



Timer/Counter IP

IP Version: v1.4.0

User Guide

FPGA-IPUG-02139-1.4

December 2024

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language [FAQ 6878](#) for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Contents	3
Acronyms in This Document	5
1. Introduction	6
1.1. Features	6
1.2. Conventions	6
1.2.1. Nomenclature	6
1.2.2. Signal Names	6
1.2.3. Host	6
1.2.4. Attribute Names	6
2. Functional Description	7
2.1. Overview	7
2.2. Block Diagram	7
2.3. Signal Description	8
2.4. Attribute Summary	9
2.5. Register Description	9
2.5.1. INT_STATUS	10
2.5.2. INT_ENABLE	11
2.5.3. INT_SET	11
2.5.4. GBL_CTRL	12
2.5.5. STATUS	12
2.5.6. CONTROL	12
2.5.7. PERIOD	13
2.5.8. SNAPSHOT	14
2.6. Modes of Operation	14
2.6.1. Software-controlled Retrigger enabled	14
2.6.2. Software-controlled Retrigger disabled	14
2.6.3. Timeout and Interrupt	14
2.6.4. Prescaler	15
2.7. Sample Configuration	15
3. IP Generation, Synthesis, and Validation	16
3.1. Licensing the IP	16
3.2. Generating and Synthesizing the IP	16
3.3. Running the Functional Simulation	18
3.4. Constraining the IP	20
Appendix A. Resource Utilization	21
References	22
Technical Support Assistance	23
Revision History	24

Figures

Figure 2.1. Timer/Counter IP Core Block Diagram	7
Figure 3.1. Module/IP Block Wizard	16
Figure 3.2. Configure User Interface of the Timer-Counter Module	17
Figure 3.3. Check Generated Result.....	17
Figure 3.4. Simulation Wizard	18
Figure 3.5. Adding and Reorder Source	19
Figure 3.6. Summary Window	19
Figure 3.7. Simulation Waveform	20

Tables

Table 2.1. Timer/Counter IP Core Signal Description	8
Table 2.2. Attributes Table	9
Table 2.3. Attribute Descriptions.....	9
Table 2.4. Summary of Timer/Counter IP Core Registers	10
Table 2.5. Access Type Definition	10
Table 2.6. Interrupt Status Register	10
Table 2.7. Interrupt Enable Register	11
Table 2.8. Interrupt Set Register.....	11
Table 2.9. Global Register	12
Table 2.10. Status Register	12
Table 2.11. Control Register	13
Table 2.12. Period Register	14
Table 2.13. Snapshot Register	14
Table 2.14. Prescaler Ratio Table.....	15
Table 3.1. Generated File List	18
Table A.1. Resource Utilization	21

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AHB	Advanced High-Performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
CPU	Central Processing Unit
FPGA	Field Programmable Gate Arrays
LINTR	Lattice Interrupt Interface

1. Introduction

The Lattice Semiconductor Timer/Counter IP is used to track timeouts in the system. It generates an interrupt for the CPU when a timeout is detected.

The Timer/Counter IP design is implemented in Verilog. It can be configured and generated using Lattice Propel™ Builder. It is targeted for all devices and implemented using the Lattice Radiant™ and Lattice Diamond® software Place and Route tools integrated with the Synopsys® Synplify Pro® synthesis tool.

1.1. Features

The Timer/Counter IP includes the following features:

- Generates up to eight timers/counters that operate individually
- Operates in either one-shot or continuous mode
- Counts up or down
- Register configuration through AMBA 3 APB Protocol v1.0 or AMBA 3 AHB-Lite Protocol v1.0
- Interrupt handling conforming to Lattice Interrupt Interface (LINTR) Standard
- User-configurable preload and prescaler value access modes
- User-configurable start and stop controls for software-controlled start and stop

1.2. Conventions

1.2.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.2.2. Signal Names

Signal names that end with:

- `_n` are active low (asserted when value is logic 0)
- `_i` are input signals
- `_o` are output signals
- `_io` are bi-directional input/output signals

1.2.3. Host

The logic unit inside the FPGA interacts with the Timer/Counter IP through APB or AHB-Lite.

1.2.4. Attribute Names

Attribute names in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Description

2.1. Overview

The Timer/Counter IP core can generate up to eight timers configured to operate individually.

It consists of a prescaler block that counts the clock source and provides outputs divided by 2, 4, 8, and so on. This is used to slow down the counting rate of the timer.

The timer block is configurable through APB or AHB-Lite register access. Refer to the [Register Description](#) and [Modes of Operation](#) sections for configuration details.

The Timer/Counter IP core also generates an interrupt whenever timers reach a timeout.

2.2. Block Diagram

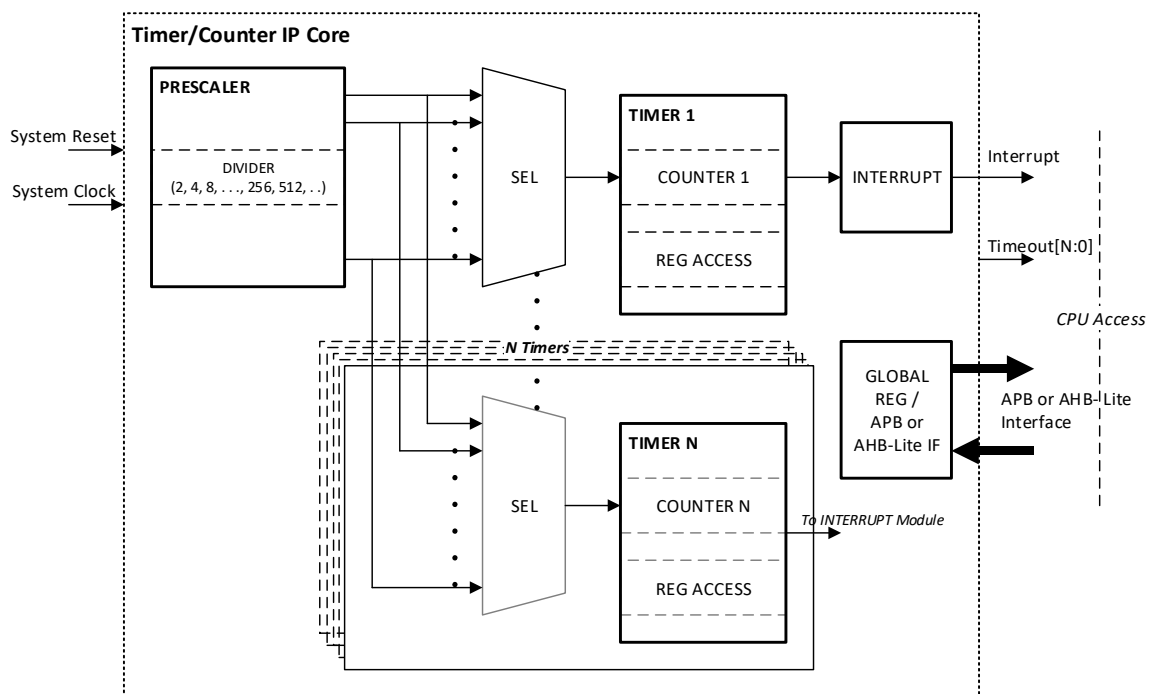


Figure 2.1. Timer/Counter IP Core Block Diagram

2.3. Signal Description

Table 2.1. Timer/Counter IP Core Signal Description

Port	Width	Direction	Description
System Clock and Reset			
clk_i	1	Input	Manager clock input.
rst_n_i	1	Input	Asynchronous reset active low. Note that reset handling should be done on the system level outside of the IP.
APB Subordinate Interface			
apb_psel_i	1	In	Select signal. Indicates that the subordinate device is selected and a data transfer is required.
apb_paddr_i	32	In	Address signal.
apb_pwdata_i	32	In	Write data signal.
apb_pwrite_i	1	In	Direction signal. Write = 1, Read = 0.
apb_penable_i	1	In	Enable signal. Indicates the second and subsequent cycles of an APB transfer.
apb_pready_o	1	Out	Ready signal. Indicates transfer completion. Subordinate uses this signal to extend an APB transfer.
apb_prdata_o	32	Out	Read data signal.
Interrupt			
int_o	1	Output	Interrupt request.
Others			
timeout_o	N-1	Output	Timeout output signal per N timer. Output a pulse whenever counter reaches timeout.
AHB-Lite Subordinate Interface			
ahbl_hsel_i	1	In	Select signal.
ahbl_hready_i	1	In	Ready Input signal.
ahbl_haddr_i	328	In	Address signal. Size: Interface Address Width.
ahbl_hburst_i	3	In	Burst Type signal. This is not supported.
ahbl_hsize_i	3	In	Transfer Size signal. This only supports Word transfer size.
ahbl_hmastlock_i	1	In	Lock signal. This is not supported.
ahbl_hprot_i	4	In	Protection Control signal. This signal is not supported.
ahbl_htrans_i	2	In	Transfer Type signal.
ahbl_hwrite_i	1	In	Direction signal. Write = High, Read = Low.
ahbl_hwdata_i	328	In	Write Data signal. Size: Interface Data Width.
ahbl_hreadyout_o	1	Out	Ready Output signal.
ahbl_hrdata_o	328	Out	Read Data signal. Size: Interface Data Width.
ahbl_hresp_o	1	Out	Transfer Response signal. This is not supported. It always returns 0.

2.4. Attribute Summary

The Timer/Counter IP Core configurable attributes are shown in [Table 2.2](#) and are described in [Table 2.3](#).

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
General			
No. of Timers	1 – 8	4	—
Prescaler Size	1 – 32	8	—
Host Interface	APB, AHB-Lite	APB	—
Timer N Settings (where N == No. of Timers)			
Direction	count-up, count-down	count-down	<i>No. of Timers == N</i>
Counter Size	1 – 32	2	<i>No. of Timers == N</i>
Timer Preloaded Value	0 – 2 ^{(Counter Size)-1}	4	<i>No. of Timers == N</i>
Disable Prescaler	enable, disable	enable	<i>No. of Timers == N</i>
Prescaler Ratio	1:2 – 1:2 ^(Prescaler Size)	1: 2147483648	<i>No. of Timers == N</i>
Software-controlled Retrigger	enable, disable	enable	<i>No. of Timers == N</i>
Timer Register Write Accessibility	enable, disable	enable	<i>No. of Timers == N, Software-controlled Retrigger == enable</i>

Table 2.3. Attribute Descriptions

Attribute	Description
General	
No. of Timers	Sets the default value of <code>tmr_en</code> field of <code>GBL_CTRL</code> register.
Prescaler Size	Sets the Prescaler counter width. The width determines the maximum divider or prescaler ratio.
Host Interface	Sets the Register interface of the IP.
Timer N Settings (where N == No. of Timers)	
Direction	Sets the default value for <code>dir</code> field of <code>CONTROL</code> register.
Counter Size	Sets the internal counter width of the timer. The width determines the maximum time range.
Timer Preloaded Value	This is the initial value of the timer internal counter. This also sets the default value for <code>load_val</code> of <code>PERIOD</code> register. The input value should be in the range that can be represented by the Counter Size.
Disable Prescaler	Sets the default value <code>pscaler_dis</code> field of <code>CONTROL</code> register.
Prescaler Ratio	Sets the default value of <code>pscaler_ratio</code> field of <code>CONTROL</code> register. See the Prescaler section for details.
Software-controlled Retrigger	When enabled, the start and stop bits of <code>CONTROL</code> register are controllable through APB access.
Timer Register Write Accessibility	When enabled, the <code>PERIOD</code> Register is writable through APB access.

2.5. Register Description

Global registers are mapped to offsets 0x000-0x004, and per-timer registers are mapped to 0xN0-0xNC, where N corresponds to the timer number, in the range 1 to *No. of Timers*.

Table 2.4. Summary of Timer/Counter IP Core Registers

Offset	Register Name	Access	Description
0x00	INT_STATUS*	RW1C	Interrupt Status Register
0x04	INT_ENABLE*	RW	Interrupt Enable Register
0x08	INT_SET*	WO	Interrupt Set Register
0x0C	GLB_CTRL	RW	Global Control Register
0xN0	STATUS	RO	Timer N Status Register
0xN4	CONTROL	RW/RZ	Timer N Control Register
0xN8	PERIOD	RW	Timer N Period Register
0xNC	SNAPSHOT	RW	Timer N Snapshot Register

The behavior of registers to write and read access is defined by their access type, as shown in [Table 2.5](#).

Table 2.5. Access Type Definition

Access Type	Behavior on Read Access	Behavior on Write Access
RO	Returns register value	Ignores write access.
WO	Returns 0	Updates register value.
RW	Returns register value	Updates register value.
RW1C	Returns register value	Writing 1'b1 on register bit clears the bit to 1'b0. Writing 1'b0 on register bit is ignored.
RZ	Returns 0	Writing 1'b1 on register returns to 0 on the next clock cycle.
RSVD	Returns 0	Ignores write access.

2.5.1. INT_STATUS

This register represents the source of interrupts.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_STATUS	RSVD															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSVD								to8_int	to7_int	to6_int	to5_int	to4_int	to3_int	to2_int	to1_int

Count-down mode – When the internal counter of timer 1 reaches zero, the interrupt bit is set to 1.

Count-up mode – When internal counter of timer 1 overflows, the interrupt bit is set to 1.

Table 2.6. Interrupt Status Register

Field	Name	Description
[0]	to1_int	Timer 1 Timeout Interrupt
[1]	to2_int*	Timer 2 Timeout Interrupt
[2]	to3_int*	Timer 3 Timeout Interrupt
[3]	to4_int*	Timer 4 Timeout Interrupt
[4]	to5_int*	Timer 5 Timeout Interrupt
[5]	to6_int*	Timer 6 Timeout Interrupt
[6]	to7_int*	Timer 7 Timeout Interrupt
[7]	to8_int*	Timer 8 Timeout Interrupt
[31:8]	RSVD	Reserved bits

***Note:** This bit is valid when *No. of Timers* > 1.

2.5.2. INT_ENABLE

This register controls whether the interrupts in the INT_STATUS register assert the int_o signal or not.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_ENABLE	RSVD															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSVD								to8_en	to7_en	to6_en	to5_en	to4_en	to3_en	to2_en	to1_en

Table 2.7. Interrupt Enable Register

Field	Name	Description
[0]	to1_en	Timer 1 Timeout Interrupt enable
[1]	to2_en*	Timer 2 Timeout Interrupt enable
[2]	to3_en*	Timer 3 Timeout Interrupt enable
[3]	to4_en*	Timer 4 Timeout Interrupt enable
[4]	to5_en*	Timer 5 Timeout Interrupt enable
[5]	to6_en*	Timer 6 Timeout Interrupt enable
[6]	to7_en*	Timer 7 Timeout Interrupt enable
[7]	to8_en*	Timer 8 Timeout Interrupt enable
[31:8]	RSVD	Reserved bits

*Note: This bit is valid when *No. of Timers* > 1. Writing to this register does not affect anything.

2.5.3. INT_SET

This register sets the interrupts in the INT_STATUS register.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_SET	RSVD															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSVD								to8_set	to7_set	to6_set	to5_set	to4_set	to3_set	to2_set	to1_set

Table 2.8. Interrupt Set Register

Field	Name	Description	Access	Default
[0]	to1_set	Timer 1 Timeout Interrupt set	RW	0
[1]	to2_set*	Timer 2 Timeout Interrupt set	RW	0
[2]	to3_set*	Timer 3 Timeout Interrupt set	RW	0
[3]	to4_set*	Timer 4 Timeout Interrupt set	RW	0
[4]	to5_set*	Timer 5 Timeout Interrupt set	RW	0
[5]	to6_set*	Timer 6 Timeout Interrupt set	RW	0
[6]	to7_set*	Timer 7 Timeout Interrupt set	RW	0
[7]	to8_set*	Timer 8 Timeout Interrupt set	RW	0
[31:8]	RSVD	Reserved bits	RO	0

*Note: This bit is valid when *No. of Timers* > 1. Writing to this register does not affect anything.

2.5.4. GBL_CTRL

Global register settings of the Timer/Counter IP Core.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GBL_CTRL	RSVD															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSVD								tmr_en							

Table 2.9. Global Register

Field	Name	Description	Access	Default
[7:0]	tmr_en	Enables or disables timer. Each bit represents each timer. Bits can be set simultaneously. [0] – Enables timer 1 when set to 1. [1] – Enables timer 2 when set to 1. [2] – Enables timer 3 when set to 1. [3] – Enables timer 4 when set to 1. [4] – Enables timer 5 when set to 1. [5] – Enables timer 6 when set to 1. [6] – Enables timer 7 when set to 1. [7] – Enables timer 8 when set to 1.	RW	No. of Timers
[31:8]	RSVD	Reserved bits	RO	0

2.5.5. STATUS

This is the status register for timer *N*.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STATUS	RSVD															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSVD															tmr_run

Table 2.10. Status Register

Field	Name	Description	Access	Default
[0]	tmr_run	When internal counter is running, this bit is read as 1. When internal counter is not running, it is read as 0.	RO	0
[31:1]	RSVD	Reserved bits	RO	0

2.5.6. CONTROL

This register controls the mode of operation, direction, counter sizes, and retriggerable ability of each timer.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CONTROL	RSVD															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	pscaler_ratio								RSVD				pscaler_dis	stop	start	dir

Table 2.11. Control Register

Field	Name	Description	Access	Default
[0]	cont	Counter mode of operation. This bit determines how the internal counter behaves when it reaches timeout. Set to 1 – Continuous: the counter keeps running until it is stopped by the stop bit. Set to 0 – One-shot: the counter stops when it reaches timeout. When the counter reaches timeout, it reloads with the value stored in PERIOD register, regardless of the setting of this bit. When attribute Software-controlled Retrigger == disable, the timer keeps running and is not affected by the value of this bit.	RW	1
[1]	dir	Direction of counter. Set to 0 – count-down. The internal counter decrements from preloaded value. Set to 1 – count-up. The internal counter increments from preloaded value to maximum counter.	RW	0
[2]	start ¹	Start the counter. Set to 1 – causes the internal counter to begin counting down or up. If the timer is stopped before reaching to zero or maximum, writing 1 to this bit causes the timer to continue counting from the number currently held in its counter. If the timer is already running, writing any value to this bit has no effect.	RZ	0
[3]	stop ¹	Stop the counter. Set to 1 – causes the internal counter to stop counting. This bit has no effect when: the timer has already stopped; a '0' is written to this bit; the attribute <i>Software-controlled Retrigger == disable</i> .	RZ	0
[4]	pscaler_dis	Disables prescaler selection. It uses 1:1 ratio of the system clock frequency.	RW	Disable Prescaler
[7:5]	RSVD	Reserved	RO	0
[15:8]	pscaler_ratio	Prescaler ratio or the divide ratio. See the Prescaler section for details. When pscaler_dis = 0, timer is not affected by this register.	RW	Prescaler Ratio
[31:16]	RSVD	Reserved	RO	0

Note:

- When a 1 is written to both start and stop bits simultaneously, the priority is the stop bit.

2.5.7. PERIOD

Preloaded value of the internal counter.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PERIOD	load_val															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	load_val															

Table 2.12. Period Register

Field	Name	Description	Access	Default
[31:0]	load_val	<p>The internal counter is loaded with the value stored in this register.</p> <p>When attribute <i>Software-controlled Retrigger == enable</i>, writing on this register updates the internal counter, and the count-down or count-up continues.</p> <p>When attribute <i>Software-controlled Retrigger == disable</i>, writing on this register does not affect the internal counter.</p> <p>When attribute <i>Timer Register Write Accessibility == disable</i>, writing on this register causes the counter to reset to the fixed value specified in the attribute <i>Timer Preloaded Value</i>.</p>	RW	<i>Timer Preloaded Value</i>

2.5.8. SNAPSHOT

Snapshot value of the internal counter.

Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SNAPSHOT	snap_val															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	snap_val															

Table 2.13. Snapshot Register

Field	Name	Description	Access	Default
[31:0]	snap_val	Specifies the snapshot value of the internal counter.	RO	0

2.6. Modes of Operation

2.6.1. Software-controlled Retrigger enabled

When attribute *Software-controlled Retrigger == enable*, timer is configurable/controllable through register access.

2.6.2. Software-controlled Retrigger disabled

When attribute *Software-controlled Retrigger == disable*, the behavior of the timer is always in continuous mode and is not affected by the settings in the CONTROL register. When it reaches the timeout event, it reloads to the value set in the attribute *Timer Preloaded Value*.

Issuing a write access to the PERIOD register causes the counter to restart to the value set in the *Timer Preloaded Value* attribute, regardless of the *Timer Register Write Accessibility* setting.

2.6.3. Timeout and Interrupt

When attribute *Direction == count-down*, a timeout event occurs when counter reaches 0x0.

When attribute *Direction == count-up*, a timeout event occurs when counter reaches the maximum count.

Whenever a timeout event occurs, the interrupt status registers (INT_STAT) mapped to the timeout signal asserts.

2.6.4. Prescaler

Table 2.14 shows the prescaler ratio with its corresponding pscaler_ratio register setting. The prescaler counter size is configurable up to 32 bits for wide time range selection.

Table 2.14. Prescaler Ratio Table

Prescaler Counter Bit	Prescaler Ratio	pscaler_ratio [7:0]								
[0]	1:2	0	0	0	0	0	0	0	0	0
[1]	1:4	0	0	0	0	0	0	0	0	1
[2]	1:8	0	0	0	0	0	0	0	1	0
[3]	1:16	0	0	0	0	0	0	0	1	1
[4]	1:32	0	0	0	0	0	1	0	0	0
[5]	1:64	0	0	0	0	0	1	0	1	0
[6]	1:128	0	0	0	0	0	1	1	0	0
[7]	1:256	0	0	0	0	0	1	1	1	0
[8]	1:512	0	0	0	0	1	0	0	0	0
[9]	1:1024	0	0	0	0	1	0	0	1	0
[10]	1:2048	0	0	0	0	1	0	1	0	0
[11]	1:4096	0	0	0	0	1	0	1	1	0
[12]	1:8192	0	0	0	0	1	1	0	0	0
[13]	1:16384	0	0	0	0	1	1	0	1	0
[14]	1:32768	0	0	0	0	1	1	1	0	0
[15]	1:65536	0	0	0	0	1	1	1	1	0
[16]	1:131072	0	0	0	1	0	0	0	0	0
[17]	1:262144	0	0	0	1	0	0	0	1	0
[18]	1:524288	0	0	0	1	0	0	1	0	0
[19]	1:1048576	0	0	0	1	0	0	1	1	0
[20]	1:2097152	0	0	0	1	0	1	0	0	0
[21]	1:4194304	0	0	0	1	0	1	0	1	0
[22]	1:8388608	0	0	0	1	0	1	1	0	0
[23]	1:16777216	0	0	0	1	0	1	1	1	0
[24]	1:33554432	0	0	0	1	1	0	0	0	0
[25]	1:67108864	0	0	0	1	1	0	0	1	0
[26]	1:134217728	0	0	0	1	1	0	1	0	0
[27]	1:268435456	0	0	0	1	1	0	1	1	0
[28]	1:536870912	0	0	0	1	1	1	0	0	0
[29]	1:1073741824	0	0	0	1	1	1	0	1	0
[30]	1:2147483648	0	0	0	1	1	1	1	0	0
[31]	1:4294967296	0	0	0	1	1	1	1	1	0

2.7. Sample Configuration

Sample use case:

If you choose a *Prescaler Ratio* of 1:512 with a system clock frequency of 50 MHz (20 ns),

1 count (a tick) of the internal counter of the timer is:

$$20 \text{ ns} \times 512 = \mathbf{10.24 \text{ us}}$$

If the *Counter Size* is set to 16 bits, maximum period is:

$$16 - \text{bit counter} \times 10.24 \text{ us} = \mathbf{671 \text{ ms}}$$

3. IP Generation, Synthesis, and Validation

This section provides information on how to generate and synthesize this module using the Lattice Radiant software. For more information on the Lattice Radiant software, refer to the Lattice Radiant software user guide and relevant tutorials.

3.1. Licensing the IP

No license is required for this module.

3.2. Generating and Synthesizing the IP

The Lattice Radiant software allows you to customize and generate modules and IPs and integrate them into the device architecture. The procedure for generating the Timer-Counter module in the Lattice Radiant software is described below.

To generate the Timer-Counter module:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the IP Catalog tab, double-click on **Timer-Counter** under the **IP, Processors_Controllers_and_Peripherals** category.
3. The **Module/IP Block Wizard** opens as shown in [Figure 3.1](#). Enter values in the **Component name** and the **Create in** fields and click **Next**.

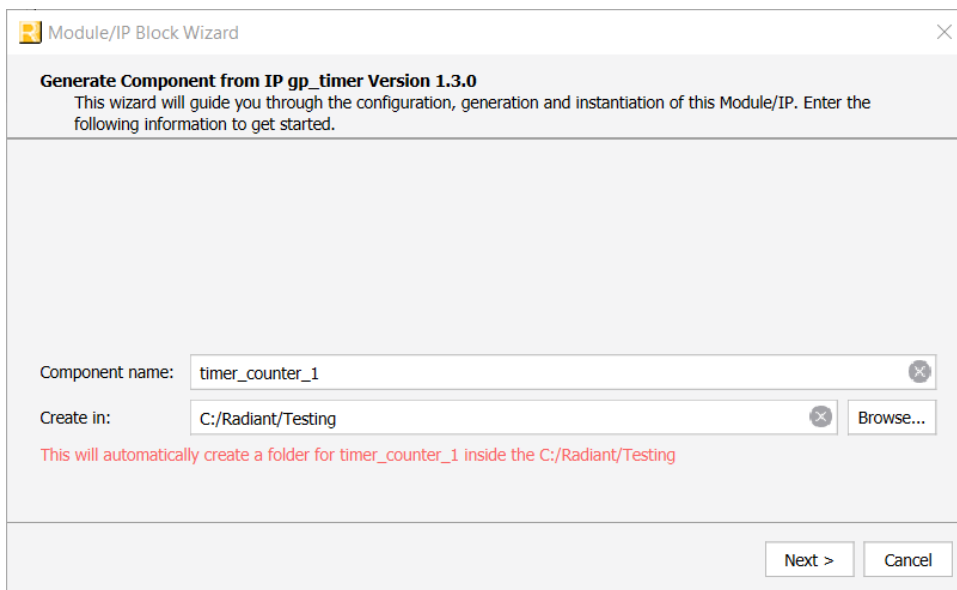


Figure 3.1. Module/IP Block Wizard

- In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected Timer-Counter module using drop-down menus and check boxes. As a sample configuration, see [Figure 3.2](#). For configuration options, see the [Attribute Summary](#) section.

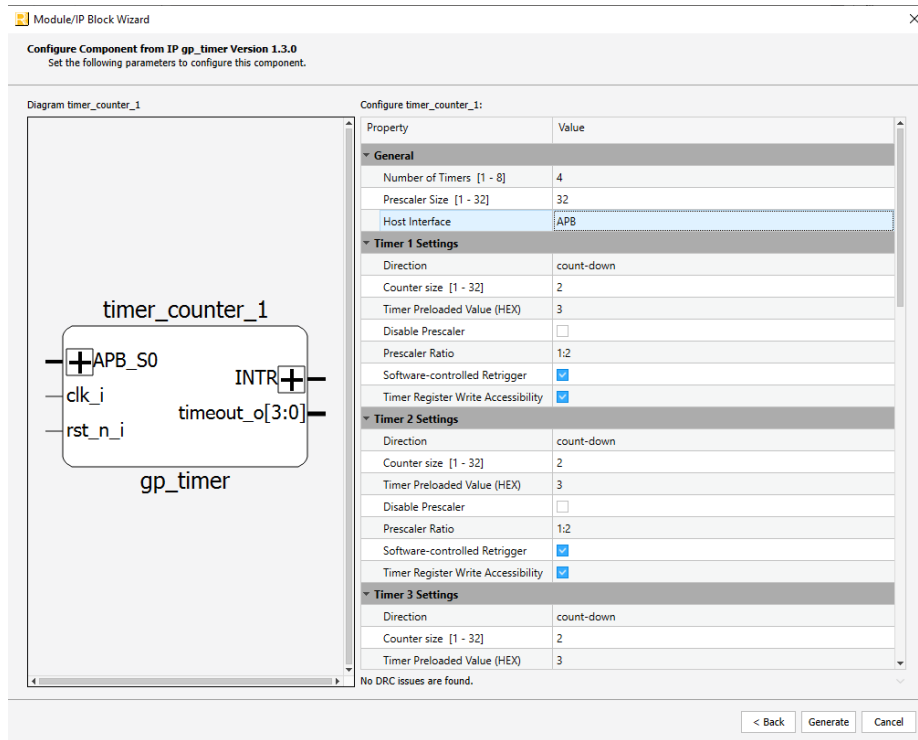


Figure 3.2. Configure User Interface of the Timer-Counter Module

- Click **Generate**. The **Check Generated Result** dialog box opens, showing design block messages and results as shown in [Figure 3.3](#).

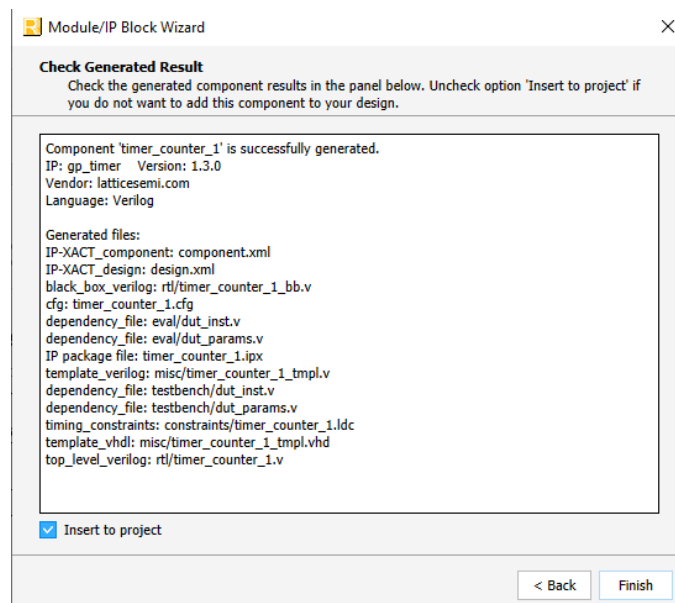


Figure 3.3. Check Generated Result

- Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in [Figure 3.1](#).

The generated Timer-Counter Module package includes the closed-box (<Instance Name>_bb.v) and instance templates (<Instance Name>_tpl.v/vhd) that can be used to instantiate the module in a top-level design. An example RTL top-level reference source file (<Instance Name>.v) that can be used as an instantiation template for the module is also provided. You may also use this top-level reference as the starting template for the top-level for your complete design. The generated files are listed in [Table 3.1](#).


Table 3.1. Generated File List

Attribute	Description
<Instance Name>.ipx	This file contains the information on the files associated to the generated IP.
<Instance Name>.cfg	This file contains the attribute values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration attributes of the IP in IP-XACT 2014 format.
rtl/<Instance Name>.v	This file provides an example RTL top file that instantiates the module.
rtl/<Instance Name>_bb.v	This file provides the synthesis closed-box.
misc/<Instance Name>_tpl.v misc /<Instance Name>_tpl.vhd	These files provide instance templates for the module.
testbench/dut_inst.v	This file contains the instance of the generated IP.
testbench/dut_params.v	This file contains the parameter settings of the generated IP.
eval/constraint.pdc	This file contains the PDC constraints for this IP. Refer to section 3.4 on how to use this file.

3.3. Running the Functional Simulation

Running functional simulation can be performed after the IP is generated.

To run functional simulation:

- Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in [Figure 3.4](#).

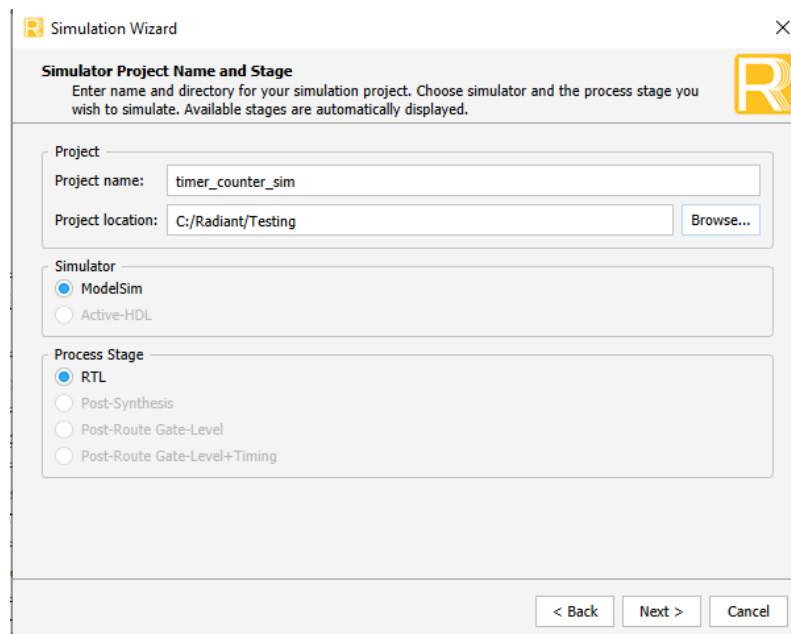


Figure 3.4. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in [Figure 3.5](#).

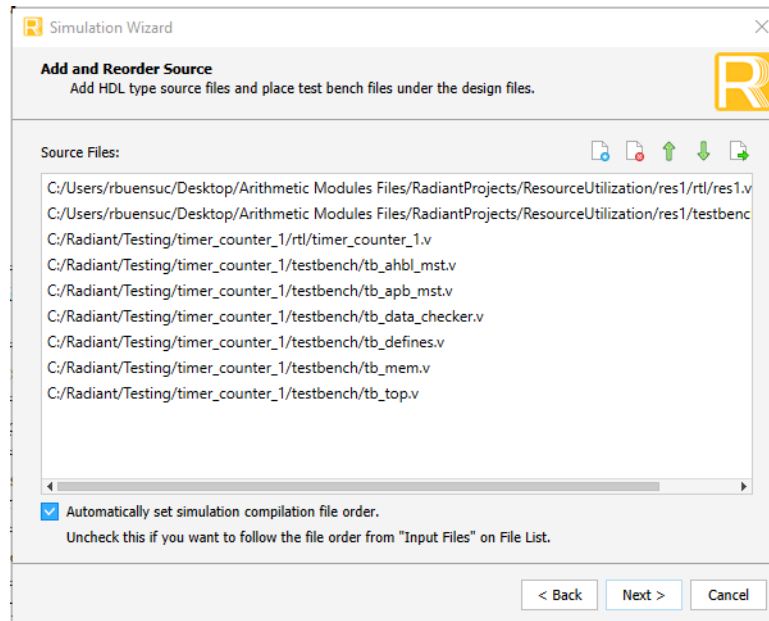


Figure 3.5. Adding and Reorder Source

3. Click **Next**. The **Summary** window is shown in [Figure 3.6](#).

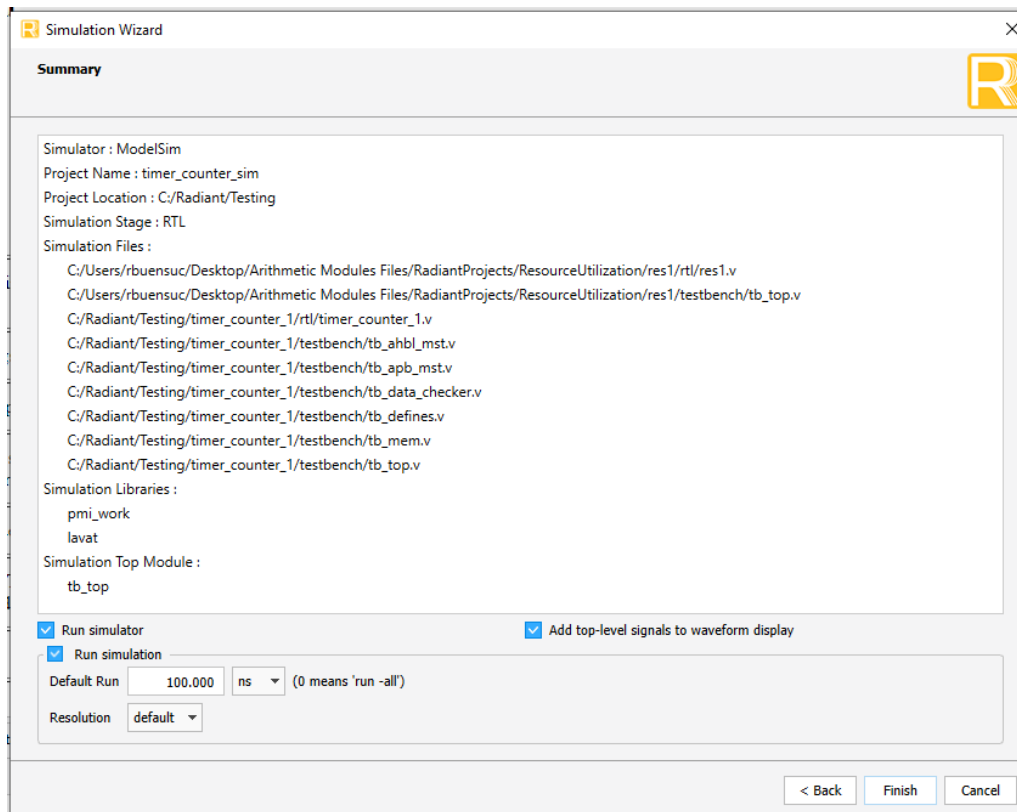


Figure 3.6. Summary Window

4. Click **Finish** to run the simulation.

Note: It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant software Suite. The results of the simulation in our example are provided in [Figure 3.7](#).

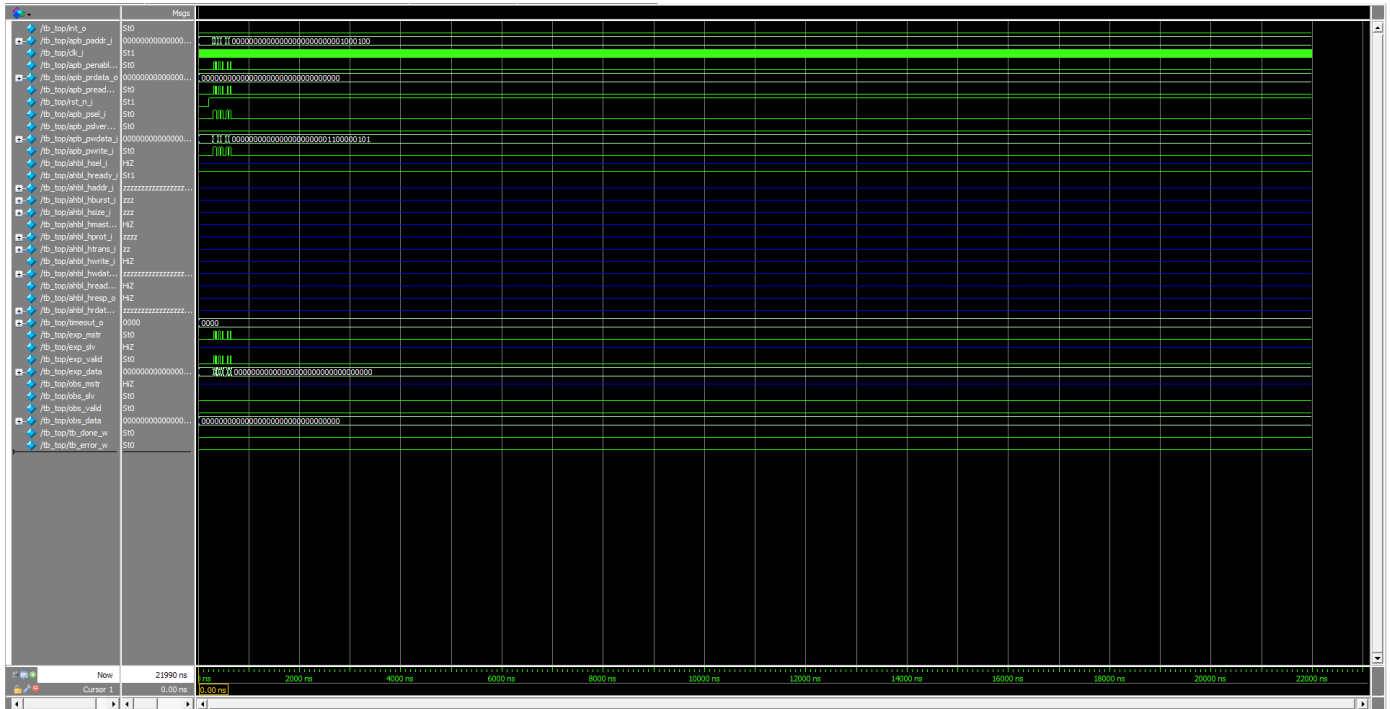


Figure 3.7. Simulation Waveform

3.4. Constraining the IP

You need to provide proper timing and physical design constraints to ensure that your design meets the desired performance goals on the FPGA. Add the content of the following IP constraint file to your design constraints: `//eval/constraint.pdc`.

The above constraint file has been verified during IP evaluation with the IP instantiated directly at the top-level module. You can modify the constraints in this file with thorough understanding of the effect of each constraint.

To use this constraint file:

- Copy the contents of `constraint.pdc` to the top-level design constraint for post-synthesis.

*Note: This example is for a standalone instance of the IP where the top level is the IP itself. If the IP is used or instantiated and the clock is connected to a top-level port that already has a constraint, then this is not necessary.

Refer to [Lattice Radiant Timing Constraints Methodology](#) for details on how to constrain your design.

Appendix A. Resource Utilization

Table A.1. Resource Utilization

No. of Timer	Registers	LUTs	EBRs	Target Device	Synthesis Tools
4	462	754	0	CrossLink™-NX	Synopsys Synplify Pro
4	446	600	0	Avant™-E	Synopsys Synplify Pro
4	469	729	0	Certus™-NX	Synopsys Synplify Pro
4	477	706	0	Certus-N2	Synopsys Synplify Pro

References

Below are some useful links related to the Timer/Counter IP Core.

- [Lattice Propel](#) Web Page
- [CrossLink-NX](#) Devices Web Page
- [Certus-NX](#) Devices Web Page
- [Certus-N2](#) Devices Web Page
- [Avant-AT-E](#) Devices Web Page
- [Lattice Solutions IP Cores](#) Web Page
- [Lattice Radiant Timing Constraints Methodology \(FPGA-AN-02059\)](#)
- [Timer/Counter IP Release Notes \(FPGA-RN-02022\)](#)
- [Lattice Insights](#) for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.4, IP v1.4.0, December 2024

Section	Change Summary
All	<ul style="list-style-type: none"> Updated document title from <i>Timer/Counter IP Core – Lattice Propel Builder</i> to <i>Timer/Counter IP</i>. Added the IP version information on the cover page.
Inclusive Language	Added boilerplate.
Functional Description	Updated the <i>rst_n_i</i> port description in Table 2.1. Timer/Counter IP Core Signal Description .
IP Generation, Synthesis, and Validation	Updated instances of <i>black box</i> to <i>closed-box</i> in the Generating and Synthesizing the IP section.
Appendix A. Resource Utilization	<p>In Table A.1. Resource Utilization:</p> <ul style="list-style-type: none"> Updated the <i>Certus™-NX</i> device information. Added the <i>Certus-N2</i> device resource utilization.
References	Added the <i>Certus-N2</i> web page, <i>Lattice Solutions IP Cores</i> web page, and <i>Timer/Counter IP Release Notes (FPGA-RN-02022)</i> .

Revision 1.3, December 2023

Section	Change Summary
Disclaimers	Updated this section.
IP Generation, Synthesis, and Validation	Added this section.

Revision 1.2, August 2023

Section	Change Summary
Functional Description	<ul style="list-style-type: none"> Table 2.1. Timer/Counter IP Core Signal Description: <ul style="list-style-type: none"> switched the contents of the <i>Width</i> and <i>Data</i> columns for those ports in the AHB-Lite Subordinate Interface; changed <i>ahbl_hburst_i[2:0]</i> to <i>ahbl_hburst_i</i>; changed <i>ahbl_hsize_i[2:0]</i> to <i>ahbl_hsize_i</i>; changed <i>ahbl_hprot_i[3:0]</i> to <i>ahbl_hprot_i</i>; changed <i>ahbl_htrans_i[1:0]</i> to <i>ahbl_htrans_i</i>. In the Register Description section, <i>changed per-timer registers are mapped to 0xNO-0xNOC</i> to <i>per-timer registers are mapped to 0xNO-0xNC</i>.
References	Newly added this section.
Technical Support Assistance	Added the link to Lattice Answer Database.

Revision 1.1, November 2022

Section	Change Summary
Introduction	<ul style="list-style-type: none"> Updated the feature regarding register configuration in the Features section to include configuration through AMBA 3 AHB-Lite Protocol v1.0. In the Host section, added AHB-Lite.
Functional Description	<ul style="list-style-type: none"> In the Overview section, added AHB-Lite to the register access applicable to the timer block. In Table 2.1. Timer/Counter IP Core Signal Description: <ul style="list-style-type: none"> added the signals of the AHB-Lite Subordinate Interface; renamed Master as Manager; renamed Slave as Subordinate. In Table 2.2. Attributes Table, added the Host Interface attribute. In Table 2.3. Attribute Descriptions, added the Host Interface attribute.

Section	Change Summary
Appendix A. Resource Utilization	Added the resource utilization information for Avant-E device.

Revision 1.0, November 2021

Section	Change Summary
All	Initial release.



www.latticesemi.com