MachXO4 DDR 7:1 Module

IP Version: v3.0.0

User Guide

FPGA-IPUG-02315-1.0

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

A list of abbreviations used in this document.

Acronyms/Abbreviations	Definition
PLL	Phase Lock Loop
GDDR	Generic Double Data Rate
RX	Receive
TX	Transmit
PIO	Programmable Input Output

1. Introduction

The high-speed Generic Double Data Rate 7:1 Input/Output (GDDR 7:1 I/O) Module interfaces are supported through the built-in gearing logic in the Programmable I/O (PIO) cells. This gearing is necessary to support high-speed I/O while reducing the performance requirement on the FPGA fabric. This DDR 7 to 1 IP supports 7:1 gearing ratio.

1.1. Overview of the IP

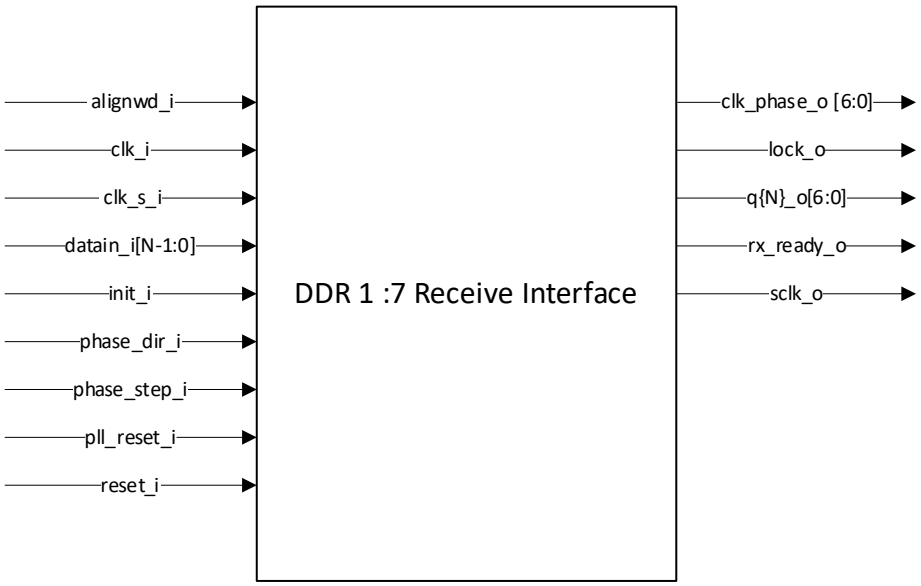
The GDDR 7:1 Module is a 7:1 gearing ratio DDR which is mainly used for video display applications. Table 1.1 provides description summary of GDDR 7:1 I/O Module interfaces. See details of the signal descriptions in the Signal Description section.

Table 1.1 Available GDDR 7:1 I/O Module Interfaces

Feature	Description
Receive Interface (RX)	Generic DDR 7:1 Receive Interface
Transmit Interface (TX)	Generic DDR 7:1 Transmit Interface

1.1.1. Receive (RX)

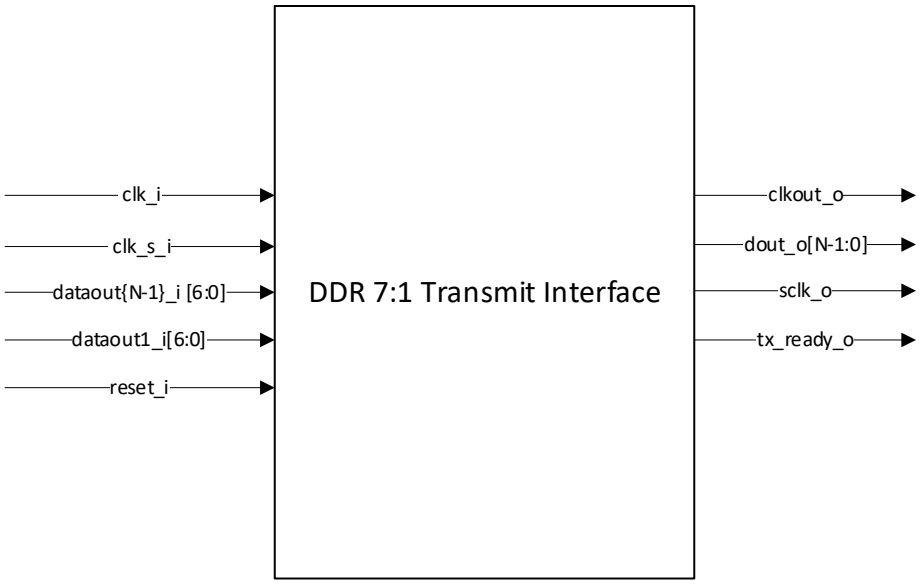
The block diagram for RX as shown in Figure 1.1.



Note: N = the number of data width/lanes.

Figure 1.1 Receive Interface Block Diagram

1.1.2. Transmit (TX)



Note: N = the number of data width/lanes.

Figure 1.2 Transmit Interface Block Diagram

1.2. Quick Facts

Table 1.2. Summary of the GDDR 7:1 I/O Module

IP Requirements	Supported Devices	MachXO4™
	IP Changes[1]	For a list of changes to the IP, refer to the MachXO4 DDR 7:1 Module Release Notes (FPGA-RN-02099).
Resource Utilization	Supported User Interface	Native Interface
Design Tool Support	Lattice Implementation	IP Core v3.0.0 – Lattice Radiant Software 2025.2
	Synthesis	Synopsys® Synplify Pro for Lattice, Lattice Synthesis Engine (LSE)
	Simulation	For a list of supported simulators, see the Lattice Radiant Software User Guide

Note:

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

Key features of the GDDR 7:1 I/O Module include:

- Receive and Transmit Interface up to 108 MHz
- 1-bit to 21-bit data bus width

1.4. Licensing and Ordering Information

The GDDR 7:1 I/O Module is provided at no additional cost with the Lattice Radiant software.

1.5. Naming Conventions

1.5.1. Nomenclature

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

1.5.2. Signal Names

- *_n* are active low (asserted when value is logic 0)
- *_i* are input signals
- *_o* are output signals

2. Functional Description

2.1. IP Architecture Overview

The GDDR 7:1 interface are supported in MachXO4 using the dedicated logic blocks and pre-defined in the software and characterized in the silicon. Table 2.1 lists all the supported interfaces and gives a brief description of each interface

Table 2.1. GDRR 7:1 I/O Module Interfaces

Mode	Interface Name	Description	Supporting Device & Sides
RX GDDR71	GDDR71_RX.ECLK.7:1	GDDR 7:1 input using ECLK	Bottom
TX GDDR71	GDDR71_TX.ECLK.7:1	GDDR 7:1 output using ECLK	Top

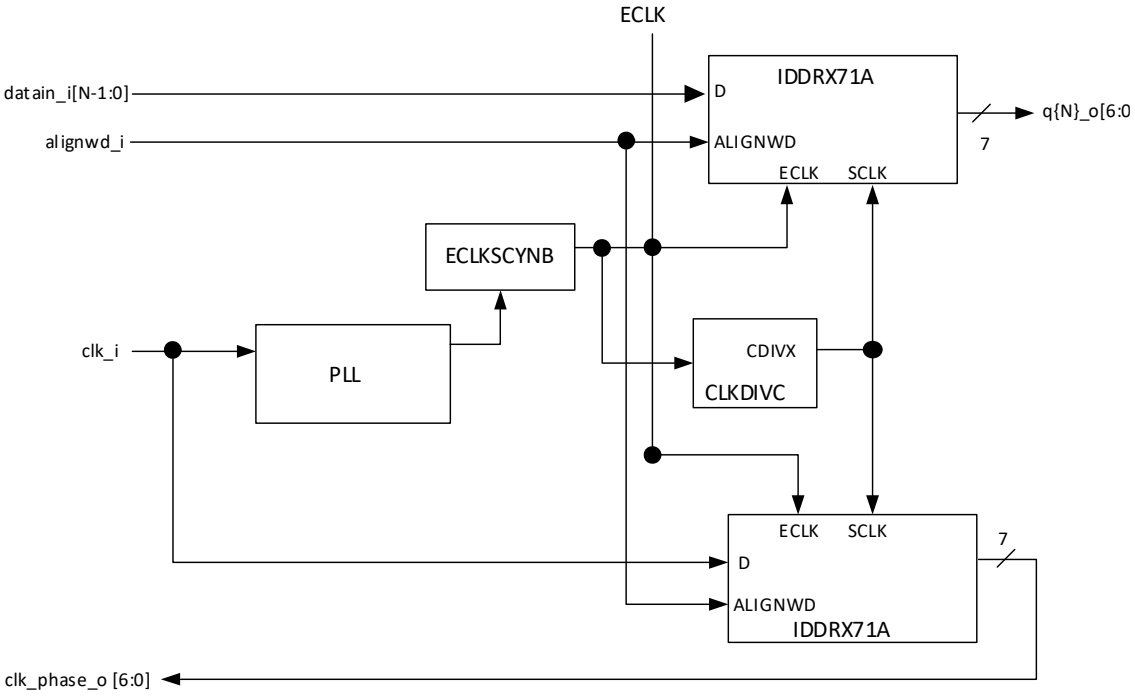
The following describes the naming conventions used for each of the interfaces listed in Table 2.1.

- G – Generic
- DDR71 – DDR 7:1 I/I register
- _RX – Receive interface
- _TX – Transmit interface
- ECLK – Uses ECLK (edge clock) clocking resource at the GDDR interface

2.2. Receive Interface

2.2.1. GDDR71_RX.ECLK.7:1

The GDDR 7:1 receive interface is unique among the supported high-speed DDR interfaces. It uses the PLL to search the best clock edge to the data opening position during bit alignment process. The PLL steps through the 16 phases to give eight sampling points per data. The data path delay is not used in this interface. CLKDIVC is used to divide down the ECLK by 3.5 due to the nature of the 1:7 deserializing requirement. This means the SCLK is running seven times slower than the incoming data rate. ECLKSYNCB element is associated with the ECLK and must be used to drive the ECLK. The complete 7:1 LVDS video display application requires bit alignment and word alignment blocks to be built in the FPGA resources in addition to the built-in I/O gearing logic and alignment logic. The CLK_PHASE signal is sent to the FPGA side to build the bit alignment logic.



Note: n = number of lanes. The range of *n* is from 1 to 21.

Figure 2.1. GDDR71_RX.ECLK.7:1 Interface

Interface rules:

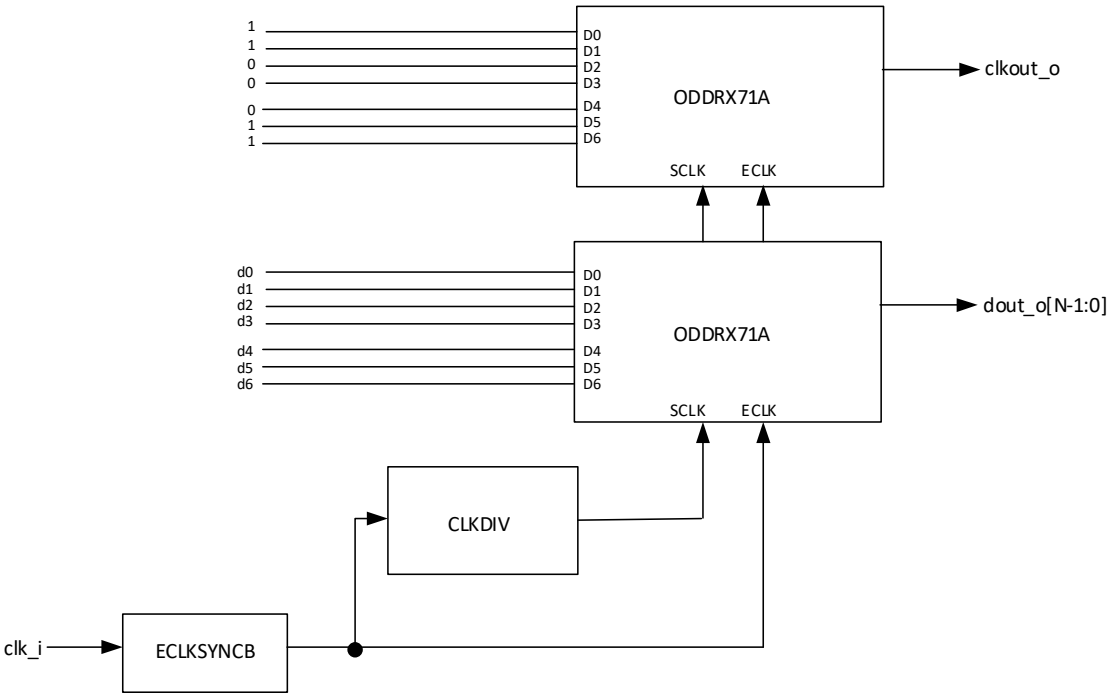
- Must use a dedicated clock pin PCLK at the bottom side as the clock source for PLL
- Clock net routed to SCLK must use primary clock net
- There are up to two PLLs per device. It limits this interface to have maximum of two clock frequencies per device.
- The data input must use A/B pair of the I/O logic cell for 7:1 gearing

This interface is supported at the bottom side of the devices.

2.3. Transmit Interface

2.3.1. GDDR71_TX.ECLK.7:1

The GDDR 7:1 transmit interface is unique among the supported high-speed DDR interfaces. It uses a specific pattern to generate the output clock, known as pixel clock. The CLKDIVC is used to divide down the ECLK by 3.5 due to the nature of the 7:1 serializing requirement. This means the SCLK is running 7 times slower than the transmit data rate. ECLKSYNCB element is associated with the ECLK and must be used to drive the ECLK.



Note: n = number of lanes. The range of n is from 1 to 21.

Figure 2.2. GDDR71_TX.ECLK.7:1 Interface

Interface rules:

- Must use edge clock routing resources for the ECLK
- The routing of SCLK must use primary clock net
- Data output must use A/B pair of the I/O logic for 7:1 gearing

This interface is supported at the top side of the device.

2.4. Functional Detail Based on Primitive Specification

Software primitives used for all the generic DDR interfaces implementation are discussed in this section. The primitives are divided according to their usage. Some of them are used for generic DDR interfaces and others are control functions. The DDR input primitives will be discussed first, followed by the DDR output primitives, then the DDR control logic primitives.

Table 2.2. MachXO4 GDDR 7:1 I/O Module Software Primitives

Type	Primitive	Usage
Data Input	IDDR71A	Generic DDR 7:1
Data Output	ODDR71A	Generic DDR 7:1

2.4.1. Input GDDR 7:1 Primitive

2.4.1.1. IDDRX71A

The primitive implements the input register block in 7:1 gearing mode. This is used only for the generic DDR 7:1 interface (gearing ratio 1:7) on the bottom side of the device. Its registers are designed to use edge clock routing on the GDDR interface and the system clock on the FPGA side. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive, like the input x4 gearing primitive, can only be used on the A/B pair of the PIO cell at the bottom side of the device.

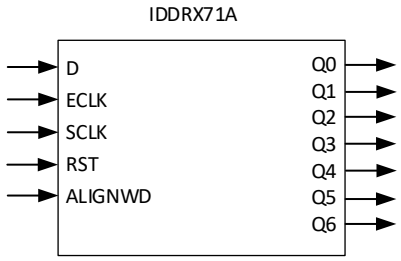


Figure 2.3. IDDRX71A Symbol

The 1:7 gearing of IDDRX71A depends on an internal control signal to select three bits or four bits data at a time. The first set of the registers is the DDR register to capture the data at both edges of ECLK. The second set is the synchronization registers to hold the data ready for clock domain transfer. The third set of registers performs the clock domain transfer from ECLK to SCLK.

2.4.2. Output GDDR 7:1 Primitive

2.4.2.1. ODDRX71A

The primitive implements the output register block in 7:1 gearing mode. This is used only for the generic DDR 7:1 interface (gearing ratio 7:1) on the top side of the device. Its registers are designed to use the system clock on the FPGA side and the edge clock routing on the GDDR interface. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive can only be used on the A/B pair of I/O cells at the top side of the device.

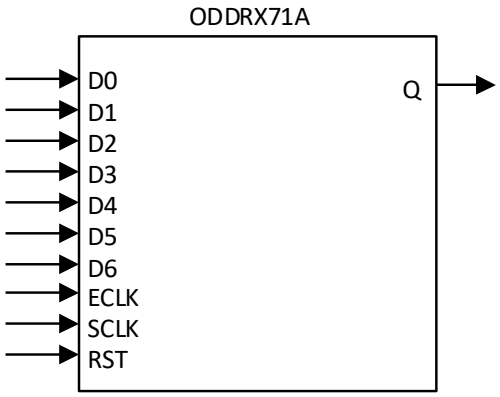


Figure 2.4. ODDRX71A Symbol

The 7:1 gearing of ODDRX71A depends on the internal control signal to select the three or four bits of data at a time for transmission. The parallel data is registered by SCLK at the first set of registers. At each update, the parallel data is clocked in by SCLK and is held by the second set of registers. The third set of registers has the data ready to be multiplexed out as serial data by ECLK.

3. IP Parameter Description

The configurable attributes of the GDDR 7:1 I/O Module are shown in the following tables. You can configure the IP by setting the attributes accordingly in the IP Catalog's Module/IP 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 GDDR 7:1 I/O Module. Attributes settings are specified using GDDR 7:1 I/O Module Configuration user interface in Lattice Radiant.

Table 3.1. General Attributes

Attribute	Selectable Values	Description
Interface type	Transmit, **Receive**	Type of interface
Data width	1-21	The number of incoming lanes (n). Default is **4.**
High Speed Clock Frequency (Pixel Clock)	10 MHz – **108 MHz**	Pixel clock frequency for GDDR 7:1 Module.

4. Signal Description

This section describes the GDDR 7:1 I/O Module signals.

4.1. GDDR 7:1 I/O Module Interface

Table 4.1. Signal Name Used in GDDR 7:1 I/O Module

Port	Type	Width (bits)	Description
Clock And Reset (Support in both Receive and Transmit Interface)			
clk_i	input	1	Source synchronous input clock. The range supports between 10 MHz to 108 MHz.
clk_s_i	input	1	Slow clock for reset synchronization. (This clock must be slower than ECLK).
reset_i	input	1	Asynchronous reset to the interface, active high.
sclk_o	output	1	System clock for the FPGA fabric.
Receive Interface			
datain_i	input	n-1	Serial data input at Receive interfaces.
alignwd_i	input	1	Word alignment control signal, active high.
init_i	input	1	Initialize reset synchronization, active high
phase_dir_i	input	1	PLL phase direction.
phase_step_i	input	1	PLL phase step.
pll_reset_i	input	1	Resets the PLL core (VCO, phase detector, and charge pump) and the output dividers.
clock_phase_o	output	7	7-bit representation of input clock phase.
lock_o	output	1	PLL lock.
q{n-1..0}_o	output	7	Parallel data output of the RX interfaces. It depends on the number of lanes.
rx_ready_o	output	1	Indicate completion of reset synchronization.
Transmit Interface			
dataout{n-1..0}_i	input	7	Parallel input data of the Transmit interfaces.
clkout_o	output	1	Source synchronous clock.
dout_o	output	n-1	Serial data output for the Transmit interfaces.
tx_ready_o	output	1	Indicate completion of reset synchronization.

Note: n = number of data width/lanes, the range is 1 to 21.

5. Designing with the IP

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

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. Generating and Instantiating the IP

You can use the Lattice Radiant software to generate IP modules and integrate them into device architecture. The steps below describe how to generate the GDDR 7:1 I/O Module in the Lattice Radiant software.

To generate the GDDR 7:1 Module :

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **GDDR 7:1** under **Architecture_Modules, IO** category. The **Module/IP Block Wizard** opens as shown in Figure 5.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.

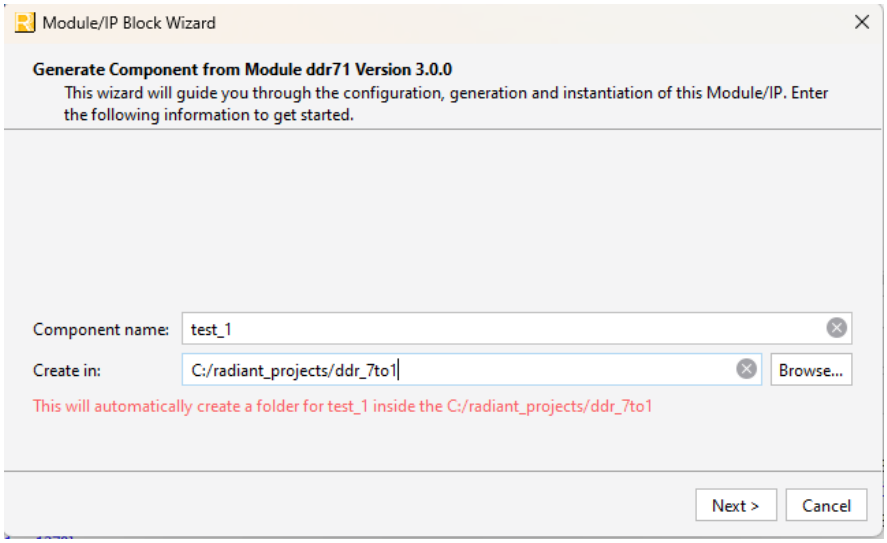


Figure 5.1. Module/IP Block Wizard

3. In the next **Module/IP Block Wizard** window, customize the selected GDDR 7:1 I/O Module using drop-down lists and fill up the values. Figure 5.1 shows an example configuration of the GDDR 7:1 Module.

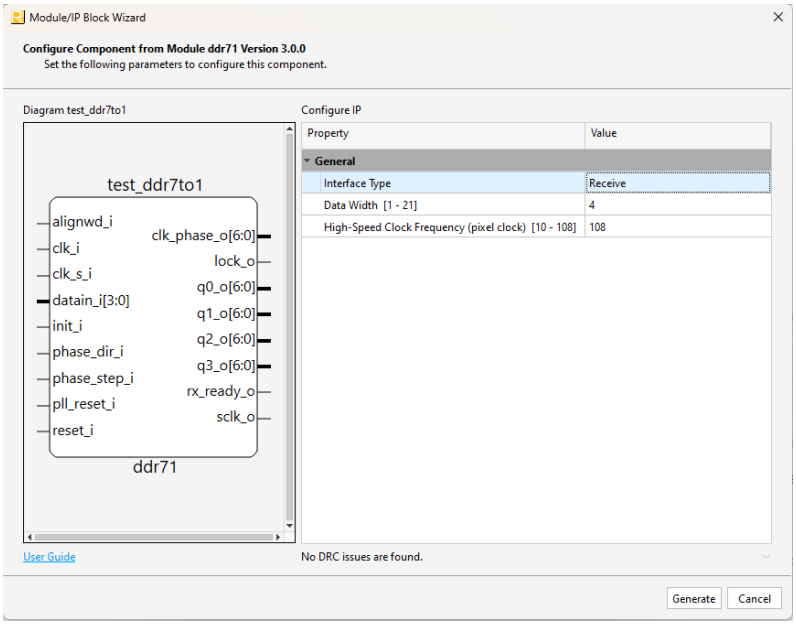


Figure 5.2. IP Configuration

Table 5.1. Interface options of the IP

Interface Option	Description	Range	Default Value
Interface Type	Types of interfaces	Transmit, Receive	Receive
Data Width	The number of incoming channels	1-21	4
High speed Clock Frequency (Pixel Clock)	Pixel clock frequency for 7:1 LVDS interface	10-108	108

4. Click **Generate**. The **Check Generated Result** dialog box opens, showing design block messages and results as shown in Figure 5.3.

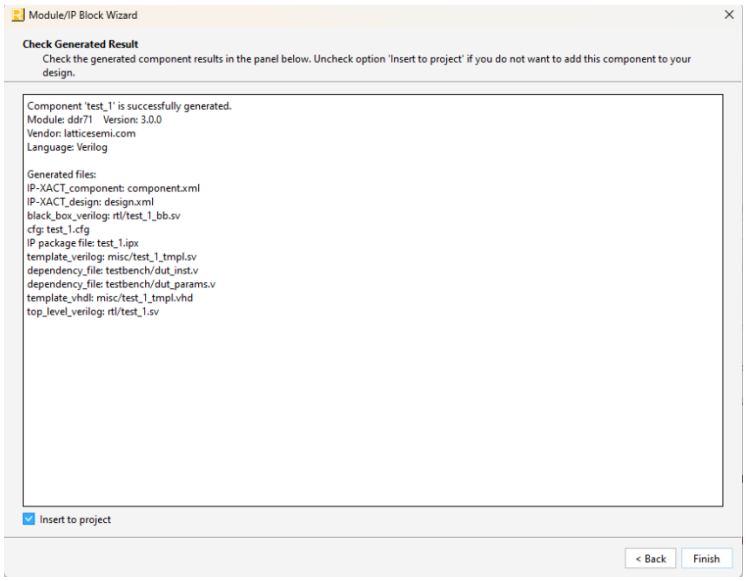


Figure 5.3. Check Generated Result

5. 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 GDDR 7:1 I/O Module package includes the closed-box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the 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 5.2.

Table 5.2. Generated File List

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

5.2. Running Functional Simulation

To run functional simulation:

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

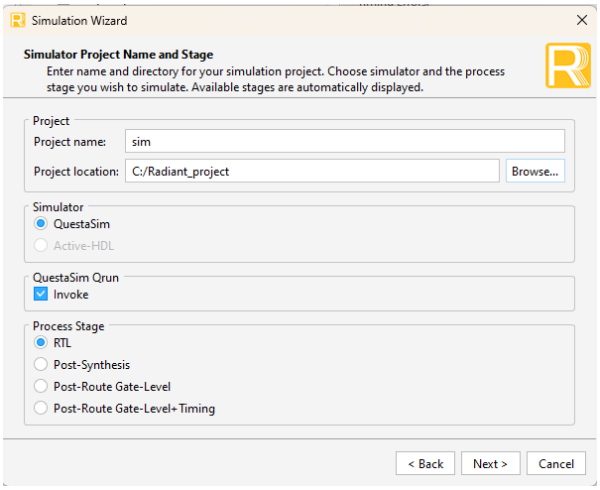


Figure 5.4. Simulation Wizard

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

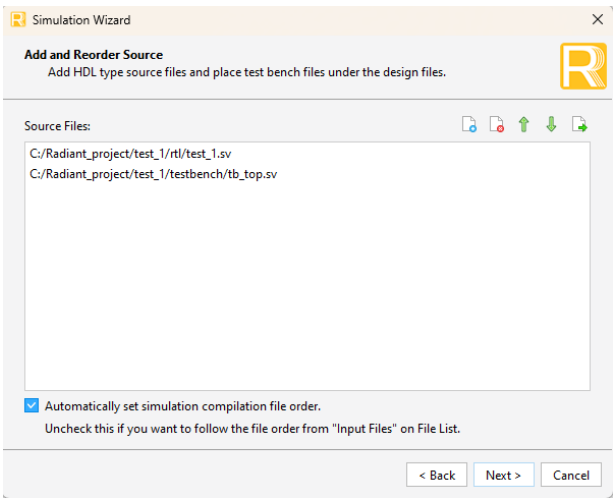


Figure 5.5. Add and Reorder Source

3. Click **Next**. The **Summary** window is shown.
4. Click **Finish** to run the simulation.

The waveform in Figure 5.6 shows an example simulation result.

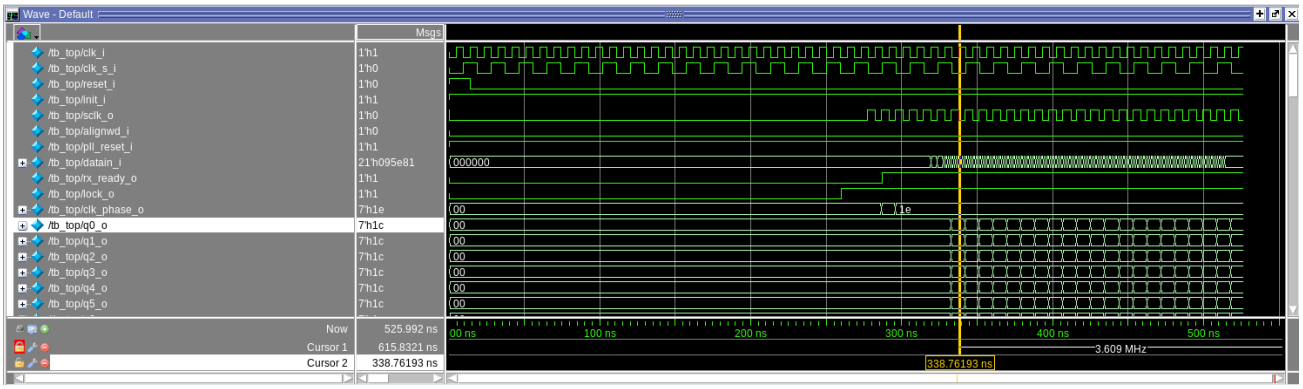


Figure 5.6. Simulation Waveform

5.2.1. Simulation Results

RX Interface Simulation

Upon successful simulation of the RX interface design, the simulator will display the following message:

```
******* RX SIMULATION PASSED ***********
```

TX Interface Simulation

Upon successful simulation of the TX interface design, the simulator will display the following message:

```
******* TX SIMULATION PASSED ***********
```

These messages serve as confirmation that the respective interface designs have passed the simulation checks defined in the IP testbench.

References

- MachXO4 DDR 7:1 Module Release Notes (FPGA-RN-02099)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- MachXO4 web page
- 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, please 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