FFT Butterfly Module - Lattice Radiant Software

User Guide

FPGA-IPUG-02119-1.3

November 2023

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 REDUDANCIES, 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
IP	Intellectual Property
LSB	Least Significant Bit
MSB	Most Significant Bit
RTL	Register Transfer Level

1. Introduction

Fast Fourier Transform (FFT) is a popular algorithm with many applications, including signal and image processing. One important block in the FFT is the butterfly module. The butterfly module breaks down a large DFT into smaller DFTs, or combines the results of smaller DFTs into a larger DFT. It is called a butterfly diagram because of the butterfly-like shape of the data flow in its simplest form.

1.1. Features

The key features of FFT Butterfly module include:

- Supports the following modes:
  - Radix-2 Decimation-in-Time
  - Radix-2 Decimation-in-Frequency
  - Radix-4 Decimation-in-Time
  - Radix-4 Decimation-in-Frequency
- Configurable input data and twiddle factor width
- Supports pipelining
- Supports complex numbers
- Supports LUT and DSP implementation of multipliers

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 famous FFT butterfly module has two popular modes: Radix-2 and Radix-4. Butterfly operation is performed on two inputs for Radix-2 and four inputs for Radix-4, respectively.

The placement of a multiplying factor, called the twiddle factor, describes the type of decimation of the butterfly module. For decimation-in-time, the twiddle factor is pre-multiplied, or before any operation is perform on the inputs. On the other hand, for decimation-in-frequency, the twiddle factor is post-multiplied, or after operation on the inputs.

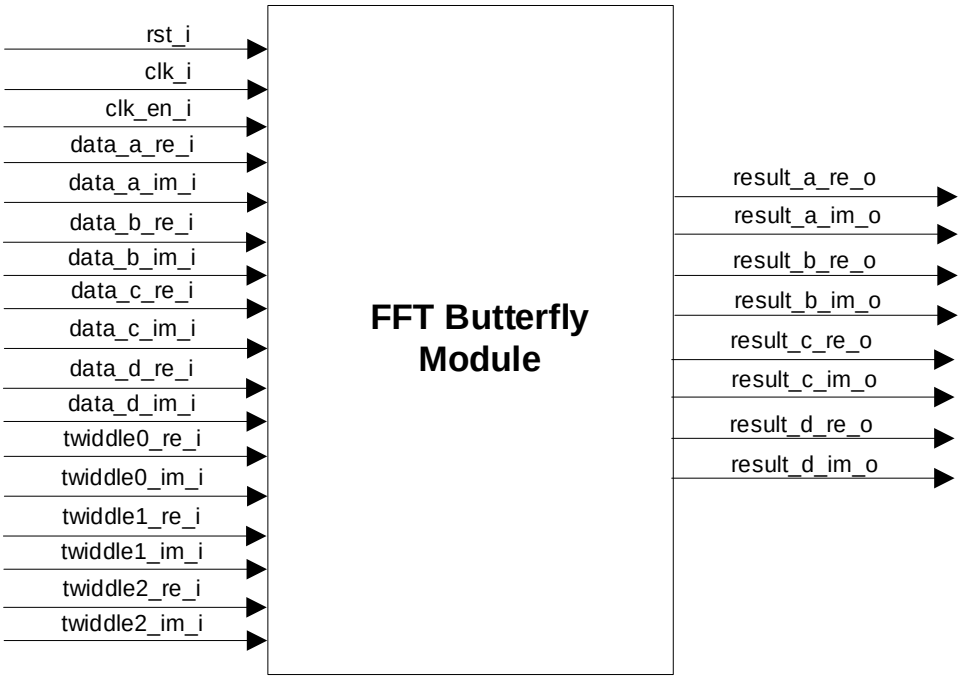

Figure 2.1. FFT Butterfly Top-Level Block Diagram

The following section describes the different modes of the butterfly module, namely:

- Radix-2 Decimation-in-Time
- Radix-2 Decimation-in-Frequency
- Radix-4 Decimation-in-Time
- Radix-4 Decimation-in-Frequency

2.1. Radix-2 Decimation-in-Time

A Radix-2 DIT butterfly module is a 2-input butterfly with the twiddle factor multiplied to the input before operation.

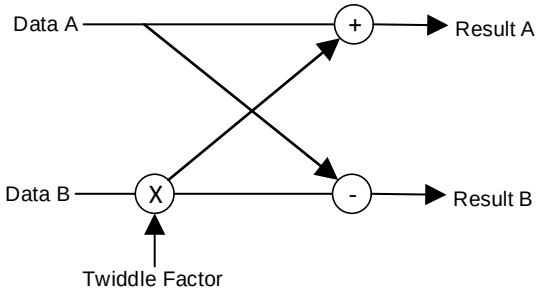

Figure 2.2. Basic Radix-2 DIT Butterfly Computation

The general formula for computing the results of Radix-2 DIT are as follow:

$$Result\ A = A + W \cdot B$$
$$Result\ B = A - W \cdot B$$

2.2. Radix-2 Decimation-in-Frequency

A Radix-2 DIF butterfly module is a 2-input butterfly unit with the twiddle factor multiplied to the result of the operation.

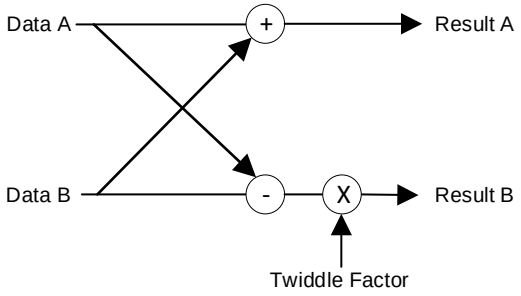

Figure 2.3. Basic Radix-2 DIF Butterfly Computation

The general formula for computing the results of Radix-2 DIF are as follow:

$$Result\ A = A + B$$
$$Result\ B = W \cdot (A - B)$$

2.3. Radix-4 Decimation-in-Time

A Radix-4 DIT is a 4-input butterfly unit with the twiddle factors multiplied to the inputs before operation.

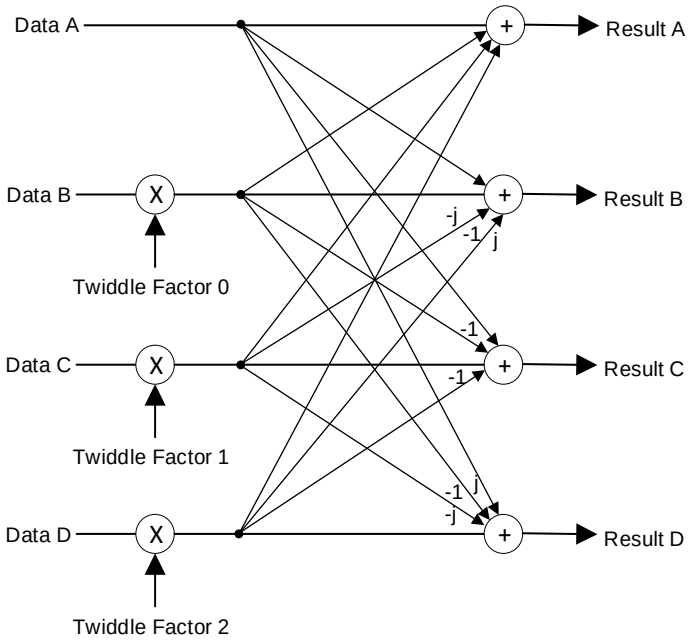

Figure 2.4. Basic Radix-4 DIT Butterfly Computation

The general formula for computing the results of Radix-4 DIT are as follow:

$$Result\ A = A + \ (B \cdot W0) + (C \cdot W1) \ + \ \ (D \cdot W2)$$
$$Result\ B = A - j(B \cdot W0) - (\ C \cdot W1) + j(D \cdot W2)$$
$$Result\ C = A - \ \ (B \cdot W0) + (\ C \cdot W1) - \ \ (D \cdot W2)$$
$$Result\ D = A + j(B \cdot W0) - (\ C \cdot W1) - j(D \cdot W2)$$

2.4. Radix-4 Decimation-in-Frequency

Figure 2.5 shows the data flow for Radix-4 DIF with the twiddle factors multiplied to the results of the operation.

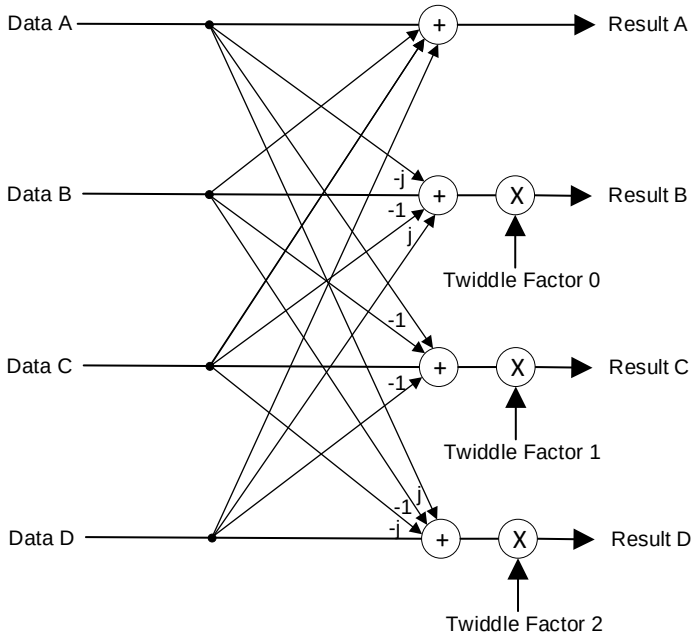

Figure 2.5. Basic Radix-4 DIF Butterfly Computation

The general formula for computing the results of radix-4 DIF are as follow:

$$Result\ A = A + B + C + D$$
$$Result\ B = (A - jB - C + jD) \cdot W0$$
$$Result\ C = (A - \ B + C - \ D) \cdot W1$$
$$Result\ D = (A + jB - C - jD) \cdot W2$$

2.5. Signal Descriptions

Table 2.1. FFT Butterfly Module Signal Description

Port Name	I/O	Width	Description
Clock and Reset Ports			
clk_i	In	1	System clock input.
rst_i	In	1	Reset input.
User Interface Ports			
clk_en_i	In	1	Clock enable input.
data_a_re_i	In	*Input Width*	Data A real input
data_b_re_i	In	*Input Width*	Data B real input
data_c_re_i[2]	In	*Input Width*	Data C real input
data_d_re_i[2]	In	*Input Width*	Data D real input
data_a_im_i[1]	In	*Input Width*	Data A imaginary input
data_b_im_i[1]	In	*Input Width*	Data B imaginary input
data_c_im_i[1,2]	In	*Input Width*	Data C imaginary input
data_d_im_i[1,2]	In	*Input Width*	Data D imaginary input
twiddle0_re_i	In	*Twiddle Factor Width*	Twiddle Factor 0 real input
twiddle1_re_i[2]	In	*Twiddle Factor Width*	Twiddle Factor 1 real input
twiddle2_re_i[2]	In	*Twiddle Factor Width*	Twiddle Factor 2 real input
twiddle0_im_i[1]	In	*Twiddle Factor Width*	Twiddle Factor 0 imaginary input
twiddle1_im_i[1,2]	In	*Twiddle Factor Width*	Twiddle Factor 1 imaginary input
twiddle2_im_i[1,2]	In	*Twiddle Factor Width*	Twiddle Factor 2 imaginary input
result_a_re_o	Out	*Output Width*[4]	Result A real part
result_b_re_o	Out	*Output Width*[4]	Result B real part
result_c_re_o[2]	Out	*Output Width*[4]	Result C real part
result_d_re_o[2]	Out	*Output Width*[4]	Result D real part
result_a_im_o[1]	Out	*Output Width*[4]	Result A imaginary part
result_b_im_o[1]	Out	*Output Width*[4]	Result B imaginary part
result_c_im_o[1,2]	Out	*Output Width*[4]	Result C imaginary part
result_d_im_o[1,2]	Out	*Output Width*[4]	Result D imaginary part

Notes:

1. Available only when *Complex Inputs* is enabled. If disabled, the imaginary part outputs are always zero.
2. Available only when *Mode*=”Radix-4 DIT” or ”Radix-4 DIF”.
3. The bit width of some signals is set by the attribute. Refer to Table 2.2 for the description of these attributes.
   a. If *Mode*=”Radix-2 DIT” or ”Radix-2 DIF”, then *Output Width* = *Input Width* + *Twiddle Factor Width*;
   b. If *Mode*=”Radix-4 DIT” or ”Radix-4 DIF”, then *Output Width* = *Input Width* + *Twiddle Factor Width*+ 1.

2.6. Attribute Summary

The configurable attributes of the FFT Butterfly 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			
FFT Butterfly Mode	Radix-2 DIT Radix-2 DIF Radix-4 DIT Radix-4 DIF	Radix-2 DIT	—
Implementation	LUT	LUT	—
Complex Inputs	True, False	True	—
Input Width	2-32	8	—
Twiddle Factor Width	2-32	8	—
Output Width	-	-	For Radix-2 DIT/DIF, OUTPUT_WIDTH = INPUT_WIDTH + TWDDL WIDTH. For Radix-4 DIT/DIF, OUTPUT_WIDTH = INPUT_WIDTH + TWDDL_WIDTH + 1.
Pipelines	0-N	1	For LUT, N = INPUT_WIDTH+1 or TWDDL_WIDTH+1, whichever is smaller. For DSP, N = 4.
Enable Input Registers	True, False	True	—
Enable Output Registers	True, False	True	—

Table 2.3. Attributes Descriptions

Attribute	Description
Configuration	
Mode	Specifies the mode of operation of the butterfly module. The default mode is *rad2_dit*. Radix-2 DIT performs 2-input butterfly operation and a twiddle factor is pre-multiplied on the inputs before any operation.
Implementation	Specifies if multiplier blocks are implemented as LUTs or the device's DSP blocks. DSP blocks perform multiplication faster than LUT blocks. The default implementation is *LUT*.
Complex Inputs	Specifies if the input data and twiddle factors are in complex form
Input Data Width	Specifies the width of the input data.
Twiddle Factor Width	Specifies the width of the twiddle factors.
Output Width	Specifies the width of the output data.
Pipelines	Specifies the number of pipelines to implement
Enable Input Registers	Enables the input registers of the butterfly module.
Enable Output Registers	Enables the output registers of the butterfly module.

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, 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 FFT Butterfly module in Lattice Radiant software is described below.

To generate the FFT Butterfly module:

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

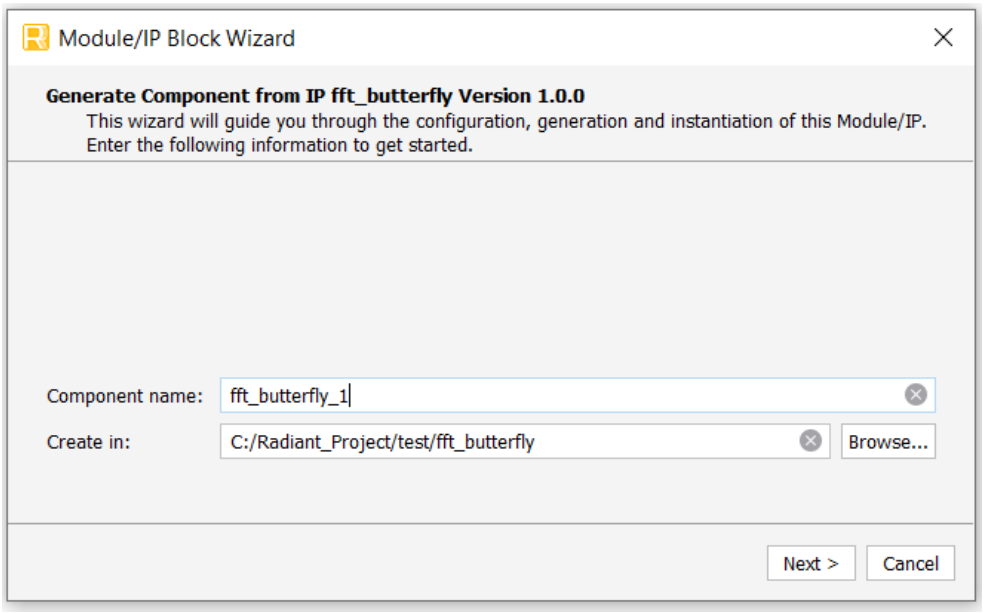

Figure 3.1. Module/IP Block Wizard

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

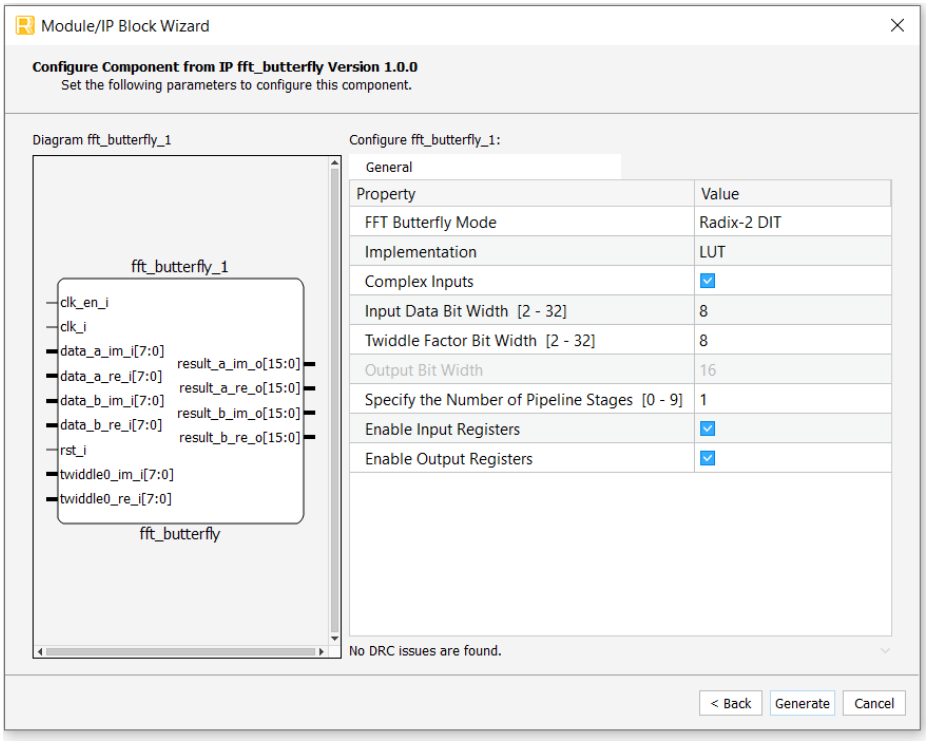

Figure 3.2. Configure User Interface of FFT Butterfly module

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

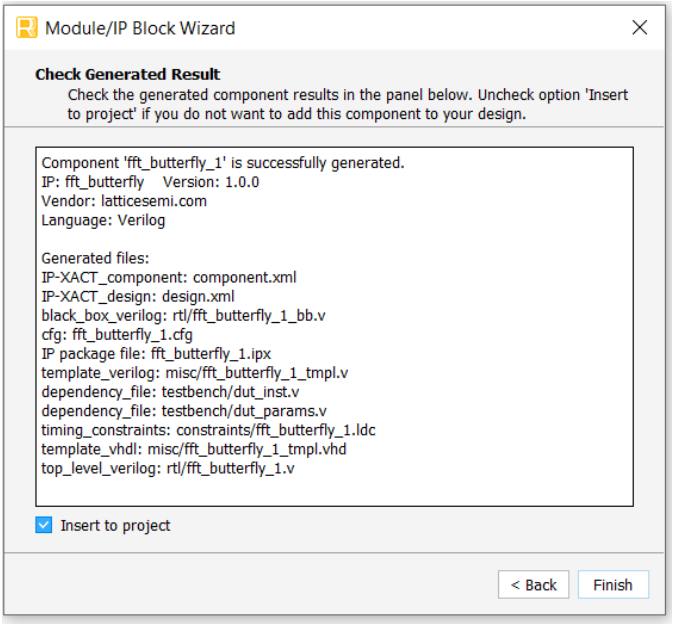

Figure 3.3. Check Generating Result

6. 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 FFT Butterfly 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.
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 Constraining the IP section 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:

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

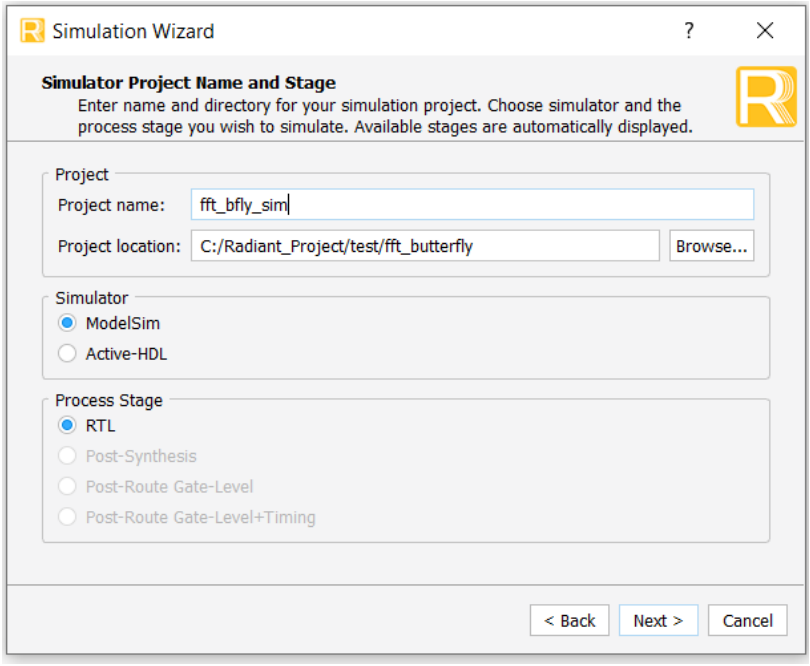

Figure 3.4. Simulation Wizard

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

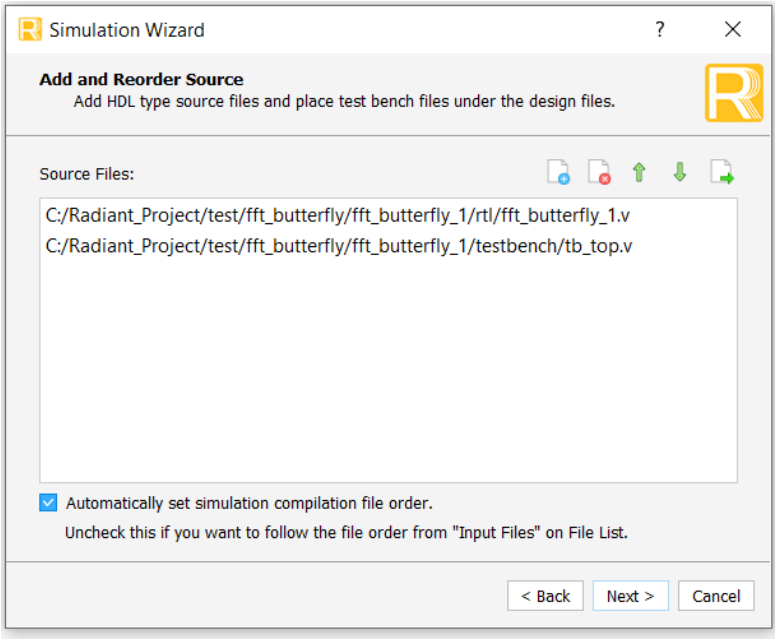

Figure 3.5. Adding and Reordering Source

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

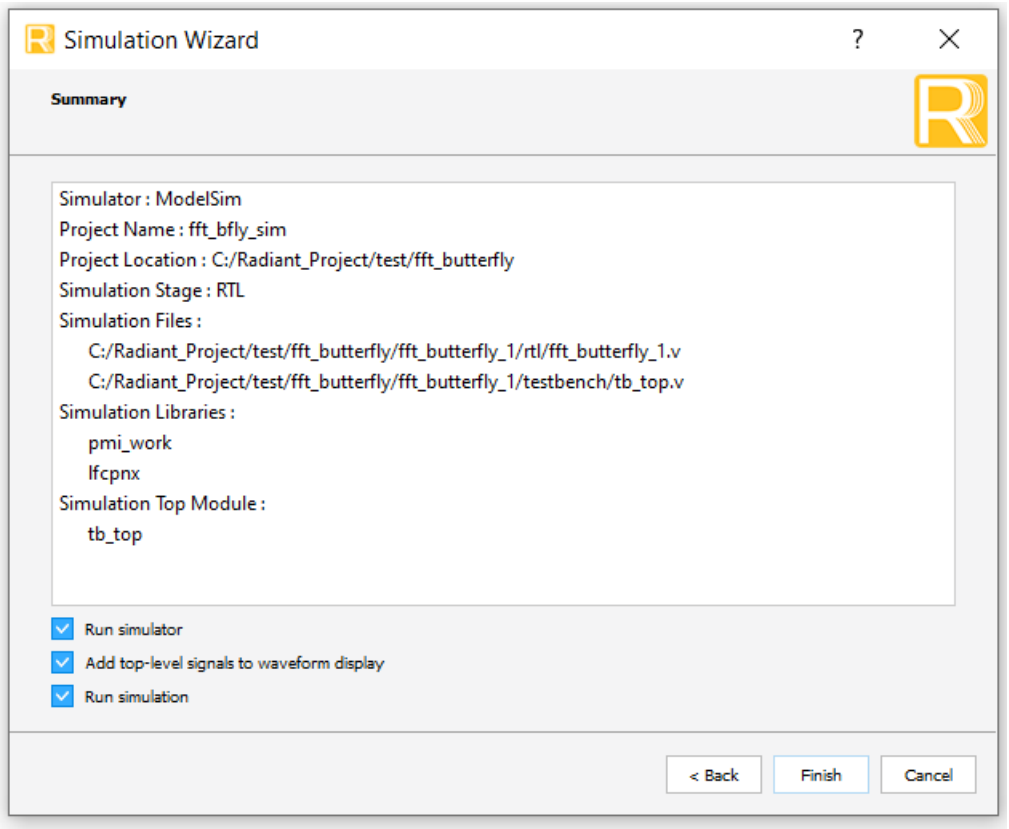

Figure 3.6. Summary WIndow

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.

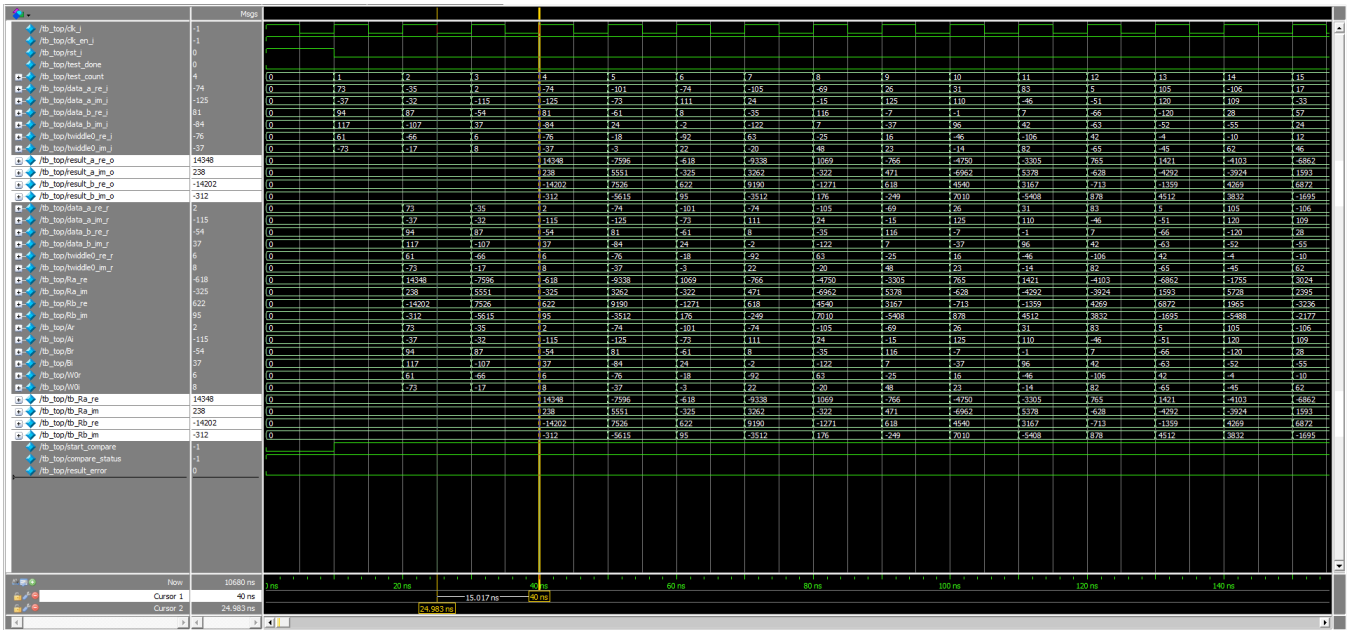

Figure 3.7. Simulation Waveform

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: <Instance_Path>/<Instance_Name>/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 provided you understand the effect of each constraint.

To use this constraint file:

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

Refer to Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details on how to constrain your design.

Appendix A. Resource Utilization

Table A.1 and Table A.2 shows the resource utilization of the FFT Butterfly module using Lattice Synthesis Engine of Lattice Radiant software. Default configuration is used, unless specified in the table. Not all configurations are shown.

Table A.1. Resource Utilization (LIFCL)

Implementation	Mode	Clk Fmax (MHz)	Registers	LUTs	EBRs	DSP Multiplier (18×18)
PIPELINES=1 (Default)						
LUT	rad2_dit	110.351	160	749	0	0
	rad2_dif	121.477	167	766	0	0
	rad4_dit	102.533	366	2443	0	0
	rad4_dif	105.363	378	2675	0	0
DSP	rad2_dit	129.735	144	154	0	1.5
	rad2_dif	120.366	143	123	0	2
	rad4_dit	114.732	318	607	0	4.5
	rad4_dif	110.546	378	496	0	9
PIPELINES=4						
LUT	rad2_dit	126.791	352	738	0	0
	rad2_dif	136.705	301	802	0	0
	rad4_dit	99.423	774	2445	0	0
	rad4_dif	119.489	811	2616	0	0
DSP	rad2_dit	200.000	336	136	0	1.5
	rad2_dif	200.000	260	160	0	2
	rad4_dit	121.095	323	951	0	4.5
	rad4_dif	167.168	760	442	0	9

Notes:

1. Performance and device utilization target an LIFCL-40-9BG400I device using Lattice Radiant 3.1 and Lattice Synthesis Engine software.
2. Fmax is generated when the FPGA design only contains the FFT butterfly module and the target frequency is 200 MHz.

The *distributed RAM* utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distribution among logic, distributed RAM, and ripple logic.

Table A.2. Resource Utilization (LFCPNX)

Implementation	Mode	Clk Fmax (MHz)	Registers	LUTs	EBRs	DSP Multiplier (18×18)
PIPELINES=1 (Default)						
LUT	rad2_dit	107.204	160	749	0	0
	rad2_dif	123.092	167	766	0	0
	rad4_dit	99.039	366	2370	0	0
	rad4_dif	101.368	378	2602	0	0
DSP	rad2_dit	123.138	144	154	0	1.5
	rad2_dif	122.895	143	123	0	2
	rad4_dit	114.142	318	534	0	4.5
	rad4_dif	107.342	378	423	0	9
PIPELINES=4						
LUT	rad2_dit	130.599	352	738	0	0
	rad2_dif	138.504	301	802	0	0
	rad4_dit	99.098	774	2372	0	0
	rad4_dif	120.120	811	2543	0	0
DSP	rad2_dit	194.250	336	136	0	1.5
	rad2_dif	250.000	260	160	0	2
	rad4_dit	118.357	323	878	0	4.5
	rad4_dif	156.740	760	369	0	9

Notes:

1. Performance and device utilization target an LFCPNX-100-9LFG672I device using Lattice Radiant 3.1 and Lattice Synthesis Engine software.
2. Fmax is generated when the FPGA design only contains the FFT butterfly module and the target frequency is 250 MHz.

The *distributed RAM* utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distribution among logic, distributed RAM, and ripple logic.

Table A.3. Resource Utilization (LAV-AT)

Implementation	Mode	Clk Fmax (MHz)	Registers	LUTs	EBRs	DSP Multiplier (18×18)
PIPELINES=1 (Default)						
LUT	rad2_dit	153.468	176	432	0	0
	rad2_dif	134.012	170	416	0	0
	rad4_dit	90.342	465	1504	0	0
	rad4_dif	84.097	414	1388	0	0
DSP	rad2_dit	181.422	144	132	0	2
	rad2_dif	111.645	166	284	0	1
	rad4_dit	108.909	315	507	0	5
	rad4_dif	78.192	394	765	0	5
PIPELINES=4						
LUT	rad2_dit	210.526	587	563	0	0
	rad2_dif	239.636	317	456	0	0
	rad4_dit	130.225	1047	1580	0	0
	rad4_dif	184.332	1051	1393	0	0
DSP	rad2_dit	176.491	336	122	0	2
	rad2_dif	224.820	273	338	0	1
	rad4_dit	150.625	723	507	0	5
	rad4_dif	147.189	760	814	0	5

Notes:

1. Performance and device utilization target an LAV-AT-E70-1LFG676C device using Lattice Radiant 2023.1 and Lattice Synthesis Engine software.
2. Fmax is generated when the FPGA design only contains the FFT butterfly module and the target frequency is 100 MHz.

The *distributed RAM utilization* is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distribution among logic, distributed RAM, and ripple logic.

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.

- Lattice Radiant Software FPGA webpage
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- 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, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.3 November 2023

Section	Change Summary
Disclaimers	Updated this section.
IP Generation, Synthesis, and Validation	• Added testbench/dut_inst.v, testbench/dut_params.v, and eval/constraint.pdc attributes. • Added Constraining the IP section.
Appendix A. Resource Utilization	Added Table A.3. Resource Utilization (LAV-AT).
References	Added web page links for Lattice Radiant software, Lattice Radiant Timing Constraints Methodology, and Lattice Insights.
Technical Support Assistance	Added link for Lattice Answer Database.

Revision 1.2, December 2021

Section	Change Summary
Appendix A. Resource Utilization	Updated section content to add table for LFCPNX.

Revision 1.1, June 2021

Section	Change Summary
References	Updated reference to Lattice Radiant software user guide.

Revision 1.0, December 2020

Section	Change Summary
All	Initial release

www.latticesemi.com