# MachXO4 SDR Module

IP Version: v3.0.0

# User Guide

FPGA-IPUG-02317-1.0

December 2025

## 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

# Figures



# Tables

# Abbreviations in This Document

| Acronyms/Abbreviations | Definition |
|---|---|
| HDL | Hardware Description Language |
| I/O | Input/Output |
| LSE | Lattice Synthesis Engine |
| 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. Overview of the IP

The SDR I/O Module is designed to support high-speed data transmission in FPGA designs using Lattice Radiant™ software. It provides configurable Receive Interface data rate of up to 2656 MBps, while the Transmit Interface with data rate up of up to 2400 MBps, supporting Single-ended or Differential Signaling, and bus widths of 1 to 128 bits. This module also operates at a frequency up to 166 MHz, supports Data Path Delay of various options, and Clock inversion.

## 1.2. Quick Facts

**Table 1.1. Summary of the SDR I/O Module**

| | | |
|---|---|---|
| **IP Requirements** | Supported Devices | MachXO4™ |
| | IP Changes[1] | Refer to the MachXO4 SDR Module Release Notes (FPGA-RN-02110). |
| **Resource Utilization** | Supported User Interface | Native interface |
| | Resources | Refer to Appendix A. Resource Utilization. |
| **Design Tool Support** | Lattice Implementation | IP Core v3.0.0 – Lattice Radiant Software 2025.2 |
| | Synthesis | Lattice Synthesis Engine (LSE),<br>Synopsys® Synplify Pro |
| | Simulation | For a list of supported simulators, see the Lattice Radiant Software User Guide. |

**Notes:**

1. In some instances, the IP may be updated without changes to the user guide. This user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

## 1.3. Features

The key features of SDR I/O Module include:

- Receive Interface of up to 21248 Mbps (2656 MBps) per x128 bit bus width
- Transmit Interface of up to 19200 Mbps (2400 MBps) per x128 bit bus width
- Selectable I/O type
  - Single-ended or Differential Signaling
- 1-bit to 128-bit data bus width
- 1 MHz to 166 MHz clock frequency depending on I/O type
- Data Path Delay that includes following options:
  - Bypass
  - Predefined (Receive Interface only)
  - User-Defined (Receive Interface only)
- Clock inversion (Receive Interface only)

## 1.4. Licensing and Ordering Information

The SDR IP is provided at no additional cost with the Lattice Radiant software. There is no restriction on the hardware evaluation of this IP.

## 1.5. Naming Conventions

### 1.5.1. Nomenclature

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

### 1.5.2. Signal Names

Signal names that end with:

- *_n* are active low
- *_i* are input signals
- *_o* are output signals
- *_io* are bi-directional input/output signals

# 2. Functional Description

## 2.1. IP Architecture Overview

**Table 2.1. SDR I/O Module Interfaces**

| Feature | Description | Comments |
|---|---|---|
| GIREG_RX.SCLK | SDR Receive Interface | • Supports Bypass, Predefined, and User-Defined<br>• Supports clock inversion logic. |
| GOREG_TX.CLK | SDR Transmit Interface | Supports bypass. |

**Note:**

The following describes the naming conventions used for each of the interfaces listed in Table 2.1.

- G – Generic
- IREG – SDR input I/O register
- OREG – SDR output I/O register
- _RX – Receive Interface
- _TX – Transmit Interface
- SCLK – Uses SCLK as primary clocking resource
- CLK – Uses CLK as primary clocking resource

*SDR GIREG_RX.SCLK* interface is used when a simple input register is required for a 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 shows a top-level block diagram of SDR I/O Module.

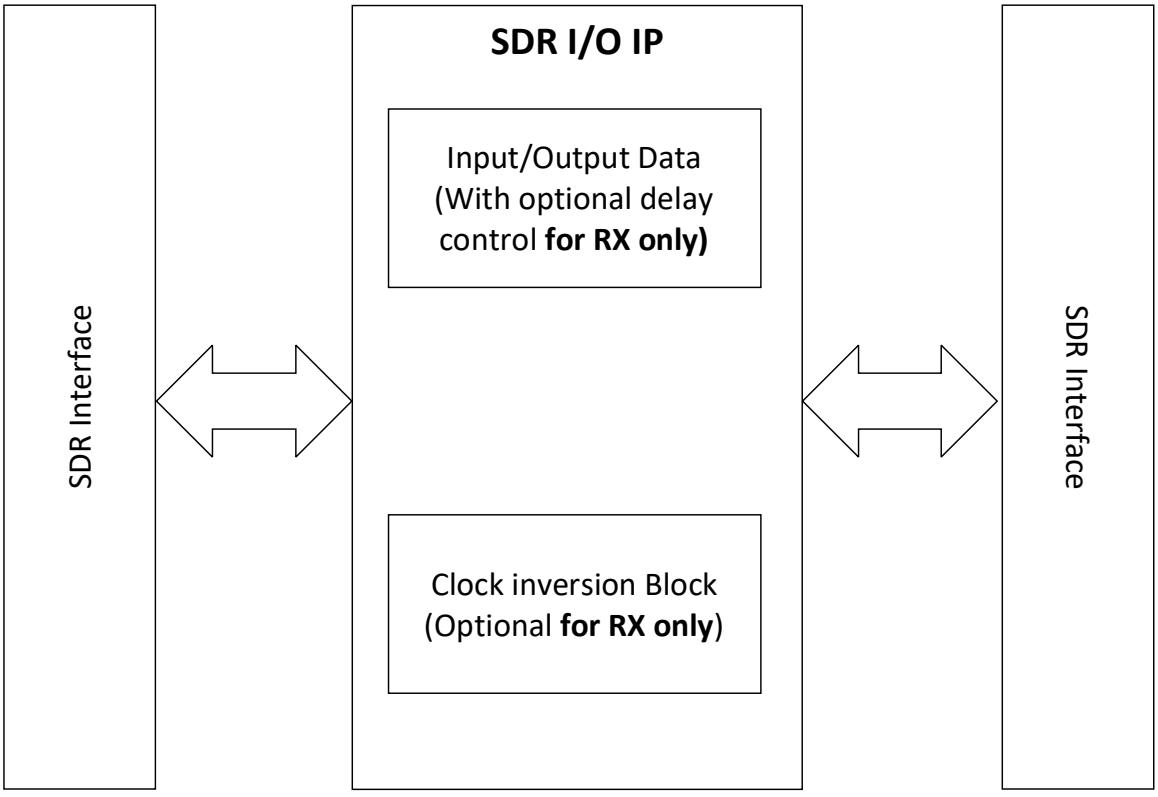
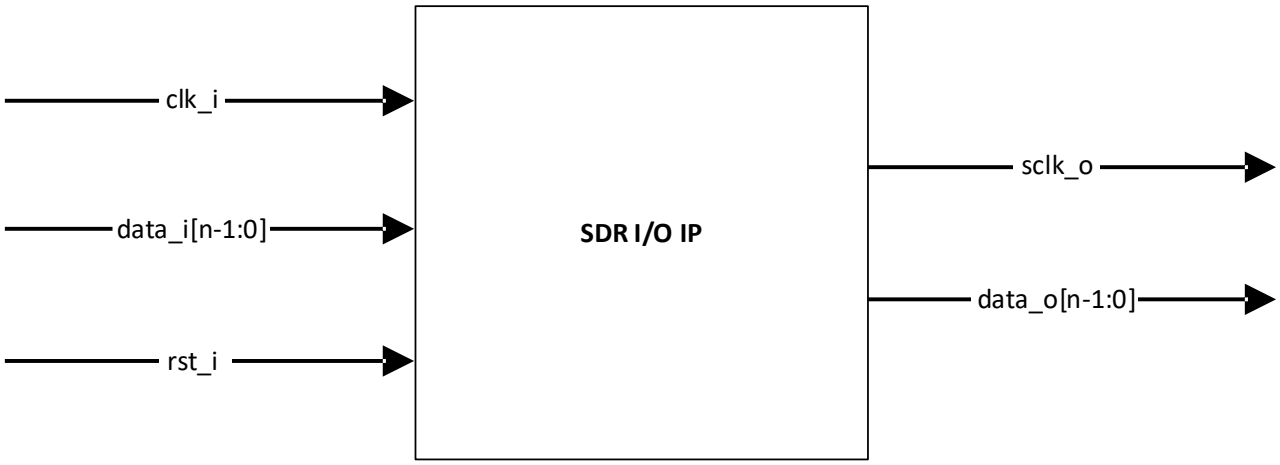



**Figure 2.1. SDR I/O Module Top Level Block Diagram**

## 2.2. Functional Diagrams

### 2.2.1. GIREG_RX.SCLK

For detailed information regarding these GIREG_RX.SCLK configurations, refer to the MachXO4 Implementing High-Speed I/O Interfaces (FPGA-TN-02410).




**Note**: n = number of lanes/bus width.

**Figure 2.2. GIREG_RX.SCLK Bypass/Predefined/User Defined Delay Block Diagram**

Figure 2.2 shows the block diagram for the configurations described in sections 2.2.1.1 and 2.2.1.2.

#### 2.2.1.1. GIREG_RX.SCLK Bypass Interface

- Reset *rst_i* is active high asynchronous reset. Change from High to Low to de-assert.
- 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.*

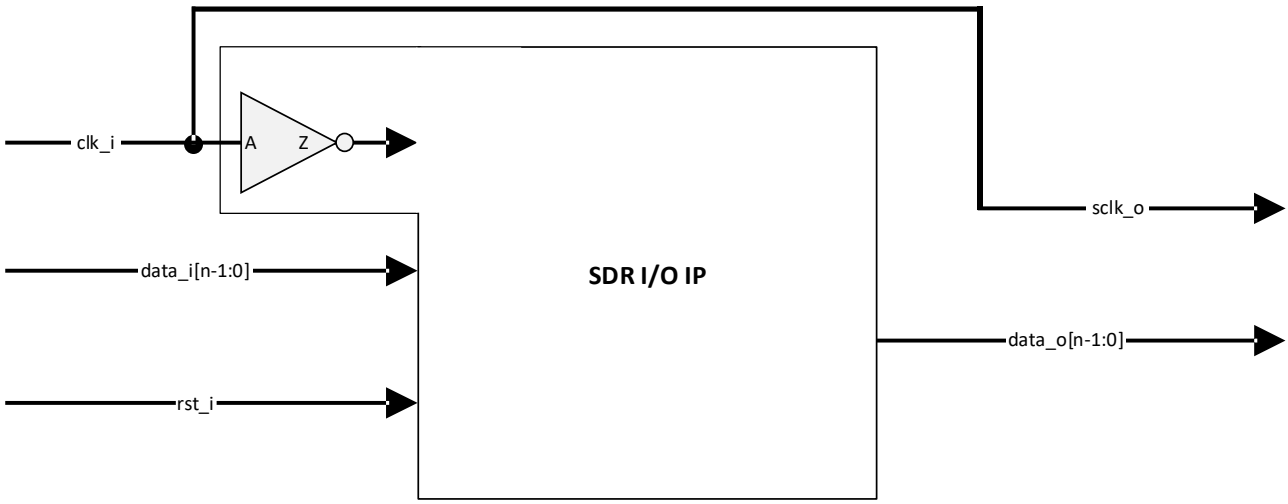
#### 2.2.1.2. GIREG_RX.SCLK Predefined/User Defined Delay Interface

- Reset *rst_i* is active high asynchronous reset. Change from High to Low to de-assert.
- 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 Predefined 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.*

**Notes**:

- The *clock injection time* refers to the delay or skew introduced when a clock signal is injected or distributed into a circuit, particularly at the I/O boundary of a digital system.
- Predefined delay value is 2.52 ns (24 steps).
- User-Defined delay value: each smallest step resolution is 105 ps and DELAY0 equals 0 ps, while each subsequent step (DELAY1 to DELAY31) adds 105 ps.

### 2.2.2. GIREG_RX.SCLK with Clock Inversion Enable




**Note**: n = number of lanes/bus width.

**Figure 2.3. GIREG_RX.SCLK Bypass/Predefined/User Defined Delay with Clock Inversion Enabled Block Diagram**

Figure 2.3 shows the block diagram for the configurations described in 2.2.2.1 and 2.2.2.2.

#### 2.2.2.1. GIREG_RX.SCLK Bypass Inversion Enabled Interface

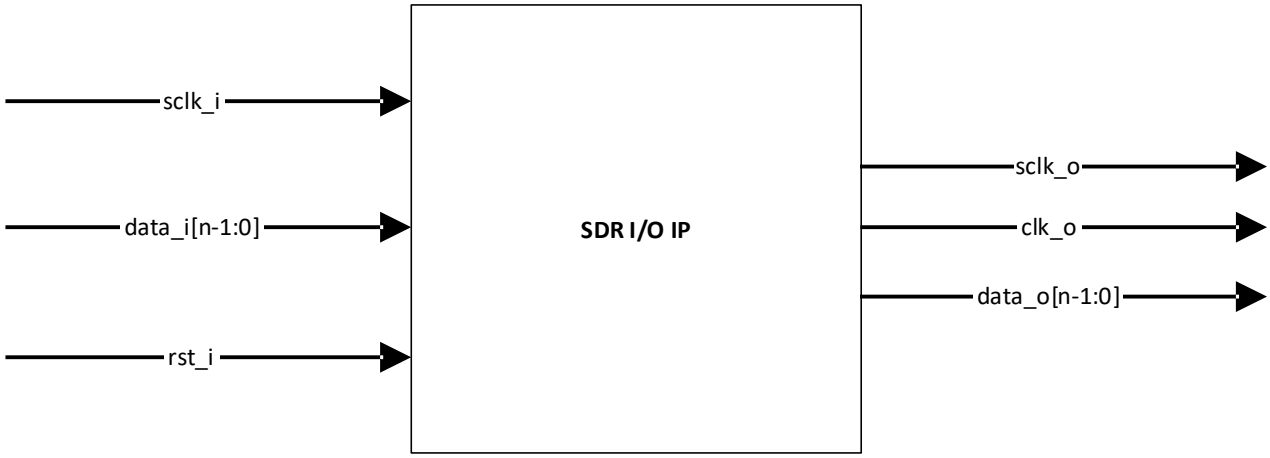
- Reset *rst_i* is active high asynchronous reset. Change from High to Low to de-assert.
- The data signal passes through a register that captures input data at the negative edge of clk_*i.*
- Output sampling clock *sclk_o* is routed from *clk_i.*

#### 2.2.2.2. GIREG_RX.SCLK Predefined/User Defined Delay Inversion Enabled Interface

- Reset *rst_i* is active high asynchronous reset. Change from High to Low to de-assert.
- 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 the delayed input data at the negative edge of *clk_i.*
- Output sampling clock *sclk_o* is routed from *clk_i.*

### 2.2.3. GOREG_TX.SCLK

For detailed information regarding these GOREG_TX.SCLK configurations, refer to the MachXO4 Implementing High-Speed I/O Interfaces (FPGA-TN-02410).




**Note**: n = number of lanes/bus width.

**Figure 2.4. GOREG_TX.SCLK Bypass Block Diagram**

Figure 2.4 shows the block diagram for the configurations described in 2.2.3.1.

#### 2.2.3.1. GOREG_TX.SCLK Bypass Interface

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

# 3. IP Parameter Description

The configurable attributes of the SDR I/O Module are shown in the following tables. You can configure the IP by setting the attributes accordingly in the IP Catalog Module/IP Block Wizard of the Lattice Radiant software.

Wherever applicable, default values are in bold.

## 3.1. General

Table 3.1 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 3.1. General Attributes**

| Attribute | Selectable Values | Default | Dependency on other Attributes |
|---|---|---|---|
| Interface Type | Receive, Transmit | **Receive** | — |
| I/O Standard for this Interface | Common:<br>LVDS25, LVTTL33, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, LVCMOS12.<br><br>For Transmit only:<br>LVDS25E, BLVDS25E, MLVDS25E, LVPECL33E, LVTTL33D, LVCMOS33D, LVCMOS25D.<br><br>For Receive only:<br>BLVDS25, MLVDS25, LVPECL33, LVTTL33D, LVCMOS33D, LVCMOS25D, LVCMOS10R25, LVCMOS10R33, LVCMOS12R25, LVCMOS12R33. | **LVDS25** | — |
| Bus Width for this Interface | 1 – 128 | **8** | — |
| Data Path Delay | Bypass<br><br>Bypass, Predefined, User Defined | **Bypass**<br><br>**Predefined** | *Interface Type* == *Transmit*<br><br>*Predefined* and *User Defined*, only supported when *Interface Type* == *Receive* |
| Fine Delay Value for User Defined | 0 – 31 | **0** | *Data Path Delay* == *User Defined* |
| Enable Clock Inversion | Checked, Not checked | **Not checked** | *Interface Type* == *Receive* |
| Clock Frequency for this Interface (MHz) | 1 – 166<br>1 – 150<br>1 – 125 | **125**<br>**125**<br>**125** | *Interface Type* == *Receive*<br>*Interface Type* == *Transmit* (grade-6 dev)<br>*Interface Type* == *Transmit* (grade-5 dev) and below |
| Bandwidth for this Interface (Mbits/s) | Calculated | **1000** | Clock Frequency × Bus Width.<br>Display for information only |

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

## 3.2. Attributes Description

Table 3.1 shows the description of the attributes used in SDR I/O Module.

**Table 3.2. Attributes Description**

| Attribute Name | Description |
|---|---|
| Interface Type | Select the interface type, such as Receive or Transmit. |
| I/O Standard for this Interface | Select an I/O standard from the list of supported Single-ended and Differential types. |
| Bus Width for this Interface | Select the total number of bus lanes. |
| Data Path Delay | Selects between *Bypass*, *Predefined*, or *User Defined* modes. |
| Fine Delay Value for User Defined | Defines the fine delay setting for the data path. This setting is only applicable when the *Data Path Delay* mode is configured as *User Defined*.<br><br>Provides 32 delay steps, from *DELAY0* to *DELAY31*. *DELAY0* equals 0 ps, while each subsequent step (*DELAY1* to *DELAY31*) adds 105 ps.<br>The maximum fine delay is 31 × 105ps = 3.255 ns. |
| Enable Clock Inversion | Enables clock inversion for the *Receive* interface type only. When enabled, the sampling clock for received data is inverted. |
| Clock Frequency for this Interface (MHz) | Specifies the clock frequency used by the selected interface. |

# 4. Signal Description

This section describes the SDR I/O Module ports.

**Table 4.1. SDR I/O Module Receive Signal Description**

| Port Name | Clock Domain | Direction | Description |
|---|---|---|---|
| **Clocks and Reset** | | | |
| rst_i | asynchronous | In | Active HIGH asynchronous reset signal. |
| sclk_o | - | Out | Received data sampling clock |
| **User Interface** | | | |
| data_o[n-1:0] | sclk_o | Out | Received input-data to fabric. |
| **I/O Pad Interface** | | | |
| clk_i | - | In | Clock input signal from I/O. |
| data_i[n-1:0] | clk_i | In | Data input signal from I/O. |

**Note:** n = number of lanes/bus width.

**Table 4.2. SDR I/O Module Transmit Signal Description**

| Port Name | Clock Domain | Direction | Description |
|---|---|---|---|
| **Clocks and Reset** | | | |
| rst_i | asynchronous | In | Active HIGH asynchronous reset signal. |
| sclk_i | - | In | Transmit data sampling clock. |
| **User Interface** | | | |
| data_i[n-1:0] | sclk_i | In | Transmit output-data to I/O. |
| **I/O Pad Interface** | | | |
| clk_o | - | Out | Clock output signal to I/O. |
| data_o[n-1:0] | clk_o | Out | Data output signal to I/O. |
| sclk_o | - | Out | System Clock output to I/O. |

**Note:** n = number of lanes/bus width.

# 5. Designing with the IP

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.

**Note:** The screenshots provided are for reference only. Details may vary depending on the version of the IP or software being used. If there have been no significant changes to the GUI, a screenshot may reflect an earlier version of the IP.

## 5.1. Generation and Instantiating the IP

You can use the Lattice Radiant software to generate IP modules and integrate them into the device architecture. The steps below describe how to generate the SDR I/O Module in the Lattice Radiant software.

To generate SDR I/O Module:

1. Create a new Lattice Radiant Software project or open an existing project.
2. Click the **IP Catalog** button to view the **IP Catalog** pane
3. On the **IP on Local** tab, double-click on **SDR** under **Module, Architecture_Modules, IO** category. The **Module/IP Block Wizard** opens as shown in Figure 5.1.

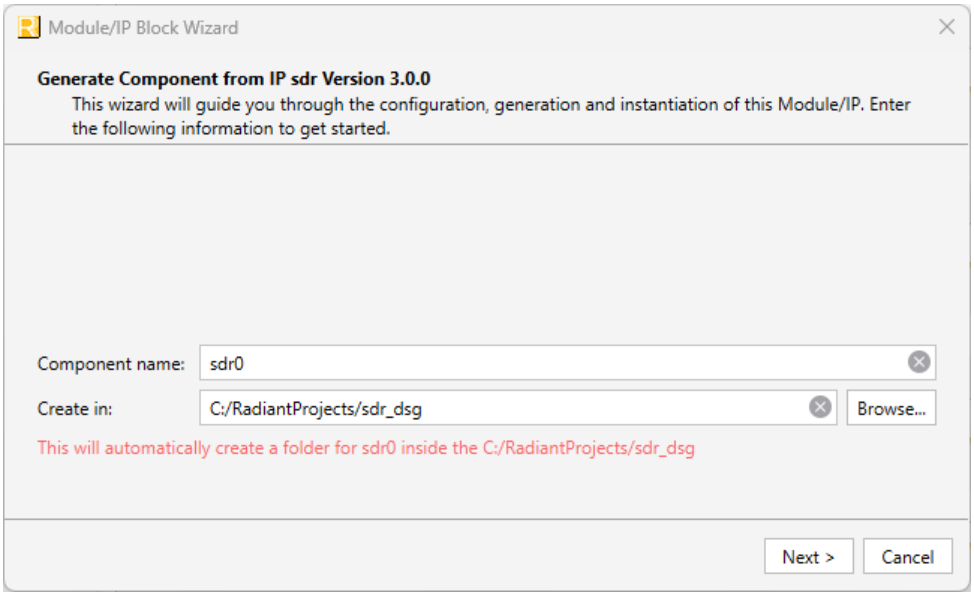



**Figure 5.1. Module/IP Block Wizard**

4. Enter value in the **Component name** and the **Create in** fields and click **Next**.
5. Customize the selected SDR I/O Module using drop-down lists and check boxes. Figure 5.2 shows an example of the SDR I/O Module. For details on the configuration options, refer to the IP Parameter Description section.

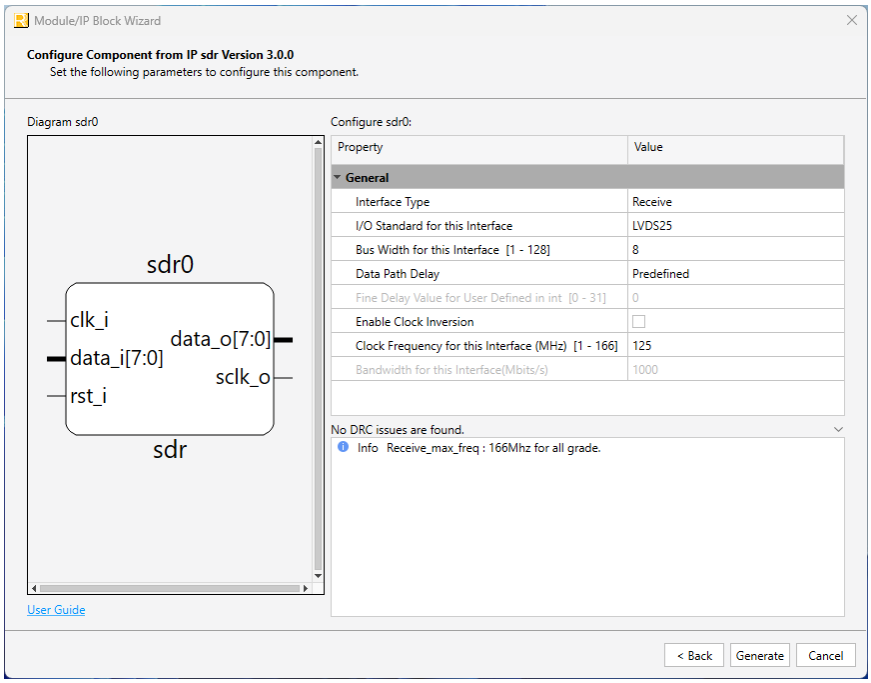



**Figure 5.2. Configure Block of SDR I/O Module**

6. Click **Generate**. The **Check Generated Result** windows opens. This window shows design block messages and results as shown in Figure 5.3.

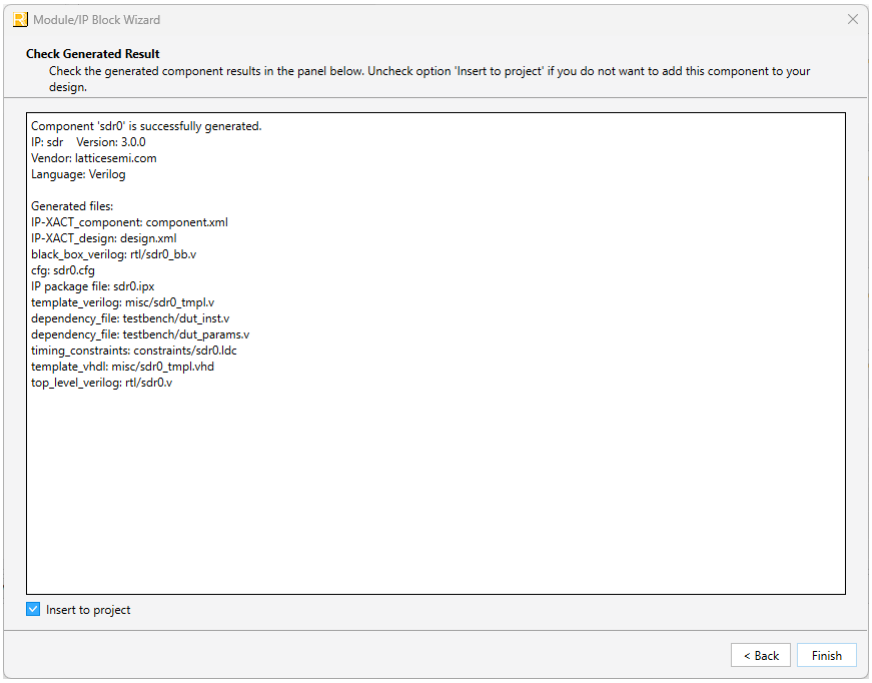



**Figure 5.3. Check Generated Result**

7. Click **Finish**. All the generated files are placed under the directory paths in the Create in and the Component name fields shown in Figure 5.1.

### 5.1.1. Generated Files and File Structure

The generated SDR I/O 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 5.1.

**Table 5.1. Generated File List**

| Attribute | Description |
|---|---|
| <Instance Name>.ipx | Contains the information on the files associated to the generated IP. |
| <Instance Name>.cfg | 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 | Provides an example RTL top file that instantiates the module. |
| rtl/<Instance Name>_bb.v | Provides the synthesis black box. |
| misc/<Instance Name>_tmpl.v<br>misc /<Instance Name>_tmpl.vhd | Provide instance templates for the module. |

## 5.2. Running Functional Simulation

After generating the IP, you can perform functional simulation using any available simulator .The default simulator, Questa® Sim, includes pre-compiled libraries for immediate simulation use. If you choose a non-default simulator, additional setup steps may be required.

**Note**: The provided testbench is intended strictly for RTL-level simulation during the functional verification stage. It does not support post-route gate-level simulation.

To run functional simulation using the default simulator:

1. Click the  button located on the **Toolbar** to initiate **Simulation Wizard** shown in Figure 5.4.

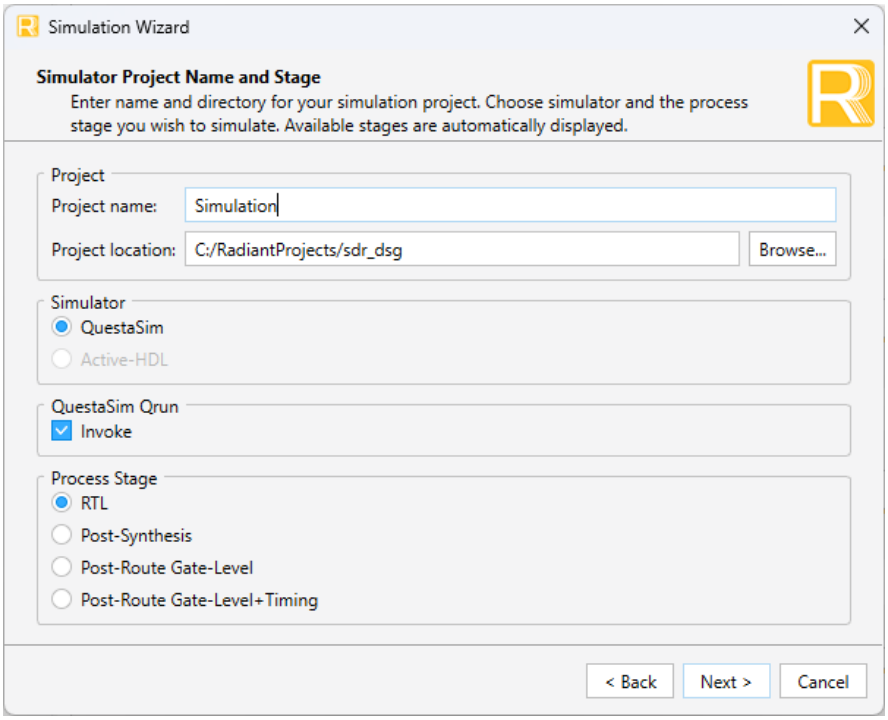



**Figure 5.4. Simulation Wizard**

2. Click **Next** to open the **Add and Reorder Source** window shown in Figure 5.5.

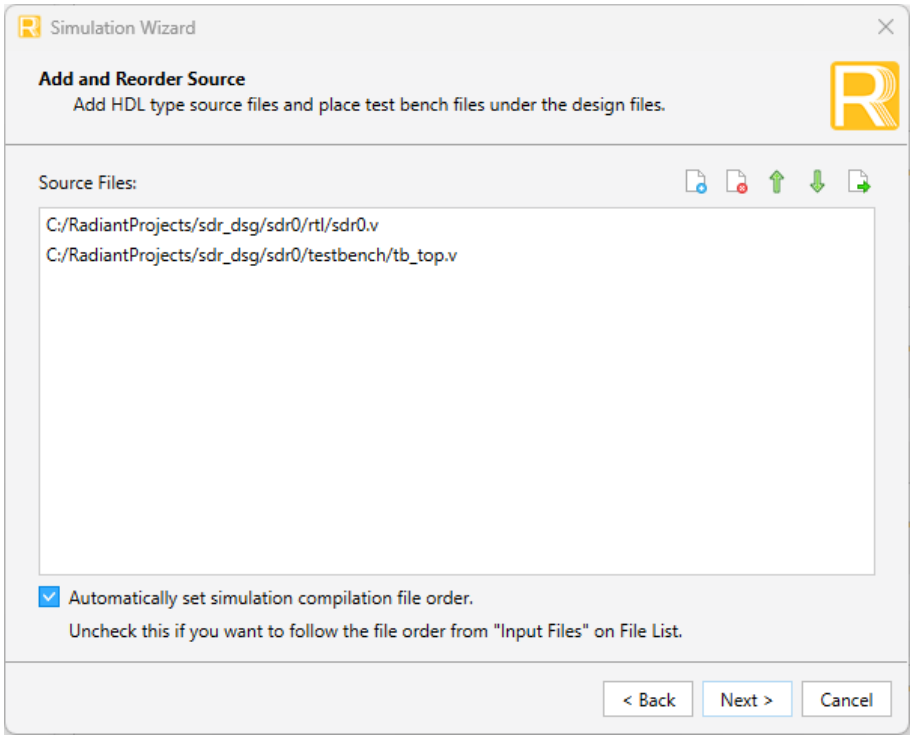



**Figure 5.5. Adding and Reordering Source**

3. Click **Next**. The Summary window is shown in Figure 5.6. In the Summary window, set the Default Run to 0 to invoke run -all.

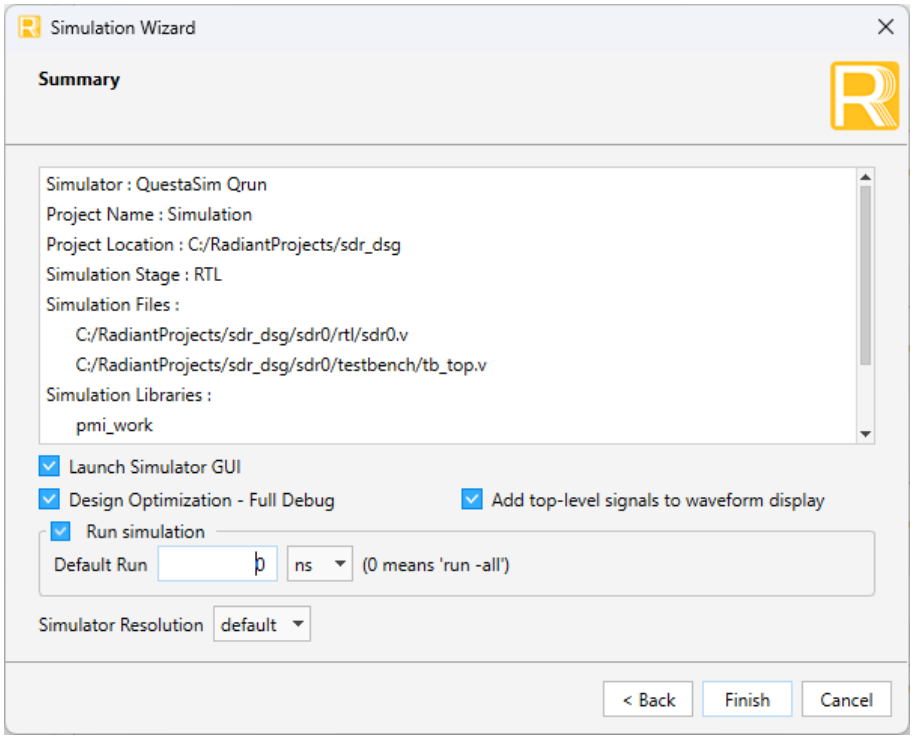



**Figure 5.6. Simulation Wizard Window**

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.

### 5.2.1. Simulation Results

Figure 5.7 shows the example simulation results.


**Figure 5.7. Simulation Result and Waveform**

# Appendix A. Resource Utilization

The SDR I/O Module resource utilization is shown in Table A.1 using MachXO4 (LFMXO4-110HE-6BBG484I) device using Lattice Radiant software 2025.2 with Lattice Synthesis Engine (LSE). 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 (LFMXO4-110HE-6BBG484I[1])**

| Configuration | Clk Fmax (MHz)[2] | Registers | LUTs[3] | EBRs | IO Buffers |
|---|---|---|---|---|---|
| Default (Interface Type == Receive) | 166 | 0 | 0 | 0 | 19/381 |
| Data Path Delay = Static User Defined, Fine Delay Value = 31, others = Default | 166 | 8 | 0 | 0 | 19/381 |
| Interface Type == Transmit, Others == Default | 150 | 0 | 2 | 0 | 20/381 |

**Notes:**

1. This table is based on the highest grade-6 LFMXO4 device.
2. Fmax is generated when the FPGA design only contains the SDR I/O Module and the target frequency is 166 MHz for Receive and 150 MHz for Transmit. These values may be reduced when user logic is added to the FPGA design.
3. 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

- MachXO4 SDR Module Release Notes (FPGA-RN-02110)
- MachXO4 Implementing High-Speed I/O Interfaces (FPGA-TN-02410)
- MachXO4 web page
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Lattice Radiant Software web page
- Lattice Solutions IP Cores web page
- Lattice Solutions Reference Designs web page
- 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

**Note:** In some instances, the IP may be updated without changes to the user guide. The user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

**Revision 1.0, IP v3.0.0, December 2025**

| Section | Change Summary |
|---|---|
| All | Initial release. |


www.latticesemi.com