|-----------|-----|---------------|---|
| AXI Write Channel Signals | | | | | |
| P7 | S_AXI_WDATA[C_S_AXI_DATA_WIDTH - 1:0] | AXI | I | - | Write data |
| P8 | S_AXI_WSTB[C_S_AXI_DATA_WIDTH/8-1:0] | AXI | I | - | Write strobes. This signal indicates which byte lanes to update in memory. ⁽¹⁾ |
| P9 | S_AXI_WVALID | AXI | I | - | Write valid. This signal indicates that valid write data and strobes are available. |
| P10 | S_AXI_WREADY | AXI | O | 0x0 | Write ready. This signal indicates that the slave can accept the write data. |
| AXI Write Response Channel Signals | | | | | |
| P11 | S_AXI_BRESP[1:0] | AXI | O | 0x0 | Write response. This signal indicates the status of the write transaction. "00" - OKAY (normal response) "10" - SLVERR (error condition) |
| P12 | S_AXI_BVALID | AXI | O | 0x0 | Write response valid. This signal indicates that a valid write response is available. |
| P13 | S_AXI_BREADY | AXI | I | - | Response ready. This signal indicates that the master can accept the response information. |
| AXI Read Address Channel Signals | | | | | |
| P14 | S_AXI_ARADDR[C_S_AXI_ADDR_WIDTH - 1:0] | AXI | I | - | Read address. The read address bus gives the address of a read transaction. |
| P15 | S_AXI_ARVALID | AXI | I | - | Read address valid. This signal indicates, when HIGH, that the read address is valid and remains stable until the address acknowledgement signal, S_AXI_ARREADY, is high. |
| P16 | S_AXI_ARREADY | AXI | O | 0x1 | Read address ready. This signal indicates that the slave is ready to accept an address. |
| AXI Read Data Channel Signals | | | | | |
| P17 | S_AXI_RDATA[C_S_AXI_DATA_WIDTH - 1:0] | AXI | O | 0x0 | Read data |
| P18 | S_AXI_RRESP[1:0] | AXI | O | 0x0 | Read response. This signal indicates the status of the read transfer. "00" - OKAY (normal response) "10" - SLVERR (error condition) |
| P19 | S_AXI_RVALID | AXI | O | 0x0 | Read valid. This signal indicates that the required read data is available and the read transfer can complete. |
| P20 | S_AXI_RREADY | AXI | I | - | Read ready. This signal indicates that the master can accept the read data and response information. |

Table 1: I/O Signal Description (Cont.)

| Port | Signal Name | Interface | I/O | Initial State | Description |
|---------------------------|--------------|-----------|-----|---------------|---|
| AXI Timer/Counter Signals | | | | | |
| P21 | CaptureTrig0 | Timer | I | - | Capture Trigger 0 |
| P22 | CaptureTrig1 | Timer | I | - | Capture Trigger 1 In cascade mode, this is not used. |
| P23 | Freeze | Timer | I | - | Freeze Count Value |
| P24 | GenerateOut0 | Timer | O | 0x0 | Generate Output 0 - Asserted whenever the timer0 wraps from all '0's to all '1's or vice-versa. In cascaded mode, this signal is asserted when the 64-bit value wraps from all '0's to all '1's or vice-versa. |
| P25 | GenerateOut1 | Timer | O | 0x0 | Generate Output 1 - Asserted whenever the timer1 wraps from all '0's to all '1's or vice-versa. In cascaded mode, this signal is asserted whenever the lower 32-bit counter (timer 0) wraps from all '0's to all '1's or vice-versa. |
| P26 | PWM0 | Timer | O | 0x0 | Pulse Width Modulation Output 0 |

Notes:

1. This signal is not used. The AXI Timer assumes that all byte lanes are active.

Design Parameters

To allow users to create the AXI Timer that is uniquely tailored for their systems, certain features can be parameterized in the AXI Timer design. This allows users to have a design that only utilizes the resources required by the system and operates at the best possible performance. The AXI Timer design parameters are shown in [Table 2](#).

In addition to the parameters listed in this table, there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see *AXI Interconnect IP Data Sheet (DS768)*.

Table 2: Design Parameters

| Generic | Feature/Description | Parameter Name | Allowable Values | Default Value | VHDL Type |
|------------------|-----------------------|--------------------|---|---------------|-----------|
| System Parameter | | | | | |
| G1 | Target FPGA family | C_FAMILY | zynq7000, artix7, kintex7, virtex7, virtex6, spartan6 | virtex6 | string |
| AXI Parameters | | | | | |
| G2 | AXI Protocol type | C_S_AXI_PROTOCOL | AXI4LITE | AXI4LITE | string |
| G3 | AXI address bus width | C_S_AXI_ADDR_WIDTH | 5 | 5 | integer |
| G4 | AXI data bus width | C_S_AXI_DATA_WIDTH | 32 | 32 | integer |

Table 2: Design Parameters (Cont.)

| Generic | Feature/Description | Parameter Name | Allowable Values | Default Value | VHDL Type |
|------------------|--|------------------|---|---------------|-----------|
| Timer Parameters | | | | | |
| G5 | The width in bits of the counters in the AXI Timer/Counter | C_COUNT_WIDTH | 8 - 32 | 32 | integer |
| G6 | Number of Timer modules | C_ONE_TIMER_ONLY | 0 = Two timers are present 1 = One timer is present (No PWM mode) | 0 | integer |
| G7 | Assertion level for CaptureTrig0 | C_TRIG0_ASSERT | '0' = CaptureTrig0 input is low-true '1' = CaptureTrig0 input is high-true | 1 | std_logic |
| G8 | Assertion level for CaptureTrig1 | C_TRIG1_ASSERT | '0' = CaptureTrig1 input is low-true '1' = CaptureTrig1 input is high-true | 1 | std_logic |
| G9 | Assertion level for GenerateOut0 | C_GEN0_ASSERT | '0' = GenerateOut0 output is low-true '1' = GenerateOut0 output is high-true | 1 | std_logic |
| G10 | Assertion level for GenerateOut1 | C_GEN1_ASSERT | '0' = GenerateOut1 output is low-true '1' = GenerateOut1 output is high-true | 1 | std_logic |

Parameter - I/O Signal Dependencies

The dependencies between the AXI timer core design parameters and I/O signals are described in [Table 3](#). In addition, when certain features are parameterized out of the design, the related logic is no longer a part of the design. The unused input signals and related output signals are set to a specified value.

Table 3: Parameter-I/O Signal Dependencies

| Generic or Port | Name | Affects | Depends | Relationship Description |
|-------------------|---------------------------------------|-------------|---------|---|
| Design Parameters | | | | |
| G3 | C_S_AXI_ADDR_WIDTH | P4, P14 | - | Defines the width of the ports |
| G4 | C_S_AXI_DATA_WIDTH | P7, P8, P17 | - | Defines the width of the ports |
| I/O Signals | | | | |
| P4 | S_AXI_AWADDR[C_S_AXI_ADDR_WIDTH-1:0] | - | G3 | Port width depends on the generic C_S_AXI_ADDR_WIDTH. |
| P7 | S_AXI_WDATA[C_S_AXI_DATA_WIDTH-1:0] | - | G4 | Port width depends on the generic C_S_AXI_DATA_WIDTH. |
| P8 | S_AXI_WSTB[C_S_AXI_DATA_WIDTH/8-1:0] | - | G4 | Port width depends on the generic C_S_AXI_DATA_WIDTH. |
| P14 | S_AXI_ARADDR[C_S_AXI_ADDR_WIDTH -1:0] | - | G3 | Port width depends on the generic C_S_AXI_ADDR_WIDTH. |
| P17 | S_AXI_RDATA[C_S_AXI_DATA_WIDTH -1:0] | - | G4 | Port width depends on the generic C_S_AXI_DATA_WIDTH. |

Register Data Types and Organization

Timer Counter registers are accessed as one of these types:

- Byte (8 bits)
- Half word (2 bytes)
- Word (4 bytes)

The AXI Timer/Counter registers are organized as little-endian data. The bit and byte labeling for the little-endian data types is shown in the [Figure 3](#).

The diagram illustrates the bit and byte labeling for three data types: Word, Halfword, and Byte, organized in a little-endian format.

| | n+3 | n+2 | n+1 | n |
|-------------------|---------|-----|-----|---------|
| Byte address | | | | |
| Byte label | 3 | 2 | 1 | 0 |
| Byte significance | MS Byte | | | LS Byte |
| Bit label | 31 | | | 0 |
| Bit significance | MSBit | | | LSBit |

| | n+1 | n |
|-------------------|---------|---------|
| Byte address | | |
| Byte label | 1 | 0 |
| Byte significance | MS Byte | LS Byte |
| Bit label | 15 | |
| Bit significance | MSBit | LSBit |

| | n | |
|-------------------|---------|-------|
| Byte address | | |
| Byte label | 0 | |
| Byte significance | MS Byte | |
| Bit label | 7 | |
| Bit significance | MSBit | LSBit |

Figure 3: Little Endian Data Types

Register Descriptions

[Table 4](#) shows all the AXI Timer registers and their addresses. Accesses to addresses inside the core address range other than these registers return an OKAY response, with reads returning 0 data values and writes having no effect.

Table 4: Register Overview

| Register Name | Address (hex) | Access Type | Default Value (hex) | Description |
|---------------|---------------|-------------|---------------------|---------------------------|
| TCSR0 | 0x00 | Read/Write | 0x0 | Control/Status Register 0 |
| TLR0 | 0x04 | Read/Write | 0x0 | Load Register 0 |
| TCR0 | 0x08 | Read | 0x0 | Timer/Counter Register 0 |
| TCSR1 | 0x10 | Read/Write | 0x0 | Control/Status Register 1 |

Table 4: Register Overview (Cont.)

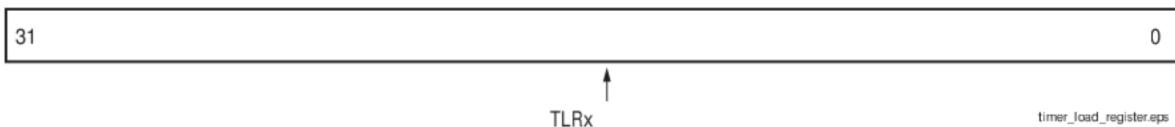
| Register Name | Address (hex) | Access Type | Default Value (hex) | Description |
|---------------|---------------|-------------|---------------------|--------------------------|
| TLR1 | 0x14 | Read/Write | 0x0 | Load Register 1 |
| TCR1 | 0x18 | Read | 0x0 | Timer/Counter Register 1 |

Load Register (TLR0 and TLR1)

When the counter width has been configured as less than 32 bits, the load register value is right-justified in TLR0 and TLR1. The least-significant counter bit is always mapped to load register bit 0.

In cascade mode, TLR0 will have the least significant 32-bits of the generate value and TLR1 should have the most significant bits of the generate value in generate mode. Similarly, in cascade mode TLR0 will have the captured value from TCR0 and TLR1 from TCR1.

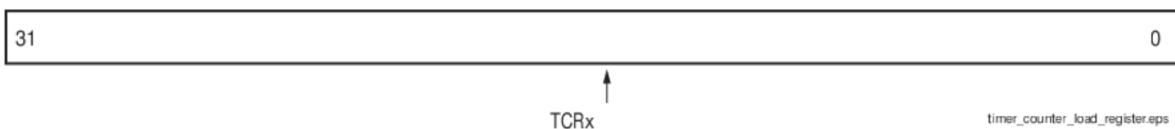
Figure 4 and Table 5 show the load register.

**Figure 4: Timer/Counter Load Register****Table 5: Load Register Bit Definitions**

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-----------------------------|-------------|-------------|-----------------------------|
| 31-0 | Timer/Counter Load Register | Read/Write | 0x0 | Timer/Counter Load register |

Timer/Counter Register (TCR0 and TCR1)

When the counter width has been configured as less than 32 bits, the count value is right-justified in TCR0 and TCR1. The least-significant counter bit is always mapped to Timer/Counter Register bit 0. Figure 5 and Table 6 show the Timer/counter register. In cascade mode, TCR0 will have the least significant 32-bits of the 64-bit counter and TCR1 will have the most significant bits.

**Figure 5: Timer/Counter Register****Table 6: Timer/Counter Register Bit Definitions**

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|------------------------|-------------|-------------|------------------------|
| 31-0 | Timer/Counter Register | Read | 0x0 | Timer/Counter register |

Control/Status Register 0 (TCSR0)

Figure 6 and Table 7 show the Control/Status register 0. Control/Status Register 0 contains the control and status bits for timer module 0.

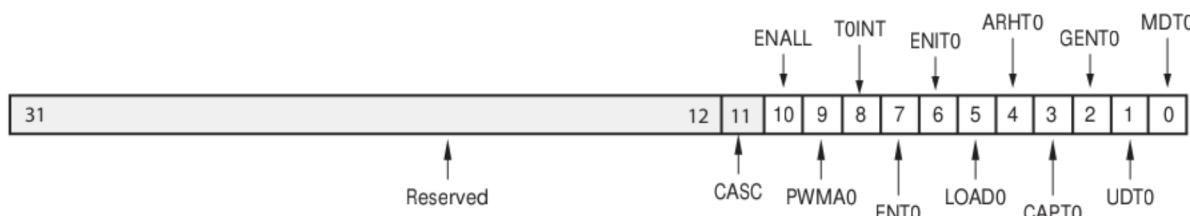


Figure 6: Control/Status Register 0

Table 7: Control/Status Register 0 Bit Definitions

| Bit(s) | Name | Core Access | Reset Value | Description |
|---------|----------|-------------|-------------|---|
| 31 - 12 | Reserved | N/A | - | Reserved |
| 11 | CASC | Read/Write | 0 | <p>Enable cascade mode of timers 0 = Disable cascaded operation 1 = Enable cascaded operation</p> <p>Cascaded operation requires using Timer0 and Timer1 together as a pair. The counting event for the Timer1 is when the Timer0 rolls over from all '1's to all '0's or vice-versa when counting down.</p> <p>TLR0 and TLR1 are used for lower 32-bit and higher 32-bit respectively. Similarly, TCR0 contains lower 32-bits for the 64-bit counter and TCR1 contains the higher 32-bits.</p> <p>Only TCSR0 is valid for both the timer/counters in this mode. This CASC bit must be set before enabling the timer/counter.</p> |
| 10 | ENALL | Read/Write | 0 | <p>Enable All Timers 0 = No effect on timers 1 = Enable all timers (counters run)</p> <p>This bit is mirrored in all control/status registers and is used to enable all counters simultaneously. Writing a '1' to this bit sets ENALL, ENT0, and ENT1. Writing a '0' to this register clears ENALL but has no effect on ENT0 and ENT1.</p> |
| 9 | PWMA0 | Read/Write | 0 | <p>Enable Pulse Width Modulation for Timer0 0 = Disable pulse width modulation 1 = Enable pulse width modulation</p> <p>PWM requires using Timer0 and Timer1 together as a pair. Timer0 sets the period of the PWM output, and Timer1 sets the high time for the PWM output. For PWM Mode, MDT0 and MDT1 must be '0' and C_GEN0_ASSERT and C_GEN1_ASSERT must be '1'.</p> |

Table 7: Control/Status Register 0 Bit Definitions (Cont.)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-------|-------------|-------------|---|
| 8 | T0INT | Read/Write | 0 | <p>Timer0 Interrupt Indicates that the condition for an interrupt on this timer has occurred. If the timer mode is capture and the timer is enabled, this bit indicates a capture has occurred. If the mode is generate, this bit indicates the counter has rolled over. Must be cleared by writing a '1'.</p> <p><i>Read:</i> 0 = No interrupt has occurred 1 = Interrupt has occurred</p> <p><i>Write:</i> 0 = No change in state of T0INT 1 = Clear T0INT (clear to '0')</p> |
| 7 | ENT0 | Read/Write | 0 | <p>Enable Timer0 0 = Disable timer (counter halts) 1 = Enable timer (counter runs)</p> |
| 6 | ENIT0 | Read/Write | 0 | <p>Enable Interrupt for Timer0 Enables the assertion of the interrupt signal for this timer. Has no effect on the interrupt flag in TCSR0. 0 = Disable interrupt signal 1 = Enable interrupt signal</p> |
| 5 | LOAD0 | Read/Write | 0 | <p>Load Timer0 0 = No load 1 = Loads timer with value in TLR0 Setting this bit loads timer/counter register(TCR0) with a specified value in the timer/counter load register(TLR0). This bit prevents the running of the timer/counter; hence, this should be cleared alongside setting Enable Timer/Counter (ENT0) bit in the Control Status Register (TCSR0).</p> |
| 4 | ARHT0 | Read/Write | 0 | <p>Auto Reload/Hold Timer0 When the timer is in Generate Mode, this bit determines whether the counter reloads the generate value and continues running or holds at the termination value. In Capture Mode, this bit determines whether a new capture trigger overwrites the previous captured value or if the previous value is held. 0 = Hold counter or capture value 1 = Reload generate value or overwrite capture value</p> |
| 3 | CAPTO | Read/Write | 0 | <p>Enable External Capture Trigger Timer0 0 = Disables external capture trigger 1 = Enables external capture trigger</p> |
| 2 | GENT0 | Read/Write | 0 | <p>Enable External Generate Signal Timer0 0 = Disables external generate signal 1 = Enables external generate signal</p> |
| 1 | UDT0 | Read/Write | 0 | <p>Up/Down Count Timer0 0 = Timer functions as up counter 1 = Timer functions as down counter</p> |
| 0 | MDT0 | Read/Write | 0 | <p>Timer0 Mode See the Timer Modes section. 0 = Timer mode is generate 1 = Timer mode is capture</p> |

Control/Status Register 1 (TCSR1)

[Figure 7](#) and [Table 8](#) show the Control/Status register 1. Control/Status Register 1 contains the control and status bits for timer module 1. This register is used only for loading the TLR1 register in cascade mode.



Figure 7: Control/Status Register 1

Table 8: Control/Status Register 1 Bit Definitions

| Bit(s) | Name | Core Access | Reset Value | Description |
|---------|----------|-------------|-------------|--|
| 31 - 11 | Reserved | N/A | - | Reserved |
| 10 | ENALL | Read/Write | 0 | <p>Enable All Timers 0 = No effect on timers 1 = Enable all timers (counters run)</p> <p>This bit is mirrored in all control/status registers and is used to enable all counters simultaneously. Writing a '1' to this bit sets ENALL, ENT0, and ENT1. Writing a '0' to this register clears ENALL but has no effect on ENT0 and ENT1.</p> |
| 9 | PWMA0 | Read/Write | 0 | <p>Enable Pulse Width Modulation for Timer1 0 = Disable pulse width modulation 1 = Enable pulse width modulation</p> <p>PWM requires using Timer0 and Timer1 together as a pair. Timer0 sets the period of the PWM output, and Timer1 sets the high time for the PWM output. For PWM Mode, MDT0 and MDT1 must be '0' and C_GEN0_ASSERT and C_GEN1_ASSERT must be '1'.</p> |
| 8 | T1INT | Read/Write | 0 | <p>Timer1 Interrupt Indicates that the condition for an interrupt on this timer has occurred. If the timer mode is capture and the timer is enabled, this bit indicates a capture has occurred. If the mode is generate, this bit indicates the counter has rolled over. Must be cleared by writing a '1'.</p> <p><i>Read:</i> 0 = No interrupt has occurred 1 = Interrupt has occurred</p> <p><i>Write:</i> 0 = No change in state of T1INT 1 = Clear T1INT (clear to '0')</p> |
| 7 | ENT1 | Read/Write | 0 | <p>Enable Timer1 0 = Disable timer (counter halts) 1 = Enable timer (counter runs)</p> |
| 6 | ENIT1 | Read/Write | 0 | <p>Enable Interrupt for Timer1 Enables the assertion of the interrupt signal for this timer. Has no effect on the interrupt flag in TCSR1. 0 = Disable interrupt signal 1 = Enable interrupt signal</p> |

Table 8: Control/Status Register 1 Bit Definitions (Cont.)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-------|-------------|-------------|--|
| 5 | LOAD1 | Read/Write | 0 | Load Timer1 0 = No load 1 = Loads timer with value in TLR1 Setting this bit loads the timer/counter register(TCR1) with a specified value in the timer/counter load register(TLR1). This bit prevents running of timer/counter; hence, this should be cleared alongside setting Enable Timer/Counter (ENT1) bit in Control Status Register (TCSR1). |
| 4 | ARHT1 | Read/Write | 0 | Auto Reload/Hold Timer1 When the timer is in generate mode, this bit determines whether the counter reloads the generate value and continues running or holds at the termination value. In capture mode, this bit determines whether a new capture trigger overwrites the previous captured value or if the previous value is held until it is read. 0 = Hold counter or capture value 1 = Reload generate value or overwrite capture value |
| 3 | CAPT1 | Read/Write | 0 | Enable External Capture Trigger Timer1 0 = Disables external capture trigger 1 = Enables external capture trigger |
| 2 | GENT1 | Read/Write | 0 | Enable External Generate Signal Timer1 0 = Disables external generate signal 1 = Enables external generate signal |
| 1 | UDT1 | Read/Write | 0 | Up/Down Count Timer1 0 = Timer functions as up counter 1 = Timer functions as down counter |
| 0 | MDT1 | Read/Write | 0 | Timer1 Mode See the Timer Modes section. 0 = Timer mode is generate 1 = Timer mode is capture |

Design Implementation

Target Technology

The intended target technologies are Zynq™-7000, Artix™-7, Kintex™-7, Virtex®-7, Virtex-6 and Spartan®-6 Field Programmable Gate Arrays (FPGAs).

Device Utilization and Performance Benchmarks

Core Performance

Because the AXI Timer core is used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the AXI Timer core is combined with other designs in the system, the utilization of FPGA resources and timing of the AXI Timer design varies from the results reported here.

The AXI Timer resource utilization for various parameter combinations measured with the Artix-7 FPGA as the target device are detailed in [Table 9](#).

Note: Resources numbers for Zynq-7000 devices are expected to be similar to 7 series device numbers.

Table 9: Performance and Resource Utilization on Artix-7 FPGA (XC7A355TDIE) and Zynq-7000 Devices

| Parameter Values | | Device Resources | | | Performance |
|------------------|------------------|------------------|------------|------|------------------------|
| C_COUNT_WIDTH | C_ONE_TIMER_ONLY | Slices | Flip-Flops | LUTs | f _{MAX} (MHz) |
| 8 | 1 | 49 | 53 | 96 | 184.196 |
| 16 | 1 | 61 | 69 | 120 | 178.891 |
| 32 | 1 | 84 | 101 | 181 | 177.999 |
| 8 | 0 | 50 | 74 | 123 | 190.803 |
| 16 | 0 | 74 | 106 | 161 | 169.549 |
| 32 | 0 | 97 | 170 | 256 | 166.806 |

The AXI Timer resource utilization for various parameter combinations measured with the Kintex-7 FPGA as the target device are detailed in [Table 10](#).

Note: Resources numbers for Zynq-7000 devices are expected to be similar to 7 series device numbers.

Table 10: Performance and Resource Utilization on Kintex-7 FPGA (xc7k410tffg676-3) and Zynq-7000 Devices

| Parameter Values | | Device Resources | | | Performance |
|------------------|------------------|------------------|------------|------|------------------------|
| C_COUNT_WIDTH | C_ONE_TIMER_ONLY | Slices | Flip-Flops | LUTs | f _{MAX} (MHz) |
| 8 | 1 | 37 | 53 | 99 | 301.659 |
| 16 | 1 | 64 | 69 | 120 | 202.143 |
| 32 | 1 | 69 | 101 | 180 | 241.371 |
| 8 | 0 | 62 | 74 | 123 | 251.762 |
| 16 | 0 | 76 | 106 | 160 | 202.02 |
| 32 | 0 | 116 | 170 | 256 | 202.306 |

The AXI Timer resource utilization for various parameter combinations measured with the Virtex-7 FPGA as the target device are detailed in [Table 11](#).

Table 11: Performance and Resource Utilization Benchmarks on a Virtex-7 FPGA (xc7v855tffg1157-3)

| Parameter Values | | Device Resources | | | Performance |
|------------------|------------------|------------------|------------|------|------------------------|
| C_COUNT_WIDTH | C_ONE_TIMER_ONLY | Slices | Flip-Flops | LUTs | f _{MAX} (MHz) |
| 8 | 1 | 51 | 53 | 96 | 203.542 |
| 16 | 1 | 62 | 69 | 119 | 221.435 |
| 32 | 1 | 92 | 101 | 180 | 186.951 |
| 8 | 0 | 46 | 74 | 124 | 251.509 |
| 16 | 0 | 83 | 106 | 160 | 202.02 |
| 32 | 0 | 118 | 170 | 256 | 185.494 |

The AXI Timer resource utilization for various parameter combinations measured with the Virtex-7 FPGA as the target device are detailed in [Table 12](#).

Table 12: Performance and Resource Utilization Benchmarks on a Virtex-6 FPGA (xc6vlx130t-1-ff1156)

| Parameter Values | | Device Resources | | | Performance |
|------------------|------------------|------------------|------------|------|------------------------|
| C_COUNT_WIDTH | C_ONE_TIMER_ONLY | Slices | Flip-Flops | LUTs | f _{MAX} (MHz) |
| 8 | 1 | 52 | 53 | 95 | 213.493 |
| 16 | 1 | 70 | 69 | 119 | 209.074 |
| 32 | 1 | 94 | 101 | 180 | 203.459 |
| 8 | 0 | 53 | 74 | 121 | 210.438 |
| 16 | 0 | 76 | 106 | 161 | 205.634 |
| 32 | 0 | 114 | 170 | 256 | 205.339 |

The AXI Timer resource utilization for various parameter combinations measured with a Spartan-6 FPGA as the target device are detailed in [Table 13](#).

Table 13: Performance and Resource Utilization Benchmarks on a Spartan-6 FPGA (xc6slx45t-2-fgg484)

| Parameter Values | | Device Resources | | | Performance |
|------------------|------------------|------------------|------------|------|------------------------|
| C_COUNT_WIDTH | C_ONE_TIMER_ONLY | Slices | Flip-Flops | LUTs | f _{MAX} (MHz) |
| 8 | 1 | 50 | 52 | 95 | 166.889 |
| 16 | 1 | 48 | 68 | 119 | 171.233 |
| 32 | 1 | 95 | 100 | 179 | 161.005 |
| 8 | 0 | 59 | 73 | 122 | 168.748 |
| 16 | 0 | 64 | 105 | 161 | 161.134 |
| 32 | 0 | 108 | 169 | 257 | 157.953 |

System Performance

To measure the system performance (F_{MAX}) of this core, this core was added to a Virtex-6 FPGA system and a Spartan-6 FPGA system as the Device Under Test (DUT).

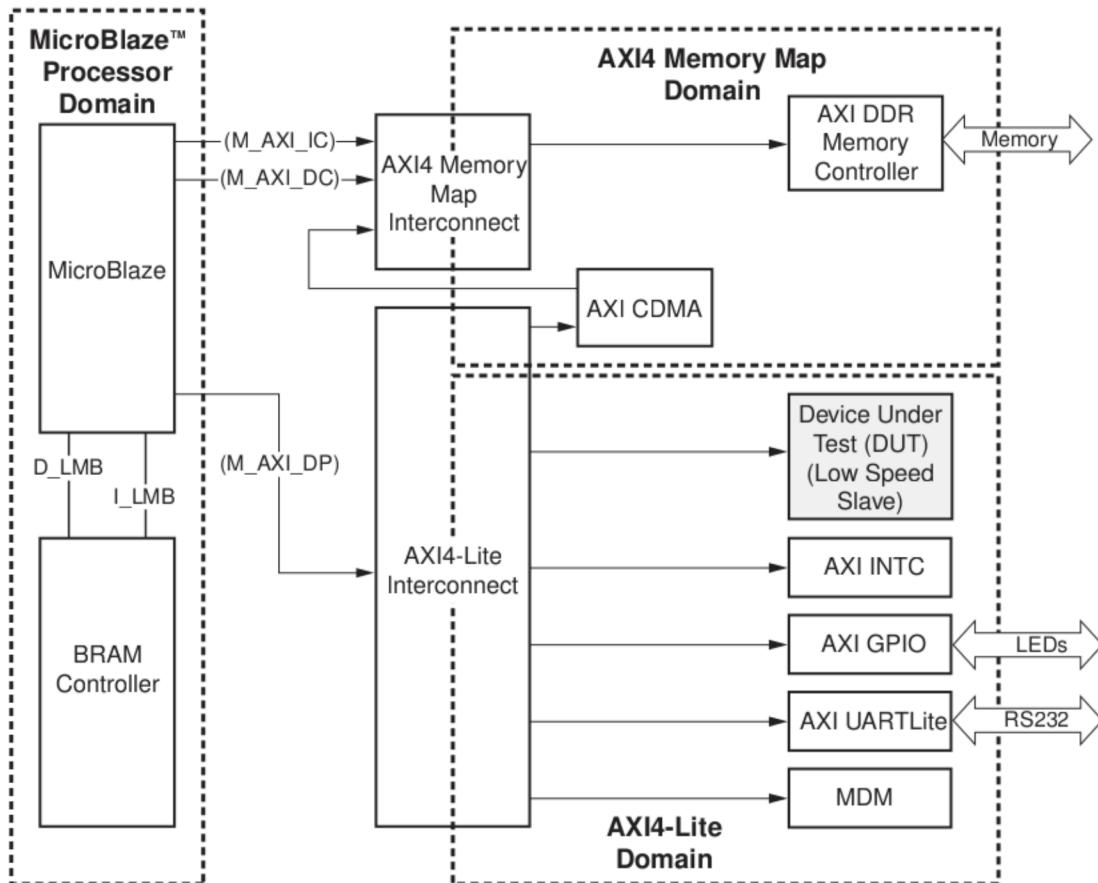


Figure 8: Virtex-6 and Spartan-6 Devices F_{MAX} Margin System

Because the AXI Timer core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the utilization of FPGA resources and timing of the design varies from the results reported here.

The target FPGA was then filled with logic to drive the Lookup Table (LUT) and block RAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in [Table 14](#).

Table 14: AXI Timer System Performance

| Target FPGA | Target F_{MAX} (MHz) |
|-------------|------------------------|
| Artix-7 | 110 |
| Kintex-7 | 180 |
| Virtex-7 | 180 |
| Spartan-6 | 110 |
| Virtex-6 | 180 |

The target F_{MAX} is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

Support

Xilinx provides technical support for this LogiCORE™ Intellectual Property (IP) product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx Vivado Design Suite and ISE Design Suite Embedded Edition software under the terms of the [Xilinx End User License](#).

Information about this and other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your local [Xilinx sales representative](#).

Reference Documents

- AXI4 AMBA AXI Protocol Version: 2.0 Specification
- LogiCORE IP AXI4-Lite IPIF Data Sheet (DS765)
- AXI Interconnect IP Data Sheet (DS768)

To search for Xilinx documentation, go to <http://www.xilinx.com/support>.

Revision History

The table shows the revision history for this document:

| Date | Version | Description of Revisions |
|----------|---------|--|
| 09/21/10 | 1.0 | First release of the core with AXI interface support. |
| 09/21/10 | 1.0.1 | Documentation only. Added inferred parameters text on page 8. |
| 12/14/10 | 1.1 | Updated to v1.01.a version; updated tools to 12.4. |
| 06/22/11 | 1.2 | Updated to v1.02.a version; updated tools to 13.2. |
| 10/19/11 | 1.3 | <p>Summary of major core changes:</p> <ul style="list-style-type: none"> Updated to v1.03.a version; updated tools to 13.3. 64-bit support has been added. <p>Summary of major documentation changes:</p> <ul style="list-style-type: none"> Updated List of Acronyms Reorganized device information so that information displays with latest devices first - Zynq-7000, Artix-7, Kintex-7, Virtex-7, Virtex-6, Spartan-6 Updated to v1.03.a core version; updated tools to 13.3 Added Zynq-7000 |
| 01/18/12 | 1.3.1 | <p>Documentation Changes Only:</p> <ul style="list-style-type: none"> Added information about supported software drivers to IP Facts table. Removed List of Acronyms. For the first occurrence of each acronym, spelled out occurrence followed by acronym. Example; Field Programmable Gate Array (FPGA) Added text to Load Timer0 description in Table 7. Added text to Load Timer1 description in Table 8. |
| 05/25/12 | 1.4 | Added support for Vivado Design Suite. Removed BASEADDR and HIGHADDR parameters. |

Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.