



Lattice Avant SDR Module

IP Version: v2.1.0

User Guide

FPGA-IPUG-02187-1.3

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
2. Functional Description.....	7
2.1. Overview	7
2.2. Functional Diagrams.....	8
2.2.1. GIREG_RX.SCLK.....	8
2.2.2. GOREG_TX.SCLK	11
2.3. Signal Description.....	15
2.4. Attribute Summary.....	16
3. IP Generation and Evaluation	18
3.1. Licensing the IP.....	18
3.2. Generation and Synthesis	18
3.3. Running Functional Simulation	22
3.4. Constraining the IP	23
3.5. Hardware Evaluation.....	23
Appendix A. Resource Utilization	24
References	25
Technical Support Assistance	26
Revision History	27

Figures

Figure 2.1. SDR I/O Module Top Level Block Diagram	7
Figure 2.2. GIREG_RX.SCLK Bypass/Static Default/Static User-Defined Delay Block Diagram	8
Figure 2.3. GIREG_RX.SCLK Bypass/Static Default/Static User-Defined Delay Inversion Enabled Block Diagram	9
Figure 2.4. GIREG_RX.SCLK Dynamic Default/Dynamic User-Defined Delay Block Diagram	10
Figure 2.5. GOREG_TX.SCLK Bypass/Static User-Defined Delay Block Diagram	11
Figure 2.6. GOREG_TX.SCLK Dynamic User-Defined Delay Block Diagram	12
Figure 2.7. GOREG_TX.SCLK Bypass/Static User Defined Delay Tristate Control Enabled Block Diagram	13
Figure 2.8. GOREG_TX.SCLK Dynamic User Defined Delay Tristate Control Enabled Block Diagram	14
Figure 3.1. Module/IP Block Wizard	18
Figure 3.2. Configure Block of SDR I/O Module	19
Figure 3.3. Check Generated Result	20
Figure 3.4. Simulation Wizard	22
Figure 3.5. Adding and Reordering Source	23
Figure 3.6. Simulation Waveform	23

Tables

Table 2.1. SDR I/O Module Interfaces	7
Table 2.2. SDR I/O Module Receive Signal Description	15
Table 2.3. SDR I/O Module Transmit Signal Description	15
Table 2.4. Attributes Summary	16
Table 2.5. Attributes Description	17
Table 3.1. Generated File List	21
Table A.1. Resource Utilization	24

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
RTL	Register Transfer Level
SDR	Single Data Rate

1. Introduction

The Lattice Semiconductor Single Data Rate Input/Output (SDR I/O) Module is designed to be used in a wide range of applications in which fast data transmission is required.

1.1. Features

The key features of Single Data Rate Input/Output (SDR I/O) Module include:

- Receive and Transmit Interface of up to 300 Mbps depending on I/O type
- Selectable I/O type
 - Single-ended or Differential Signaling
- 1-bit to 256-bit data bus width
- 1 MHz to 300 MHz clock frequency depending on I/O type
- Data Path Delay that includes following options:
 - Bypass
 - Static Default (Receive Interface only)
 - Dynamic Default (Receive Interface only)
 - Static User-Defined
 - Dynamic User-Defined
- Tristate control (Transmit Interface only)
- Clock inversion (Receive Interface only)

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` is active low
- `_i` are input signals
- `_o` are output signals
- `_io` are bi-directional input/output signals

2. Functional Description

2.1. Overview

Table 2.1. SDR I/O Module Interfaces

Feature	Description	Comments
GIREG_RX.SCLK	SDR Receive Interface	Supports bypassed, static, and dynamic data path delay. Supports clock inversion logic
GOREG_TX.SCLK	SDR Transmit Interface	Supports bypassed, static, and dynamic data path delay. Supports tristate control.

Notes:

- G – Generic
- IREG – SDR input I/O register
- OREG – SDR output I/O register
- _RX – Receive interface
- _TX – Transmit interface
- .SCLK – Uses SCLK (primary clock) clocking resource

SDR GIREG_RX.SCLK interface is used when a simple input register is required for the design. The clock input to the input register can be optionally inverted if required. These interfaces always use SCLK.

SDR GOREG_TX.SCLK interface is used for an SDR data output implementation with tight specifications on clock out to data out skew. The same clock is used for both data and clock generation.

Single Data Rate applications capture data on one edge of a clock only.

Figure 2.1 presents a top-level block diagram of SDR I/O Module.

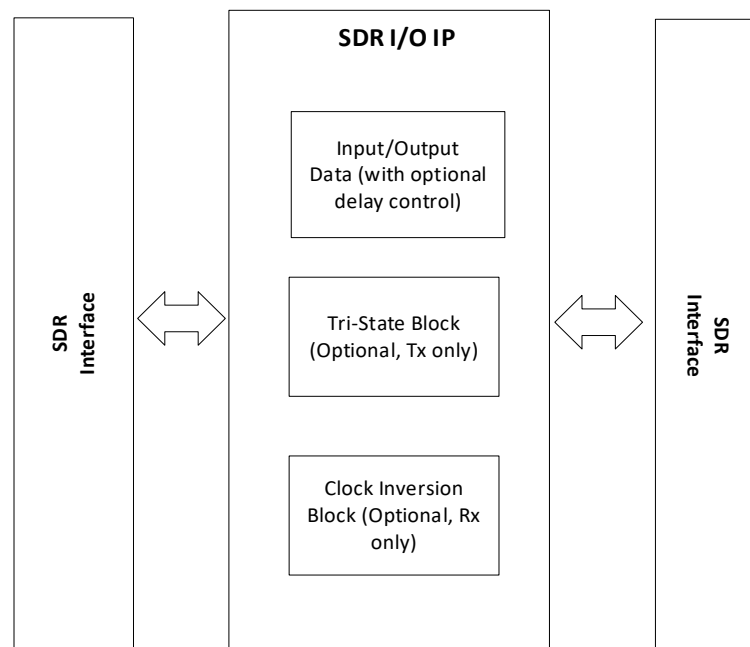


Figure 2.1. SDR I/O Module Top Level Block Diagram

2.2. Functional Diagrams

2.2.1. GIREG_RX.SCLK

Refer to the [Lattice Avant High-Speed I/O and External Memory Interface User Guide](#) for detailed information regarding these GIREG_RX.SCLK configurations.

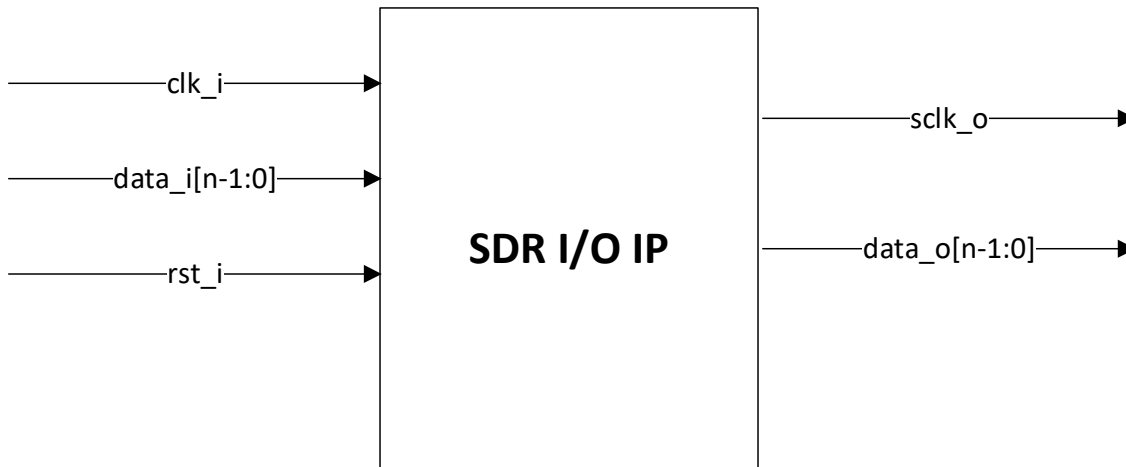


Figure 2.2. GIREG_RX.SCLK Bypass/Static Default/Static User-Defined Delay Block Diagram

Figure 2.2 shows the block diagram for the configurations described in [GIREG_RX.SCLK Bypass Interface](#) and [GIREG_RX.SCLK Static Default/Static User-Defined Delay Interface](#) sections.

2.2.1.1. GIREG_RX.SCLK Bypass Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*.
- Output sampling clock *sclk_o* is routed from *clk_i*.

2.2.1.2. GIREG_RX.SCLK Static Default/Static User-Defined Delay Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The input data signal is routed internally through a delay I/O module to remove clock injection time. The delay value may be equal to the default or user-defined value configured through the Module/IP Block Wizard.
- The delayed input signal then passes through a register that captures input data at the positive edge of *clk_i*.
- Output sampling clock *sclk_o* is routed from *clk_i*.

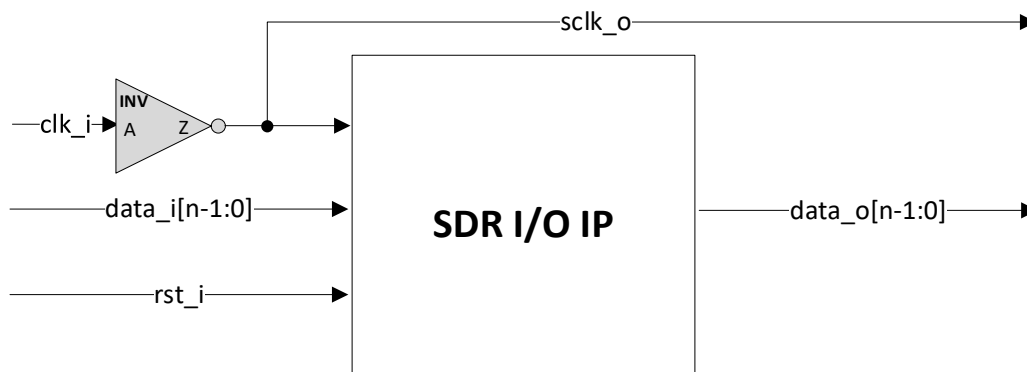


Figure 2.3. GIREG_RX.SCLK Bypass/Static Default/Static User-Defined Delay Inversion Enabled Block Diagram

Figure 2.3 shows the block diagram for the configurations described in [GIREG_RX.SCLK Bypass Inversion Enabled Interface](#) and [GIREG_RX.SCLK Static Default/Static User-Defined Delay Inversion Enabled Interface](#) sections.

2.2.1.3. GIREG_RX.SCLK Bypass Inversion Enabled Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*.
- Output sampling clock *sclk_o* is routed from inverted *clk_i*.

2.2.1.4. GIREG_RX.SCLK Static Default/Static User-Defined Delay Inversion Enabled Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The input data signal is routed internally through a delay I/O module to remove clock injection time. The delay value may be equal to the default or user-defined value configured through the Module/IP Block Wizard.
- The delayed input signal then passes through a register that captures input data at the positive edge of *clk_i*.
- Output sampling clock *sclk_o* is routed from inverted *clk_i*.

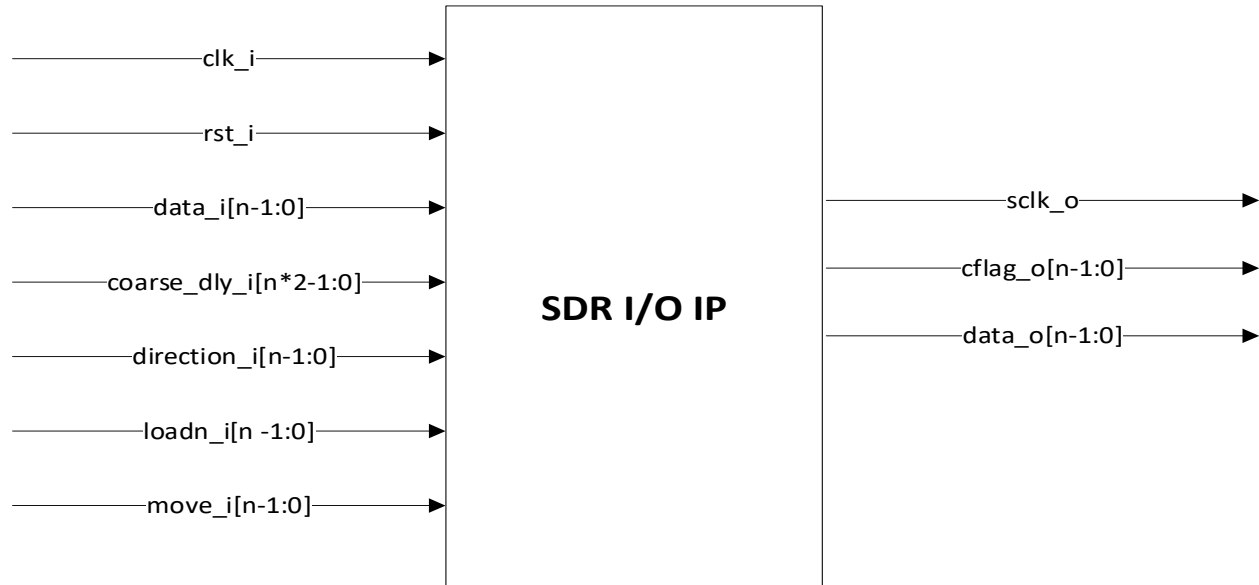


Figure 2.4. GIREG_RX.SCLK Dynamic Default/Dynamic User-Defined Delay Block Diagram

Figure 2.4 shows the block diagram for the configurations described in [GIREG_RX.SCLK Dynamic Default/Dynamic User-Defined Delay Interface](#) and [GIREG_RX.SCLK Dynamic Default/Dynamic User-Defined Delay Inversion Enabled Interface](#) sections.

2.2.1.5. GIREG_RX.SCLK Dynamic Default/Dynamic User-Defined Delay Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The input data signal is routed internally through a delay I/O module to remove clock injection time. The delay value may be equal to the default or user-defined value configured through the Module/IP Block Wizard. The delay value may also be dynamically changed by setting *loadn_i* input to reset the data path delay setting to default. The inputs *coarse_dly_i*, *direction_i*, and *move_i* are used to update the data path delay after resetting it to default. See [Table 2.2](#) for the description of these signals.
- The delayed input signal then passes through a register that captures data at the positive edge of *clk_i*.
- Output sampling clock *sclk_o* is routed inverted *clk_i*.

2.2.1.6. GIREG_RX.SCLK Dynamic Default/Dynamic User-Defined Delay Inversion Enabled Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The input data signal is routed internally through a delay I/O module to remove clock injection time. The delay value may be equal to the default or user-defined value configured through the Module/IP Block Wizard. The delay value may also be dynamically changed by setting *loadn_i* input to reset the data path delay setting to default. The inputs *coarse_dly_i*, *direction_i*, and *move_i* are used to update the data path delay after resetting it to default. See [Table 2.2](#) for the description of these signals.
- The delayed input signal then passes through a register that captures data at the positive edge of *clk_i*.
- Output sampling clock *sclk_o* is routed inverted *clk_i*.

2.2.1.7. Static and Dynamic Default Delay Settings for RTL

The SDR RX static and dynamic default configurations use 0 and 0NS fine and coarse delay settings respectively for RTL simulation. However, this does not produce default delay settings on silicon due to the unavoidable difference in behavior between the simulation model and the silicon. Therefore, different default fine and coarse delay settings derived from silicon characterization data are applied on DELAYB and DELAYA for static and dynamic default configuration respectively. Performed during the Radiant implementation flow, this ensures that the default delay is correctly produced in a silicon. This may prevent the post-routed netlist simulation from producing default delay and, thus, fail the simulation with the testbench that comes with the IP.

2.2.2. GOREG_TX.SCLK

Refer to the [Lattice Avant High-Speed I/O and External Memory Interface User Guide](#) for detailed information regarding these GOREG_RX.SCLK configurations.

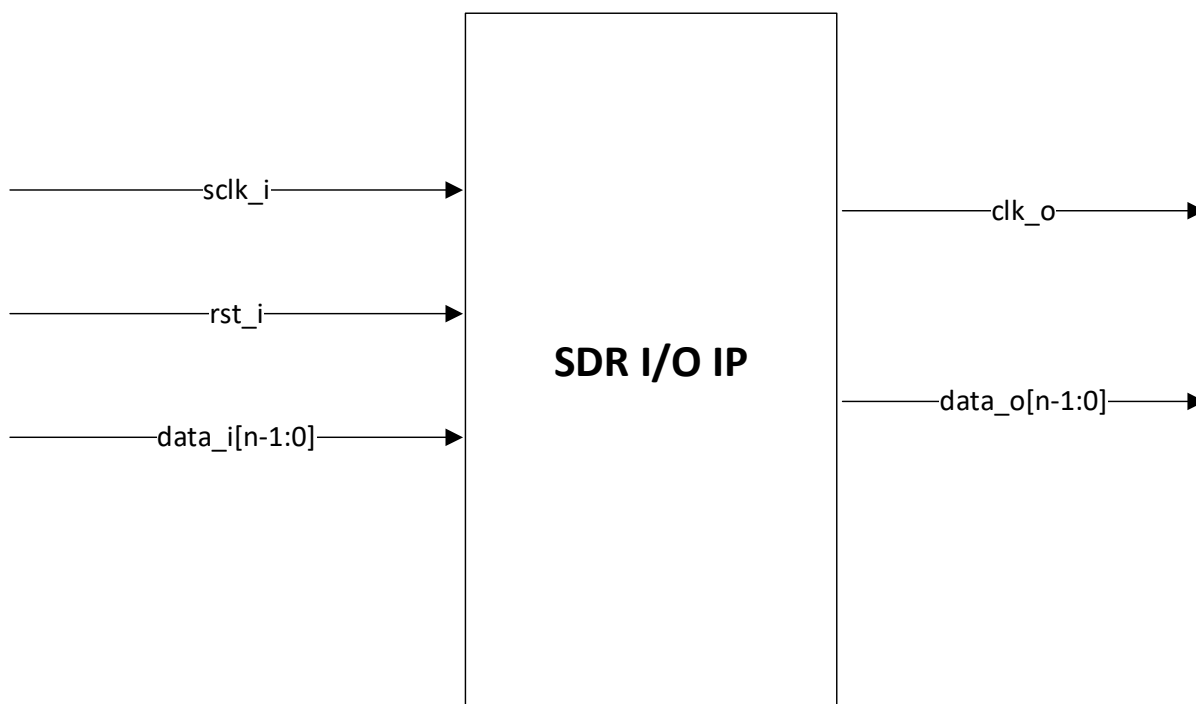


Figure 2.5. GOREG_TX.SCLK Bypass/Static User-Defined Delay Block Diagram

Figure 2.5 shows the block diagram for the configurations described in [GOREG_TX.SCLK Bypass Interface](#) and [GOREG_TX.SCLK Static User-Defined Delay Interface](#) sections.

2.2.2.1. GOREG_TX.SCLK Bypass Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i* and outputs it to *data_o*.
- Output sampling clock *clk_o* is internally routed through a generic X1 ODDR primitive from *sclk_i*.

2.2.2.2. GOREG_TX.SCLK Static User-Defined Delay Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*. The registered data is internally routed through a delay module before it is output to *data_o*. The delay value is equal to the user-defined value configured through the Module/IP Block Wizard.
- Output sampling clock *clk_o* is internally routed through a generic X1 ODDR primitive from *sclk_i*.

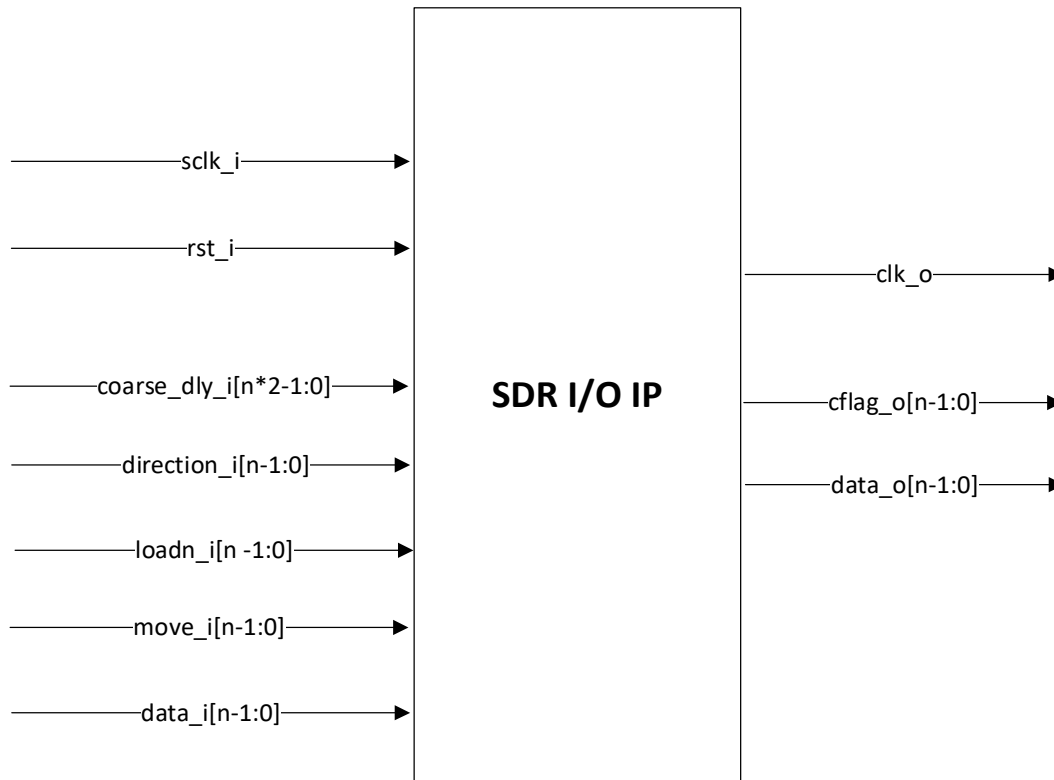


Figure 2.6. GOREG_TX.SCLK Dynamic User-Defined Delay Block Diagram

Figure 2.6 shows the block diagram for the configurations described in [GOREG_TX.SCLK Dynamic User-Defined Delay Interface](#) section.

2.2.2.3. GOREG_TX.SCLK Dynamic User-Defined Delay Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*. The registered data is internally routed through a delay module before it is output to *data_o*. The delay value may be equal to the default or user-defined value configured through the Module/IP Block Wizard. The delay value may also be dynamically changed by setting *loadn_i* input to reset the data path delay setting to default. The inputs *coarse_dly_i*, *direction_i*, and *move_i* are used to update the data path delay after resetting it to default. Refer [Table 2.3](#) for the description of these signals.
- Output sampling clock *clk_o* is internally routed through a generic X1 ODDR primitive from *sclk_i*.

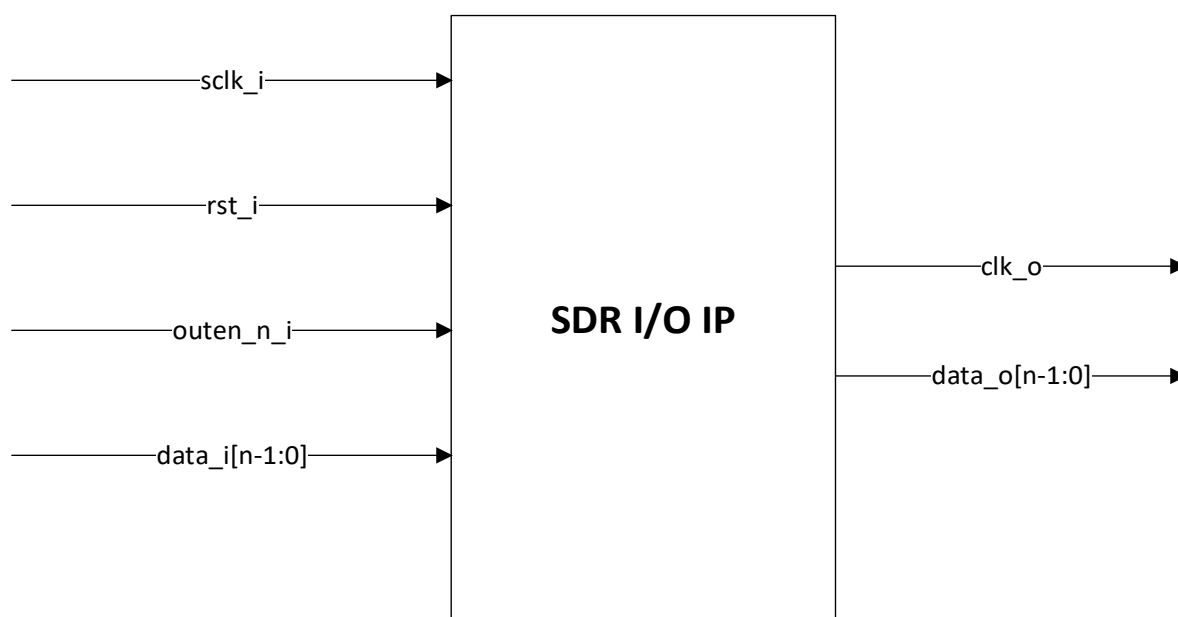


Figure 2.7. GOREG_TX.SCLK Bypass/Static User Defined Delay Tristate Control Enabled Block Diagram

Figure 2.7 shows the block diagram for the configurations described in [GOREG_TX.SCLK Bypass Tristate Control Enabled Interface](#) and [GOREG_TX.SCLK Static User Defined Delay Tristate Control Enabled Interface](#) sections.

2.2.2.4. GOREG_TX.SCLK Bypass Tristate Control Enabled Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*. This signal is input to the tristate buffer for *data_o*.
- Input to tristate buffer for *clk_o* output is internally routed through a generic X1 ODDR primitive from *sclk_i*.
- *data_o* and *clk_o* are both output from a tristate buffer. When *outen_n_i* is set to Low, data is passed to both outputs. When *outen_n_i* is set to High, both outputs are Hi-Z.

2.2.2.5. GOREG_TX.SCLK Static User Defined Delay Tristate Control Enabled Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*. The registered data is internally routed through a delay module before it is output to *data_o*. The delay value is equal to user-defined value configured through the Module/IP Block Wizard. The delayed signal is input to the tristate buffer for *data_o*.
- Input to tristate buffer for *clk_o* output is internally routed through a generic X1 ODDR primitive from *sclk_i*.
- *data_o* and *clk_o* are both output from a tristate buffer. When *outen_n_i* is set to Low, data is passed to both outputs. When *outen_n_i* is set to High, both outputs are Hi-Z.

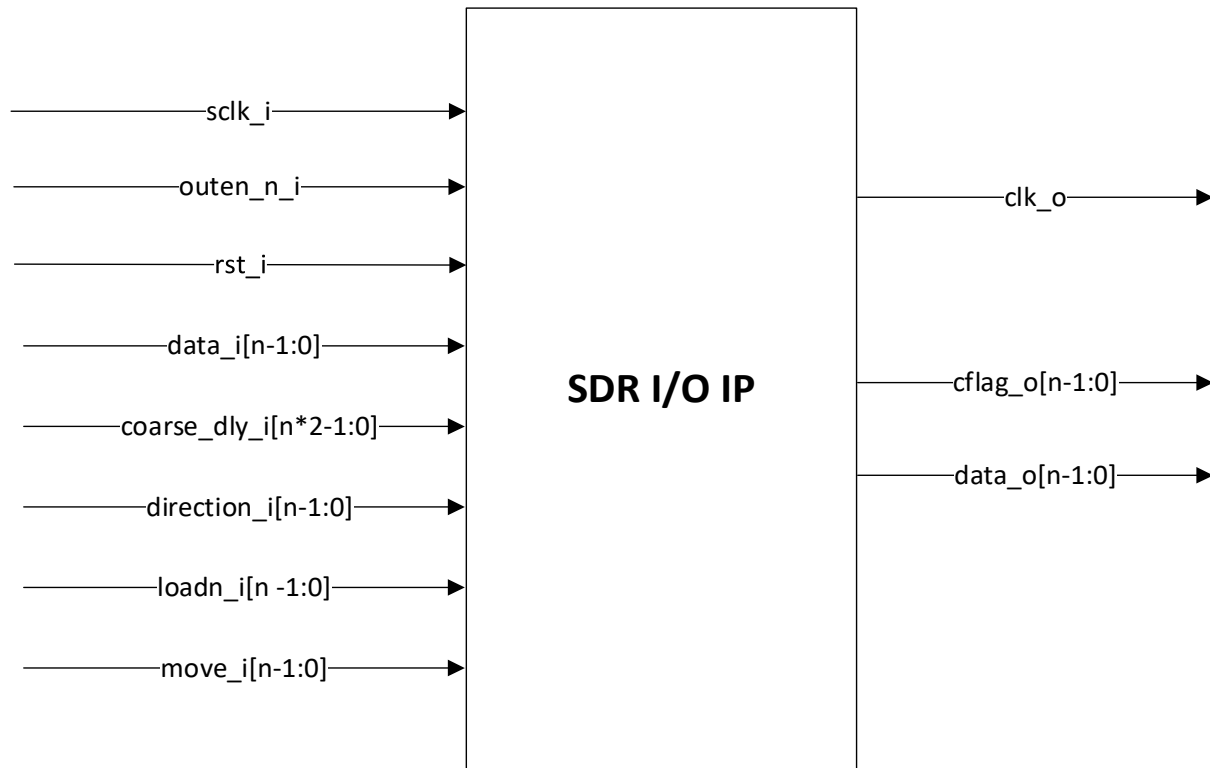


Figure 2.8. GOREG_TX.SCLK Dynamic User Defined Delay Tristate Control Enabled Block Diagram

Figure 2.8 shows the block diagram for the configurations described in [GOREG_TX.SCLK Dynamic User Defined Delay Tristate Control Enabled Interface](#) section.

2.2.2.6. GOREG_TX.SCLK Dynamic User Defined Delay Tristate Control Enabled Interface

- Reset *rst_i* is active High asynchronous reset. Change from High to Low to deassert.
- The data signal passes through a register that captures input data at the positive edge of *clk_i*. The registered data is internally routed through a delay module before it is output to *data_o*. The delay value may be equal to the default or user-defined value configured through the Module/IP Block Wizard. The delay value may also be dynamically changed by setting *loadn_i* input to reset the data path delay setting to default. The inputs *coarse_dly_i*, *direction_i*, and *move_i* are used to update the data path delay after resetting it to default. Refer [Table 2.3](#) for the description of these signals. The delayed signal is input to the tristate buffer for *data_o*.
- Input to tristate buffer for *clk_o* output is internally routed through a generic X1 ODDR primitive from *sclk_i*.
- *data_o* and *clk_o* are both output from a tristate buffer. When *outen_n_i* is set to Low, data is passed to both outputs. When *outen_n_i* is set to High, both outputs are Hi-Z.

2.3. Signal Description

Table 2.2. SDR I/O Module Receive Signal Description

Port Name	Direction	Description
Clocks and Reset		
rst_i	In	Active high asynchronous reset signal.
sclk_o	Out	Received data sampling clock
User Interface		
data_o[n-1:0]	Out	Received input data to fabric.
loadn_i[n-1:0]	In	Active low signal to reset data path delay setting to default. Only available during dynamic default or dynamic user-defined data path delay.
move_i[n-1:0]	In	Increments or decrements data path delay setting depending on <i>direction_i</i> . Only available during dynamic default or dynamic user-defined data path delay.
direction_i[n-1:0]	In	1 to decrease and 0 to increase data path delay. This is only available during dynamic default or dynamic user-defined data path delay.
cflag_o[n-1:0]	Out	Underflow or overflow flag to indicate that minimum or maximum data path delay adjustment is reached. This is only available during dynamic default or dynamic user-defined data path delay.
I/O Pad Interface		
clk_i	In	Clock input signal from I/O.
data_i[n-1:0]	In	Data input signal from I/O.

Note: n = bus width.

Table 2.3. SDR I/O Module Transmit Signal Description

Port Name	Direction	Description
Clocks and Reset		
rst_i	In	Active high asynchronous reset signal.
sclk_i	In	Transmit data sampling clock.
User Interface		
data_i[n-1:0]	In	Transmit output data going to I/O.
outen_n_i	In	Active low signal output enable. Only available when Tri-State mode is enabled.
loadn_i[n-1:0]	In	Active low signal to reset data path delay setting to default. Only available during dynamic default or dynamic user-defined data path delay.
move_i[n-1:0]	In	Increments or decrements data path delay setting depending on <i>direction_i</i> . Only available during dynamic default or dynamic user-defined data path delay.
direction_i[n-1:0]	In	1 to decrease and 0 to increase data path delay. This is only available during dynamic default or dynamic user defined data path delay.
cflag_o[n-1:0]	Out	Underflow or overflow flag to indicate that minimum or maximum data path delay adjustment is reached. This is only available during dynamic default or dynamic user-defined data path delay.
I/O Pad Interface		
clk_o	Out	Clock output signal to I/O.
data_o[n-1:0]	Out	Data output signal to I/O.

Note: n = bus width.

2.4. Attribute Summary

Table 2.4 provides a list of user configurable attributes for the SDR I/O Module. Attribute settings are specified using SDR I/O Module Configuration user interface in Lattice Radiant.

Table 2.4. Attributes Summary

Attribute	Selectable Values	Default	Dependency on other Attributes
Interface Type	Receive, Transmit	Receive	—
I/O Standard for this Interface	LVDS, SSTL135D, HSUL12D, LVCMOS09, LVCMOS10, LVCMOS33, LVCMOS25, LVCMOS18 and LVCMOS12	LVDS	If clock frequency is greater than 200MHz, selectable values are only LVDS, HSUL12D, SSTL135D, LVCMOS09, LVCMOS10. If clock frequency is greater than 100MHz, selectable values are only LVDS, HSUL12D, SSTL135D, LVCMOS09, LVCMOS10, LVCMOS25, LVCMOS18 and LVCMOS33.
Bus Width for this Interface	1 – 256	8	—
Data Path Delay	Bypass, Static Default, Dynamic Default, Static User Defined, Dynamic User Defined	Bypass	<i>Static Default</i> and <i>Dynamic Default</i> are supported for <i>Interface Type == Receive</i>
Fine Delay Value for User Defined	0 – Maximum Fine Delay	0	Selectable only if <i>Data Path Delay == Static User Defined</i> or <i>Dynamic User Defined</i> . Maximum Fine Delay = $(1 / \text{Clock Frequency}) / 14\text{ps}$ Note: In silicon, targeted delay step is 12.5 ps. In functional simulation, delay step that can be observed is 10ps. Meanwhile 14ps is the observed delay step on timing characterization.
Enable Tristate Control	Checked, Not checked	Not checked	<i>Interface Type == Transmit</i>
Enable Clock Inversion	Checked, Not checked	Not checked	<i>Interface Type == Receive</i>
Clock Frequency for this Interface (MHz)	1 – 300	200	—
Bandwidth for this Interface (Mbits/s)	Calculated	1600	Clock Frequency × Bus Width. Display for information only.

Note: All attributes can be configured from the General tab of the Lattice Radiant Software user interface.

Table 2.5 shows the description of the attributes used on SDR I/O module.

Table 2.5. Attributes Description

Attribute Name	Description
Interface Type	Selects interface type as Receive or Transmit.
I/O Standard for this Interface	List of Single-ended or Differential I/O supported.
Bus Width for this Interface	Total number of lanes/bus width.
Data Path Delay	Selects among Bypass, Static Default, Static User Defined, Dynamic Default or Dynamic User Defined. Allows setting default data path fine delay value when selected as Static User Defined or Dynamic User Defined.
Fine Delay Value for User Defined	Default data path fine delay setting. This is only valid when Data Path Delay is Static User Defined or Dynamic User Defined. For STATIC mode, value overrides the default delay setting. For DYNAMIC mode, this is the initial value of the fine delay element. Ideally, a 9-bit binary string, 12.5 ps per step. $\text{max_fine_delay} = 511 \times 12.5 \text{ ps} = 6387.5\text{ps}$ Note: The silicon is designed to have 12.5ps delay per step but on simulation, only 10ps can be observed.
Enable Tristate Control	Tristate enabled during transmission.
Enable Clock Inversion	Clock Inversion enabled. If enabled, received data sampling clock signal is inversed.
Clock Frequency for this Interface (MHz)	Clock frequency to be used in selected interface.

3. IP Generation and Evaluation

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

3.1. Licensing the IP

No license is required for this module.

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 SDR I/O module in Lattice Radiant Software is described below.

To generate SDR I/O Module:

1. Create a new Lattice Radiant Software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **SDR** under **Module, Architecture_Modules, IO** 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**.

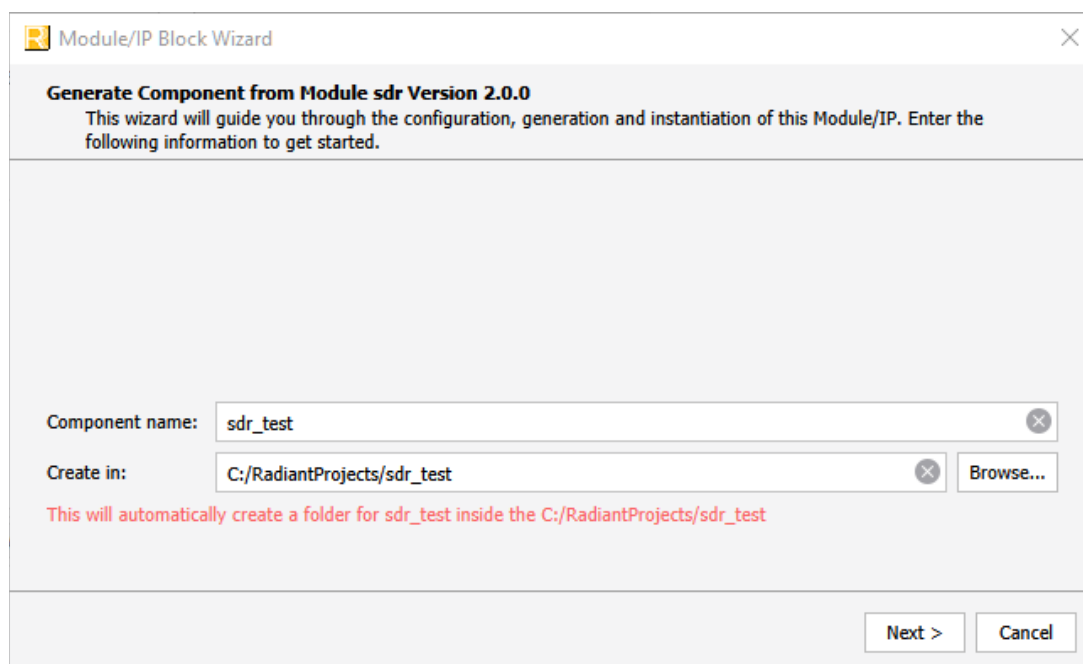


Figure 3.1. Module/IP Block Wizard

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

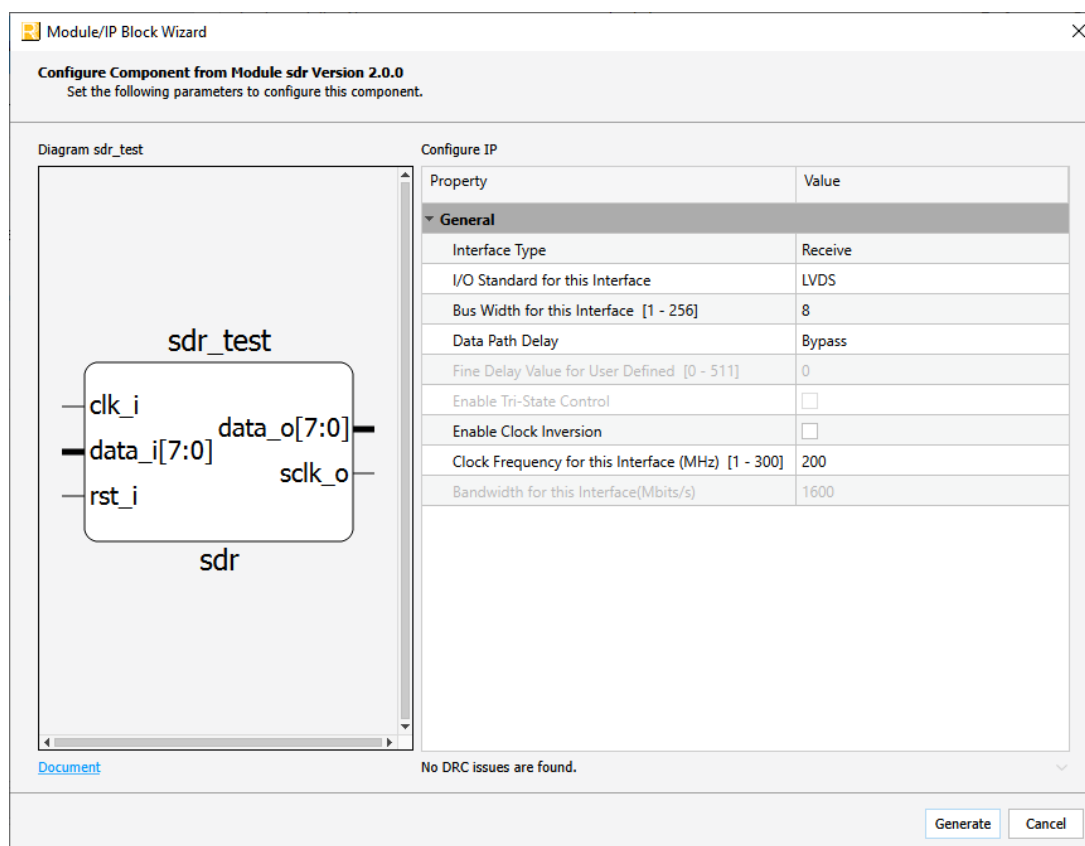


Figure 3.2. Configure Block of SDR I/O Module

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

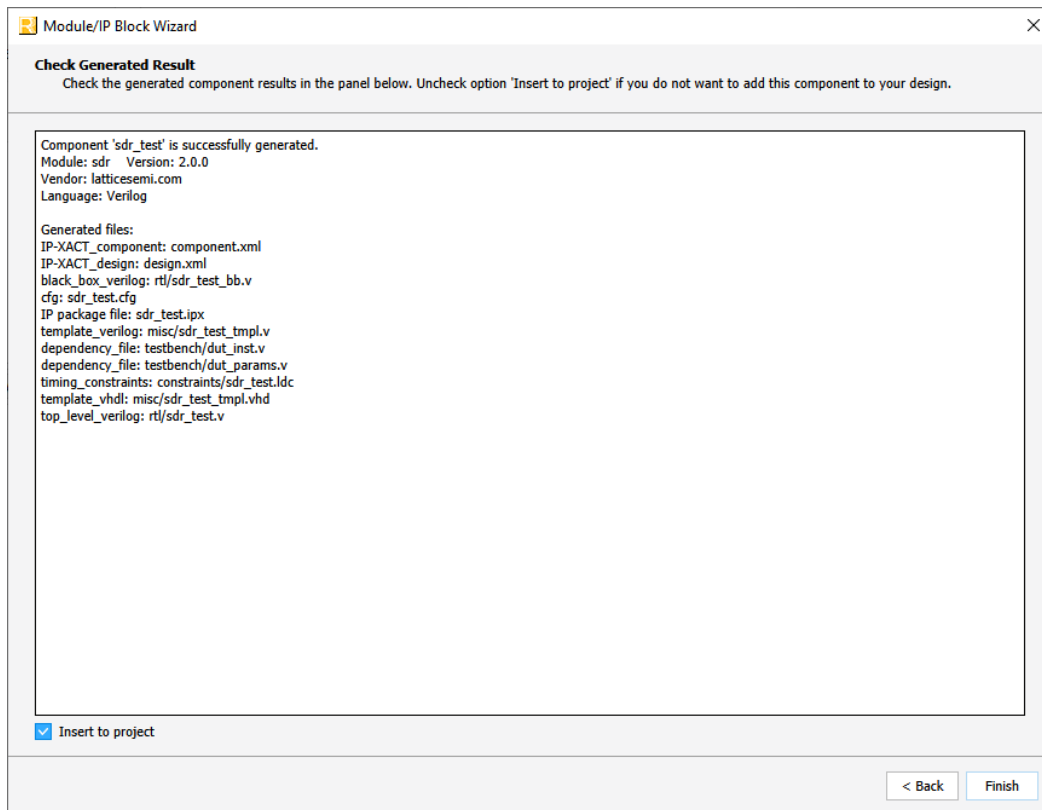


Figure 3.3. Check Generated Result

- Click 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 SDR I/O module package includes the closed-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 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 module.
rtl/<Instance Name>_bb.v	This file provides the synthesis closed-box.
misc/<Instance Name>_tmpl.v misc /<Instance Name>_tmpl.vhd	These files provide instance templates for the module.
testbench/tb_top.v	Test bench template; you can edit this to match your specific needs.
testbench/dut_params.v	Instantiated version of the <IP_name>.v file for simulation use
testbench/dut_ints.v	Top level parameters of the generated RTL file
eval/constraint.pdc	This file provides information on how to constrain the IP in your design. Refer to 3.4 Constraining the IP on how to use this file.

3.3. Running Functional Simulation

After the IP is generated, running functional simulation can be performed using different available simulators. The default simulator already has pre-compiled libraries ready for simulation. Choosing a non-default simulator, however, may require additional steps.

To run functional simulation using the default simulator:

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

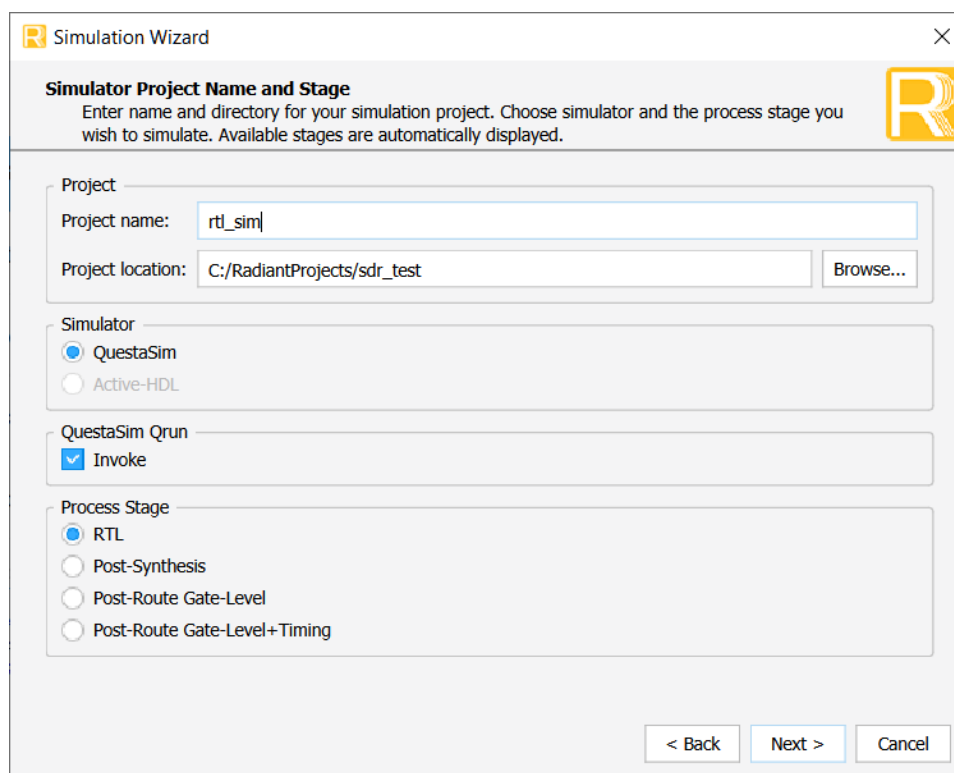


Figure 3.4. Simulation Wizard

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

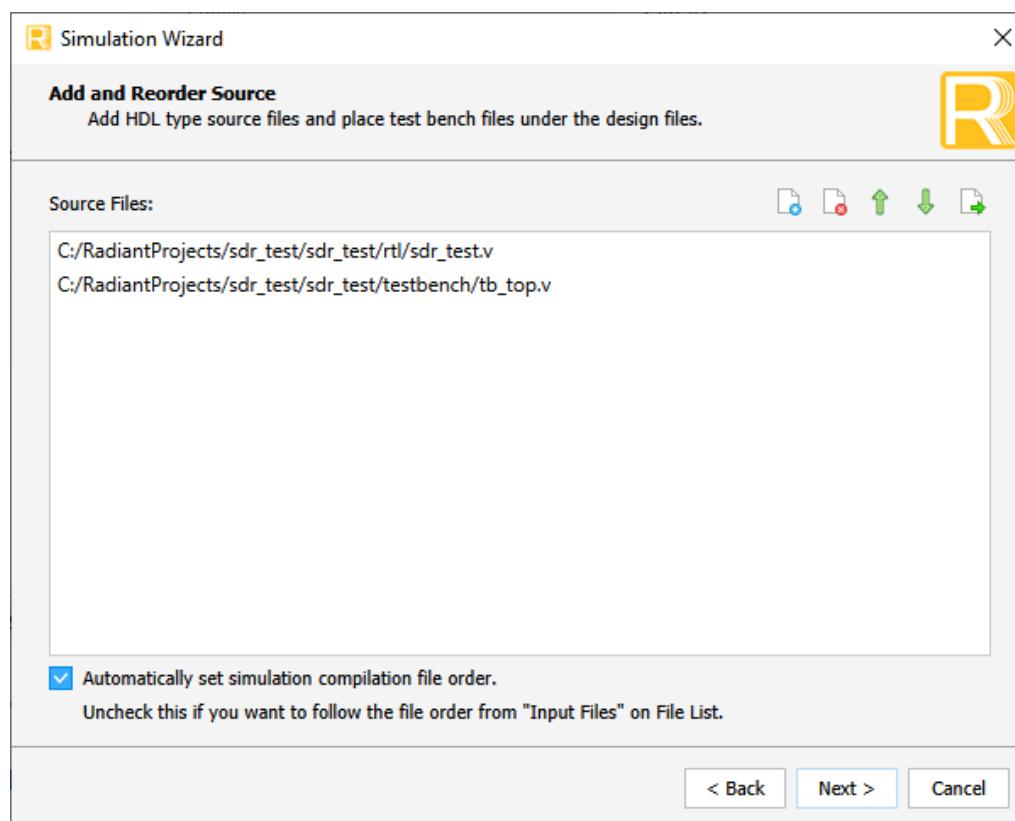


Figure 3.5. Adding and Reordering Source

- 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.

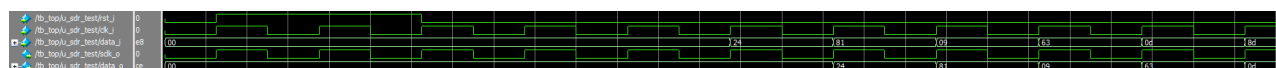


Figure 3.6. Simulation Waveform

3.4. 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 as shown in this pathway: `<IP_Instance_Path>/<IP_Instance_Name>/eval/constraint.pdc`.

The constraint file has been verified during IP evaluation with the IP instantiated directly in 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 content of `constraint.pdc` to the top-level design constraint for post-synthesis.

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

3.5. Hardware Evaluation

There is no restriction on the hardware evaluation of this module.

Appendix A. Resource Utilization

The SDR module resource utilization is shown in [Table A.1](#) using LAV-AT-E70-1LFG676C device with Lattice Synthesis Engine of Lattice Radiant software 2024.1. 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

Configuration	Clk Fmax (MHz) ¹	Registers	LUTs ²	EBRs	DSPs
Default	250	16/398343	9/397440	0	0
Data Path Delay = Static User Defined, Fine Delay Value = 10, others = Default	250	16/398343	9/397440	0	0
Interface Type = Transmit, others = Default	250	16/398343	9/397440	0	0
Interface Type = Transmit, Data Path Delay = Dynamic User Defined, Fine Delay Value = 10, others = Default	250	16/398343	17/397440	0	0

Notes:

1. Fmax is generated when the FPGA design only contains the SDR module, and the target frequency is 200 MHz. These values may be reduced when user logic is added to the FPGA design.
2. 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

- [Lattice Radiant Software](#) web page.
- [Lattice Avant-E](#) web page.
- [Lattice Insights](#) web page 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, IP v2.1.0, December 2024

Section	Change Summary
All	- Made editorial fixes. - Added the IP version information on the cover page.
Functional Description	Updated the <i>I/O Standard for this interface</i> attributes under <i>Selectable Values</i> and <i>Dependency on other Attributes</i> in Table 2.4. Attributes Summary .
References	Updated this section.

Revision 1.2, June 2024

Section	Change Summary
All	- Updated the document title from <i>Avant SDR Module - Lattice Radiant Software</i> to <i>Lattice Avant SDR Module</i>. - Made editorial fixes.
Introduction	- Updated the Features section: - Updated the value of <i>clock frequency</i> from 250 MHz to 300 MHz.
Functional Description	Updated the following attributes under <i>Selectable Values</i> and <i>Dependency on other Attributes</i> in Table 2.4. Attributes Summary: - I/O Standard for this interface - Fine Delay Value for User Defined - Clock Frequency for this Interface (MHz) - Bandwidth for this interface (Mbits/s)
IP Generation and Evaluation	- Updated Figure 3.2. Configure Block of SDR I/O Module. - Updated Figure 3.4. Simulation Wizard.
Resource Utilization	- Updated the <i>Registers</i> and <i>LUTs²</i> values in Table A.1. Resource Utilization. - Updated device name in Resource Utilization section from 1LFG672C to 1LFG676C.

Revision 1.1, December 2023

Section	Change Summary
All	- Changed the document name from <i>Avant SDR Module - Lattice Radiant Software</i> to <i>Lattice Avant SDR Module - Lattice Radiant Software</i>. - Updated the device name from LAV-AT-500E to LAV-AT-E70. - Minor adjustments to ensure the document is consistent with Lattice Semiconductor's inclusive language policy.
Disclaimers	Updated boilerplate.
Inclusive Language	Added boilerplate.
Functional Description	- Updated the content of 2.2 Functional Diagrams. - Added sections 2.2.1.1 – 2.2.1.7, and sections 2.2.2.1 – 2.2.2.6. - Updated Figure 2.3 – Figure 2.8. - Updated Figure 3.1 – Figure 3.6. - Removed Figure 2.9 – Figure 2.21. - Updated the description for <i>rst_i</i> in Table 2.2 and Table 2.3.
IP Generation and Evaluation	- Added attribute <i>eval/constraint.pdc</i> to Table 3.1. - Added 3.4 Constraining the IP. - Update the header number for 3.5 Hardware Evaluation.
Resource Utilization	Updated the <i>Clk Fmax</i> value for all configurations from 200 MHz to 250 MHz in Table A.1.
References	Added this section.
Technical Support Assistance	Added link to the Lattice Answer Database.

Revision 1.0, October 2022

Section	Change Summary
Resource Utilization	Added Appendix A. Resource Utilization section.

Revision 0.8, May 2022

Section	Change Summary
All	Initial release.



www.latticesemi.com