Lattice Avant DDR 7:1 Module

User Guide

FPGA-IPUG-02189-1.4

January 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

Contents 3
Abbreviations 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. GDDRX71_RX.ECLK 9
2.2.2. RX Output Data Mapping 11
2.2.3. GDDRX71_TX.ECLK 12
2.2.4. TX Input Data Mapping 13
2.3. Signal Description 14
2.4. Attribute Summary 15
3. IP Generation, Simulation, and Validation 17
3.1. Generating the IP 17
3.2. Running Functional Simulation 20
3.3. Constraining the IP 21
3.4. IP Evaluation 22
Appendix A. Resource Utilization 23
References 24
Technical Support Assistance 25
Revision History 26

Figures

Figure 2.1. GDDR 7:1 I/O Soft IP Top-level Block Diagram 7
Figure 2.2. DDR 7:1 Timing Diagram (During Startup) 8
Figure 2.3. DDR 7:1 Timing Diagram (During Data Transaction) 8
Figure 2.4. GDDR_SYNC Timing Diagram 8
Figure 2.5. GDDRX71_RX.ECLK with GDDR_SYNC and GPLL (without BW_ALIGN) Block Diagram 9
Figure 2.6. GDDRX71_RX.ECLK with GDDR_SYNC, GPLL and BW_ALIGN Block Diagram 9
Figure 2.7. GDDRX71_RX.ECLK with GDDR_SYNC, GPLL, BW_ALIGN and Data Delay Control Block Diagram 10
Figure 2.8. Rx Output Data Mapping 11
Figure 2.9. Rx Timing Diagram 11
Figure 2.10. GDDRX71_TX.ECLK Static Delay Block Diagram 12
Figure 2.11. Tx Input Data Mapping 13
Figure 2.12. Tx Timing Diagram 14
Figure 3.1. Module/IP Block Wizard 17
Figure 3.2. Configure Block of GDDR 7:1 I/O Module 18
Figure 3.3. Check Generated Result 19
Figure 3.4. Simulation Wizard 20
Figure 3.5. Adding and Reordering Source 21
Figure 3.6. Simulation Waveform 21

Tables

Table 2.1. Available GDDR 7:1 I/O Module Interfaces 7
Table 2.2. Summary of the Soft Logic 7
Table 2.3. GDDR 7:1 I/O Module Receive Signal Description 14
Table 2.4. GDDR 7:1 I/O Module Transmit Signal Description 15
Table 2.5. Attributes Table 15
Table 2.6. Attributes Description 16
Table 3.1. Generated File List 19
Table A.1. Resource Utilization using LAV-AT-E70-3LFG1156I 23
Table A.2. Resource Utilization using LAV-AT-E70-1LFG1156I 23

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
GDDR	Generic Double Data Rate
GUI	Graphical User Interface
I/O	Input/Output
RTL	Register Transfer Level

1. Introduction

The Lattice Avant™ Generic Double Data Rate 7:1 Input/Output (GDDR 7:1 I/O) module is designed to be used mainly for flat panel display interface.

1.1. Features

Key features of the Generic Double Data Rate 7:1 Input/Output Module include:

- Receive and Transmit Interface up to 1,050 Mbps
- 1-bit to 16-bit data bus width
- Optional bit word alignment and data delay control (for 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

GDDR 7:1 I/O Module is a specific case of Generic DDR I/O Module designed to be used for flat panel display interface.

Table 2.1 provides a summary of GDDR 7:1 I/O Interfaces.

Table 2.1. Available GDDR 7:1 I/O Module Interfaces

Feature	Description	Comments
GDDRX71_RX.ECLK	Generic DDR 7:1 Receive Interface	Supports bypassed and dynamic data path delay. Optional BW_ALIGN support soft logic. Optional Data Delay Control support soft logic. Required GDDR_SYNC support soft logic.
GDDRX71_TX.ECLK	Generic DDR 7:1 Transmit Interface	Supports bypassed data path delay. Required GDDR_SYNC support soft logic.

***Notes:**
G – Generic
RX – Receive interface
TX – Transmit interface
ECLK – Uses edge clock clocking resource

The following Soft Logic modules are utilized by the Generic DDR 7:1 IP for synchronization and alignment. The GDDR_SYNC module is used by all configurations of the IP and automatically included when generating the IP. The module BW_ALIGN, on the other hand is optional but is only available when the interface of the IP is receive.

Table 2.2. Summary of the Soft Logic

Module	Description
GDDR_SYNC	Needed to tolerate large skew between stop and reset input.
BW_ALIGN	This soft IP is used to perform bit and word alignment using PLL's dynamic phase shift interface and dynamic data delay adjustment.

Figure 2.1 presents top-level diagram describing GDDR 7:1 I/O Module.

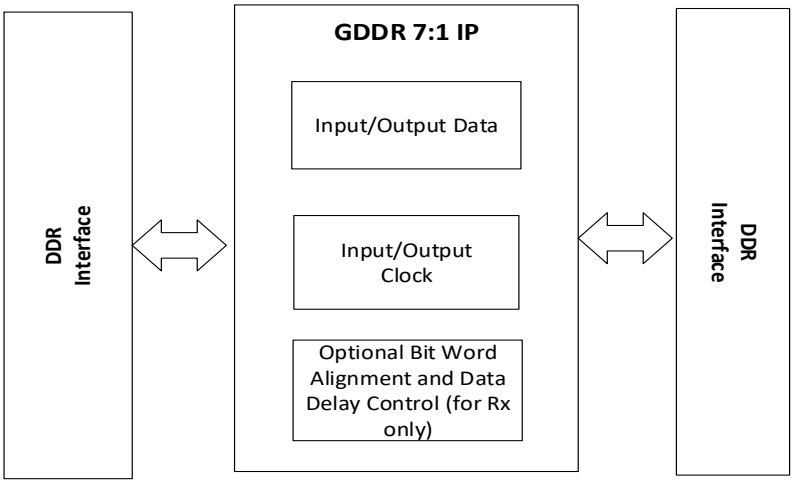


Figure 2.1. GDDR 7:1 I/O Soft IP Top-level Block Diagram

2.2. Functional Diagrams

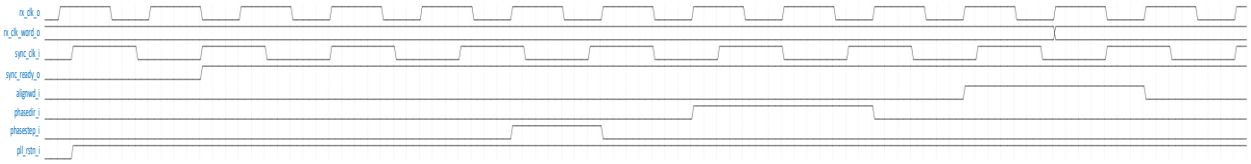


Figure 2.2. DDR 7:1 Timing Diagram (During Startup)

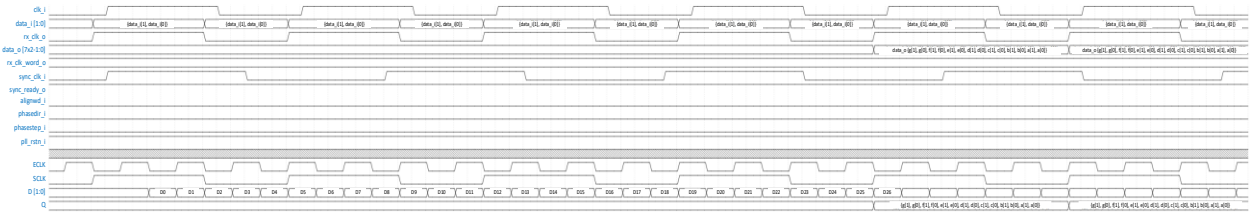


Figure 2.3. DDR 7:1 Timing Diagram (During Data Transaction)

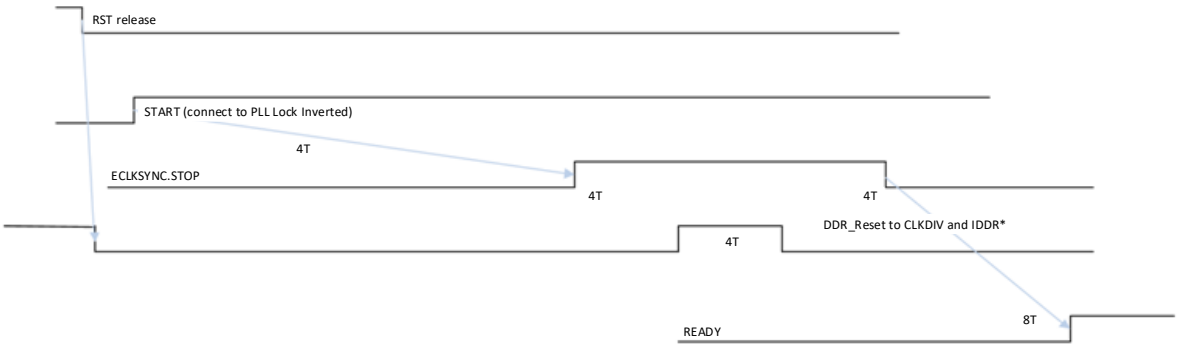


Figure 2.4. GDDR_SYNC Timing Diagram

Figure 2.2 and Figure 2.3 show the timing diagram of the signals of the DDR 7:1 IP and their relationship with one another. The configuration above uses the module GDDR_SYNC; hence, the signals sync_clk_i, sync_rst_i, sync_start_i, and sync_ready_o are present. Note that these signals are independent to the IP and are only used by the GDDR_SYNC module. During start-up, GDDR_SYNC perform STOP and DDR RESET sequences to synchronize CLKDIV and DDR elements. Once completed, the module asserts the READY signal, which is mirrored in the signal sync_ready_o to signify that the IP is ready for operation. Figure 2.3 shows the timing diagram of the GDDR_SYNC module. During data transaction phase, you can observe that the input data, data_i is level sensitive with the signal rx_clk_o. The signal rx_clk_o is the output of ECLKDIVA and is used by the IDDR171A module as the SCLK in which the output data is sampled. With every change in the level of rx_clk_o, a new data comes in. The output data, data_o, updates every sclk_o cycle and follows a first-in first-out condition, which means that the first output data came from the first input data and so on. These transactions are explained in Section 2.2.2 and Section 2.2.4. The signals phasedir_i, phasestep_o, and pll_rstn_i all came from the PLL instantiated within the IP and followed the behavior of such IP. The phasedir_i dictates which direction to shift the clock, the phasestep_o is the command to adjust the clock, and the pll_rstn_i is PLL's own reset. Note that the timing behavior may vary on the actual board.

2.2.1. GDDRX71_RX.ECLK

This is a specialized receive interface (called 7:1 LVDS, FPD-Link, or OpenLDI) using 1:7 gearing and ECLK. The incoming input clock is multiplied 3.5X using a PLL. The multiplied clock is used to capture the data at the IDDR71 receiver module.

Figure 2.5 shows a GDDR 7:1 I/O receive interface with bit and word alignment, and with data delay control options disabled.

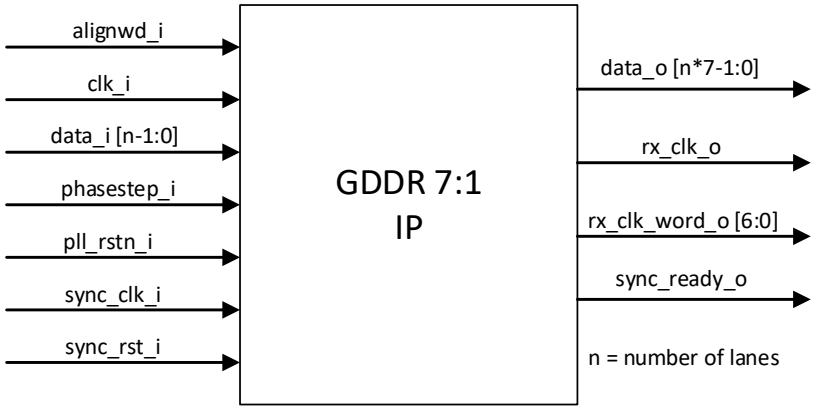


Figure 2.5. GDDRX71_RX.ECLK with GDDR_SYNC and GPLL (without BW_ALIGN) Block Diagram

The component of this configuration is described below:

- **PLL IP**, which is automatically instantiated on this IP configuration, generates the edge clock from *clk_i*. The frequency of the generated edge clock of clk_i is 3.5x clk_i, it has a dedicated reset input signal, *pll_rstn_i*. This IP configuration provides you with control over clock phase shifting through the *phasedir_i* and *phasestep_i* signals. The value of phasedir_i dictates whether to delay or advance the phase of the output clock; 0 to delay the phase, 1 to advance. To activate these changes in the output clock dictated by phasedir_i, phasestep_i should be asserted.
- **Clock IP** components provide edge clock alignment. It also derives the slow clock, *rx_clk_o*, from the edge clock divided by 3.5 to support the gearing data ratio. The divided clock is available externally, but its frequency is similar to clk_i. The only difference is rx_clk_o is aligned to the edge clock.
- **DDR Clock IP** component generates the receiver clock word as parallel output.
- **DDR Data IP** component captures the input data. The edge clock is also the clock used to capture data.
- **GDDR_SYNC Soft IP** logic performs a sequence to synchronize all the resets of the IP components after PLL lock is achieved. The synchronization occurs before the *RX* operation starts. Once completed, it asserts the *sync_ready_o* signal to indicate that an operation be initiated. This component has its own clock input, sync_clk_i, and reset input, *sync_rst_i*.

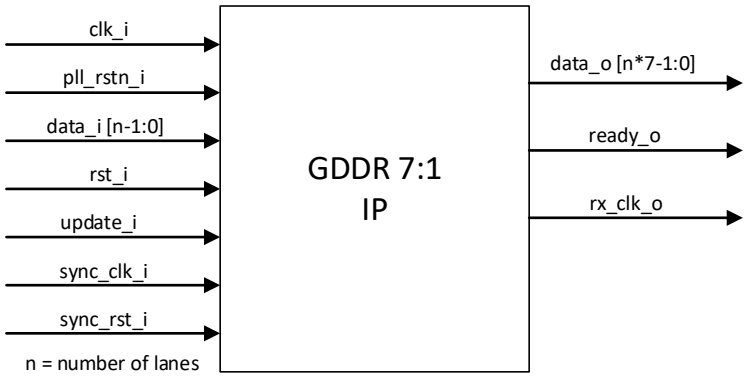


Figure 2.6. GDDRX71_RX.ECLK with GDDR_SYNC, GPLL and BW_ALIGN Block Diagram

Figure 2.6 shows the GDDR 7:1 I/O receive interface with bit and word alignment feature enabled.

When the bit and word alignment feature is enabled, the BW_ALIGN soft IP logic is added to the receive interface IP components. This component will control the PLL dynamic clock phase shifting, and DDR components will align input to perform the bit and word alignment, respectively. Notice that in this configuration the *phasedir_i*, *phasestep_i*, and *alignwd_i* signals are no longer available on the interface. During bit alignment, the PLL generated edge clock will be placed in the center of the valid window of the DDR clock and data words. Word alignment automatically controls the alignment on the DDR clock component by slipping the clock word until *7'h63* clock word value is achieved. Once synchronization and alignment are performed, the *ready_o* signal will be asserted by the IP.

Figure 2.7 shows GDDR 7:1 I/O receive interface with data delay control option enabled.

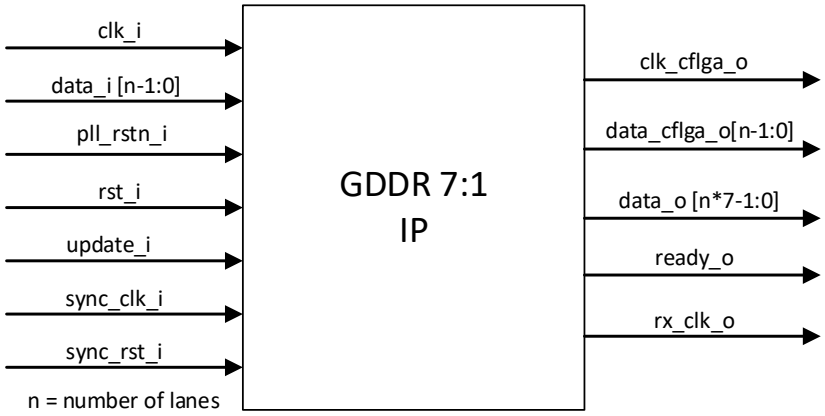


Figure 2.7. GDDRX71_RX.ECLK with GDDR_SYNC, GPLL, BW_ALIGN and Data Delay Control Block Diagram

When the data delay control option is enabled, it provides the IP automatic control to add data delay on the DDR data path. For this feature, the IP checks if the data is on the center of the edge clock. It delays the data to determine the valid window, then places the data in the center of the window. Once synchronization, alignment, and data delay are all performed, the *ready_o* signal will be asserted by the IP.

2.2.2. RX Output Data Mapping

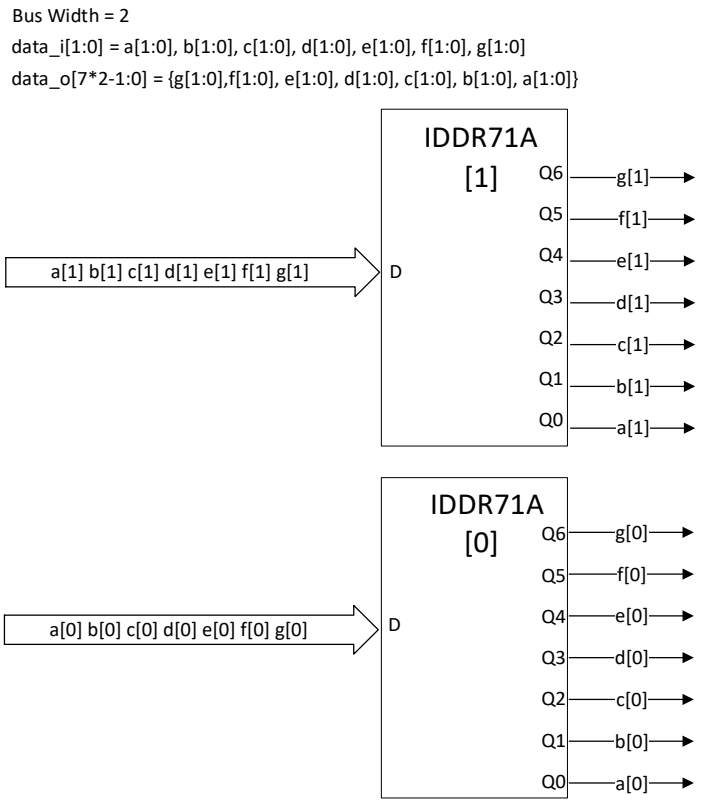


Figure 2.8. Rx Output Data Mapping

In Figure 2.8 illustrates how the IP generates two IDDR71A modules for GDDR 7:1 Rx configuration when the bus width is 2 bits. In this example for 3.5X, the number of *data_i* batches is seven. Refer to FPGA-TN-02300 Lattice Avant High Speed I/O and External Memory Interface User Guide for more details.

The first batch of incoming *data_i[a1:a0]* is captured on the rising edge of the fast clock. The next batch of *data_i[b1:b0]* is captured on the falling edge of the fastest clock and so on. The fast clock in this case is ECLK since it has the greater frequency.

In the figure, this translates to the batch of the input data's slowest bits *data_i[g1:g0]* placed on the *data_o* vector's highest bits [13:12]. Similarly, *data_i[f1:f0]* bits are placed on *data_o[11:10]* and so on. The sum of all the IDDR71 outputs is a *data_o[13:0]* vector.

Figure 2.9 shows the timing diagram of the said example.

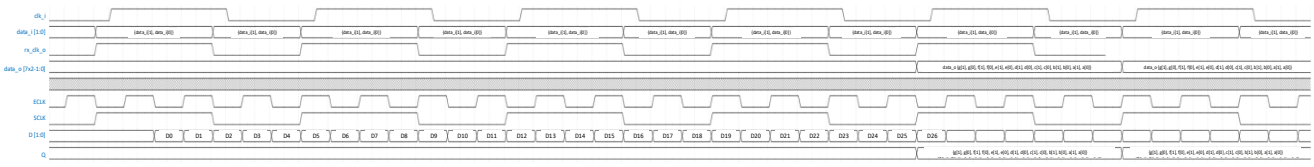


Figure 2.9. Rx Timing Diagram

2.2.3. GDDRX71_TX.ECLK

This specialized transmit interface (called 7:1 LVDS, FPD-Link, or OpenLDI) uses 7:1 gearing and ECLK. The output clock is divided by 3.5 using ECLKDIV. The divided clock is then used to capture data at the ODDR71 module. The transmit side for the 7:1 LVDS interface DDR uses 7:1 gearing with ECLK, ensuring the clock output is aligned with the data output.

Figure 2.10 describes the GDDR 7:1 I/O Transmit Interface.

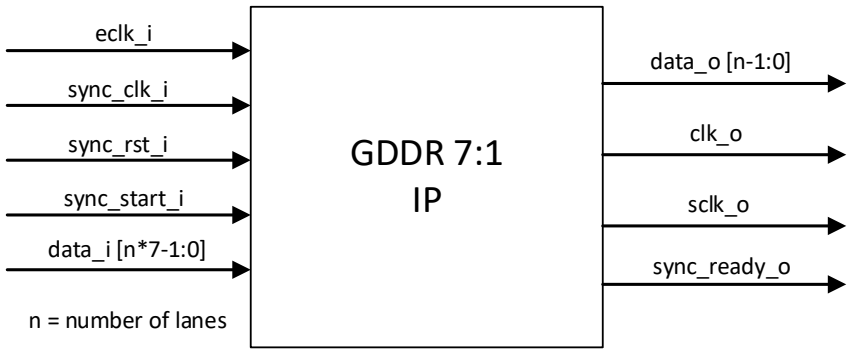


Figure 2.10. GDDRX71_TX.ECLK Static Delay Block Diagram

Described below are the components of this configuration:

- Clock IP components provides edge clock alignment. It also derives the slow clock, *sclk_o*, from the edge clock, divided by 3.5, to support the gearing data ratio. The divided clock is available externally. Similar to the output clock, clk_o is derived by dividing the edge clock by 3.5.
- DDR Clock IP component generates the receiver clock word as parallel output.
- DDR Data IP component generates the output data. Output is generated with the slow clock.
- GDDR_SYNC Soft IP logic performs sequence to synchronize all the resets of the IP components once *sync_start_i* signal is asserted. The synchronization happens before TX operation starts. When done, it asserts *sync_ready_o* signal to indicate that an operation can already be started. This component has its own clock input, sync_clk_i and reset input, *sync_rst_i*.

2.2.4. TX Input Data Mapping

Bus Width = 4
data_i [7 * 4 - 1 : 0]
data_o [4-1:0]

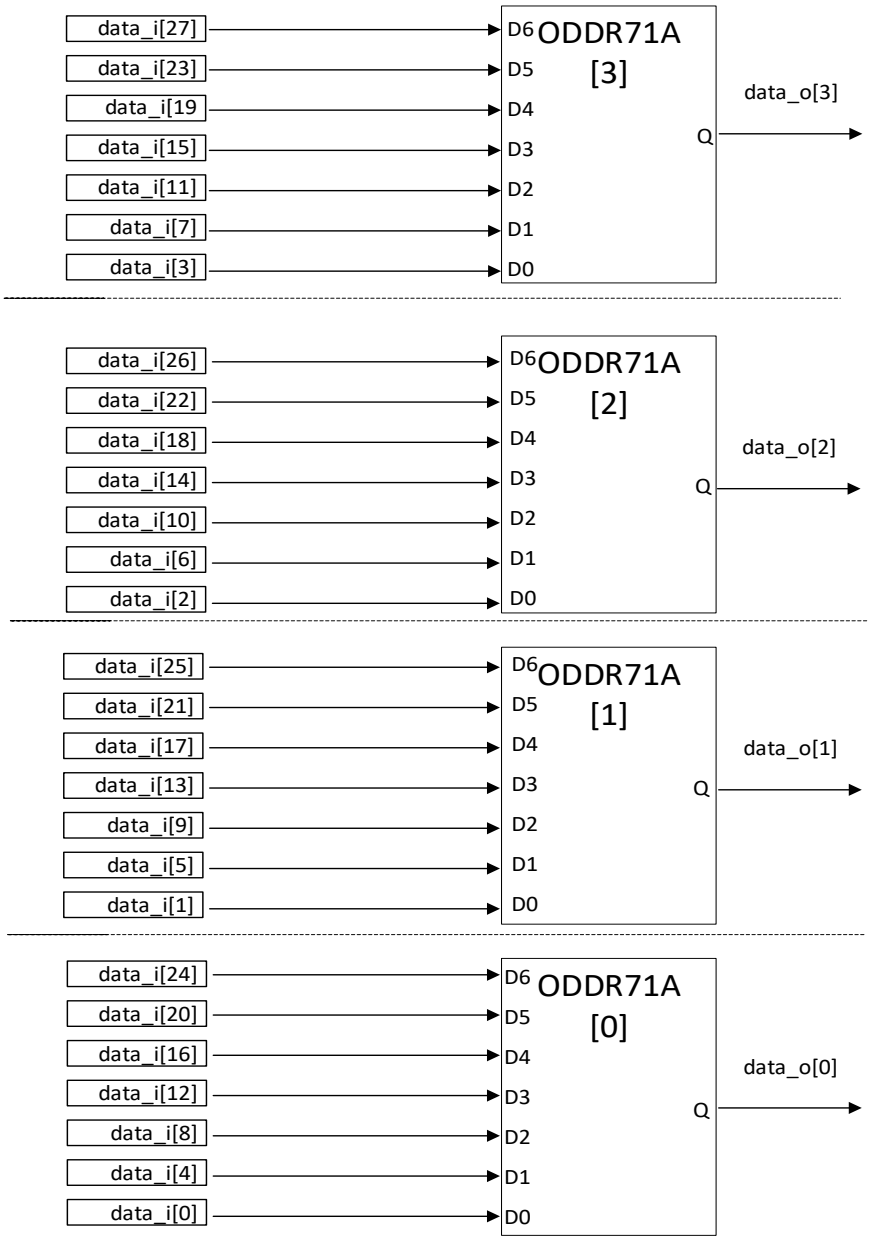
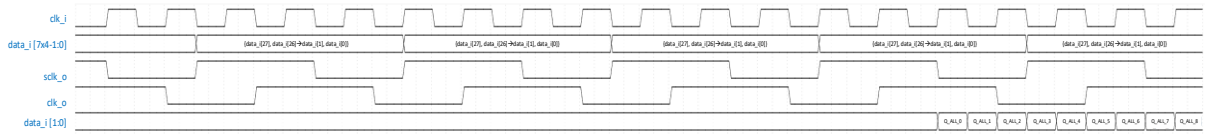


Figure 2.11. Tx Input Data Mapping

As shown in Figure 2.11, for GDDR 7:1 Tx configuration, when the bus width is 4 bits, the IP generates four ODDR71A modules. The entire input data is broken down into 7 groups, each group with the size of 4 bits. The group with the fastest bits *data_i[3:0],* is transmitted first, followed by *data_i[7:4]* and so on.

Refer to FPGA-TN-02300 Lattice Avant High Speed I/O and External Memory Interface User Guide for more details.

Figure 2.12 shows the timing diagram of the said example.



Where: Q_ALL = {Q[3], Q[2], Q[1], Q[0]}

Figure 2.12. Tx Timing Diagram

2.3. Signal Description

Table 2.3. GDDR 7:1 I/O Module Receive Signal Description

Port Name	Direction	Width(bits)	Description
Clock and Reset			
rx_clk_o	OUT	1	Clock output for receive interface. Divided RX clock from 7:1 RX interface produced by ECLKDIV. This is only available when *Enable Bit and Word Alignment Soft IP* is enabled.
sync_clk_i	IN	1	Startup clock. A continuously running clock with a low speed input. Its frequency depends on the input clock but must be significantly lower.
sync_rst_i	IN	1	Active high reset signal. It can be used asynchronously with the input clock, but it is recommended to de-asserted it synchronously with sync_clk_i.
rst_i	IN	1	Active high reset signal when *Enable Bit and Word Alignment Soft IP* is enabled.
User Interface			
pll_rstn_i	IN	1	Active low reset of internal PLL. This is available only when *Interface Type* is receive.
phasedir_i	IN	1	Phase rotation direction of internal PLL. This is available only when *Interface Type* is receive, *Enable Bit and Word Alignment Soft IP* is disabled, and *Enable Data Delay Control* is disabled.
phasestep_i	IN	1	Rotate phase of internal PLL. This is available only when *Interface Type* is receive, *Enable Bit and Word Alignment Soft IP* is disabled, and *Enable Data Delay Control* is disabled.
update_i	IN	1	Start bit and word alignment, or restart the procedure if optimization is needed again. This is available only when *Interface Type* is receive and *Enable Bit and Word Alignment Soft IP* is enabled.
alignwd_i	IN	1	This signal is used for word alignment. It shifts word by one bit. This is only available when *Enable Bit and Word Alignment Soft IP* is disabled.
data_o	OUT	n*7	Received input data to fabric.
sync_ready_o	OUT	1	Indicate that startup is finished, and RX circuit is ready to operate. Available only when *Enable Bit and Word Alignment Soft IP* is disabled.
ready_o	OUT	1	Indicates alignment is done, startup is finished, and RX circuit is ready to operate. This is only available when *Enable Bit and Word Alignment Soft IP* is enabled.

Port Name	Direction	Width(bits)	Description
rx_clk_word_o	OUT	7	Valid receiver clock word size allowance. Parallel data output. Available only when *Enable Bit and Word Alignment Soft IP* is disabled.
data_cflag_o	OUT	n	Underflow or overflow flag to indicate the minimum or maximum data path delay adjustment is reached. This is available only when *Interface Type* is receive, *Enable Bit and Word Alignment Soft IP* is enabled, and *Enable Data Delay Control* is enabled.
clk_cflag_o	OUT	1	Underflow or overflow flag to indicate the minimum or maximum clock path delay adjustment is reached. This is available only when *Interface Type* is Receive, *Enable Bit and Word Alignment Soft IP* is enabled, and *Enable Data Delay Control* is enabled.
I/O Pad Interface			
clk_i	IN	1	Clock input signal from I/O.
data_i	IN	n	Data input signal from I/O.

Note:

1. n = number of lanes.

Table 2.4. GDDR 7:1 I/O Module Transmit Signal Description

Port Name	Direction	Width(bits)	Description
Clock and Reset			
eclk_i	IN	1	Transmit data sampling clock.
sclk_o	OUT	1	Clock output for *Interface Type* is transmit.
sync_clk_i	IN	1	A continuously running clock with a low-speed input.
sync_rst_i	IN	1	Active high reset signal.
User Interface			
data_i	IN	n*7	Transmit output data going to I/O.
sync_ready_o	OUT	1	Indicates that startup is finished, and TX circuit is ready to operate.
sync_start_i	IN	1	Waits for the PLL lock signal to start the synchronization.
I/O Pad Interface			
clk_o	OUT	1	Clock output signal to I/O.
data_o	OUT	n	Data output signal to I/O.

Note:

1. n = number of lanes

2.4. Attribute Summary

Table 2.5 provides a list of user-configurable attributes for the GDDR 7:1 I/O Module. Attributes settings are specified using the GDDR 7:1 I/O Module Configuration user interface in Lattice Radiant™ software.

Table 2.5. Attributes Table

Attribute	Selectable Values	Default	Dependency on other Attributes	Additional Requirements
Interface Type	Receive, Transmit	Receive	—	—
Bus Width	1–16	8	—	—
Interface Bandwidth (Mbps)	70-1050 (Transmit) 110-1050 (Receive)	945	—	—
Clock Frequency (MHz)	10–150 (Transmit) 16- 150 (Receive)	135	*Bandwidth/7*	Display for information only
Enable Bit and Word Alignment Soft IP	Checked, Unchecked	Unchecked	*Interface Type = Receive*	—

Attribute	Selectable Values	Default	Dependency on other Attributes	Additional Requirements
Enable Data Delay Control	Checked, Unchecked	Unchecked	*Interface Type = Receive* and *Enable Bit and Word Alignment Soft IP* is checked	—

Note:

1. The attributes can be configured from the General Tab of the Lattice Radiant software user interface.

Table 2.6. Attributes Description

Attribute Name	Description
Interface Type	RECEIVE or TRANSMIT interface type
Bus Width	Total number of lanes/bus width
Interface Bandwidth (Mbps)	Interface clock frequency
Enable Bit and Word Alignment Soft IP	Optional bit and word alignment Soft IP (BW_ALIGN) module. The bit alignment module centers the edge clock to the middle of the data eye. The word alignment module is used to achieve the 7-bit word alignment.
Enable Data Delay Control	When enabled, BW_ALIGN Soft IP module automatically controls the movement of the data on DELAY module.

3. IP Generation, Simulation, and Validation

This section provides information on how to generate the IP 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.

3.1. Generating the IP

The Lattice Radiant software allows you to customize, generate modules, and IPs and integrate them into the device's architecture. The procedure for generating the GDDR 7:1 I/O module in Lattice Radiant software is described below.

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **GDDR 7:1** under **Module, Architecture_Modules, I/O** 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**.

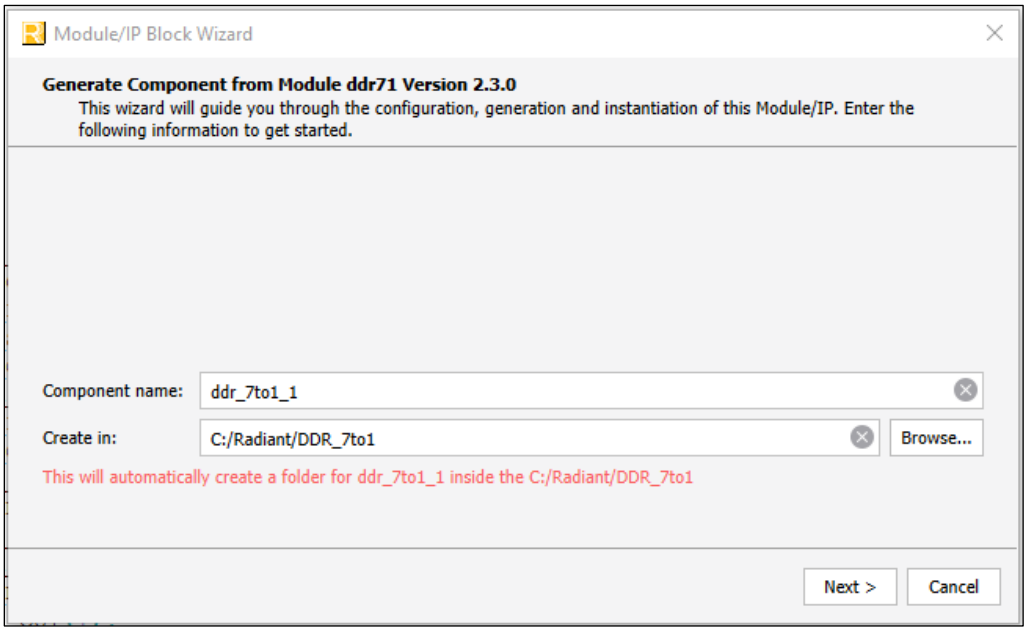


Figure 3.1. Module/IP Block Wizard

3. In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected GDDR 7:1 I/O module using the drop-down menu and check box. As see Figure 3.2 sample configuration. For configuration options, see the Attribute Summary section.

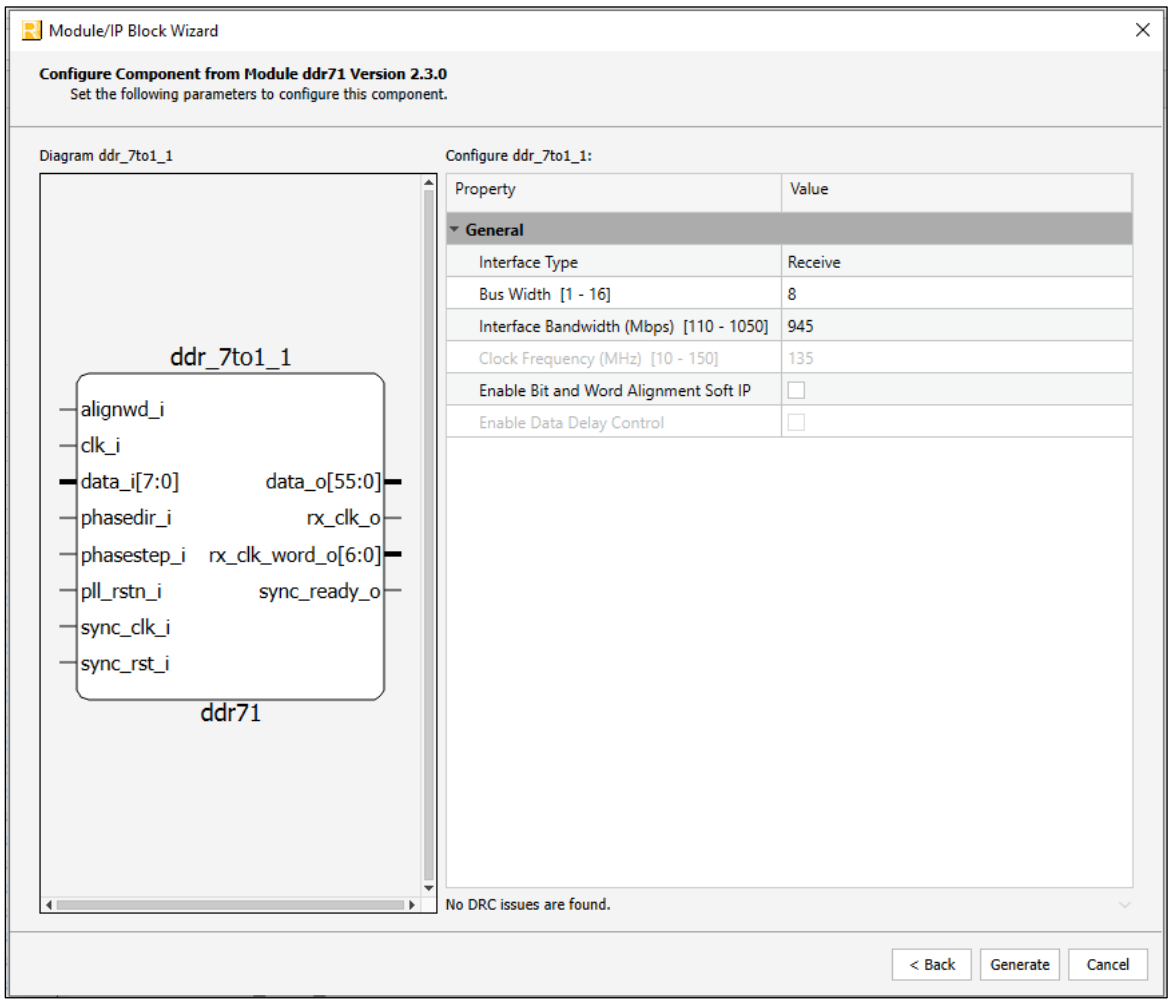


Figure 3.2. Configure Block of GDDR 7:1 I/O Module

4. Click **Generate**. The **Check Generated Result** dialog box opens, showing design block messages and results. See Figure 3.3 sample.

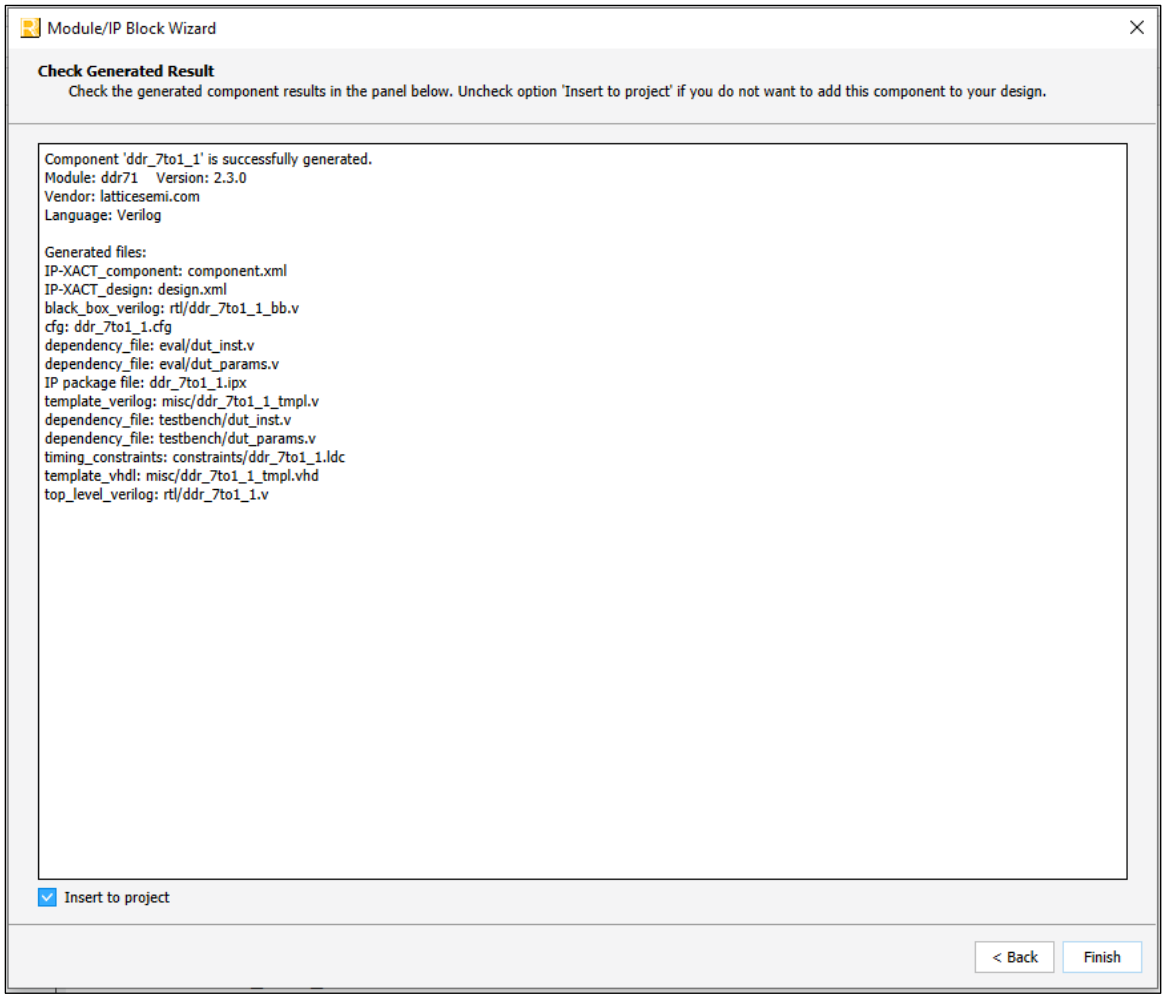


Figure 3.3. Check Generated Result

5. Click the **Finish** button. All the generated files are located in the directory paths specified in the **Create in** and **Component name** fields shown in Figure 3.1.

The generated GDDR 7:1 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 with the generated IP.
<Instance Name>.cfg	This file contains the parameter values used in IP configuration.
component.xml	Contains the ip-xact 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.

Attribute	Description
misc/<Instance Name>_tmpl.v misc /<Instance Name>_tmpl.vhd	These files provide instance templates for the module.
eval/constraints.pdc	This file provides information on how to constrain the IP in your design. Refer to the Constraining the IP section on how to use this file.
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

3.2. 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.

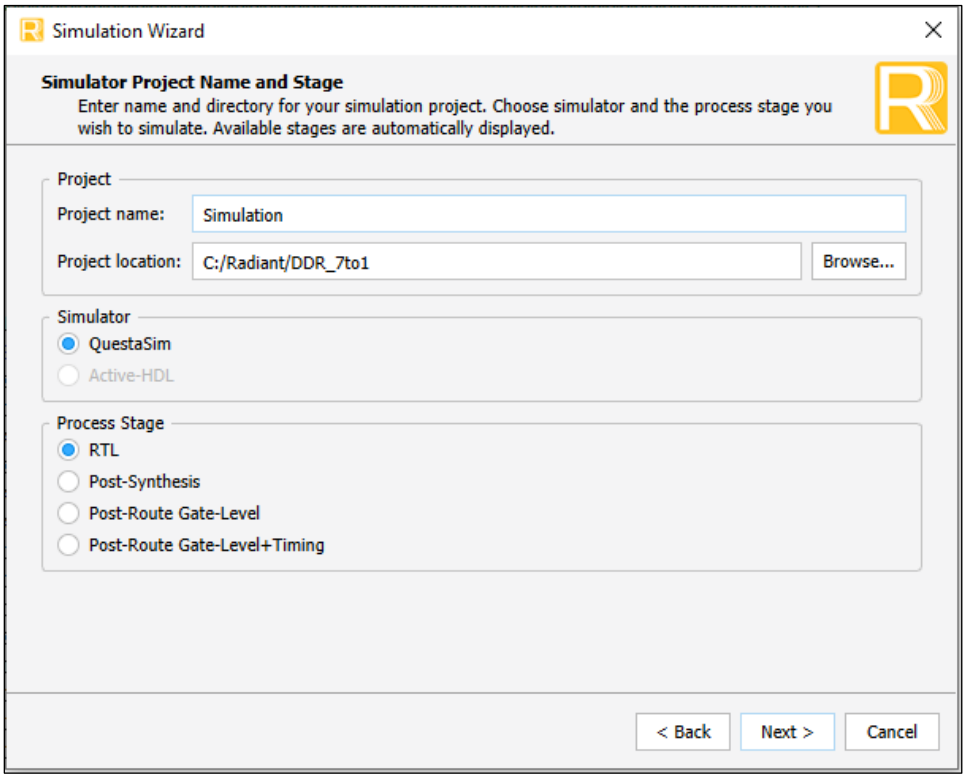


Figure 3.4. Simulation Wizard

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

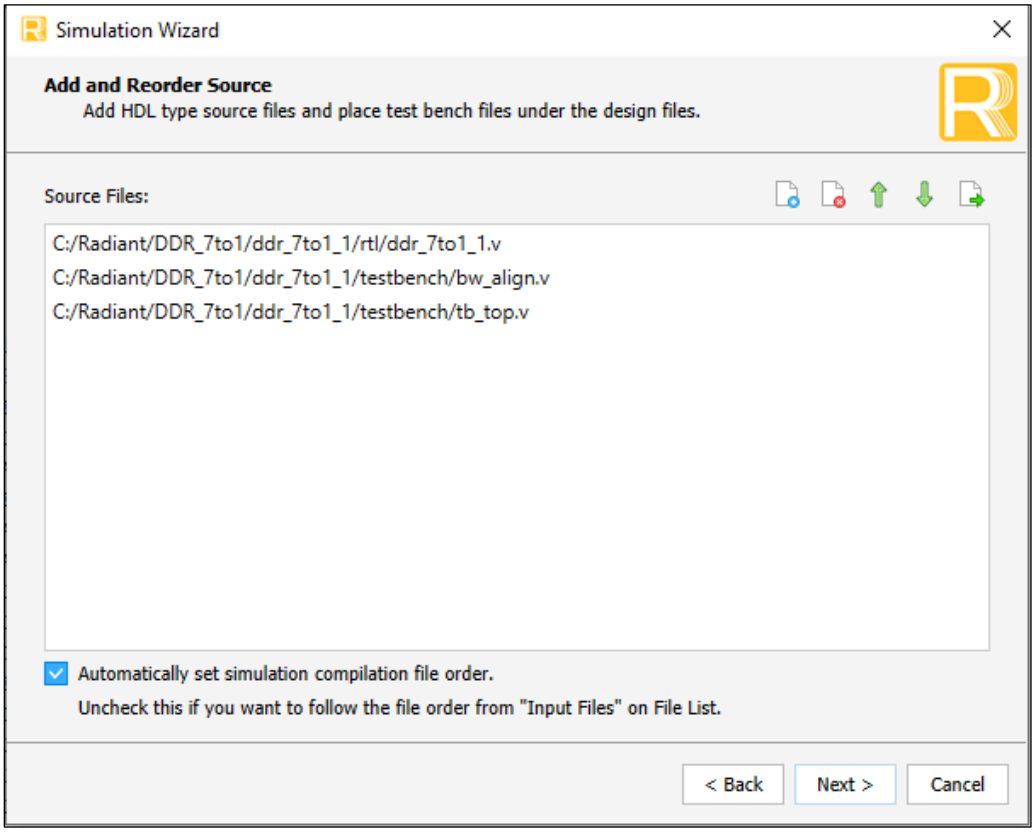


Figure 3.5. Adding and Reordering Source

3. Click **Next**. The Summary window is shown. 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.

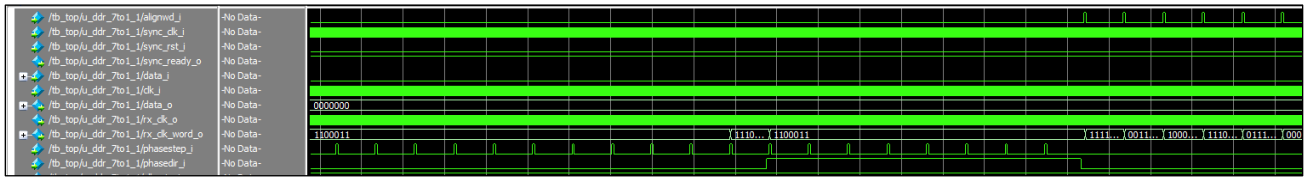


Figure 3.6. Simulation Waveform

3.3. 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: *<IP_Instance_Path>/<IP_Instance_Name>/eval/constraints.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 a thorough understanding of the effect of each constraint.

To use this constraint file, copy the content of *constraints.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.

3.4. IP Evaluation

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

Appendix A. Resource Utilization

DDR 7:1 module resource utilization is shown in Table A.1 and in Table A.2 using LAV-AT-E70-3LFG1156I, and LAV-AT-E70-1LFG1156I devices with Synplify Pro® of Lattice Radiant software. The 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 using LAV-AT-E70-3LFG1156I

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs[2]	EBRs	DSPs
Interface type is receive Interface bandwidth is 1,050 Mbps, others are default	150	12	25	0	0
Interface type is transmit, Interface bandwidth is 1050Mbps, others are default	150	12	25	0	0

Notes:

1. Fmax is generated when the FPGA design only contains the DDR7:1 module and the target frequency is 150 MHz. DDR7:1 module supports up to 1,050 MHz ECLK and 150 MHz SCLK only. 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 distributed among logic, distributed RAM, and ripple logic.

Table A.2. Resource Utilization using LAV-AT-E70-1LFG1156I

Configuration	Clk Fmax (MHz)[1]	Registers	LUTs[2]	EBRs	DSPs
Interface Type is receive, Interface bandwidth is 1,050 Mbps, others are default	150	12	25	0	0
Interface Type is transmit, Interface bandwidth is 1,050 Mbps, others are default	150	12	25	0	0

Notes:

1. Fmax is generated when the FPGA design only contains the DDR7:1 module and the target frequency is 150 MHz. DDR7:1 module supports up to 1,050 MHz ECLK and 150 MHz SCLK only. 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 distributrd among logic, distributed RAM, and ripple logic.

References

- Lattice Avant High-Speed I/O and External Memory Interface User Guide (FPGA-TN-02300)
- Lattice Avant sysCLOCK PLL Design and User Guide (FPGA-TN-02298)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Avant-E web page
- Lattice Avant Platform
- Lattice Radiant Software User Guide
- Lattice Radiant FPGA Design software
- 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 https://www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.4, January 2025

Section	Change Summary
All	Minor editorial fixes.
Abbreviation in This Document	Replaced *Acronyms* with *Abbreviations*.
Functional Description	Updated Figure 2.5. GDDRX71_RX.ECLK with GDDR_SYNC and GPLL (without BW_ALIGN) Block Diagram.

Revision 1.3, June 2024

Section	Change Summary
All	• Updated the document title from *Lattice Avant DDR 7:1 Module - Lattice Radiant Software* to *Lattice Avant DDR 7:1 Module*. • Made editorial fixes. • Added missing figure captions.
Functional Description	• Added description to Table 2.2 Summary of the Soft Logic. • Added Figure 2.2. DDR 7:1 Timing Diagram (During Startup) - Figure 2.4. GDDR_SYNC Timing Diagram and updated the figure numbers of remaining figures accordingly. • Updated the description for Figure 2.5. GDDRX71_RX.ECLK with GDDR_SYNC and GPLL (without BW_ALIGN) Block Diagram. • Updated the description for section GDDRX71_TX.ECLK. • Added Figure 2.12. Tx Timing Diagram and updated the description. • Updated the description for Figure 2.10. GDDRX71_TX.ECLK Static Delay Block Diagram. • Updated the description for the following ports *sync_clk_i* and *sync_rst_i* in Table 2.3. GDDR 7:1 I/O Module Receive Signal Description. • Updated the *Selectable Values* for the following attributes *Interface Bandwidth (Mbps)* and *Clock Frequency (Mhz)* in Table 2.5. Attributes Table. • Removed Figure 2.7 GDDRX71_TX.ECLK Static Delay Interface.
IP Generation, Simulation, and Validation	• Replaced Figure 3.1. Module/IP Block Wizard - Figure 3.5. Adding and Reordering Source. • Updated the Running Functional Simulation section.

Revision 1.2, December 2023

Section	Change Summary
All	Minor adjustments to ensure the document is consistent with Lattice Semiconductor's inclusive language policy.
Disclaimers	Updated boilerplate.
Inclusive Language	Added boilerplate.
Functional Description	• Changed the description of module BW_ALIGN in Table 2.2 from dynamic DELAY adjustment to dynamic data delay adjustment. • Updated Figure 2.1. Updated the paragraphs of GDDRX71_RX.ECLK section. Removed *Figure 2.3*, *Figure 2.5*, *Figure 2.7*, and *Figure 2.8* from GDDRX71_RX.ECLK section. Updated the paragraphs of GDDRX71_TX.ECLK section.
IP Generation, Simulation, and Validation	• Updated Figure 3.1, Figure 3.2, and Figure 3.3 in Generating the IP section. • Added attribute *eval/constraints.pdc* and its description to Table 3.1. • Added Constraining the IP section.
Resource Utilization	Updated the devices name from *LAV-AT-500E-3LFG1156I* to *LAV-AT-E70-3LFG1156I*, and from *LAV-AT-500E-1LFG1156I* to *LAV-AT-E70-1LFG1156I*.
References	Added link to the Lattice Insights web page.

Revision 1.1.1, May 2023

Section	Change Summary
All	Changed document title to *Lattice Avant DDR 7:1 Module - Lattice Radiant Software*.
Acronyms in This Document	Corrected GDDR definition.
Functional Description	Updated the following figures: Figure 2.3. GDDRX71_RX.ECLK with GDDR_SYNC and GPLL (without BW_ALIGN) Interface Figure 2.5. GDDRX71_RX.ECLK with GDDR_SYNC, GPLL and BW_ALIGN Interface Figure 2.7. GDDRX71_RX.ECLK with GDDR_SYNC, GPLL, BW_ALIGN and Data Delay Control Interface Figure 2.11. GDDRX71_TX.ECLK Static Delay Interface
IP Generation, Simulation, and Validation	Updated Figure 3.1. Module/IP Block Wizard, Figure 3.2. Configure Block of GDDR 7:1 I/O Module, and Figure 3.3. Check Generated Result.
Technical Support Assistance	Added reference to the Lattice Answer Database on the Lattice website.
All	Minor adjustments in formatting and style.

Revision 1.1, November 2022

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

Revision 1.0, November 2022

Section	Change Summary

Functional Description	In Table 2.3. GDDR 7:1 I/O Module Receive Signal Description:
	Revised the description for the signal sync_clk_i
	• Added the following User Interface signals and their details:
	  • pll_rstn_i
	  • phasedir_i
	  • phasestep_i
	  • update_i
	  • data_cflag_o
	  • clk_cflag_o

IP Generation, Simulation, and Validation	• Revised the title from 'IP Generation and Evaluation' to 'IP Generation, Simulation, and Validation' • Deleted the section 'Licensing the IP' • Revised the title of section 3.1 from 'Generation and Synthesis' to 'Generating the IP' • Revised the figures Figure 3.1. Module/IP Block Wizard, Figure 3.2. Configure Block of GDDR 7:1 I/O Module, and Figure 3.3. Check Generated Result • Revised the title of section 3.3 from 'Hardware Evaluation' to 'IP Evaluation'

Revision 0.80, May 2022

Section	Change Summary
All	Initial release

www.latticesemi.com