Divider IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02130-1.1

June 2021

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
FPGA	Field Programmable Gate Array
RTL	Register Transfer Level
DSP	Digital Signal Processing
LUT	Look Up Table

1. Introduction

The Divider IP core is a one-clock divider which completes one integer division every clock. It supports signed or unsigned inputs and provides configurable output latency.

1.1. Quick Facts

Table 1.1 presents a summary of the Divider IP Core.

Table 1.1. Quick Facts

IP Requirements	Supported FPGA Families	CrossLink™-NX, Certus™-NX, CertusPro™-NX
Resource Utilization	Targeted Devices	LIFCL-40, LIFCL-17, LFD2NX-40, LDF2NX-40, LFCPNX-100
	Resources	See Table A.1, Table A.2, Table A.3
Design Tool Support	Lattice Implementation	IP Core v1.x.x – Lattice Radiant® Software 2.1 or later
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro® for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software User Guide.

1.2. Features

The key features of Divider IP Core include:

- Supports signed or unsigned numerator and denominator
- Supports numerator and denominator data width 4-64
- Supports forced positive remainder
- Supports configurable output latency
- Optional clock enable and data valid ports

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:

- _*i* are input signals
- _*o* are output signals

1.3.3. Host

The logic unit inside the FPGA interacts with the Divider IP Core

1.3.4. Attribute

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

2. Functional Description

2.1. Overview

The Divider IP core implements integer division with the formula:

*Numerator = Denominator * Quotient + Remainder*

The Numerator and the Denominator can be signed or unsigned integers. When either the Numerator or the Denominator is a signed integer, the Quotient and the Remainder are also in signed integer format. When the Remainder is configured as *Always positive remainder*, it becomes a positive signed integer value.

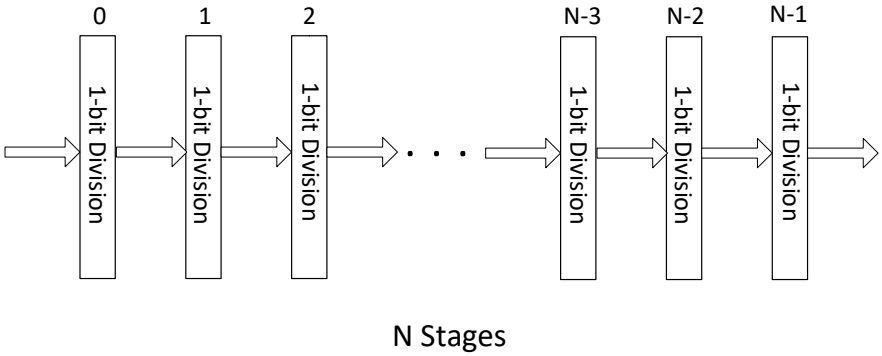

Figure 2.1. Divider IP Core Functional Diagram

2.2. Primary I/O

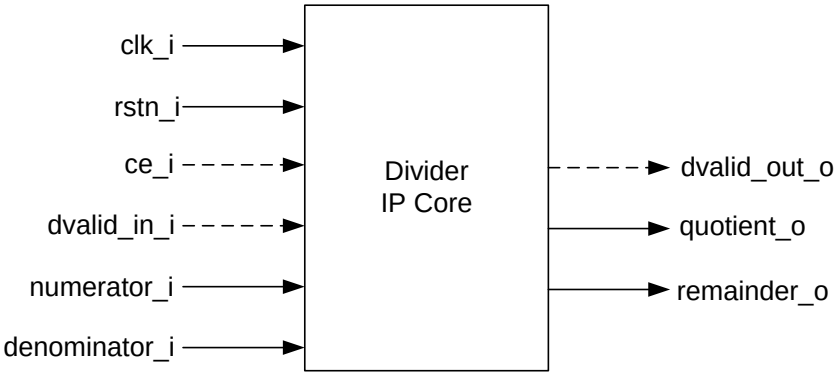

Figure 2.2. Divider IP core I/O Diagram

2.3. Signal Description

Table 2.1 lists the input and output signals for Divider IP Core.

Table 2.1. Divider IP Core Signal Description

Ports Name	Size	Direction	Description
clk_i	1	I	Input clock
rstn_i	1	I	Asynchronous active-low reset signal
numerator_i	4-64	I	Input numerator value
denominator_i	4-64	I	Input denominator value
dvalid_in_i	1	I	Optional input data valid signal, active-high
ce_i	1	I	Clock enable, active high
dvalid_out_o	1	O	Optional output data valid signal, active high
quotient_o	4-64	O	Output quotient
remainder_o	4-64	O	Output remainder

2.4. Attributes Summary

The configurable attributes of the Divider 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
Structure			
Numerator data type	Unsigned, Signed	Unsigned	—
Denominator data type	Unsigned, Signed	Unsigned	—
Numerator data width	4-64	20	—
Denominator data width	4-64	10	—
Output Latency	1-64	20	Depends on Quotient width
Always positive remainder	Checked, Unchecked	Unchecked	Depends on Numerator data type
Clock enable port	Checked, Unchecked	Checked	—
Data valid ports	Checked, Unchecked	Checked	—

Table 2.3. Attributes Descriptions

Attribute	Description
Structure	
Numerator data type	Numerator data type specifies the data type of input Numerator
Denominator data type	Denominator data type specifies the data type of input Denominator
Numerator data width	Numerator data width specifies the data width of input Numerator.
Denominator data width	Numerator data width specifies the data width of input Denominator.
Output Latency	Output latency specifies the output latency of the Divider core. Its value can be between 1 and Numerator data width.
Always positive remainder	The Remainder is a positive signed integer value.
Clock enable port	The Clock enable port check box specifies whether the core has a clock enable port.
Data valid ports	The Data valid ports check box specifies whether the core has an input data valid and output data valid ports

2.5. Operations Details

2.5.1. General Divider Operation

The Divider IP core uses a non-restoring division algorithm to implement the integer division operation. There are N stages of 1-bit division in an integer division operation, where N is the width of the quotient. Each stage generates a 1-bit quotient and partial-remainder. In the last stage, the final quotient and remainder are generated. 1-bit division uses an adder-subtractor to compare the partial remainder and denominator to get a new partial remainder. Quotient-digit selection is based on the sign of the partial remainder. In the last stage, the partial remainder is corrected to get the final remainder.

The Divider IP core supports configurable output latency. The latency can be any number of clock cycles from 1 to N. When latency is set to the value M, M stages of output registers are uniformly distributed into the N stages of 1-bit division operation. The final division stage always has output registers.

2.6. Timing Specifications

The Divider IP core is a one-clock divider. It can accept a numerator and denominator every clock cycle and generate a quotient and remainder every clock cycle.

When the input numerator and denominator are in an unsigned format, the output quotient and remainder are in an unsigned format. When either the numerator or denominator is in a signed format, the output quotient and remainder are always in a signed format.

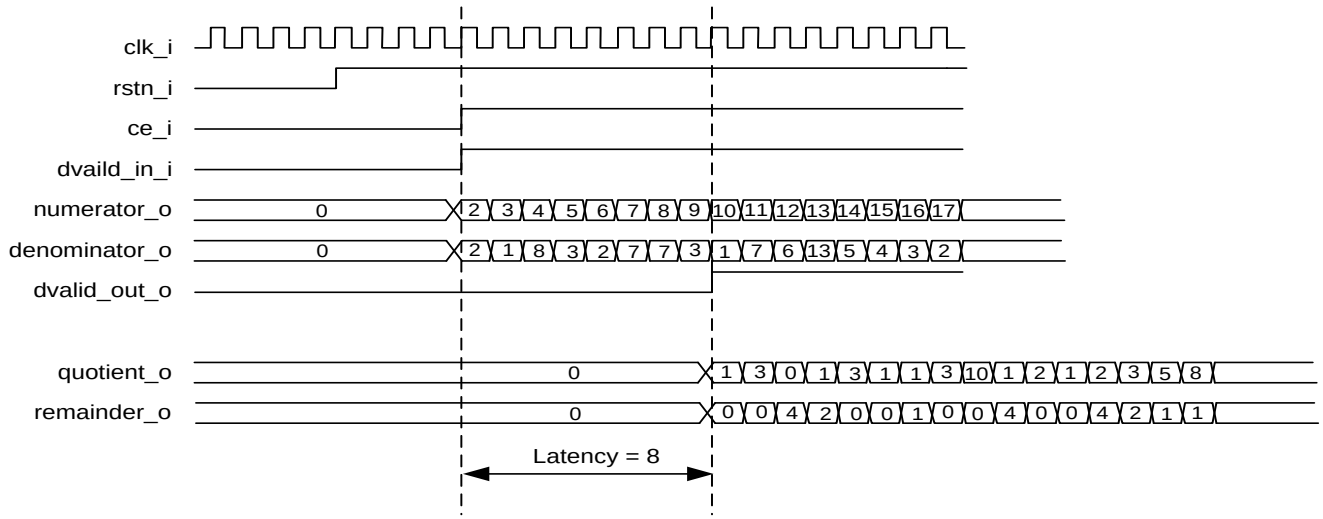

Figure 2.3. Unsigned Division for Latency 8 Diagram

3. Core Generation, Simulation, and Validation

This section provides information on how to generate the Divider 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 User Guide.

3.1. Licensing the IP

An IP core-specific license string is required enable full use of the Divider 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 Divider IP Core in Lattice Radiant software is described below.

To generate the Divider IP Core:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **Divider** under **DSP** category. 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**.

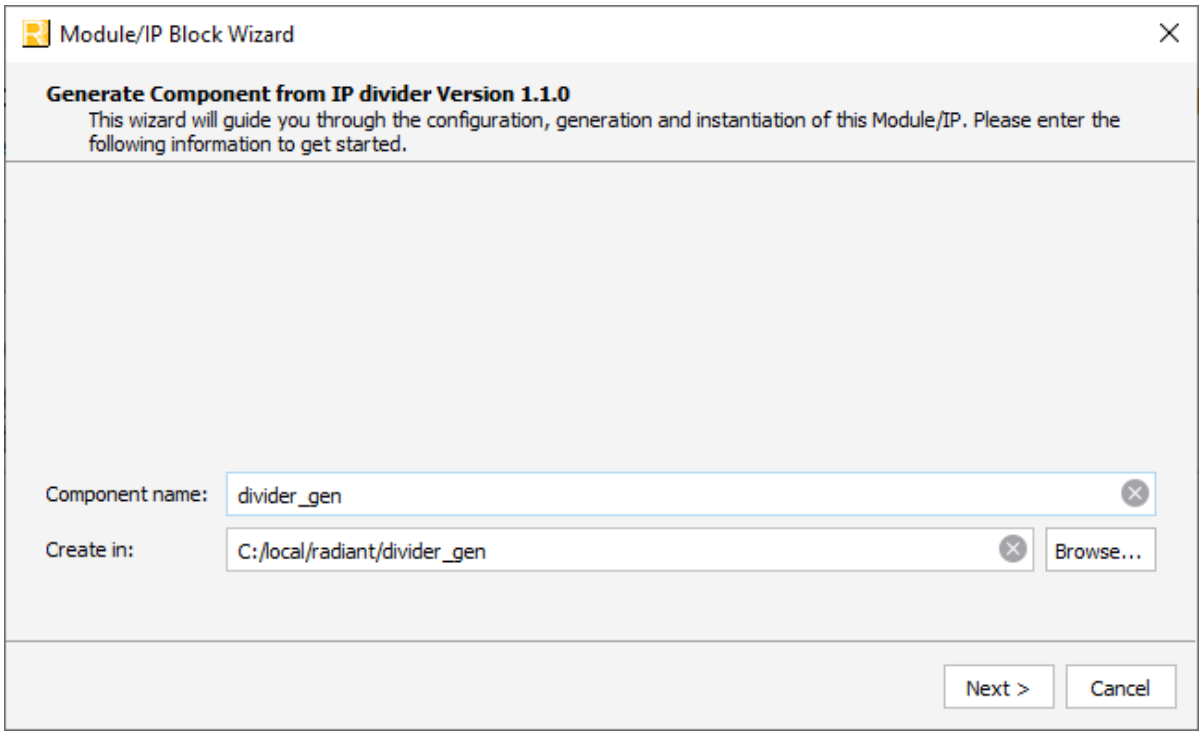

Figure 3.1. Module/IP Block Wizard

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

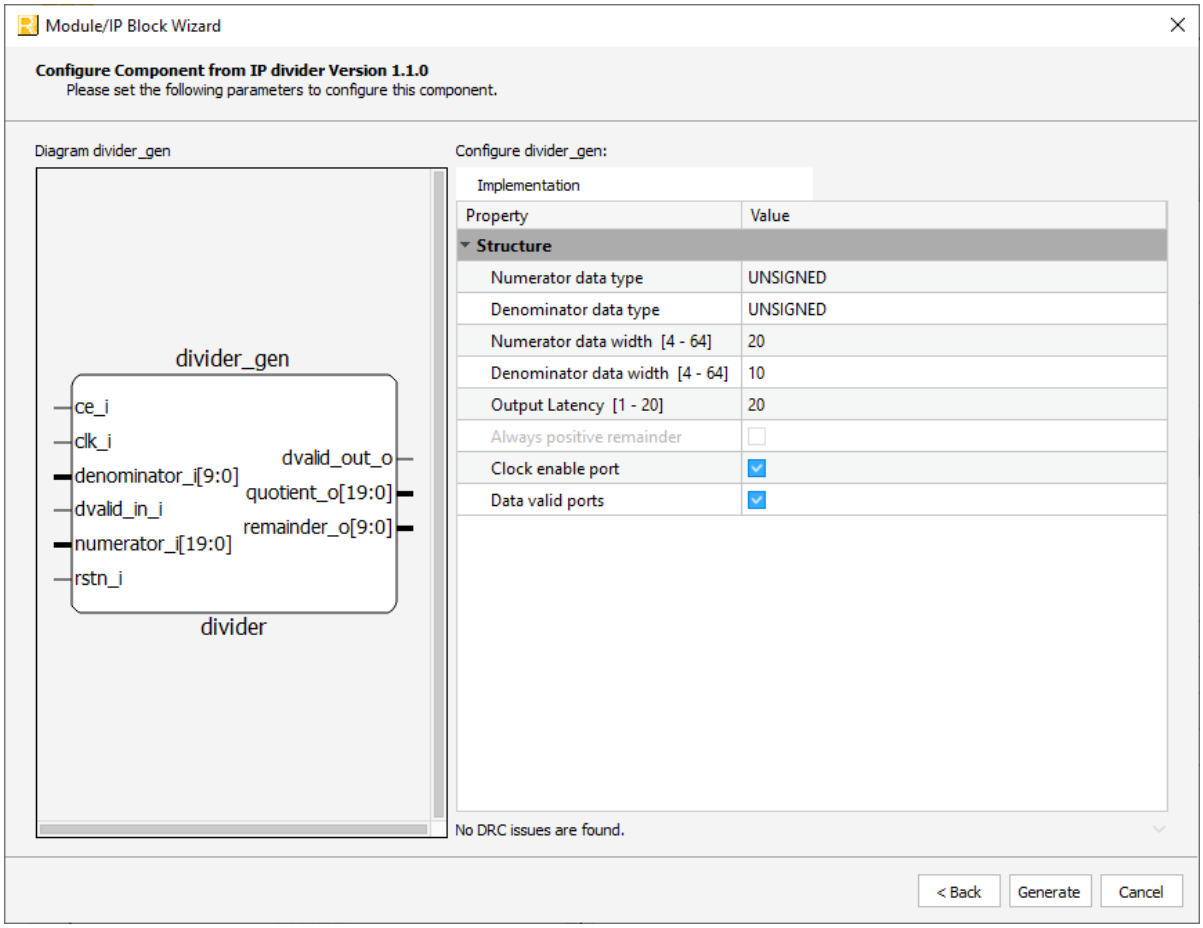

Figure 3.2. Configure User Interface of Divider IP Core

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

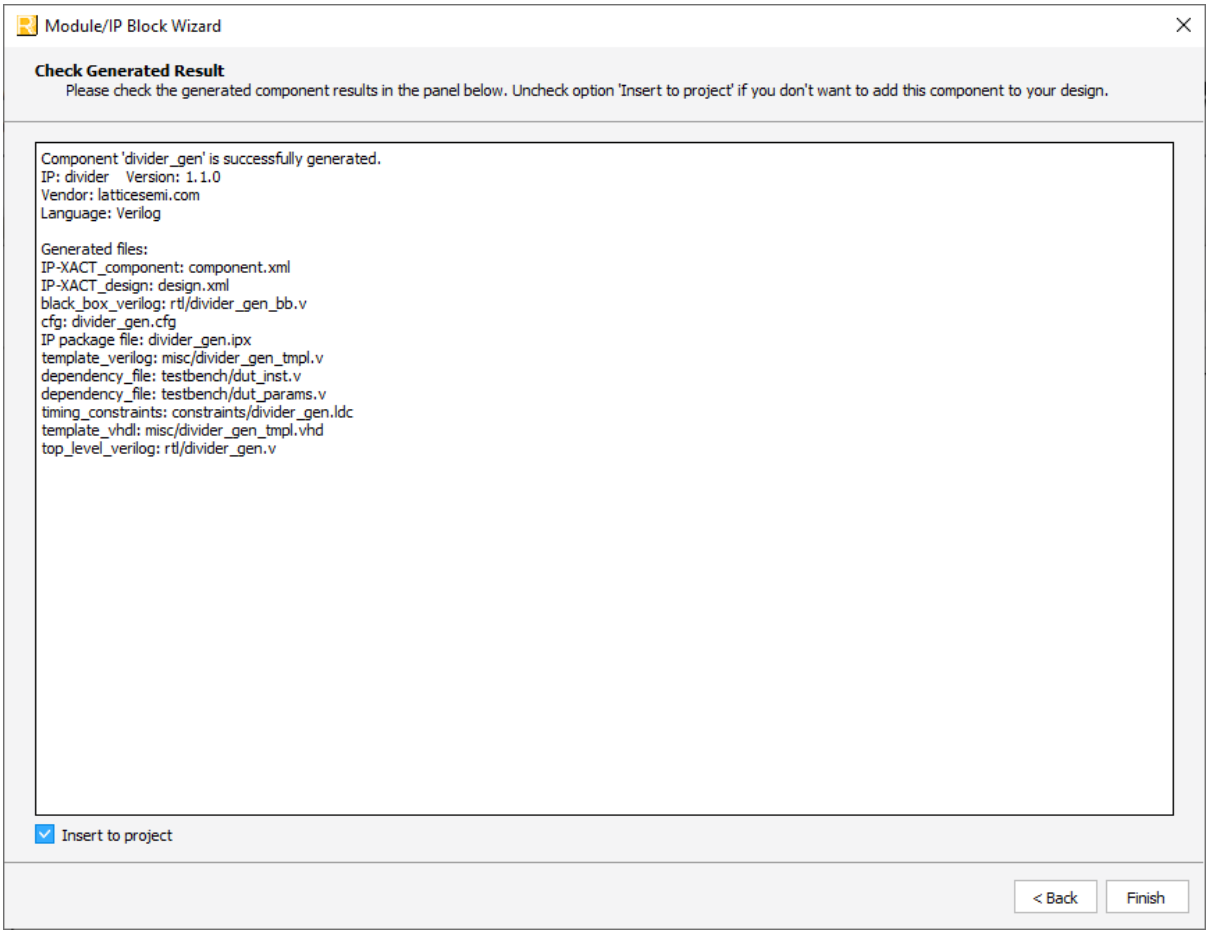

Figure 3.3. Check Generating Result

5. Click the **Finish** button.

All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 3.1.

The generated Divider IP Core package includes the black box (<Component name>_bb.v) and instance templates (<Component 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 (<Component 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
<Component name>.ipx	This file contains the information on the files associated to the generated IP.
<Component 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/<Component name>.v	This file provides an example RTL top file that instantiates the IP core.
rtl/<Component name>_bb.v	This file provides the synthesis black box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the IP core.

3.3. Running Functional Simulation

To run functional simulation:

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

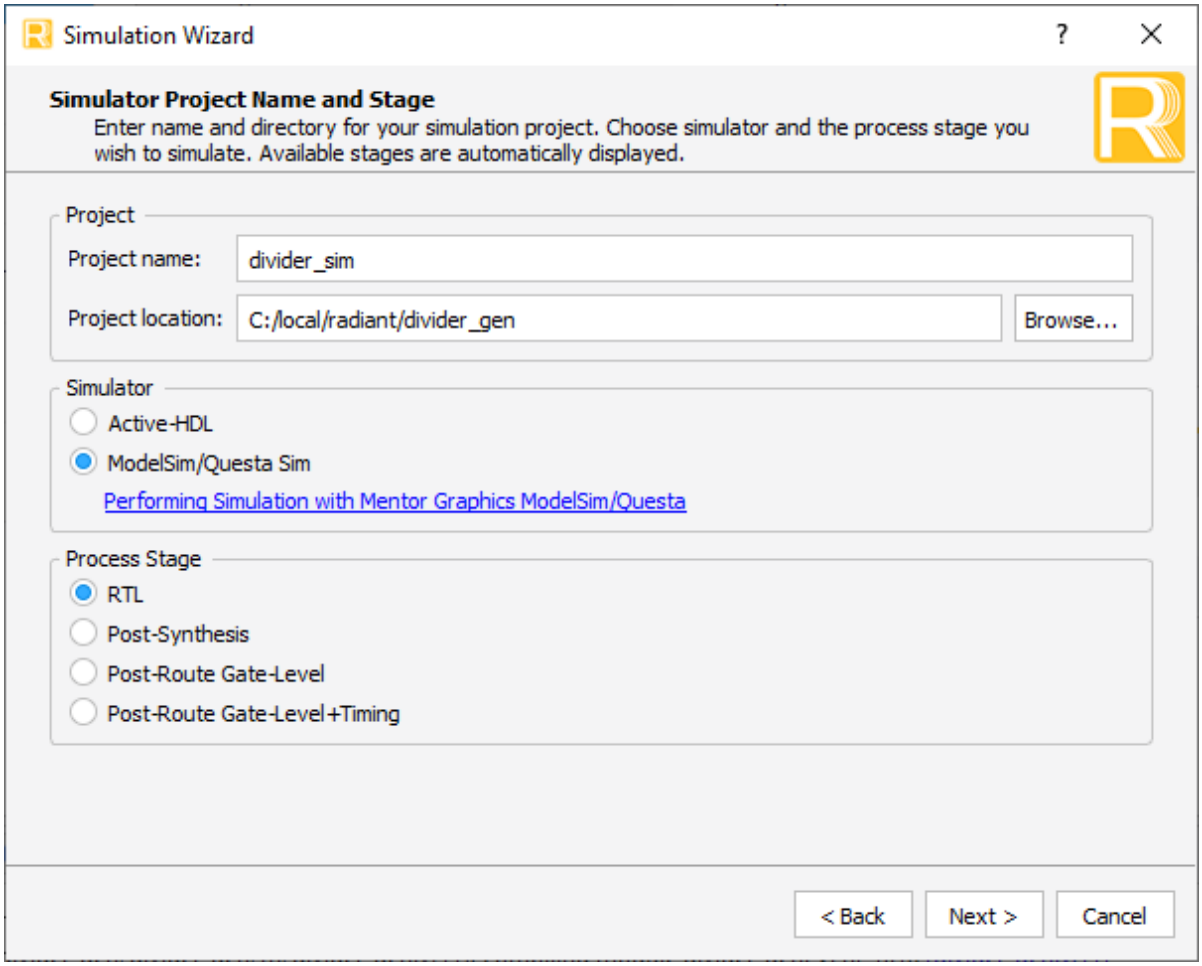

Figure 3.4. Simulation Wizard

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

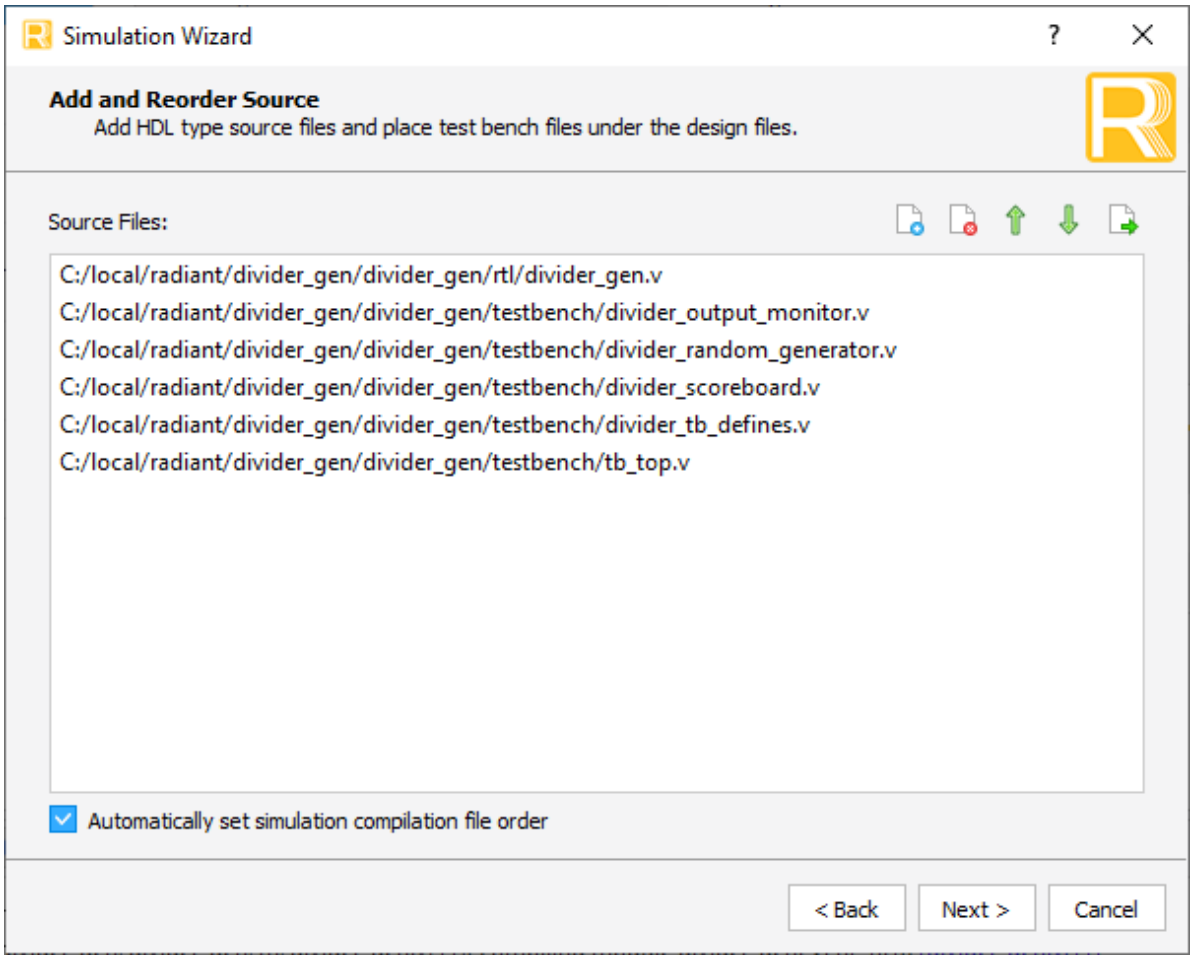

Figure 3.5. Adding and Reordering Source

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

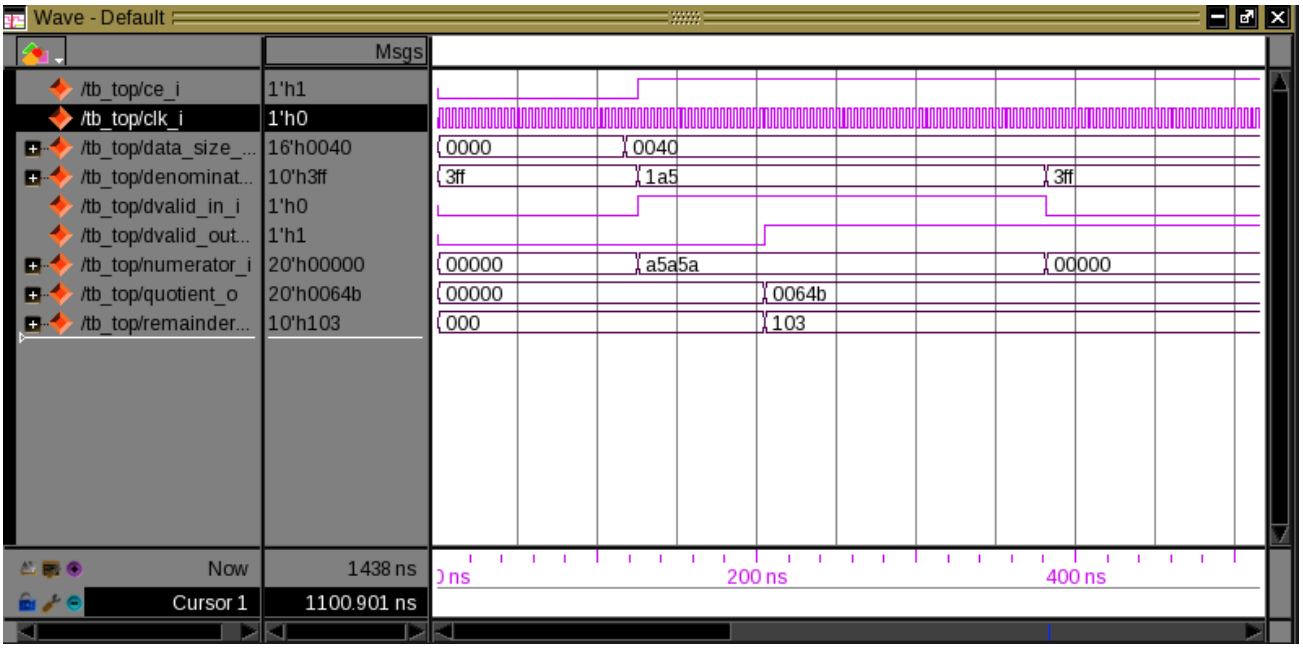

Figure 3.6. Simulation Waveform

3.4. Hardware Evaluation

The Divider IP Core supports Lattice's IP hardware evaluation capability when used with Lattice FPGA devices built on the Lattice Nexus™ platform. 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.

4. Ordering Part Number

The Ordering Part Number (OPN) for this IP Core are the following:

- DIVIDE-CNX-U – Divider for CrossLink-NX - Single Design License
- DIVIDE-CNX-UT – Divider for CrossLink-NX - Site License
- DIVIDE-CTNX-U – Divider for Certus-NX - Single Design License
- DIVIDE-CTNX-UT – Divider for Certus-NX - Site License
- DIVIDE-CPNX-U - Divider for CertusPro-NX - Single Design License
- DIVIDE-CPNX-UT - Divider for CertusPro-NX - Site License

Appendix A. Resource Utilization

Table A.1 shows configuration and resource utilization for LIFCL-40-9BG400I using Synplify Pro of Lattice Radiant software 2.1.

Table A.1. Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	EBRs
Configuration 1: 20-bit Numerator, 10-bit Denominator, 20 output Latency	200	770	341	0
Configuration 2: 24-bit Numerator, 12-bit Denominator, 12 output Latency	200	528	422	0
Configuration 3: 32-bit Numerator, 32-bit Denominator, 32 output Latency	200	2976	1281	0

***Note:** Fmax is generated when the FPGA design only contains Divider IP Core and the target Frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.2 shows configuration and resource utilization for LIFCL-17-8BG256I using Synplify Pro of Lattice Radiant software 2.1.

Table A.2. Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	EBRs
Configuration 1: 20-bit Numerator, 10-bit Denominator, 20 output Latency	200	770	341	0

***Note:** Fmax is generated when the FPGA design only contains Divider IP Core and the target Frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.3 shows configuration and resource utilization for LFD2NX-40-8BG256I using Synplify Pro of Lattice Radiant software 2.1.

Table A.3. Resource Utilization

Configuration	Clk Fmax (MHz)*	Registers	LUTs	EBRs
Configuration 1: 20-bit Numerator, 10-bit Denominator, 20 output Latency	200	770	341	0
Configuration 2: 24-bit Numerator, 12-bit Denominator, 12 output Latency	200	528	422	0
Configuration 3: 32-bit Numerator, 32-bit Denominator, 32 output Latency	200	2976	1281	0

***Note:** Fmax is generated when the FPGA design only contains Divider IP Core and the target Frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.

References

- CrossLink-NX web page at www.latticesemi.com
- Certus-NX web page at www.latticesemi.com
- CertusPro-NX web page at www.latticesemi.com

Technical Support Assistance

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

Revision History

Revision 1.1, June 2021

Section	Change Summary
Introduction	Updated Table 1.1. Quick Facts. • Revised Supported FPGA Families • Revised Targeted Devices • Revised Lattice Implementation
Core Generation, Simulation, and Validation	Replaced specific devices with *Lattice FPGA devices built on the Lattice Nexus platform* In the Hardware Evaluation sections.
Ordering Part Number	Added part numbers.
References	Added reference to the CertusPro-NX web page.

Revision 1.0, October 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com