Adder Tree Module - Lattice Radiant Software

User Guide

FPGA-IPUG-02087-1.4

November 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, with all faults and associated risk the responsibility entirely of the 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
IP	Intellectual Property
RTL	Register Transfer Level

1. Introduction

The Adder Tree Module is used to add large numbers of digits at one time. The module is pipelined and can operate on each clock cycle.

1.1. Features

The key features of Adder Tree Module include:

- Configurable data width
- Supports multiple number of inputs
- Configurable Reset Mode
- Supports Enable/Disable of Fully Pipeline Mode
- Supports Enable/Disable of Input and Output Registers

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

1.2.3. Attribute Names

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

2. Functional Description

The Adder Tree Module is a pipelined adder which implements "*data0_i + data1_i + ... + datan_i = result_o*" function.

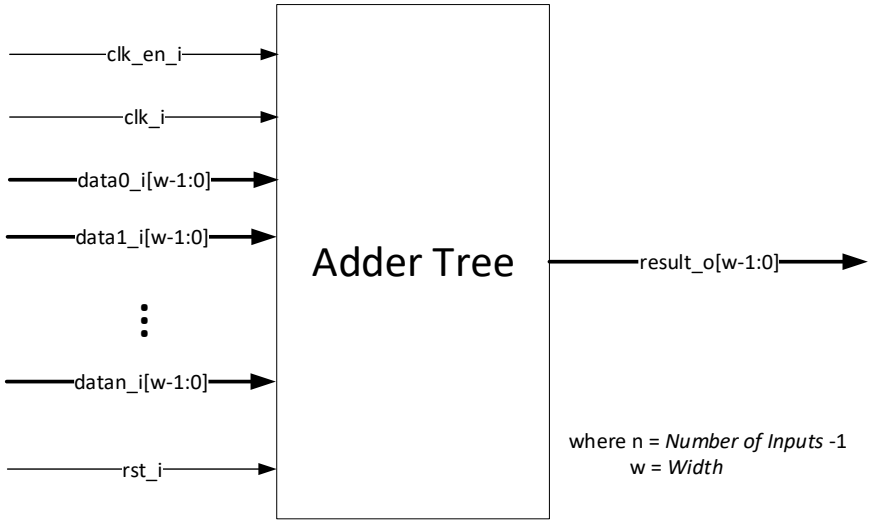

Figure 2.1. Adder Tree Block Diagram

2.1. Signal Descriptions

Table 2.1. Adder Tree Module Signal Description

Port Name	I/O	Width	Description
Clock and Reset Ports			
clk_i	In	1	System clock input. Available if any of the following attributes is/are enabled: *Enable Fully Pipelined Mode, Enable Input Register or Enable Output Register.*
rst_i	In	1	Reset input. Available if any of the following attributes is/are enabled: *Enable Fully Pipelined Mode, Enable Input Register or Enable Output Register.*
User Interface Ports			
clk_en_i	In	1	Clock enable input. Available if any of the following attributes is/are enabled: *Enable Fully Pipelined Mode, Enable Input Register or Enable Output Register.*
data[n]_i	In	*Width**	Data input.
result_o	Out	*Width**	Output Result.

***Note:** The bit width of some signals is set by the attribute. Refer to Table 2.2 for the description of these attributes.

2.2. Attribute Summary

The configurable attributes of the Adder Tree Module 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
Configuration			
Width	2 – 36	8	—
Number of inputs	2 – N*	6	—
Reset Mode	Sync, Async	Sync	Active if: *Enable Fully Pipelined Mode* == Checked or *Enable Input Register* == Checked or *Enable Output Register* == Checked
Enable Fully Pipelined Mode	Checked, Unchecked	Unchecked	—
Enable Input Register	Checked, Unchecked	Checked	—
Enable Output Register	Checked, Unchecked	Checked	Active if *Enable Fully Pipelined Mode* == Unchecked

***Note:** The maximum *Number of Inputs* depends on the Targeted Device Type. If the Device Type is LIFCL-17, N is 24 else 28.

Table 2.3. Attributes Descriptions

Attribute	Description
Configuration	
Width	Specifies the width of input and output data.
Number of inputs	Specifies the number of inputs to be added together.
Reset Mode	Specifies the mode of reset to be used. *Reset Mode* can only be configured if any of the following attributes is/are enabled: *Enable Fully Pipelined Mode, Enable Input Register or Enable Output Register.*
Enable Fully Pipelined Mode	Specifies if data pipeline is enabled or not. When this is enabled, *Enable Output Register* attribute is automatically enabled.
Enable Input Register	Specifies if input register is enabled or not.
Enable Output Register	Specifies if output register is enabled or not.

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 on Lattice Radiant software, please 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's architecture. The procedure for generating the Adder Tree Module in Lattice Radiant software is described below.

To generate the Adder Tree Module:

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

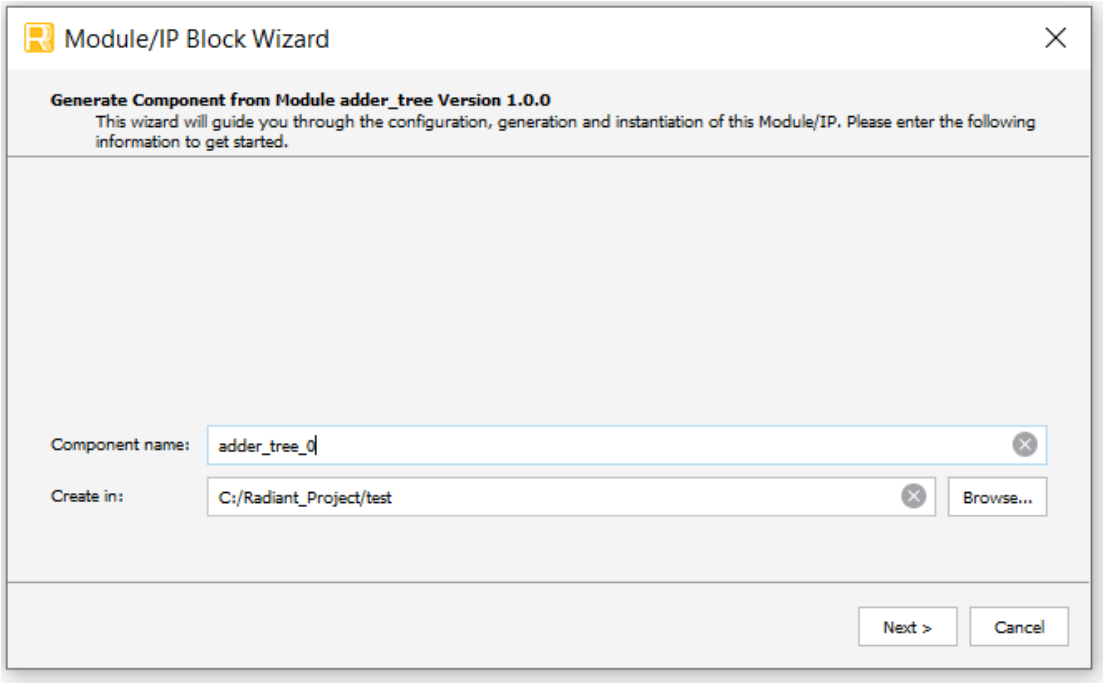

Figure 3.1. Module/IP Block Wizard

4. In the module's dialog box, customize the selected Adder Tree module. A sample configuration is shown in Figure 3.2. For configuration options, see the Attribute Summary section.

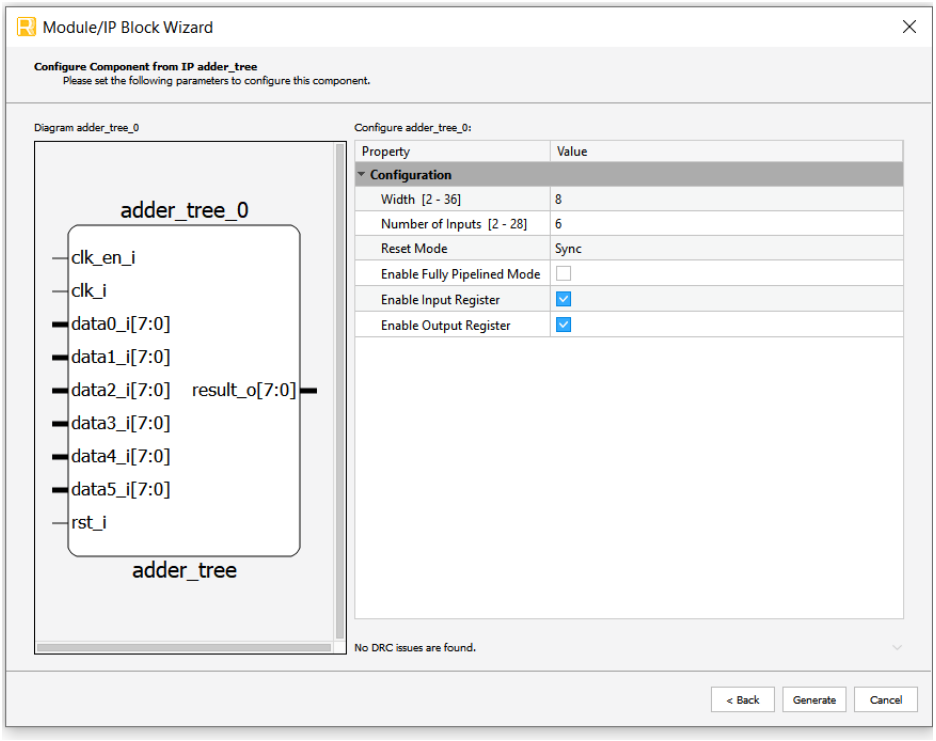

Figure 3.2. Configure User Interface of Adder Tree Module

5. Click **Generate**. The **Check Generating Result** dialog box opens. Design block messages and results are shown in Figure 3.3.

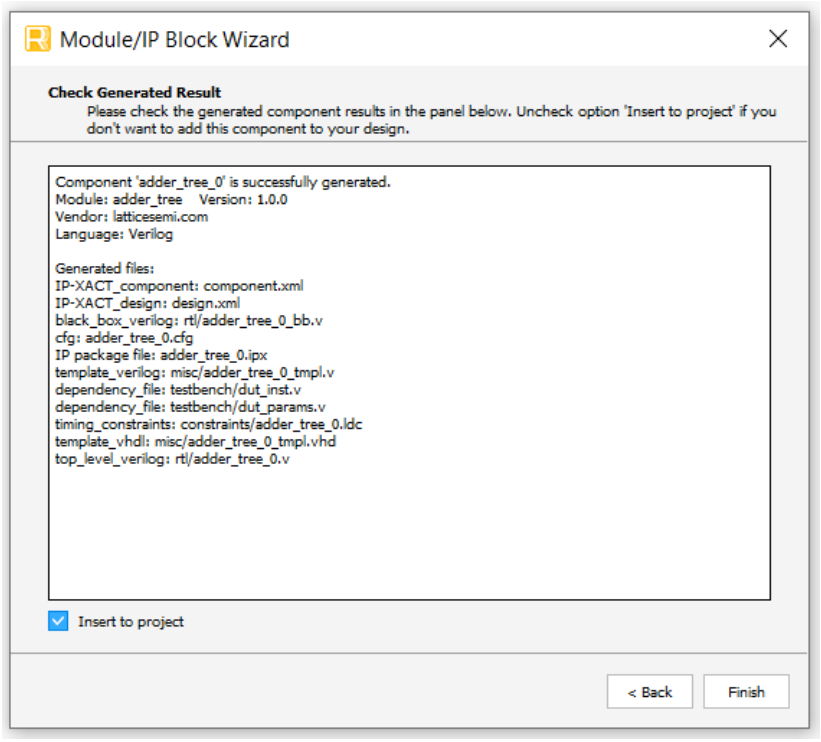

Figure 3.3. Check Generating Result

6. Click **Finish**. 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 Adder Tree Module package includes the black box (<Instance Name>_bb.v) and instance templates (<Instance Name>_tmpl.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 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 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 black box.
misc/<Instance Name>_tmpl.v misc /<Instance Name>_tmpl.vhd	These files provide instance templates for the module.

3.3. Running Functional Simulation

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

To run functional simulation:

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

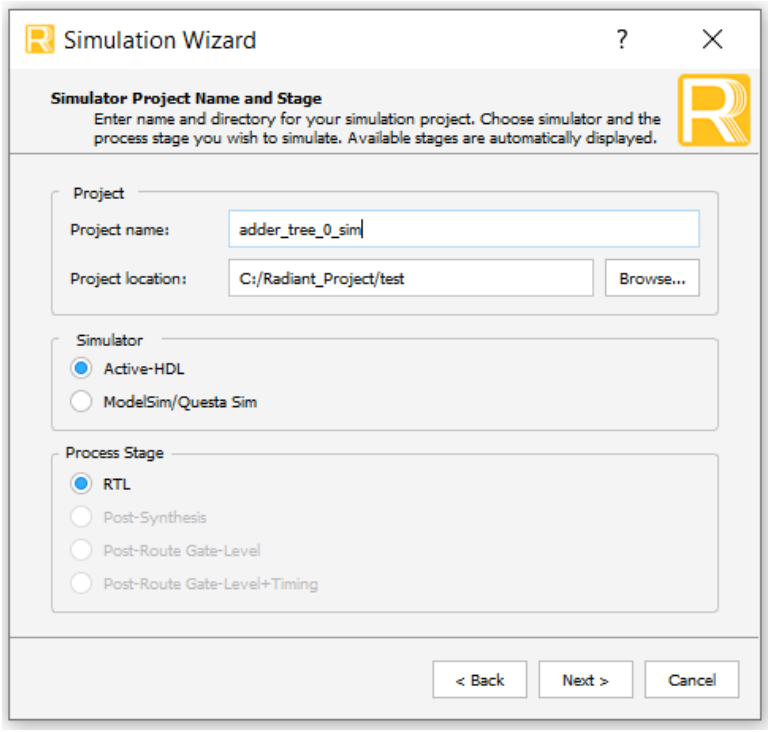

Figure 3.4. Simulation Wizard

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

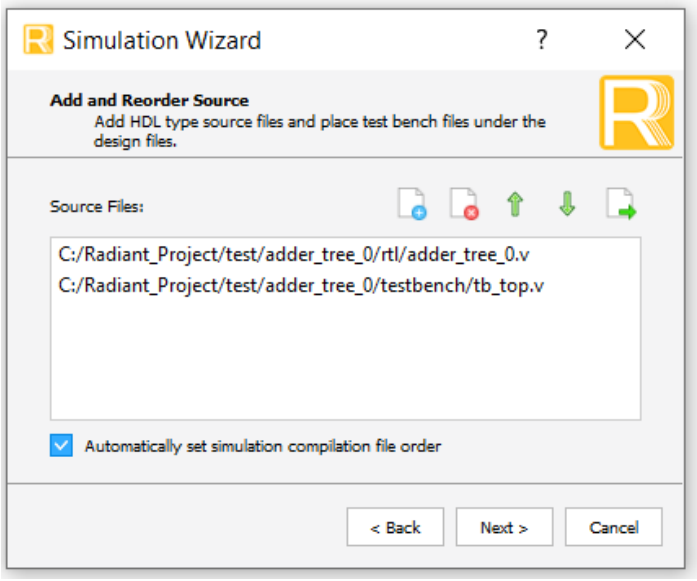

Figure 3.5. Adding and Reordering Source

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.

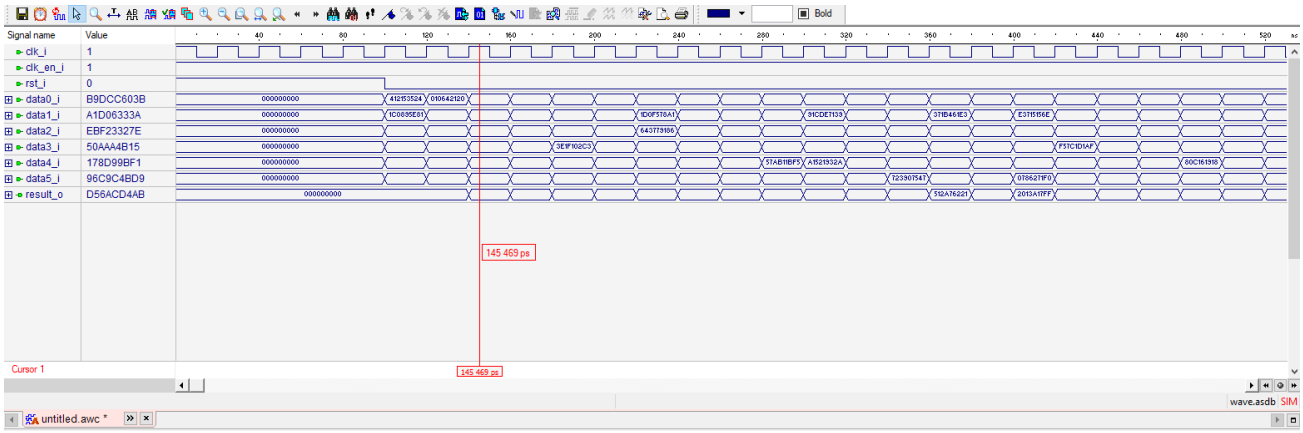

Figure 3.6. Simulation Waveform

Appendix A. Resource Utilization

Table A.1, and Table A.2 show the resource utilization of the Adder Tree Module for the LIFCL-40-9BG400I and LFCPNX-100-9LFG672I devices, using Lattice Synthesis Engine of the Lattice Radiant software. Default configuration is used and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.1. Resource Utilization (LIFCL)

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP (ACC54)
Default	200 MHz	0	1	3
Number of Inputs = 5, *Reset Mode* = Async, Others = Default	200 MHz	0	1	2
Enable Fully Pipelined Mode = Checked, *Width* = 10, Others = Default	200 MHz	40	1	3
Enable Input Register = Unchecked, *Enable Output Register* = Unchecked, *Width* = 10, Others = Default	N/A	0	1	3

***Note:** Fmax is generated when the FPGA design only contains Adder Tree Module and the target frequency is 160 MHz. These values may be reduced when user logic is added to the FPGA design.

Table A.2. Resource Utilization (LFCPNX)

Configuration	Clk Fmax (MHz)*	Registers	LUTs	DSP (ACC54)
Default	250 MHz	0	1	3
Number of Inputs = 5, *Reset Mode* = Async, Others = Default	250 MHz	0	1	2
Enable Fully Pipelined Mode = Checked, *Width* = 10, Others = Default	250 MHz	40	1	3
Enable Input Register = Unchecked, *Enable Output Register* = Unchecked, *Width* = 10, Others = Default	N/A	0	1	3

***Note:** Fmax is generated when the FPGA design only contains Adder Tree Module and the target frequency is 250 MHz. These values may be reduced when user logic is added to the FPGA design.

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 user guide.

Technical Support Assistance

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

Revision History

Document Revision 1.4, Lattice Radiant SW version 3.1, November 2021

Section	Change Summary
Appendix A. Resource Utilization	Added Table A.1. Resource Utilization (LIFCL) and updated Table A.2. Resource Utilization (LFCPNX).

Document Revision 1.3, Lattice Radiant SW version 3.0, June 2021

Section	Change Summary
Introduction	Remove Quick Facts section.
References	Updated this section.

Document Revision 1.2, Lattice Radiant SW version 2.3, December 2020

Section	Change Summary
Introduction	Update Table1.1 to add LFD2NX-17 as targeted device.
Functional Description	Updated dependency of *Number of Inputs* in Attribute Summary.
References	Updated this section.

Document Revision 1.1, Lattice Radiant SW version 2.1, June 2020

Section	Change Summary
Introduction	Updated Table 1.1. • Added Certus-NX support. • Updated Table 1.1 to add LFD2NX-40 as targeted device. • Updated Lattice Implementation to Lattice Radiant 2.1.
Functional Description	Updated dependency of *Number of Inputs* to support LFD2NX-40 in Attribute Summary.
Appendix A. Resource Utilization	Updated device to LIFCL-40-9BG400I.

Document Revision 1.0, Lattice Radiant SW version 2.0, February 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com