Watchdog Timer IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02097-1.0

December 2019

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 and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
WDT	Watchdog Timer
NMI	Non-Maskable Interrupt
FPGA	Field Programmable Gate Array
RTL	Register Transfer Level

1. Introduction

The Watchdog Timer IP Core is designed for use as an indicator that a corrective action is needed in response to a computer or a processor malfunction. The design features a two-stage timer which supports non-maskable interrupt and a hard reset.

This design is implemented in Verilog. It can be targeted to CrossLink-NX™ FPGA devices and implemented using the Lattice Radiant® Software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.1. Quick Facts

Table 1.1 presents a summary of the Watchdog Timer IP Core.

Table 1.1. Quick Facts

IP Requirements	Supported FPGA Family	CrossLink-NX
Resource Utilization	Targeted Device	LIFCL-40
	Resources	See Table A.1
Design Tool Support	Lattice Implementation	IP Core v1.0.x – Lattice Radiant Software 2.0
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro, O-2018.09LR-SP1
	Simulation	For a list of supported simulators, see the Lattice Radiant Software 2.0 User Guide.

1.2. Features

Key features of the Watchdog Timer IP Core include:

- Seemlessly work with sleep mode
- Programmable timer spec 50-500 ms (10 ms)
- Two-stage timer that supports warm boot and cold boot resets
- You can specify in cycles or time
- Supports trigger input, reset and NMI output

1.3. Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

Signal Names that end with:

- *_n* are active low
- *_i* are input signals
- *_o* are output signals

1.3.3. Attribute

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

2. Functional Descriptions

2.1. Overview

The Watchdog Timer IP Core performs the following functions:

- Send a non-maskable interrupt to the computer (nmi_o)
- Send a reset to the computer for system restart (reset_o)

WDT IP Core is a two-stage timer that will start to count depending on the computer status. The computer continuously sends a kick signal to the watchdog timer as its status, absence of the kick signal means there is a hardware malfunction or program error in the computer.

The Watchdog Timer IP Core uses an up-counter that counts from zero to a timeout value at a rate that is determined by the configurable mode and the clock frequency.

2.1.1. Watchdog Timer Modes

Cycle

When in cycle mode, the user can configure the timeout in terms of the number of cycles for each stage of timer.The number of cycle refers to the count needed by the watchdog timer to reach in order to assert the timeout signal.

Time

Using the time mode, the user can configure the timeout in milliseconds and the clock frequency ranging from 10kHz to 150 MHz.

Reset ports

The Watchdog Timer IP Core supports both warm boot (nmi_o) and cold boot (reset_o) resets. Reset ports will be asserted once the timeout value is reached. The warm boot reset is a non-maskable interrupt which serves as a warning to the computer that a system reset is expected. When this port is asserted, the computer should either debug the program error or store the important files in running programs. The cold boot or the hard reset, as the name implies, will reset the whole system once the second timeout value has reached.

Figure 2.1 shows a functional diagram for the Watchdog Timer IP Core.

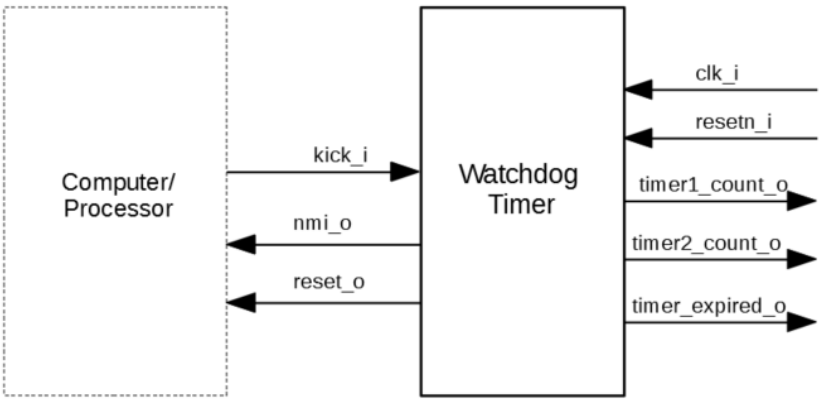

Figure 2.1. Functional Block Diagram

2.2. Signal Description

Table 2.1. Watchdog Timer IP Core Signal Description

Port Name	I/O	Width	Description
Clock and Reset			
clk_i	In	1	System clock.
resetn_i	In	1	Active low reset.
reset_o	Out	1	Cold boot output reset
Interrupt Port			
nmi_o	Out	1	Non-maskable interrupt signal.
WDT Ports			
kick_i	In	1	Kick signal connected from an external computer/processor.
timer1_count_o	Out	25	Up counter for Timer Stage 1
timer2_count_o	Out	25	Up counter for Timer Stage 2
timer_expired_o	Out	1	Timer Expiration bit. 1'b0 – watchdog timer is still counting 1'b1 – watchdog timer count expires

2.3. Attribute Summary

The configurable attributes of the Watchdog Timer IP Core are shown in Table 2.2 and are described in Table 2.3. The attributes can be configured through the IP Catalog's Module/IP wizard of the Lattice Radiant Software.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
General			
Mode	("Cycle", "Time")	"Cycle"	—
Frequency Unit	("kHz", "MHz")	"MHz"	Active if *Watchdog Timer Mode* == Time
System Clock Frequency	(10, 999), (1,150)	100	Active if *Watchdog Timer Mode* == Time, else will use the testbench clock frequency provided by the user. Selectable values are dependent on *Frequency Unit*
Timer Stage 1			
Timeout (number of cycles)	(500, 75000000)	5000	Active if *Watchdog Timer Mode* == Cycle
Timeout (ms)	(50, 500)	50	Active if *Watchdog Timer Mode* == Time
Timer Stage 2			
Timeout (number of cycles)	(500, 75000000)	5000	Active if *Watchdog Timer Mode* == Cycle
Timeout (ms)	(50, 500)	50	Active if *Watchdog Timer Mode* == Time
Calculated			
Timeout S1 (number of cycles)	N/A	N/A	—
Timeout S2 (number of cycles)	N/A	N/A	—

Table 2.3. Attributes Descriptions

Attribute	Description
General	
Mode	Specifies the Watchdog Timer Mode.
Frequency Unit	Specifies the unit for *System Clock Frequency.*
System Clock Frequency	Specifies the user clock frequency.
Timer Stage 1	
Timeout (number of cycles)	Specifies the timeout of Timer Stage 1 in terms of number of cycles when the selected *Mode* is Cycle.
Timeout (ms)	Specifies the timeout of Timer Stage 1 in terms of ms.
Timer Stage 2	
Timeout (number of cycles)	Specifies the timeout of Timer Stage 2 in terms of number of cycles when the selected *Mode* is Cycle.
Timeout (ms)	Specifies the timeout of Timer Stage 2 in terms of ms.
Calculated	
Timeout S1 (number of cycles)	Specifies the calculated number of cycles for Timer Stage 1 when the selected *Mode* is Time.
Timeout S2 (number of cycles)	Specifies the calculated number of cycles for Timer Stage 2 when the selected *Mode* is Time.

2.4. Timing Diagrams

During normal operation, the connected computer/processor regularly sends a “kick” signal to the watchdog timer using kick_i port. When the kick of computer suddenly stopped, timer stage 1 will begin to count. Timer Stage 1 will eventually equal to the timeout value which will assert the nmi_o port. This port will trigger the counting of Timer Stage 2 and simultaneously notify the computer by means of the warm boot (*nmi_o*) that a reset is imminent. After the Timer Stage 2 times out, cold boot (*reset_o*) will assert which will forcefully restart the computer. See Figure 2.2 for a sample Watchdog Timer timing diagram.

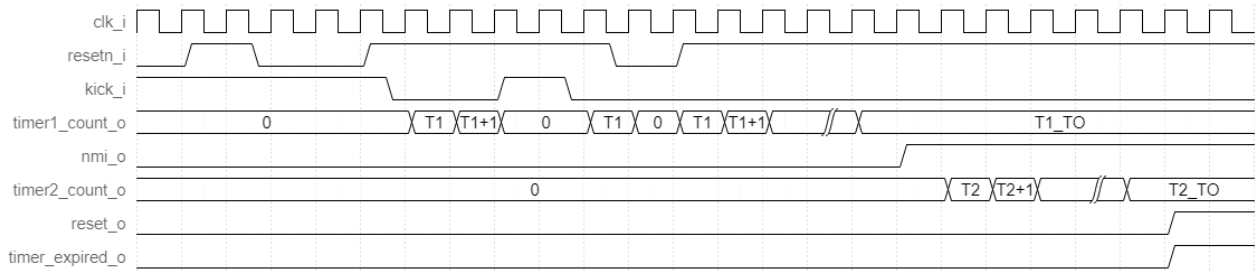

Figure 2.2. Timing Diagram

3. IP Generation and Evaluation

This section provides information on how to generate the Watchdog Timer IP Core using the Lattice Radiant Software and how to run simulation and synthesis. For more details on the Lattice Radiant Software, refer to the Lattice Radiant Software 2.0 User Guide.

3.1. Licensing the IP

An IP core-specific license string is required enable full use of the Watchdog Timer IP Core in a complete, top-level design. You can fully evaluate the IP core through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core, which operate in hardware for a limited time (approximately four hours) without requiring an IP license string. See Hardware Evaluation section for further details. However, a license string is required to enable timing simulation and to generate bitstream file that does not include the hardware evaluation timeout limitation.

3.2. Generation and Synthesis

The Lattice Radiant Software allows you to customize and generate modules and IPs and integrate them into the device's architecture. The procedure for generating the Watchdog Timer IP Core in Lattice Radiant Software is described below.

To generate the Watchdog Timer IP Core:

1. Create a new Lattice Radiant Software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **Watchdog_Timer** under **IP, Processors_Controllers_and_Peripherals** category. The **Module/IP Block Wizard** opens as shown in Figure 3.1. Enter values in the **Instance name** and the **Create in** fields and click **Next**.

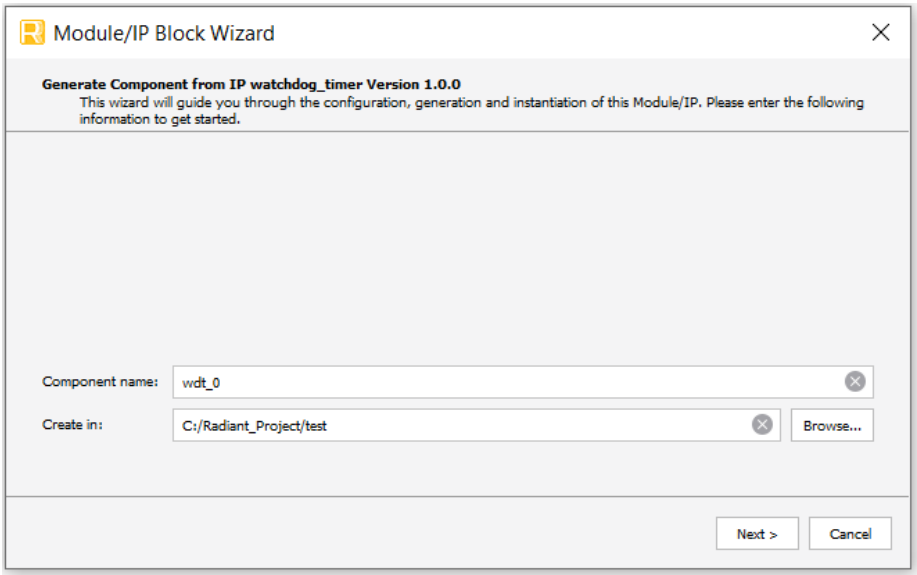

Figure 3.1. Module/IP Block Wizard

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

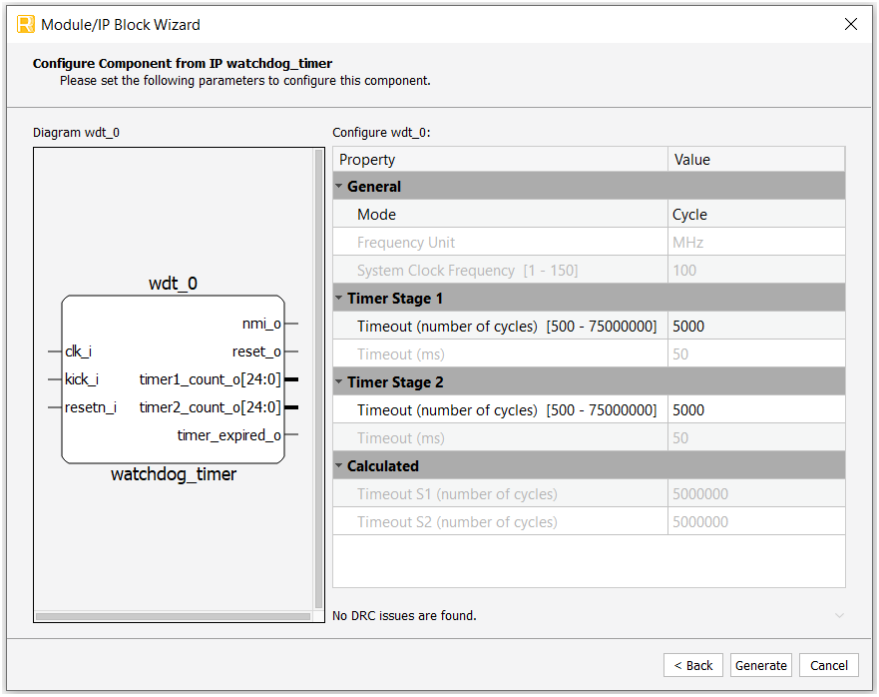

Figure 3.2. Configure User Interface of Watchdog Timer IP Core

4. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in Figure 3.3.

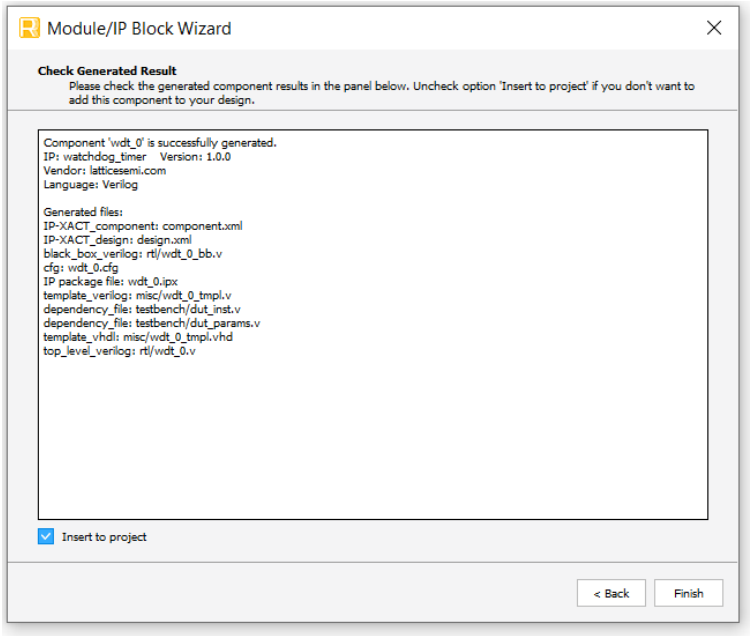

Figure 3.3. Check Generating Result

5. Click the **Finish** button. All the generated files are placed under the directory paths in the **Create in** and the **Instance name** fields shown in Figure 3.1.

The generated Watchdog Timer IP Core package includes the black box (<Instance Name>_bb.v) and instance templates (<Instance Name>_tmpl.v/vhd) that can be used to instantiate the core 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 IP core is also provided. You may also use this top-level reference as the starting template for the top-level for their 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 parameter values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Instance Name>.v	This file provides an example RTL top file that instantiates the IP core.
rtl/<Instance Name>_bb.v	This file provides the synthesis black box.
misc/<Instance Name>_tmpl.v misc /<Instance Name>_tmpl.vhd	These files provide instance templates for the IP core.

3.3. Running Functional Simulation

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

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 3.4.

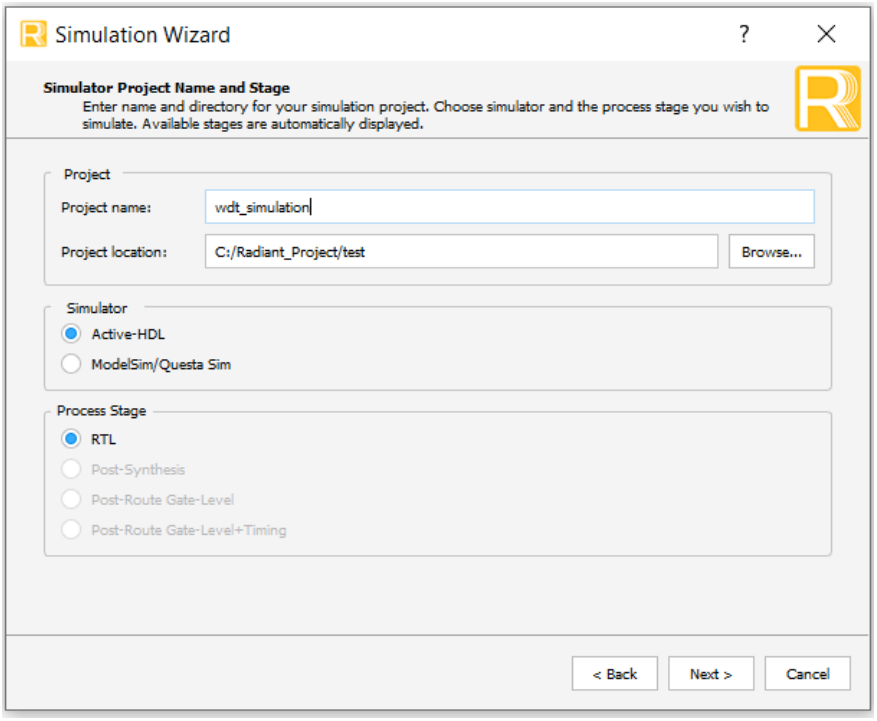

Figure 3.4. Simulation Wizard

1. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.5.

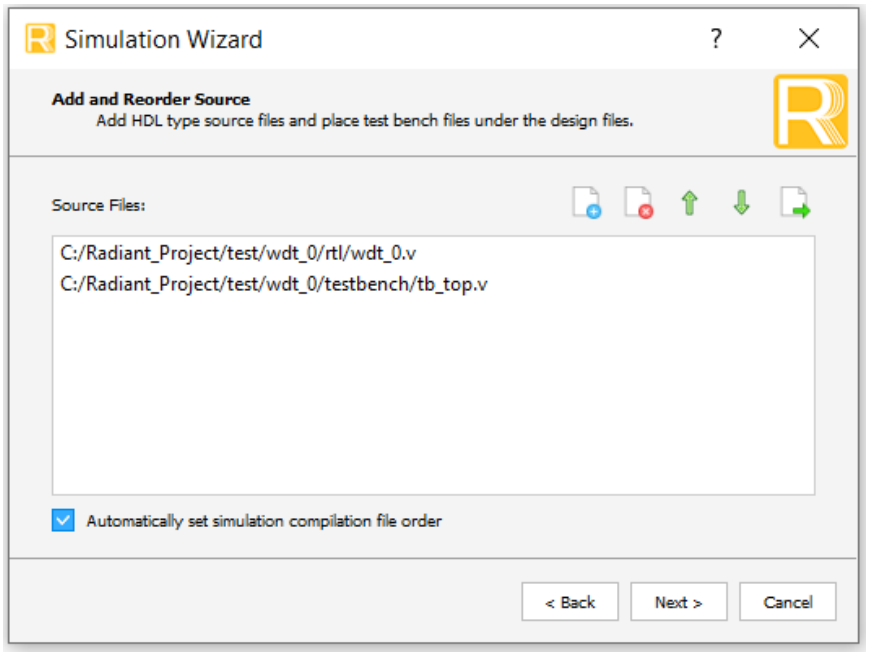

Figure 3.5. Adding and Reordering Source

2. Click **Next**. The Summary window is shown. 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 3.6.

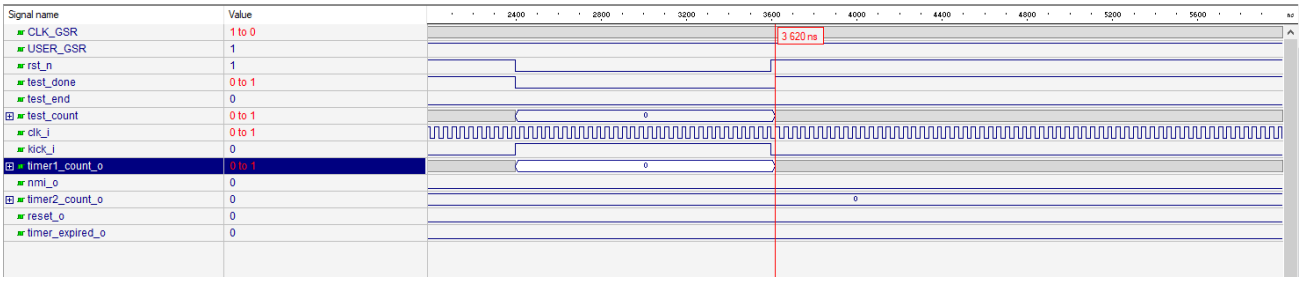

Figure 3.6. Simulation Waveform

3.4. Hardware Evaluation

The Watchdog Timer IP Core supports Lattice's IP hardware evaluation capability when used with LIFCL devices. This makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.

References

For complete information on Lattice Radiant Project-Based Environment, Design Flow, Implementation Flow and Tasks, as well as on the Simulation Flow, see the Lattice Radiant Software 2.0 User Guide.

Technical Support Assistance

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

Appendix A. Resource Utilization

Table A.1 shows the resource utilization of the Watchdog Timer IP Core for the LIFCL-40 device, using Lattice Synthesis Engine of Lattice Radiant Software. Default configuration is used and some attribues are changed from the default value to show the effect on the resource utilization.

Table A.1. Resource Utilization

Configuration	Slice Registers	LUTs	EBRs
Default	52	116	0
Timer Stage 1 *Timeout* = 1050, Others = Default	52	117	0
Mode Time, *Clock Frequency* = 100, Others = Default	52	128	0
Mode Time, Timer Stage 2 *Timeout* = 100, Others = Default	52	126	0

***Note:** Fmax is generated when the FPGA design only contains Watchdog Timer IP Core. These values may be reduced when user logic is added to the FPGA design.

Revision History

Revision 1.0, December 2019

Section	Change Summary
All	Initial release.

www.latticesemi.com