DDR3 SDRAM Controller IP Core for Nexus Devices

IP Version: v2.3.1

User Guide

FPGA-IPUG-02086-2.3

April 2026

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.

Abbreviation	Definition
SDRAM	Synchronous Dynamic Random Access Memory
APB	Advanced Peripheral Bus
BL	Burst Length
CA	Column Address
CS	Chip Select
DDR3	Double Data Rate Generation 3
DFI	DDR PHY
DM	Data Mask
DQ	Data
DQS	Data Strobe
ECC	Error Correction Code
ECLK	Edge Clock
FPGA	Field Programmable Gate Array
I/F	Interface
JEDEC	Joint Electron Device Engineering Council
MC	Memory Controller
MR	Mode Register
MRS	Mode Register Set
ODT	On-Die Termination
PRBS	Pseudorandom Binary Sequence
PVT	Process, Voltage, and Temperature
RTL	Register Transfer Level
SCLK	System Clock
SDR	Single Data Rate
SDRAM	Synchronous Dynamic Random Access Memory
SSTL	Stub-Series Terminated Logic
TCL	Tool Command Language
VREF	Voltage Reference

1. Introduction

The Lattice Double Data Rate Synchronous Dynamic Random Access Memory (DDR3 SDRAM) Controller IP Core is a general-purpose memory controller that interfaces with industry standard DDR3 memory devices compliant with JESD79-3C, DDR3 SDRAM Standard. This IP provides a generic command interface to user applications.

DDR3 SDRAM Controller IP reduces the effort required to integrate the DDR3 memory controller with the user application design and minimizes the need to directly deal with the DDR3 memory interface.

1.1. Quick Facts

The following table presents a summary of the DDR3 Memory Controller for Nexus Devices.

Table 1.1. Quick Facts

IP Requirements	Supported Devices	CrossLink™-NX (LIFCL-40), Certus™-NX (LFD2NX-15, LFD2NX-25, LFD2NX-28, LFD2NX-40), MachXO5™-NX (LFMXO5-15D, LFMXO5-15D-AQA, LFMXO5-15D-HBN, LFMXO5-25, LFMXO5-55T, LFMXO5-55TD, LFMXO5-100T), CertusPro™-NX
	IP Changes[1]	For a list of changes to the IP, refer to the DDR3 SDRAM Controller IP Release Notes (FPGA-RN-02032).
Resource Utilization	Supported User Interfaces	Data Access (Native or AXI4) Configuration Access (Native or APB)
	Resources	Refer to Table A.1
Design Tool Support	Lattice Implementation[2]	IP Core v2.3.1 – Lattice Radiant Software 2025.2.1
	Synthesis	Lattice Synthesis Engine Synopsys® Synplify Pro for Lattice
	Simulation	Questasim
Driver Support	API Reference	Refer to the Nexus DDR3 Memory Controller Driver API Reference (FPGA-TN-02401).

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.
2. Lattice Implementation indicates the IP version release coinciding with the software version release. Check the software for IP version compatibility with earlier or later software versions.

1.2. IP Support Summary

Table 1.2. DDR3 SDRAM Controller IP Support Readiness

Device Family	IP	Rank	Gearing Ratio	DDR Width	Data Rate (Mbps)	Radiant Timing Model
Nexus	DDR3	Single	4:1	x8	600	Preliminary
					666	
					800	
				x16	600	
					666	
					800	
				x24	600	
					666	
					800	
				x32	600	
					666	
					800	

Device Family	IP	Rank	Gearing Ratio	DDR Width	Data Rate (Mbps)	Radiant Timing Model
			8:1	x8	600	
					666	
					800	
					1,066	
				x16	600	
					666	
					800	
					1,066	
				x24	600	
					666	
					800	
					1,066	
				x32	600	
					666	
					800	
					1,066	

The DDR3 SDRAM Controller IP Core example design supports both simulation and deployment in development boards. For instructions on running the example design in hardware or simulation environments, see Designing and Simulating the IP section.

1.3. Features

The key features of DDR3 SDRAM Controller IP Core include:

- Memory data path widths of 8, 16, 24, 32 bits
- Selectable gearing ratios: 4:1, 8:1
- x8 and x16 device configurations
- Programmable burst lengths of 8 (fixed), chopped 4 or 8 (on-the-fly), or chopped 4 (fixed)
- Programmable read and write CAS latency set
- Read burst type of nibble sequential or interleave
- Automatic DDR3 SDRAM initialization and refresh
- Automatic write levelling for each DQS
- Automatic read training for each DQS
- Power Down mode with no DRAM data retention
- Dynamic On-Die Termination (ODT) controls
- I/O primitives manage read skews (read levelling equivalent)
- Automatic programmable interval refresh or user-initiated refresh
- Periodic Voltage/Temperature (VT) compensation of I/O delay
- Option for controlling memory reset outside the controller

The DDR3 SDRAM Controller IP Core supports the following devices:

- Interfaces to industry standard DDR3 SDRAM components and modules compliant with JESD79-3C, DDR3 SDRAM Standard
- Interfaces to DDR3 SDRAM at speeds up to 533 MHz/1,066 Mbps

Table 1.3. Features Overview

Key Features	DDR3 Support Details
Device Format	Component
Data Widths	X8, x16, x24[1], x32
Data User Interface	NATIVE, AXI4
Configuration Interface	NATIVE, APB
Maximum Command Speed	533 Mbps
Maximum Data Speed	1,066 Mbps
HW Managed Periodic Events[2]	
Refresh	All bank auto refresh
ZQ Calibration	Yes
Low Power Features	Self-refresh with power down
Periodic VT compensation of I/O delay	Yes
Other Features[2]	
Error Correction Code (ECC)	No
Dual rank	No
Data Bus Inversion (DBI)	No
Temperature Tracking	Yes
Refresh Adaptation (derate) to Temperature Variation	Yes
On-Die Termination (ODT)	Yes
Training[2]	
Initialization	Yes
Command Training	No
Write Leveling	Yes
Read Training	Yes
Write Training	Yes
VREF Training	No

Notes:

1. Data Width x24 is supported only when data user interface is configured as NATIVE.
2. *Yes* implies that a configurable option exists to enable or disable the feature. *No* implies that the feature is currently not supported.

1.4. Licensing and Ordering Information

The DDR3 Memory Controller IP for Nexus Devices is provided at no additional cost with the Lattice Radiant Software.

1.5. Minimum Device Requirements

The following table summarizes the minimum device requirements for the Memory Controller IP Core.

Table 1.4. Minimum Device Requirements

DDR3 Interface Speed	Gearing Ratio	DDR3 Data Width	Supported Speed Grades	
			Certus-NX, CrossLink-NX and MachXO5-NX	CertusPro-NX
300 MHz (600 Mbps) to 333 MHz (666 Mbps)	4:1	x8, x16, x24, x32	8, 9	7, 8, 9
400 MHz (800 Mbps)	4:1	x8, x16, x24, x32	9	9
300 MHz (600 Mbps) to 400 MHz (800 Mbps)	8:1	x8, x16, x24, x32	7, 8, 9	7, 8, 9
533 MHz (1,066 Mbps)	8:1	x8, x16, x24, x32	8, 9	8, 9

1.6. Naming Conventions

This section provides information regarding terminology used within this document.

1.6.1. Nomenclature

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

1.6.2. 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

1.7. Limitations

- The AXI interface supports a maximum burst length of 64 beats.
- The AXI interface lacks a dedicated clock; the output clock port is sclk_o.
- The Lattice LSE synthesis engine causes difficulty in closing timing, especially for the eval_top's RISC-V and its interfacing modules. The DDR3 SDRAM Controller IP Core closes timing correctly.
- The Example Design does not support (both on hardware and simulation) having Gearing Ratio set to 4:1 and Burst Length set to On-the-fly. However, this configuration is a supported feature of the IP.

2. Functional Description

2.1. IP Architecture

The DDR3 memory controller consists of four submodules: AXI4/APB Bridge Interface Module, Soft Memory Controller (MC) module, Soft Physical Interface (PHY) module and Soft Training Engine. The following subsections briefly describe the operation of each submodule. Figure 2.1 provides a high-level block diagram illustrating the main functional blocks used to implement the DDR3 SDRAM Controller IP Core functions.

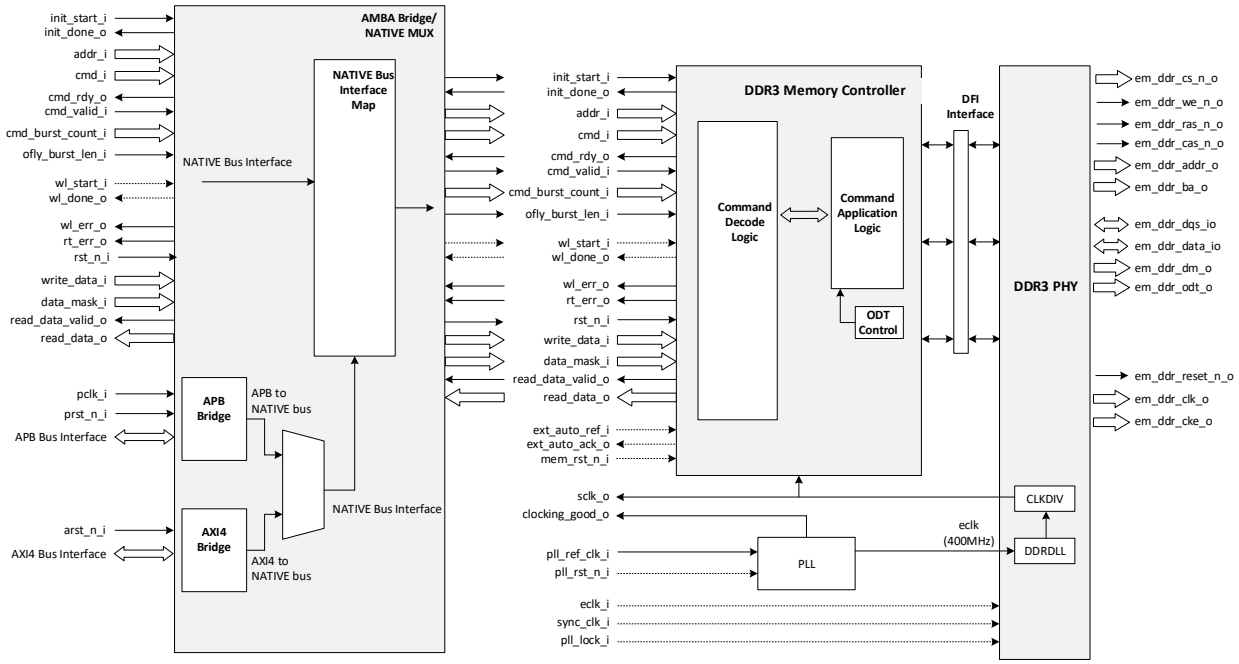

Figure 2.1. Memory Controller IP Core Functional Diagram

2.1.1. AXI4/APB Bridge Interface

The DDR3 memory controller contains the APB bridge interface for handling configuration access and AXI4 bridge interface for handling data access.

2.1.1.1. APB Bridge Interface

This interface module implements the APB slave interface, handling all configuration-related operations. This involves initialization and executing user commands. You can access the status register via the APB interface module. This interface runs on its own clock and reset. For further details, refer to the APB section.

2.1.1.2. AXI4 Bridge Interface

This interface module implements the AXI4 slave interface, handling write and read operations. This interface runs on SCLK but has a separate reset from the system reset. For further details, refer to the AXI4 section.

2.1.2. Soft Memory Controller

The DDR3 Memory Controller module contains Command Decode Logic (CDL) block, the Command Application Logic (CAL) block, and On-Die Termination (ODT) Control block.

2.1.2.1. Command Decode Logic

The Command Decode Logic (CDL) block accepts user commands from the local interface and decodes them to generate a sequence of internal memory commands based on the current command and the status of current bank and row. The intelligent bank management logic tracks the open/close status of every bank and stores the row address of every opened bank. The controller implements a command pipeline to improve throughput. With this capability, the next command in the queue is decoded while the current command is presented at the memory interface.

2.1.2.2. Command Application Logic

The Command Application Logic (CAL) block accepts the decoded internal command sequence from the Command Decode Logic and translates each sequence into memory commands that meet the operational sequence and timing requirements of the memory device. The CDL and CAL blocks work in parallel to fill and empty the command queue, respectively.

2.1.2.3. On-Die Termination Control

The ODT feature improves the signal integrity of the memory channel by allowing the DDR3 SDRAM Controller IP Core to independently turn on or off the termination resistance for any or all DDR3 SDRAM devices.

2.1.3. Soft PHY and Soft Training Engine

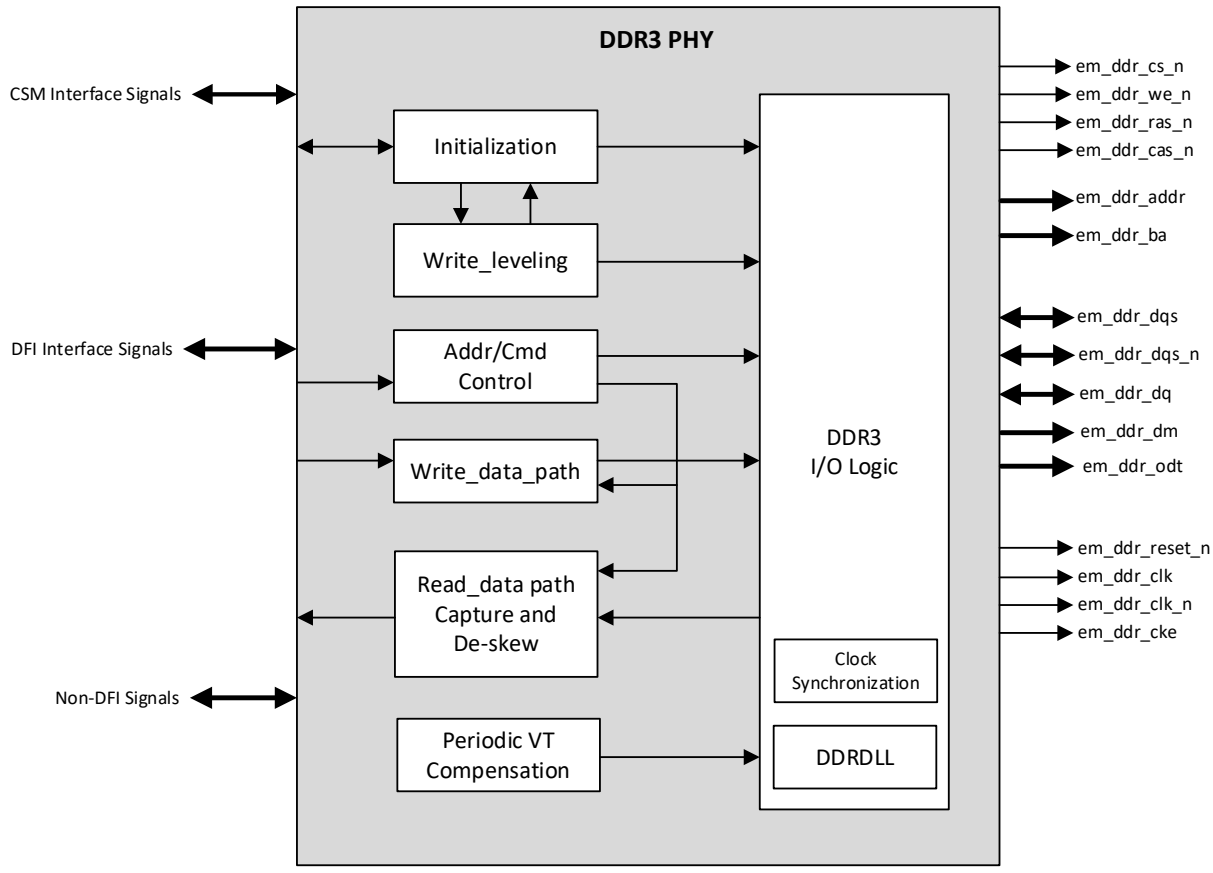

Figure 2.2. DDR3 PHY Block Diagram

2.1.3.1. DDR3 PHY Components

- **Clock Synchronization Module (CSM)**
  Refer to the DDR3 Calibration section for more details on the clock synchronization logic.
- **Initialization Module**
  Refer to the Initialization and Training Sequence section for more details on the initialization block.
- **Write Leveling**
  Refer to the Write Leveling section for more details on the write leveling block.
- **Read Training**
  Refer to the Read Training (DQS Gate) section for more details on the read training block.
- **Data Logic Path**
  The Data Path Logic (DPL) block interfaces with the DDR3 I/O modules and is responsible for generating the read data and read data valid signals during read operations. This block implements all the logic needed to ensure that the data write/read to and from the memory is transferred to the local user interface in a deterministic and coherent manner
- **Write Data Path**
  The write data path block interfaces with the DDR3 I/O modules and is responsible for loading the write data along with write data control signals to the DDR3 I/O primitives during write operations. This block implements all the logic needed to ensure that the data write to the memory is transferred from the DFI in a deterministic and coherent manner.
- **Read Data Path**
  The read data path block interfaces with the DDR3 I/O modules and is responsible for extracting the read data and read data valid signals during read operations. This block implements all the logic needed to ensure that the data read from the memory is transferred to the DFI in a deterministic and coherent manner. In addition, this block has the logic to deskew the read data delays between different data lanes.
- **DDR3 I/O Logic**
  The DDR3 I/O logic block provides the physical interface to the memory device. This block consists mainly of the CrossLink-NX DDR3 I/O primitives supporting compliance to DDR3 electrical and timing requirements. These primitives implement all the interface signals required for memory access and convert the single data rate (SDR) DFI data to double data rate DDR3 data for the write operations. In read mode, they perform the DDR3-to-SDR conversion.
- **Periodic VT Compensation**
  The DDR3 I/O delay changes when the voltage and temperature (VT) changes. Without compensation, the change in delay reduces the available DQS-DQ data eye and may cause data corruption in prolonged operation. To compensate for this, the DDR3 I/O Logic has DDRDLL which measures DDR clock period and produces a DLL code which means the number of delay taps to shift the DQ from the DQS by 45^{O}. This code is used by the DDR3 I/O Logic to adjust the DQ delay with respect to the DQS. The Periodic VT Compensation logic updates the DLL code such that the new DLL code value is still 45^{O}. This is shown in Figure 2.3. The actual DLL code update is done during refresh to avoid glitch on the DQ/DM signals.

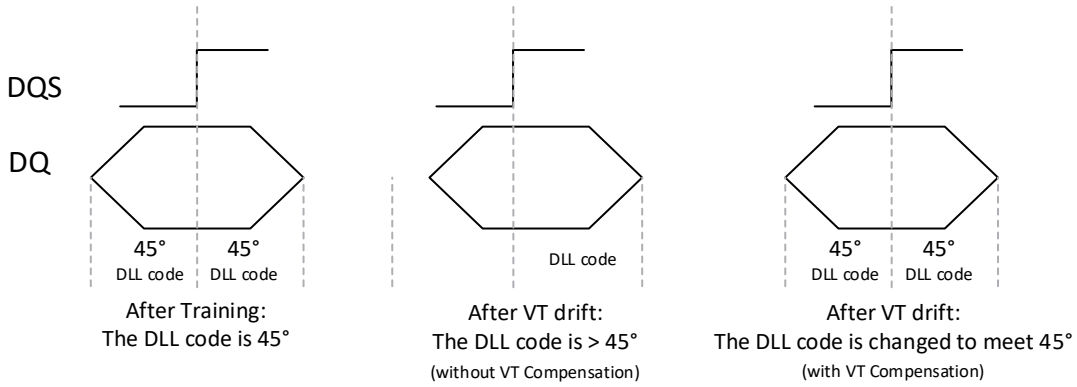

Figure 2.3. Periodic VT compensation

2.2. Clocking and Reset

You can reset the PLL module independently if the PLL lock is lost. For this setup, both initialization and read training are skipped, and only write leveling is required. Check the following attributes in the Module/IP Block Wizard to enable this setup: *Enable PLL and Enable External PLL Reset.*

Figure 2.4. shows the recommended steps for resetting the PLL and resuming the valid operation of the controller.

1. The clocking_good_o deasserts when PLL lock is lost due to the PLL reference clock (clk_i),
2. Assert pll_rst_n_i to reset the PLL module after the PLL reference clock returns to normal
3. Wait until the clocking_good_o asserts, indicating a stable clocking condition.
4. Assert ext_auto_ref to execute a memory refresh. Hold ext_auto_ref then deassert the signal at the same edge of the sampling clock when ext_auto_ref_ack_o goes high.
5. Assert wl_start_i for 1 sclk_o cycle to perform write leveling. Wait until wl_done_o goes high, indicating the completion of the write leveling.

Note: You can skip both step 4 and step 5 if you do not select the Write Leveling attribute.

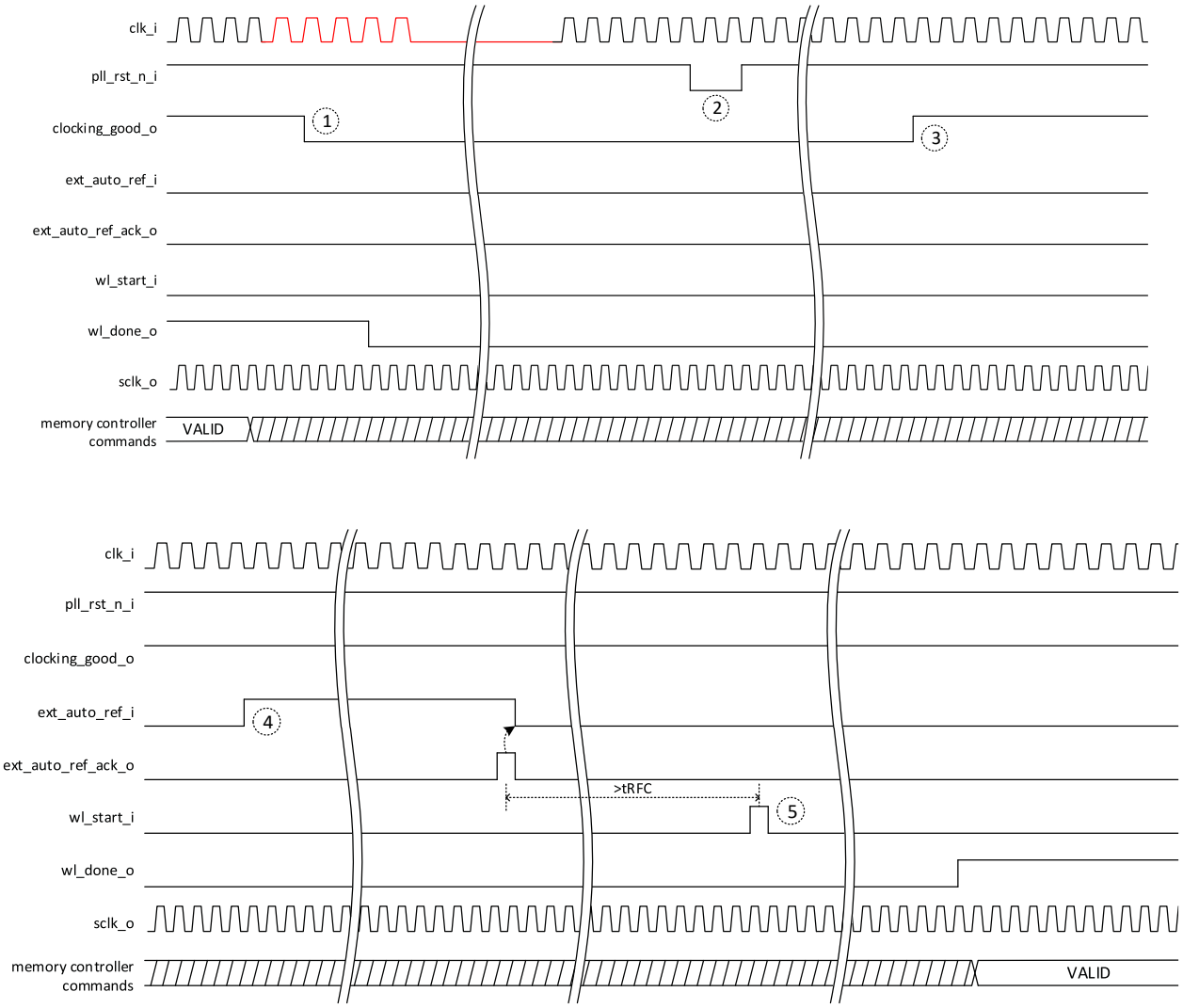

Figure 2.4. External PLL Reset Steps

2.3. User Interfaces

2.3.1. Data Interface Protocols

The Data Path Logic interfaces with the DDR3 I/O modules and generates the read data and read data valid signals during read operations. This block implements all the logic needed to ensure that the data writes and reads to and from the memory are transferred to the local user interface in a deterministic and coherent manner.

2.3.1.1. Native

The native interface provides direct access to the Soft Memory Controller command, address, command valid, data and data valid ports. Refer to the DDR3 Operation Description for more details on how to use the *native* interface for data access.

2.3.1.2. AXI4

The AXI4 interface is available in IP Core v2.1.0 and operates on the sclk_o signal. However, it is reset via the arst_n_i and rst_n_i signals. Resetting this interface via the arst_n_i triggers a reset to the Soft Memory Controller but not the Soft PHY and Soft Training Engine. Resetting the rst_n_i resets this interface, the Soft Memory Controller, and the Soft PHY and Soft Training Engine. Refer to the Clocking and Reset section of this user guide for more details.

The Memory Controller IP supports the following AXI4 types of burst transfers:

- Burst Type:
  - SINGLE: single burst (1 beat)
  - INCR: incrementing burst that does not wrap at address boundaries (4 or 8 beats)
- Burst Size :
  - [8, 16, 32, 64] beats
- Burst Address :
  - Unaligned addressing is supported.

The AXI4 protocol supports only a single outstanding transaction per bus master. Both read and write operations cannot happen concurrently and are arbitrated in a round-robin fashion, with both given equal priority. All responses are returned in order. For more information regarding the AXI4 protocol, refer to the AMBA AXI4 Protocol Specification.

Table 2.1. Supported AXI4 Transactions

Transaction Type	AXI Data Width	LENGTH	AxSIZE	Comment
Read/Write	32	0	[0,1,2]	[1,2,4] Byte transferred
Read/Write	32	1	[0,1,2]	[2,4,8] Byte transferred
Read/Write	32	32	[0,1,2]	[32,64,128] Byte transferred
Read/Write	32	64	[0,1,2]	[64,128,256] Byte transferred
Read/Write	64	0	[0,1,2,3]	[1,2,4,8] Byte transferred
Read/Write	64	1	[0,1,2,3]	[2,4,8,16] Byte transferred
Read/Write	64	32	[0,1,2,3]	[32,64,128,256] Byte transferred
Read/Write	64	64	[0,1,2,3]	[64,128,256,512] Byte transferred
Read/Write	128	0	[0,1,2,3,4]	[1,2,4, 8,16] Byte transferred
Read/Write	128	1	[0,1,2,3,4]	[2,4,8, 16,32] Byte transferred
Read/Write	128	32	[0,1,2,3,4]	[32,64,128,256,512] Byte transferred
Read/Write	128	64	[0,1,2,3,4]	[64,128,256,512,1024] Byte transferred
Read/Write	256	0	[0,1,2,3,4,5]	[1,2,4, 8,16,32] Byte transferred
Read/Write	256	1	[0,1,2,3,4,5]	[2,4,8,16,32,64] Byte transferred
Read/Write	256	32	[0,1,2,3,4,5]	[32,64,128,256,512,1024] Byte transferred
Read/Write	256	64	[0,1,2,3,4,5]	[64,128,256,512,1024,2048] Byte transferred

Table 2.2. Address Mapping

Memory Address	Size	Local Address Map	
Row	ROW_WIDTH = 13, 14, 15, 16 (depending on DDR Density and Configuration	ROW_H = ROW_L + ROW_WIDTH-1 ROW_L = BANK_H + 1	AxADDR[ROW_H: ROW_L]
Bank	BSIZE = 3	BANK_H = COL_H +BSIZE -1 BANK_L = COL_H + 1	AxADDR[BANK_H: BANK_L]
Column	COL_WIDTH = 10, 11, 12 (depending on DDR Density and Configuration)	COL_H = COL_L + ACTUAL_COL_WIDTH – 1 COL_L = 0	AxADDR[COL_H: COL_L]
Actual Column Width	If *DDR Bus Width* == 32; ACTUAL_COL_WIDTH = COL_WIDTH+2 If *DDR Bus Width* == 16: ACTUAL_COL_WIDTH = COL_WIDTH+1 If *DDR Bus Width* == 8: ACTUAL_COL_WIDTH = COL_WIDTH	N/A	

Table 2.3. AXI4 to Memory Address Mapping Example

Memory Address	Example User Value	Actual Line Size	Local Address Map
Actual Column Width	*DDR Bus Width* = 32	12	–
Column Width (COLW)	DDR_DENSITY = 1Gb Configuration = x16	10	axi_axaddr_i[11:0]
Bank Width (BANKW)	3 (Fixed for DDR3)	3	axi_axddr_i[14:12]
Row Width (ROWW)	DDR_DENSITY = 1Gb Configuration=x16	13	axi_axddr_i[27:15]
Rank Width (RANKW)	*Number of Ranks* = 1	0	–
Total Local Address Line Size		30	*addr_i[29:0]

2.3.2. Configuration Interface Protocol

2.3.2.1. Native

The native interface provides direct access to the Soft Memory Controller command, address, command valid, data and data valid ports. Refer to the DDR3 Operation Description for more details on how to use the *native* interface for configuration access.

2.3.2.2. APB

The APB interface is available in IP Core v2.1.0 and operates on its own clock and reset. This interface implements a register map for control configuration, status readback, and executing user commands. Refer to the Register Description section for more details.

2.4. DDR3 Calibration

The DDR3 PHY module provides PHY interface to the memory device. It implements soft logic in the FPGA fabric for initialization, write leveling, read training, and read/write data paths. It utilizes hard logic, called DDR3 I/O modules, for a 4:1 or 8:1 gearing ratio and DDR3 memory interface. The DDR3 I/O modules are hardware primitives that directly interface with the DDR3 memory, including the IDDR/ODDR/TDDR resource indicated in Table A.1. These primitives implement all the interface signals required for memory access. They convert the single data rate (SDR) data to double rate (DDR3) data for write operation and perform the DDR3 to SDR conversion in read mode.

The DDR3 PHY also ensures that the clock domain crossing margin between ECLK to SCLK stays the same for the IDDR and ODDR buses that produce 4:1 or 8:1 gearing ratio. Without proper synchronization, the bit order on different elements might be out of sync with each other, and the entire bus could be scrambled. Clock synchronization ensures that all DDR components start from the same edge clock cycle.

For 400 MHz DDR3 memory clock operation and a 4:1 gearing ratio, the Memory Controller Module operates with a 200 MHz system clock (SCLK), and the I/O logic in the DDR3 PHY Module works with a 400 MHz edge clock (ECLK). This operating clock ratio and the double data rate transfer lead to a user-side data bus that is four times the width of the memory-side data bus. For example, a 32-bit memory-side data width requires a 128-bit read data bus and a 128-bit write data bus at the user-side interface.

For 400 MHz DDR3 memory clock operation and an 8:1 gearing ratio, the Memory Controller Module operates with a 100 MHz system clock (SCLK), and the I/O logic in the DDR3 PHY Module works with a 400 MHz edge clock (ECLK). This operating clock ratio and the double data rate transfer leads to a user-side data bus that is eight times the width of the memory-side data bus. For example, a 32-bit memory-side data width requires a 256-bit read data bus and a 256-bit write data bus at the user-side interface.

2.4.1. Initialization and Training Sequence

The Initialization block performs the DDR3 memory initialization sequence as defined by the JEDEC protocol. After powering on or resetting the DDR3 controller, you must initialize the memory before sending any command to the controller. You need to assert the init_start input to the DDR3 controller to start the memory initialization sequence. The init_done_o signal indicates the completion of initialization.

2.4.1.1. Write Leveling

The write leveling block adjusts the DQS-to-CLK relationship for each memory device using the write level mode of the DDR3 SDRAM when the fly-by topology is implemented. Write leveling is always done immediately after a memory initialization sequence. When the init_done_o signal is asserted after the initialization process, it also indicates the completion of write leveling. If the write leveling process is not successful, the wl_err_o signal is also asserted along with the init_done_o signal. To ensure the DQS-to-CLK alignment is optimal, the first DQ sample coming back from the DDR3 must not be a *1*. If it is a *1*, there will be an auto adjustment to the CLK delay, in which case the CLK will be delayed until the first DQ sample coming back from the DDR3 is a *0*. This happens before the actual write-leveling operation begins. The CLK delay should not be arbitrarily set to a large number. The default should be used. In cases where the default does not work, set it to *0*, and let the IP auto-adjust the CLK delay to the correct value. Setting it high may cause write-leveling to fail, as it will be unable to successfully align DQS to CLK.

The DDR3 memory module adopts fly-by topology for the address, command, control, and clock signals for better signal integrity. This reduces the number of stubs and length but causes flight time skew between the DQS and CLK. Therefore, DDR3 allows the controller to compensate for the skew of the DQS signal delays to those signals at the DDR3 DRAM side through its write leveling capability. Route the PCB for the on-board memory application using the fly-by topology. Otherwise, write leveling failures occur due to the lack of guaranteed DQS to CLK edge relationship at the beginning of write level training. For this reason, disable the write leveling option if the PCB does not use fly-by routing for write leveling.

The write leveling scheme of the DDR3 SDRAM Controller IP core follows all the steps stipulated in the JEDEC specification. For more details on write leveling, refer to the JEDEC specification JESD79-3C.

2.4.1.2. Read Training (DQS Gate)

For every read operation, you must initialize the DDR3 I/O primitives of the LIFCL family device at the appropriate time to identify the incoming DQS preamble. Upon proper detection of the preamble, the primitive DQSBUF extracts a clean signal from the incoming DQS signal from the memory and generates BTDETECT, BURSTDETECT and DATAVALID output signals that indicate the correct timing window of the valid read data.

The memory controller generates a positioning signal, READ[3:0], to the primitive DQSBUF that is used for the above-mentioned operation. In addition to the READ[3:0] input, a fine control input signal RDCLKSEL[3:0] and an output signals BTDETECT and BURSTDETECT of the DQSBUF block are provided to the controller to accomplish the READ[3:0] signal positioning.

Due to the DQS round trip delay, which includes PCB routing and I/O pad delays, proper internal positioning of the READ[3:0] signal with respect to the incoming preamble is crucial for successful read operations. The LIFCL family DQSBUF block supports a dynamic READ[3:0] signal positioning function called read training. This function enables the memory controller to position the internal READ[3:0] signal within an appropriate timing window by progressively shifting the READ[3:0] signal and monitoring positioning results.

This read training is performed as part of the memory initialization process, after the write leveling operation is complete. During the read training, the memory controller generates the READ[3:0] pulse, positions this signal using RDCLKSEL[3:0] and monitors the BTDETECT output of DQSBUF for the result of the current position. The READ[3:0] signal is set high before the read preamble starts. When the READ[3:0] pulse is properly positioned, the preamble is detected correctly, and the BTDETECT and BURSTDETECT are asserted. This guarantees that the generated DATAVALID signal indicates the correct read valid time window.

The READ[3:0] signal is generated in the system clock (sclk_o) domain and stays asserted for the total burst length of the read operation.

A minimum burst length of four times the memory bus length is used in the read training process. The memory controller can determine proper position alignment when there are no failures on BTDETECT assertions during the multiple trials. If there is a failure, the memory controller shifts the READ[3:0] signal position and tries again until it detects no BTDETECT failure.

The memory controller stores the delay value of the successful position of the READ[3:0] signal for each DQS group. It uses these delay values during a normal read operation to correctly detect the preamble first, followed by the generation of DATAVALID signal.

2.4.1.3. Extended Training

The extended training provides more robust calibration performance, where the DQ/DM-to-DQS phase adjustment is done to achieve the optimal phase relationship. The extended training is only performed with the example-design. The result will be displayed and should be used to configure the IP parameters appropriately.

2.4.1.3.1 Write Training

Write training delays each DQS relative to DQ during write operations to optimize the data window. During this stage, a coarse adjustment is performed to obtain an initial valid DQ-DQS phase alignment on both write and read. Once a valid DQ-DQS coarse alignment is achieved for both write and read, the fine-tuning of the write DQ-DQS adjustment is performed. This is done by performing a write followed by a read, and DQ is delayed in the positive (incremental) direction. This is done repeatedly until the data being read back no longer matches the data written. This determines the maximum boundary. The DQ-DQS phase is then reset to the default position. The same sequence is performed, but this time with DQ being delayed in the negative (decremental) direction. A data mismatch indicates the minimum boundary. The optimal position is then calculated as the midpoint between the maximum and the minimum boundary.

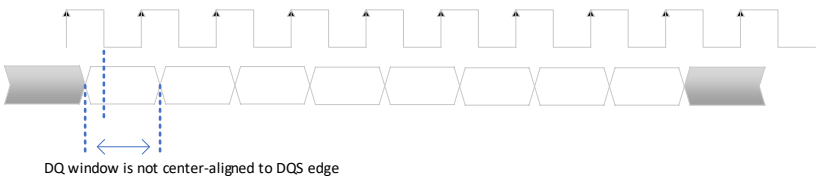

Figure 2.5. Before Write Training

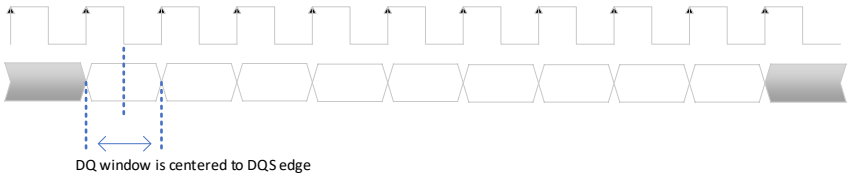

Figure 2.6. After Write Training

The write training is performed as part of the example-design sequence, and the resulting window and optimal position will be displayed, as the example below:

```
-----------------------------------------------
Starting Write DQ/DM Calibration
-----------------------------------------------
DLL Code: 44

W
Write DQ/DM Maximum Delay Value: -12
Write DQ/DM Minimum Delay Value: -78
Write DQ/DM Optimal Delay Value: -53
```

Figure 2.7. Result

This example illustrates a window size of 66 delay taps (one tap equals ~12.5 ps), with the maximum boundary at -12 and the minimum boundary at –78. In this case, the minimum boundary hits underflow and the correct minimum cannot be found due to not having enough delay taps. The optimal DQ-DQS phase is taken based on the maximum boundary and the DLL Code. If the correct minimum boundary was found, the optimal delay will be the average of the maximum and minimum boundaries.

The corresponding parameters from the IP configuration GUI should be updated accordingly:

- Write DQ/DM Delay Direction: Decrement (negative indicates decrement)
- Write DQ/DM Delay Value: 46

2.4.1.3.2 Read Data Eye Training

The read data eye training center-aligns DQS relative to the DQ window. There are two scenarios in this step.

- **Default DQ-DQS position results in a bad read:** During this step, a write is performed followed by a read, and DQ is delayed in the positive (incremental) direction. This is repeated until the data read back matches the data written. The minimum boundary is determined. This step is repeated until the data read back no longer matches the data written. The maximum boundary is determined. The optimal DQ-DQS phase relationship is calculated by taking the midpoint between the maximum and the minimum boundary.
- **Default DQ-DQS position results in a good read:** During this step, a write is performed followed by a read, and DQ is delayed in the positive (incremental) direction. This is repeated until the data read back does not match the data written. The maximum boundary is determined. The same sequence is repeated, but this time DQ is delayed in the negative (decremental) direction. A data mismatch indicates the minimum boundary. The optimal DQ-DQS phase relationship is calculated by taking the midpoint between the maximum and the minimum boundary.

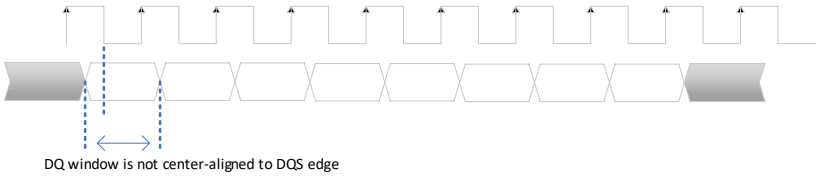

Figure 2.8. Before Read Data Eye Training

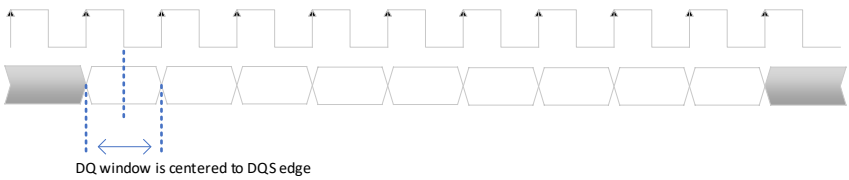

Figure 2.9. After Read Data Eye Training

The read data-eye training is performed as part of the example-design sequence, and the resulting window and optimal position will be displayed as shown in the example below:

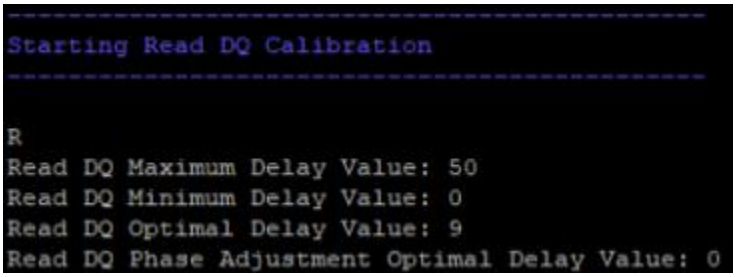

Figure 2.10. Result

This example shows how calibration determines the optimal delay for DQ relative to DQS.

- Window size: 82 delay taps (each tap = 12.5 ps)
- Boundaries:
  - Maximum: +50 taps
  - Minimum: –32 taps (calculated, not displayed)
- Display values:
  - Maximum delay shown: 50 taps
  - Minimum delay shown: 0 tap (the physical delay element cannot apply negative delay).

The negative boundary (–32) is theoretical and calculated. You can achieve an effective negative delay by adjusting the internal DQ sampling phase (reducing the DQ-to-DQS phase below 90°), as described in the IP user guide. The display shows the minimum delay value (0), which differs from the minimum boundary (–32).

Optimal Delay

- The midpoint of the window is +9 taps, indicating that the best alignment is achieved by adding nine taps to DQ.
- Since this is a positive delay, no phase adjustment is needed. Therefore:
  - Read DQ Delay Value: 9
  - Read DQ Phase Adjustment Direction: Increment
  - Read DQ Phase Adjustment Value: 0

Read DQ Phase Adjustment Value Explained

- Why zero (0) is correct:
  The delay element cannot apply negative delay. In this case, the optimal delay (+9 taps) is achievable by the DQ delay element alone, so no extra phase shift is needed.
- When it would be nonzero:
  If the calculated optimal delay were negative (such as –10 taps), the tool would set the DQ delay to zero (0) and apply a phase adjustment equivalent to –10 taps internally.

2.4.2. Initialization and Training with APB Interface

When the APB interface is enabled, the init_start_i port is no longer be available. Initialization is initiated using the APB interface. Refer to the Register Description section for more details on the register map. You should execute the following steps to start the initialization and training sequence of the DDR3 SDRAM:

1. Initiate an APB write transaction to address offset 0x00. The data to be written should be 'b1001. Set bit 3 to enable configuration mode and bit 0 to trigger the initialization.
2. Perform an APB read transaction to address offset 0x04. Check if bit 0 is driven HIGH.
3. Repeat step 2 until bit 0 is driven HIGH. This indicates that initialization is complete.
4. Perform an APB write transaction to address offset 0x00. The data to be written should be 'b0000. Set bit 3 to '0' to switch from configuration mode to data mode.

2.5. DDR3 Operation Description

The native interface of the DDR3 SDRAM Controller IP Core consists of five independent functional groups. Each functional group and its associated local interface signals as listed in Table 2.4.

Table 2.4. Native Interface Functional Groups

Functional Group	Native Interface Signals
Initialization Control	init_start_i, init_done_o, rt_done_o, rt_err_o, wl_err_o
Command and Address	addr_i, cmd_i, cmd_rdy_o, cmd_valid_, cmd_burst_cnt_i
Data Write	datain_rdy_o, write_data_i, data_mask_i
Data Read	read_data_o, read_data_valid_o
Auto Refresh	ext_auto_ref_i, ext_auto_ref_ack_o

When the AXI4 and APB Interfaces are enabled, these modules interfaces directly with the native interface within the memory controller. The APB Interface drives the Initialization Control group and the Command and Address group. The AXI4 interface drives the Command and Address, Data Write, and Data Read groups. Auto Refresh is available only on the native interface. To handle the resource contention between AXI4 and APB on the Command and Address group, use the Enable APB Config bit from the APB Control Reg. Write a '1' to this bit to give the APB Interface access to the Command and Address group. Writing a '0' to this bit returns access to the AXI4 Interface. Refer to the APB section for more details.

2.5.1. Initialization Control

You must initialize DDR3 memory devices before the memory controller can access them. The memory controller starts the memory initialization sequence when the init_start_i signal is asserted. Once asserted, the init_start_i signal needs to be held high until the initialization process is completed. The init_done_o signal is asserted high for one clock cycle, indicating that the core has completed the initialization sequence and is now ready to access the memory. You must negate the init_start_i signal as soon as the init_done_o signal is sampled high at the rising edge of sclk_o. Memory initialization is required once, immediately after the system reset.

The JESD79-3C standard specifies the following minimum reset assert time requirements:

- During power-up initialization: 200 ns
- During reset initialization with stable power: 100 ns

You are responsible for ensuring that the above minimum reset assert duration is met.

As part of Initialization, the memory controller ensures a minimum gap of 500 µs between em_ddr_reset_n_i de-assertion and em_ddr_cke_o assertion.

The IP Core performs write levelling for all available ranks and stores the write level delay values.

Read training is also performed during the initialization process to find the best-read pulse position that detects the incoming read DQS preamble timing. Since DDR3 memory does not use a DLL function, the clock-to-DQS driving time can vary significantly with process, voltage, and temperature (PVT) variations. Because of this, periodic retraining of the read pulse position may be necessary to guarantee stable read transactions over the PVT variations during normal operation.

2.5.2. Command and Address

Once the memory initialization is complete, the core waits for user commands to set up and access the memory. The user logic needs to provide the command, address, and control signals to the core. Commands and addresses are delivered to the core using the Command Decoding Registers.

The DDR3 SDRAM Controller IP Core informs the user logic that it is ready to receive a command by asserting the cmd_rdy_o signal for one cycle. If the core finds the cmd_valid_i signal asserted by the user logic while its cmd_rdy_o is asserted, it takes the cmd_i input as a valid user command. Usually, cmd_valid_i is de-asserted at the rising edge of the clock that samples cmd_rdy_o high. The core also accepts the addr_i input as a valid start address or mode register programming data, depending on the command type. Along with the addr_i input, the core also accepts the signals cmd_burst_cnt_i and ofly_burst_len_i. If cmd_valid_i is not asserted, the cmd_i and addr_i inputs become invalid, and the core ignores them. The cmd_i, addr_i, cmd_burst_cnt_i, ofly_burst_len_i and cmd_valid_i inputs are ignored while cmd_rdy_o is de-asserted. The cmd_rdy_o signal is asserted again to accept the next command.

The core is designed to ensure maximum throughput at a burst length of eight by asserting cmd_rdy_o once every two-clock cycle, unless the command queue is full or there is an intervention on the memory interface (such as Auto-Refresh cycles.)

When the core is in the command burst operation, it extensively occupies the data bus. During this time, the IP Core negates cmd_rdy_o until the command burst is completed. While the IP Core is operating in command burst mode, it can keep maximum throughput by internally replicating the command. The memory controller can repeat the given READ or WRITE command up to 32 times. The cmd_burst_cnt_i[4:0] input is used to set the number of repeats of the given command. The core allows the command burst function to access the memory addresses within the current page. When the core reaches the boundary of the current page while accessing the memory in the command burst mode, the next address to be accessed by the core becomes the beginning of the same page. This causes overwriting of contents or reading unexpected data. You must therefore track the accessible address range in the current page, while the command burst operation is performed. If an application requires a fixed command burst size, using 2-, 4-, 8-, 16- or 32-burst with burst-aligned column address is recommended to ensure that the command burst accesses do not cross the page boundary.

When cmd_burst_cnt_i and ofly_burst_len_i are 0, the controller performs 32 commands (reads or writes). The cmd_burst_cnt_i input is sampled the same way as the cmd signal. The timing diagram of Command and Address group signals is shown in Figure 2.11. When cmd_burst_cnt_i is sampled with a value greater than *00001*, and the command queue becomes full, the cmd_rdy_o signal is not asserted, and the memory address is automatically increased by the core until the current command burst cycle is completed.

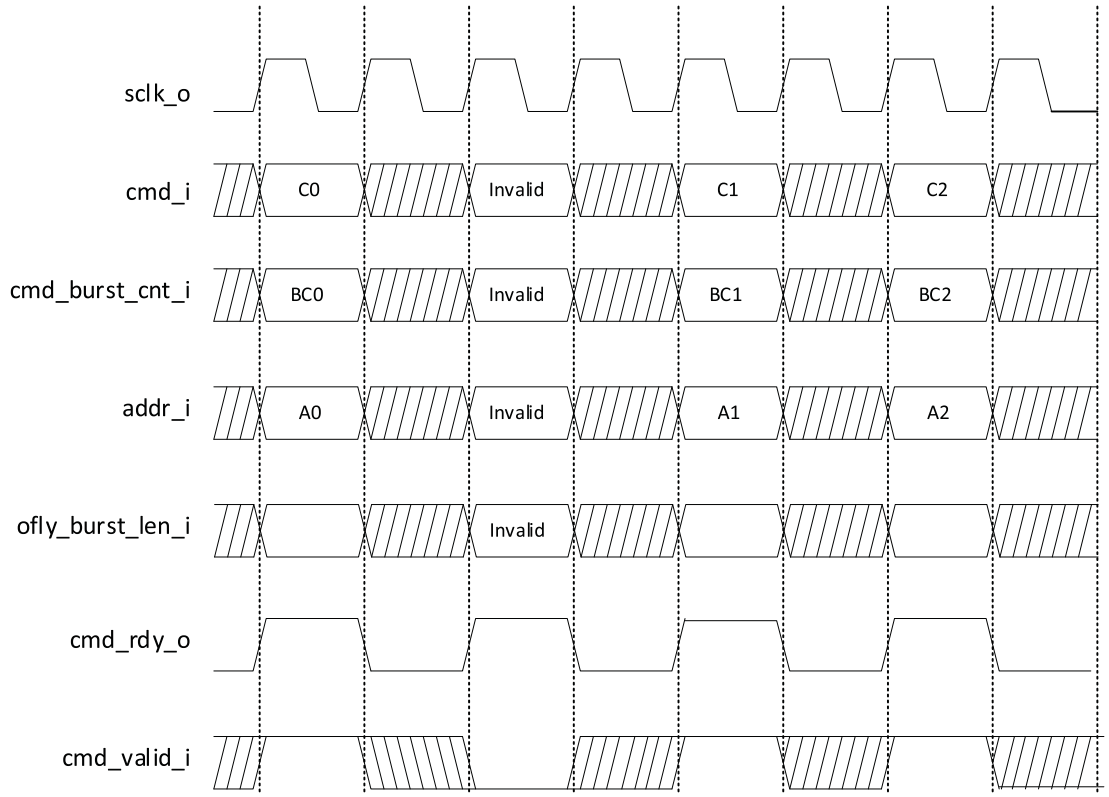

Figure 2.11. Timing of Command and Address

2.5.2.1. Address Mapping

Mapping local addresses to memory addresses is an important part of system design when implementing a memory controller function. You must know how the local address lines from the memory controller connect to those address lines from the memory because proper local-to-memory address mapping is crucial to meet system requirements in applications such as a video frame buffer controller. Even for other applications, careful address mapping is generally necessary to optimize system performance. On the memory side, the address (A), bank address (BA) and chip select (CS) inputs are used for addressing a memory device. You can obtain this information from a given data sheet. Figure 2.12 shows the local-to-memory address mapping of the Lattice DDR3 memory controller cores.

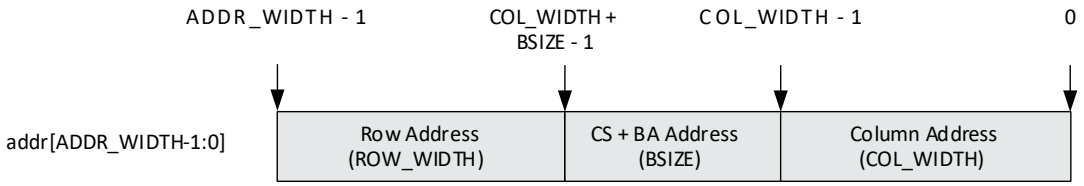

Figure 2.12. Local-to-Memory Address Mapping for Memory Access

ADDR_WIDTH is calculated by the sum of COL_WIDTH, ROW_WIDTH and BSIZE. BSIZE is determined by the sum of the BANK_WIDTH and CS_WIDTH. For DDR3 devices, the bank address size is always 3. When the number of chip select is 1, 2, or 4, the chip select address size becomes 0, 1, or 2, respectively. An example of a typical address mapping is shown in Table 2.5 and Figure 2.13.

Table 2.5. Address Mapping Example

Attribute Name	Example User Value	Actual Line Size	Local Address Map
Column Size	11	11	addr_i[10:0]
Bank Size[1]	8	3	addr_i[13:11]
Rank Size (or Chip Select Size)	Dual	1	addr_i[14]
Row Size	14	14	addr_i[28:15]
Total Local Address Line Size		29	addr_i[28:0]

Note:

1. Bank Size is not set in Module/IP Block Wizard, this is fixed for DDR3.

Figure 2.13. Mapped Address for the Example

2.5.3. User Commands

You initiate a request to the memory controller by loading a specific command code in cmd input along with other information such as memory address. The command on the *cmd bus* must be a valid command. Lattice defines a set of valid memory commands as shown in Table 2.6. All other values are reserved and considered invalid.

Table 2.6. Defined User Commands

Command	Mnemonic	cmd_i[3:0]
Read	READ	4'b0001
Write	WRITE	4'b0010
Read with Auto Pre-charge	READA	4'b0011
Write with Auto Pre-charge	WRITEA	4'b0100
Power down Entry	PDOWN_ENT	4'b0101
Load Mode Register	LOAD_MR	4'b0110
Self-Refresh Entry	SEL_REF_ENT	4'b1000
Self-Refresh Exit	SEL_REF_EXIT	4'b1001
Power down Entry	PDOWN_EXIT	4'b1011
ZQ Calibration Long	ZQ_LNG	4'b1100
ZQ Calibration Short	ZQ_SHRT	4'b1101

Notes:

- The controller accepts only the command codes listed above as legal commands. It discards any other command code as an invalid command.
- The controller discards Self-Refresh Entry or Power Down Entry command if the memory is already in Self Refresh mode or Power Down mode respectively.
- The controller discards Self-Refresh Exit or Power Down Exit command if the memory is already not in Self Refresh mode or Power Down mode respectively.
- The controller issues a refresh before every self-refresh entry command.

2.5.4. WRITE

You initiate a memory write operation by asserting cmd_valid_i along with the WRITE or WRITEA command and the address. After the WRITE command is accepted, the memory controller core asserts the datain_rdy_o signal when it is ready to receive the write data from the user logic to write into the memory. Since the duration from the time a write command is accepted to the time the datain_rdy_o signal is asserted is not fixed, the user logic needs to monitor the datain_rdy_o signal. Once datain_rdy_o is asserted, the core expects valid data on the write_data bus one or two clock cycles after the datain_rdy signal is asserted. You can program the write data delay by setting the value for *Data ready to Write Data delay* attribute, providing flexible backend application support. For example, setting the value to *2* ensures that the core takes the write data in proper time when the local user interface of the core is connected to a synchronous FIFO module inside the user logic. Figure 2.14 shows two examples of the local user interface data write timing. Both cases are in BL8 (Burst Length 8) mode. The upper diagram shows the case of one clock cycle delay of write data, while the lower one displays a two clock-cycle delay case. The memory controller considers D0, DM0 through D5, DM5 valid write data.

The controller decodes the addr input to extract the current row and current bank addresses and checks if the current row in the memory device is already opened. If there is no opened row in current bank, the controller generates an ACTIVE command to the memory to open the current row first. Then the memory controller issues a WRITE command to the memory. If there is already an opened row in the current bank and the current row address is different from the opened row, the controller generates a PRECHARGE command to close opened row in the bank. This is followed by an ACTIVE command to open the current row. Then the memory controller issues a WRITE command to the memory. If the current row is already opened, the controller sends only a WRITE command (without any ACTIVE or PRECHARGE commands) to the memory.

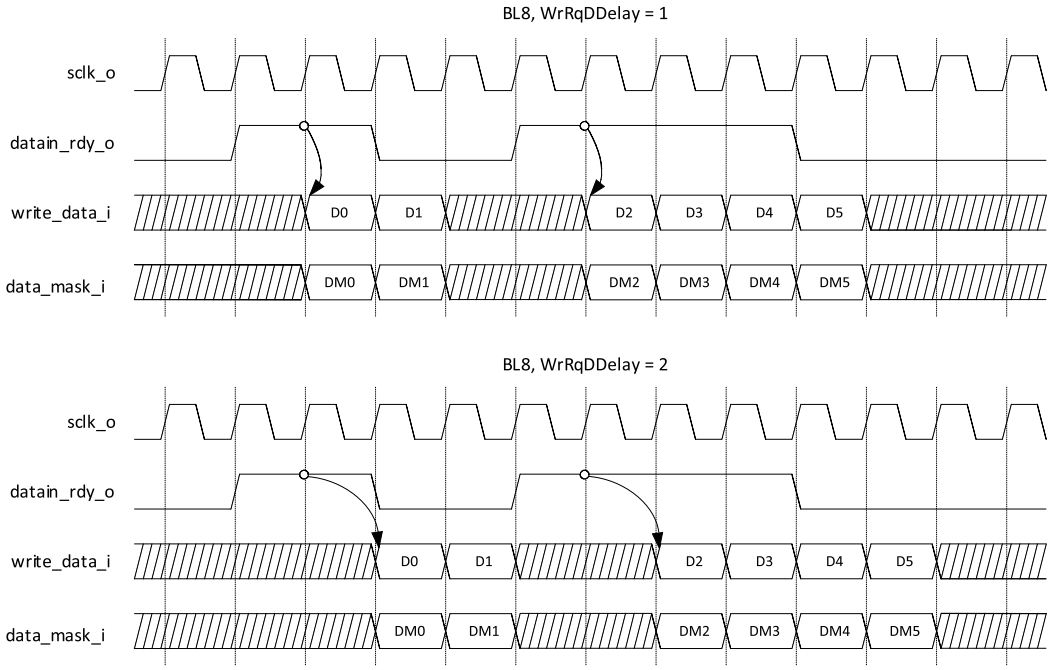

Figure 2.14. One-Clock vs. Two-Clock Write Data Delay

Note: WrRqDDelay is *Data ready to Write Data delay* attribute, which is currently fixed to 1.

2.5.5. WRITEA

WRITEA is treated the same way as the WRITE command, except that the IP Core issues a Write with Auto Precharge command to the memory, instead of just a Write command. This causes the memory to automatically close the current row upon completing the write operation. The AXI4 Interface does not send the WRITEA command for write transactions.

2.5.6. READ

When the READ command is accepted, the memory controller core accesses the memory to read the addressed data and brings the data back to the local user interface. Once the read data is available on the local user interface, the memory controller core asserts the read_data_valid_o signal to tell the user logic that the valid read data is on the read_data_o bus. The read data timing on the local user interface is shown in Figure 2.15. The read operation follows the same row status checking scheme as mentioned in the write operation. Depending on the current row status, the memory controller generates ACTIVE and PRECHARGE commands, as required. Refer to the description mentioned in *Write operation* for more details.

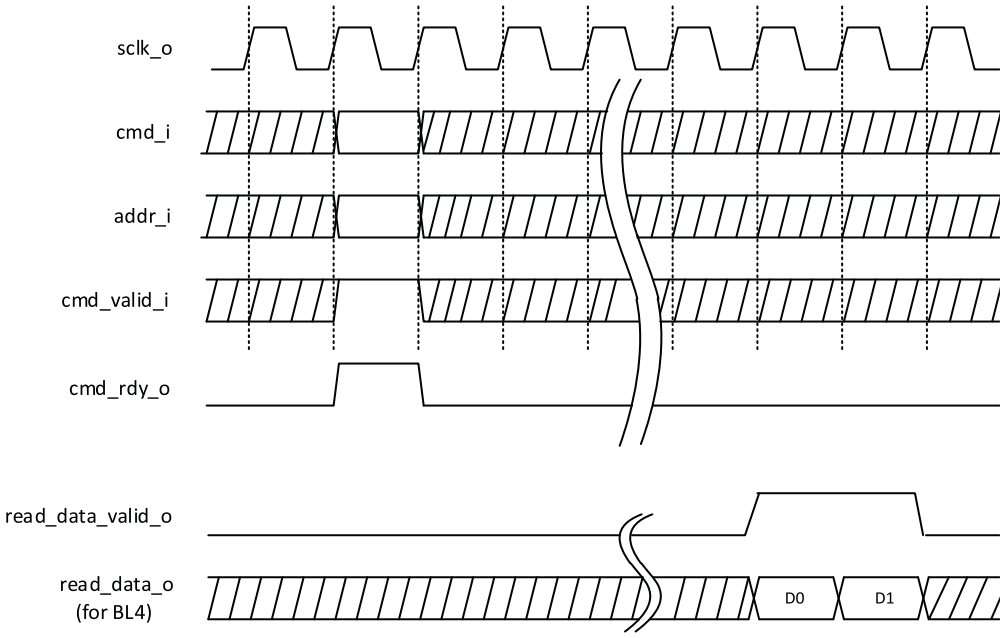

Figure 2.15. User-Side Read Operation

2.5.7. READA

READA is treated the same way as the READ command except that the IP Core issues a Read with Auto Precharge command to the memory instead of a Read command. This makes the memory automatically close the current row after completing the read operation. The AXI4 Interface doesn't send the READ command for read transactions.

2.5.8. Mode Register Programming

The DDR3 SDRAM memory devices are programmed using the mode registers MR0, MR1, MR2, and MR3. The bank address signal (em_ddr_ba_o) chooses one of the mode registers, while the programming data is delivered through the address signal (em_ddr_addr_o). The memory data signal is not used for the Mode Register programming.

The Lattice DDR3 SDRAM Controller IP Core uses the local address bus, addr_i, to program these registers. The core accepts a user command, LOAD_MR, to initiate the programming of mode registers. When LOAD_MR is applied on the cmd_i signal, the user logic must provide the information for the targeted mode register and the programming data on the addr_i signal. When the target mode register is programmed, the memory controller core is also configured to support the new memory setting. Figure 2.16 shows how the local address lines are allocated for the programming of memory registers.

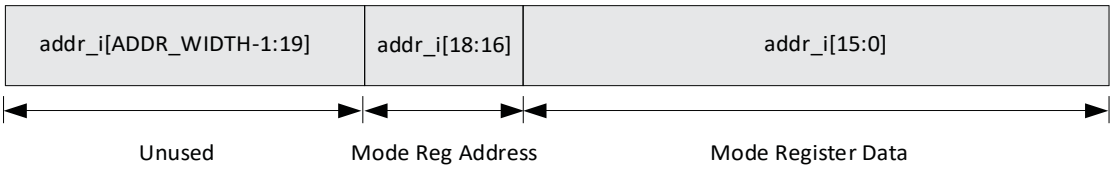

Figure 2.16. User-to-Memory Address Mapping for MR Programming

The lower side of the addr_i provides the register programming data, starting from bit 0 for LSB. The programming data requires 16 bits of the local address lines. You need three more bits to choose a target register as listed in Table 2.7. All other upper address lines remain unused during LOAD_MR command.

Table 2.7. Transmit MAC Statistics Vector

Mode Register	(addr_i[18:16])
MR0	3'b000
MR1	3'b001
MR2	3'b010
MR3	3'b011

The initialization process uses the mode register initial values selected through the Module/IP Block Wizard during IP configuration. If the user logic does not further program these registers using the LOAD_MR user command, they remain in the configurations programmed during the initialization process.

Table 2.8 shows the list of available parameters and their initial default values.

Table 2.8. Initialization Default Values for Mode Register Setting

Mode Register	Register Field	Default Value	Description	Local Address	Module/IP Block Wizard Setting
MR0	Burst Length	2'b00	BL = 8	addr_i[1:0]	Yes
	Read Burst Type	1'b0	Sequential	addr_i[3]	Yes
	CAS Latency	3'b000	CL = 5	addr_i[6:4], addr_i[2]	Yes
	Mode	1'b0	Normal	addr_i[7]	No
	DLL Reset	1'b1	DLL Reset = Yes	addr_i[8]	No
	WR Recovery	3'b010	6	addr_i[11:9]	Yes
	DLL Control for Precharge PD	1'b1	Fast	addr_i[12]	Yes
	All Others	0	–	addr_i[ROW_WIDTH-1:13]	No
MR1	DLL Enable	1'b0	DLL Enable	addr_i[0]	No
	ODI Control	2'b00	RZQ/6	addr_i[5],addr_i[1]	Yes
	RTT_Nom	3'b001	RZQ/4	addr_i[9],addr_i[6],addr_i[2]	Yes
	Additive Latency	2'b00	Disabled	addr_i[4:3]	No
	Write Level Enable	1'b0	Disabled	addr_i[7]	No
	TDQS Enable	1'b0	Disabled	addr_i[11]	No
	Qoff	1'b0	Enable	addr_i[12]	No
	All Others	0	–	addr_i[ROW_WIDTH-1:13]	No
MR2	CAS Write Latency	3'b000	5	addr[5:3]	Yes
	Rtt_WR	2'b01	RZQ/4	addr_i[10:9]	Yes
	All Others	0	–	addr_i[ROW_WIDTH-1:11], addr_i[8:6], addr_i[2:0]	No
MR3	All	0	–	addr_i[ROW_WIDTH-1:0]	No

2.5.9. REFRESH Support

Since DDR3 memories have at least an 8-deep Auto Refresh command queue as per JEDEC specification, Lattice's DDR3 memory controller core supports up to eight Auto Refresh commands in one burst. The core has an internal auto refresh generator that sends out a set of consecutive Auto Refresh commands to the memory at once when it reaches the time period of the refresh intervals (*TREFI* attribute) times the *Auto Refresh Burst Count* attribute as selected in the Module/IP Block Wizard.

It is recommended that the maximum number be used if the DDR3 interface throughput is a major concern of the system. If it is set to 8, for example, the core sends a set of eight consecutive Auto Refresh commands to the memory at once when it reaches the time period of the eight refresh intervals (*TREFI* × 8). Bursting refresh cycles increases the DDR3 bus throughput because it helps keep core intervention to a minimum. When a refresh burst is used, the controller issues a Precharge command only for the first Refresh command, and the subsequent Refresh commands of the burst are issued without the associated Precharge commands. This is to improves the DDR3 throughput. The TREFI is internally compensated with a minor margin (2 to 4 SCLK cycles, depending on DDR clock frequency) to ensure that, cumulatively, it never exceed 8 × TREFI maximum allowable refresh period.

Alternatively, you enable the *External Auto Refresh Port*, which adds an input signal ext_auto_ref and an output signal ext_auto_ref_ack to the core. In this case the internal auto refresh generator is disabled, and the core sends out a burst of refresh commands, as directed by Auto refresh burst count, every time the ext_auto_ref is asserted. The output signal ext_auto_ref_ack indicates the completion of the refresh burst.

In an application where explicit memory refresh is not necessary, you enable the External Auto Refresh Port and keep the ext_auto_ref signal deasserted.

3. IP Parameter Description

Table 3.6 shows the configurable attributes of the DDR3 SDRAM Controller IP Core along with the description for each attribute. You can configure the attributes through the IP Catalog's Module/IP Block Wizard of the Lattice Radiant software. The attributes are arranged into tabs, and related attributes are grouped together. The three tabs are as follows:

- **General Tab**
  The General tab contains the attributes for configuring the target memory device and the IP Core features. These attributes are static; you can only set them in the Module/IP Block Wizard. You must regenerate the DDR3 SDRAM Controller IP Core to change the features set by these attributes.
- **Memory Device Setting Tab**
  The Memory Device Setting Tab contains the attributes for configuring the target memory device/module. The attributes under Mode Register Initial Setting Group are dynamic, meaning the reset values are set from the Module/IP Block Wizard. Refer to JESD79-3, DDR3 SDRAM Standard, for allowed values.
- **Memory Device Timing Tab**
  The default attributes displayed in this tab are the default values of the Micron DDR3 1Gb-187E memory module. You can modify these attributes by checking the Manual Adjust attribute. It is important to adjust the attribute values in this tab to the timing parameters of the memory device for the target application. The DDR3 SDRAM Controller IP Core also uses these timing parameters when generating memory commands.

The *Calculate* button at the bottom is used to calculate the optimum parameter values for the PLL.

3.1. General

The following section describes the parameters available in the General tab of the IP parameter editor.

Table 3.1. Device Information Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Interface Type	DDR3, DDR3L	DDR3	—
Simulation Mode Enable	Checked, Unchecked	Unchecked	—
Gearing Ratio	4:1, 8:1	8:1	—
I/O Buffer Type	SSTL15_I, SSTL15_II	SSTL15_I	If Interface Type == DDR3
I/O Buffer Type	SSTL135_I, SSTL135_II	SSTL135_I	If Interface Type == DDR3L
Select Memory	Micron DDR3 1Gb-187E, Micron DDR3 2Gb-187E, Micron DDR3 4Gb-187E, Custom	Micron DDR3 1Gb-187E	—
MemClock (MHz)	300, 333, 400, 533	400	If *Gearing Ratio* == 8:1, 533 is available

Table 3.2. Clock Settings Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Enable PLL	Checked, Unchecked	Checked	—
Enable External PLL Reset	Checked, Unchecked	Unchecked	—
RefClock (MHz)	25, 50, 75, 100, 111, 125, 150, 200	100	—
PLL Reference Clock from Pin	Checked, Unchecked	Checked	—
I/O Standard for Reference Clock	SLVS, SSTL15_I, SSTL15_II, SSTL135_I, SSTL135_II	SLVS	—
MemClock Actual Frequency (MHz) [100-800]	—	400	—

Table 3.3. Memory Configuration Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Memory Type	On-board Memory	On-board Memory	—
Memory Data Bus Size	8, 16, 24, 32	16	If Local Bus Type == Native Selectable values may change depending on the target device.
	8, 16, 32	16	If Local Bus Type == AMBA Selectable values may change depending on the target device.
Configuration	x8, x16	x8	x16 is only available when the size of the Memory Data Bus is 16 or 32.
Rank Size	Single	Single	—
Clock Width	1, 2	1	If Rank Size == Single: [1, 2] If Memory Data Buse Size == 32: [1]

Table 3.4. Local Interface Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Bus Interface Type	Native, AMBA	AMBA	—
Enable APB Interface	Checked, Unchecked	Checked	If Local Bus Type == AMBA
AXI Data Width	32, 64, 128, 256	Calculated	The default value is the largest selectable value. The maximum selectable value is calculated based on the formula: *Memory Data Bus Size × Gearing Ratio.* The smallest selectable value is 32. All other supported values are powers of 2, between the smallest and largest selectable values.
AXI Address Width	Calculated	Calculated	Display only. Calculated based off selection for DDR density.
AXI Maximum Burst Length	64	64	Display only.
AXI ID Width	1, 2, 3, 4, 5, 6, 7, 8	4	—
AXI Outstanding Command Support	1,2	2	—
FIFO RD DEPTH	Calculated	Calculated	Calculated based off selection for DDR Bus Width and AXI Data Width.

Table 3.5. Additional Configuration Group Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Data ready to Write Data delay	1, 2	1	Display only
Controller reset to Memory	Checked/Unchecked	Checked	—
MC CK/Command Delay Value	0 to 127	40	—
MC DQS/DQ ODT Value	OFF, 40, 50, 60, 75	60	—
Enable Extended Training[1]	Checked, Unchecked	Unchecked	If local bus type == AMBA and enable APB interface == Checked.
Write DQ/DM Delay Direction	Increment, Decrement	Calculated	This is visible only when the enable extended training is Checked. The default value depends on the Simulation Mode Enable and Gearing Ratio.

Attribute	Selectable Values	Default	Dependency on Other Attributes
Write DQ/DM Delay Value	0–63	Calculated	This is visible only when the enable extended training is Unchecked. Calculated based on the Simulation Mode Enable, Gearing Ratio and MemClock.
Read DQ Delay Value	0–63	Calculated	Calculated based on the Simulation Mode Enable and MemClock.
Read DQ Phase Adjustment Direction	Increment, Decrement	Increment	This is visible only when the enable extended training is unchecked.
Read DQ Phase Adjustment Value	0–63	Calculated	This is visible only when the enable extended training is unchecked. Calculated based on the Simulation Mode Enable and MemClock.

Note:

1. If this attribute is checked, eval_top must be run, and the printed delay values for Write DQ/DM Delay Value, Write DQ/DM Delay Direction, Read DQ Delay Value, Read DQ Phase Adjustment Direction, and Read DQ Phase Adjustment Value need to be used to reconfigure the corresponding configuration parameters of this IP, with the Enable Extended Training set to unchecked.

Table 3.6. General Definitions

Attribute	Description
Device Information Group	
Interface Type	Specifies the DDR3 Memory interface based on the target DRAM IO: DDR3 or DDR3L.
Simulation Mode Enable	Shorten simulation and training for simulation. This should be unchecked for implementation.
Gearing Ratio	Specifies the number of DDR data transfers in one system clock cycle. 4:1 – four DDR data transfers (2 DDR clock cycles) in one system clock cycle. 8:1 – eight DDR data transfers (4 DDR clock cycles) in one system clock cycle.
I/O Buffer Type	Specifies the I/O buffer types of the DDR signals.
Select Memory	Some attribute default values depend on this attribute. The Micron DDR3 1GB-187E is the default DDR3 memory device, and the timing parameters of this memory device are listed in the Memory Device Timing tab as default values. The other available options are Micron DDR3 2Gb-187E and Micron DDR3 4Gb-187E.
Clock Settings Group	
Enable PLL	Enables the internal PLL instance. *Checked:* A PLL is instantiated inside the IP Core. You should provide the PLL reference clock input (clk_i). *Unchecked:* The IP Core do not have PLL, you should instantiate the PLL and connect the following signals from the PLL: eclk_i, sync_clk_i, pll_lock_i
Enable External PLL Reset	Enables the external reset of PLL. Refer to the Clocking and Reset section for more details.
PLL Reference Clock from Pin	Select this option if you want to connect the PLL reference clock to an I/O pin.
I/O Standard for Reference Clock	Specifies the I/O buffer type for the PLL reference clock.
RefClock (MHz)	Specifies the reference input clock to PLL which generates the system clock (sclk_o) and memory clock (em_ddr_clk_o).
MemClock (MHz)	Specifies the frequency of the memory clock to the memory device. The allowed values are 300 MHz, 333 MHz, 400 MHz, and 533 MHz. This is the PLL output frequency, which depends on the corresponding value of RefClock. For example, for a MemClock value of 333 MHz, you should set the PLL RefClock to 111 MHz.
Memory Configuration Group	
Memory Type	This attribute is for information only. Only On-board Memory type is supported.
Memory Data Bus Size (DATA_WIDTH)	Specifies the bit width of DDR3 data bus (em_ddr_data_io). If the memory module has a wider data bus than required, only the required data width should be selected.
Configuration	Selects the device configuration of the on-board memory. The memory controller supports device configurations x8, and x16.

Attribute	Description
Rank Size (CS_WIDTH)	Select the number of Chip selects (em_ddr_cs_n_o) required – Single or Dual. This also specifies the bit width of em_ddr_cke_o and em_ddr_odt_o.
Clock Width (CLKO_WIDTH)	Specifies the number of clock signals (em_ddr_clk_o) with which the IP Core drives the memory. The clock signals are converted to differential pairs in the FPGA pins. Note that the differential pair signals are not shown in simulation.
Local Interface	
Bus Interface Type	Specifies the user interface on FPGA fabric side. Only the Native Interface is currently supported. AMBA is also supported.
Enable APB I/F	Enables APB interface. This is useful for systems with a CPU. For systems without a CPU, you can disable this option.
AXI Data Width	Specifies the width of the AXI4 data signal.
AXI Address Width	Specifies number of address pins in memory interface, dependent on the SDRAM density.
AXI Maximum Burst Length	Specifies the maximum supported AXI4 burst length.
AXI ID Width	Specifies the width of the AXI4 ID signal.
AXI Outstanding Command Support	Specifies the number of supported outstanding AXI4 transactions.
FIFO RD DEPTH	Shows the FIFO read depth information.
Additional Configuration Group	
Data ready to Write Data delay	This option is for information only. The user logic sends the write data to the controller after a one-clock cycle delay with respect to the datain_rdy_o signal.
Controller reset to Memory	When you disable (uncheck) this option, the reset signals mem_rst_n_i and em_ddr_reset_n_o are no longer available, and external logic should take care of the DDR3 Memory reset. If you enable the option, the IP Core responds to the mem_rst_n_i signal. When it is 1'b0, it asserts the m_ddr_reset_n_o signal for a minimum of 200 µs as per the DDR3 Memory requirement.
MC CK/Command Delay Value	Specifies the DDR clock delay so that the first DQS toggle is at CK = 0 during write leveling. It is recommended to increase this value if write leveling fails.
MC DQS/DQ ODT Value	Memory controller termination resistance value at FPGA.
Write DQ/DM Delay Direction	Specifies the write DQ/DM delay adjustment direction.
Write DQ/DM Delay Value	Specifies the write DQ/DM delay adjustment value.
Read DQ Delay Value	Specifies the read DQ delay adjustment value
Read DQ Phase Adjustment Direction	Specifies the read DQ delay adjustment direction.
Read DQ Phase Adjustment Value	Specifies the read DQ delay adjustment value.

3.2. Memory Device Setting

Table 3.7. Address Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Row Size	13, 14, 15, 16	Calculated	Default value is set by Select Memory and Configuration: Micron DDR3 1Gb-187E:13, 14 Micron DDR3 2Gb-187E:14, 15 Micron DDR3 4Gb-187E:15,16 Custom: 13 Editable when Select Memory = Custom
Column Size	10, 11, 12	10	Editable when Select Memory = Custom

Table 3.8. Auto Refresh Control Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Auto Refresh Burst Count[1]	1, 2, 3, 4, 5, 6, 7, 8	8	—
External Auto Refresh Port	Checked/Unchecked	Unchecked	There is a known issue on clash of internal and external refresh when this is enabled.

Note:

1. When auto refresh burst count is set to 8, the refresh period is reduced to approximately *7.6 µs* instead of *7.8 µs* to allow margin for internal arbitration or queueing delays, in order to prevent postponement of the refresh beyond 8 × TREFI.

Table 3.9. Mode Register Initial Setting Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Burst Length	Fixed 4 (BC4), Fixed 8 (BL8), On the fly	Fixed 8 (BL8)	—
CAS Latency	5, 6, 7, 8	Calculated	Selectable and default values are updated based on Select Memory and MemClock. For CAS Latency = 8, there is a known issue in design and should be avoided.
Burst Type	Sequential/Interleave	Sequential	—
Write Recovery	5, 6, 7, 8, 10, 12	6	If MemClock = 533, the minimum and default value are both 8.
DLL Control for PD	Slow Exit/Fast Exit	Slow Exit	—
ODI Control	RZQ/6, RZQ/7	RZQ/6	—
RTT_Nom (Ohm)	RZQ/2, RZQ/4, RZQ/6, RZQ/8, RZQ/12, Disabled	RZQ/6	—
CAS Write Latency	5, 6, 7, 8, 10, 12	5	Selectable and default values are updated based on Select Memory and MemClock.
RTT_WR	RZQ/2, RZQ/4, Off	RZQ/4	—

Table 3.10. Memory Device Setting

Attribute	Description
Address Group	
Row Size (ROW_WIDTH)	Indicates the default row address size used in the selected memory configuration.
Column Size	Indicates the default column address size used in the selected memory configuration.
Auto Refresh Control Group	
Auto Refresh Burst Count	Indicates the number of Auto Refresh commands that the DDR3 SDRAM Controller IP Core is set to send in a single burst. Refer to REFRESH Support for more details.

Attribute	Description
External Auto Refresh Port	Specifies the generation of refresh commands to the memory. If *unchecked,* the controller automatically generates refresh commands to the memory at intervals defined by the Auto Refresh Burst Count and memory refresh timing requirements. If *checked*, the user logic generates a refresh request to the controller via ext_auto_ref_i signal. Refer to REFRESH Support for more details.
Mode Register Initial Setting Group	
Burst Length	Sets the Burst length value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
CAS Latency	Sets the CAS Latency value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
Burst Type	Sets the Burst Type value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
Write Recovery	Sets the Write Recovery value in Mode Register 0 during initialization. The value is in terms of Memory clock. This value remains until you write a different value to the Mode Register.
DLL Control for PD	Sets the DLL Control for Pre-charge PD value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
ODI Control	Sets the Output Driver Impedance Control value in Mode Register 1 during initialization. This value remains until you write a different value to the Mode Register.
RTT_Nom (Ohm)	Sets the nominal termination, Rtt_Nom, value in Mode Register 1 during initialization. This value remains until you write a different value to the Mode Register.
CAS Write Latency	Sets the CAS Write Latency, CWL, value in Mode Register 2 during initialization. This value remains until you write a different value to the Mode Register.
RTT_WR	Sets the Dynamic ODT termination, Rtt_WR, value in Mode Register 2 during initialization. This value remains until you write a different value to the Mode Register.

3.3. Memory Device Timing

The following section describes the parameters available in the Memory Device Timing tab of the IP parameter editor. The default value for the attributes listed under Memory Device Timing Setting Group differs for each DDR Command Frequency value and is based on the JESD209-4C SDRAM standard. You may need to manually adjust these values based on the selected LPDDR4 device.

Table 3.11. Command and Address Timing Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Manually Adjust	Checked/Unchecked	Unchecked	—
TRTP (tCLK)	4–65,536	4	Enabled when manually adjust is checked
TWTR (tCLK)	4–65,536	4	Enabled when manually adjust is checked
TMRD(tCLK)	4–65,536	4	Enabled when manually adjust is checked
TMOD (tCLK)	12–65,536	12	Enabled when manually adjust is checked
TRCD (tCLK)	4–65,536	8	Enabled when manually adjust is checked
TRP (tCLK)	4–65,536	8	Enabled when manually adjust is checked
TRC (tCLK)	15–65,536	20	Enabled when manually adjust is checked
TRAS (tCLK)	12–65,536	16	Enabled when manually adjust is checked
TFAW (tCLK)	12–65,536	16	Enabled when manually adjust is checked
TRRD (tCLK)	4–65,536	4	Enabled when manually adjust is checked

Table 3.12. Calibration Timing Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
TZQINIT(tCLK)	512–65,536	512	Enabled when manually adjust is checked
TZQCS (tCLK)	64–65,536	64	Enabled when manually adjust is checked
TZQOPER (tCLK)	256–65,536	256	Enabled when manually adjust is checked

Table 3.13. Refresh, Reset and Power Down Timing Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
TCKE (tCLK)	3–65,536	4	Enabled when manually adjust is checked
TRFC (tCLK)	44–28,080	44	Enabled when manually adjust is checked Default value is set by *Select Memory*: • Micron DDR3 1Gb-187E:44 • Micron DDR3 2Gb-187E:64 • Micron DDR3 4Gb-187E:104 Custom: 140
TCKESR (tCLK)	4–65,536	4	Enabled when manually adjust is checked
TPD (tCLK)	3–65,536	4	Enabled when manually adjust is checked
TXPDLL (tCLK)	10–65,536	12	Enabled when manually adjust is checked
TXPR (tCLK)	48–65,536	48	Enabled when manually adjust is checked Default value is set by *Select Memory*: • Micron DDR3 1Gb-187E:48 • Micron DDR3 2Gb-187E:68 • Micron DDR3 4Gb-187E:108 Custom: 144
TREFI (tCLK)	44–4,160	3,120	Enabled when manually adjust is checked

Table 3.14. Write Leveling and ODT Timing Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
TWLMRD (tCLK)	40–65,536	40	Enabled when manually adjust is checked
TWLDQSEN (tCLK)	25–65,536	28	Enabled when manually adjust is checked
TWLO (ns)	0–9	4	Enabled when manually adjust is checked
ODTH4 (tCLK)	4	4	Display only
ODTH8 (tCLK)	6	6	Display only

Note: When you *check* the manually adjust option, the system rounds all timing parameters in tCLK up to the nearest value divisible by 2 when the Gearing Ratio is 4:1, and divisible by 4 when the Gearing Ratio is 8:1.

Table 3.15. Memory Device Timing Definitions

Attribute	Description
Command and Address Timing Group	
Manually Adjust	Check this box to manually set any of the memory timing parameters. If you need to change any of the default values, you must check the Manual Adjust checkbox. This selection enables you to modify the memory timing parameters.
TRTP (tCLK)	Internal READ Command to PRECHARGE Command delay.
TWTR (tCLK)	Delay from start of internal write transaction to internal read command.
TMRD(tCLK)	Mode Register Set command cycle time.
TMOD (tCLK)	Mode Register Set command update delay.
TRCD (tCLK)	ACT to internal read or write delay time.
TRP (tCLK)	PRE command period.
TRC (tCLK)	ACT to ACT or REF command period.
TRAS (tCLK)	ACTIVE to PRECHARGE command period.

Attribute	Description
TFAW (tCLK)	Four activate window for 1 kB/2 kB page size.
TRRD (tCLK)	ACTIVE to ACTIVE command period for 1 kB/2 kB page size.
Calibration Timing Group	
TZQINIT(tCLK)	Power-up and RESET calibration time.
TZQCS (tCLK)	Normal operation Short calibration time.
TZQOPER (tCLK)	Normal operation Full calibration time.
Refresh, Reset, and Power Down Timing Group	
TCKE (tCLK)	CKE minimum pulse width.
TRFC (tCLK)	REF command to ACT or REF command time.
TCKESR (tCLK)	Minimum CKE low width for Self-Refresh entry to exit timing.
TPD (tCLK)	Power Down Entry to Exit Timing.
TXPDLL (tCLK)	Exit Precharge Power Down with DLL frozen to commands requiring a locked DLL.
TXPR (tCLK)	Exit Reset from CKE HIGH to a valid command.
TREFI (tCLK)	Average periodic refresh interval.
Write Leveling and ODT Timing Group	
TWLMRD (tCLK)	First DQS/DQS# rising edge after write leveling mode is programmed.
TWLDQSEN (tCLK)	DQS/DQS# delay after write levelling mode is programmed.
TWLO (ns)	Write leveling output delay.
ODTH4 (tCLK)	ODT high time without write command or with write command and BC4.
ODTH8 (tCLK)	ODT high time with Write command and BL8.

3.4. Example Design

This section describes the type of board to be used with the example design. This allows for automated pin location assignment in the constraints file.

Table 3.16 Example Design

Attribute	Selectable Values	Default	Dependency on Other Attributes
Select Development Board	None, Certus-NX_Versa_Board	None	—

4. Signal Description

The following section covers the input and output signals of the DDR3 Memory Controller for Nexus Devices. The available signals depend on the selected configuration of the DDR3 Memory Controller IP.

4.1. Clock and Reset

The following section describes the interface ports for clock and reset in the DDR3 Memory Controller IP.

Table 4.1. Clock and Reset Port Definitions

Port Name	I/O	Width	Description
pll_refclk_i	In	1	Reference clock to the PLL. This signal is available when the *Enable PLL* attribute is checked.
eclk_i	In	1	Clock of the DDR primitives. The frequency should be the same as the *MemClock* attribute. This signal is available when you uncheck the *Enable PLL* attribute.
sync_clk_i	In	1	Clock for initializing the DDR primitives. This clock is slow and asynchronous to eclk_i. This signal is available when you uncheck the *Enable PLL* attribute.
pll_lock_i	In	1	This signal indicates that the PLL has achieved a lock state and the eclk_i is now stable. The initialization of DDR primitives starts when this signal asserts. This signal is available when you uncheck the *Enable PLL* attribute.
clocking_good_o	Out	1	Signal from PLL indicating stable clock condition.
sclk_o	Out	1	System clock used by the controller's core module. You may use this clock for DDR3 controller interface logic.
pclk_i	In	1	Clock for the APB Interface module. This is a slow clock, meant for configuration only.
rst_n_i	In	1	This signal provides an asynchronous reset. By default, this signal resets the IP core and DDR3 memory when asserted, including the AXI4 interface, but not the APB interface. It is recommended that arst_n_i be asserted whenever rst_n_i is asserted to ensure consistency between the DDR3 SDRAM AXI4 interface and corresponding AXI4 manager.
srst_n_o	Out	1	Reset synchronous deassertion copy of rst_n_i. Deassertion edge is synchronous to sclk_o.
prst_n_i	In	1	This signal provides an asynchronous reset for the APB Interface module. It only resets the APB Interface module.
arst_n_i	In	1	This signal provides an asynchronous reset for the AXI4 Interface module. When asserted, it resets the AXI4 Interface and the Soft Memory Controller.
pll_rst_n_i	In	1	This signal provides an asynchronous reset when you check the *Enable External PLL Reset* option in the Module/IP Block Wizard. It allows you to reset the PLL only and does not reset the memory controller.
mem_rst_n_i	In	1	This signal provides an asynchronous reset when you check the *Controller Reset to Memory* in the Module/IP Block Wizard. It allows you to reset the memory device only and does not reset the memory controller.
em_ddr_reset_n_o	Out	1	This signal provides an asynchronous reset signal from the controller to the memory device. The controller asserts this signal during of power-on reset, or when rst_n_i, or mem_rst_n_i is active.

4.2. APB Config Interface

Table 4.2. APB Interface Port Definitions

Port Name	I/O	Width	Description
apb_paddr_i	in	32	Address. APB address bus.
apb_penable_i	in	1	Enable. Indicates subsequent cycles of an APB transaction.
apb_pwdata_i	in	32	Write Data. This bus drives the intended data to be written.
apb_pwrite_i	in	1	Direction. Indicates if it is a WRITE or READ transaction. 1: Write 0: Read
apb_psel_i	in	1	Select. This signal indicates that the subordinate is selected for the current transaction.
apb_pready_o	out	1	Ready. This signal is used to extend a transaction.
apb_prdata_o	out	32	Read Data. This bus drives the intended data to be read.
apb_pslverr_o	out	1	Error. Indicates a transaction failure.

4.3. AXI4 Data Interface

Table 4.3. AXI4 Interface Port Definitions

Port Name	I/O	Width	Description
axi_awvalid_i	in	1	Write Address Valid. This signal indicates that the write address and control information are valid.
axi_awid_i	in	AXI_ID_WIDTH	Write Address ID. This signal provides an identification tag for the write address and control information.
axi_awaddr_i	in	AXI_ADDR_WIDTH	Write Address. This signal provides the address of the first beat in a burst transfer.
axi_awlen_i	in	8	Burst Length. This signal specifies the number of beats in a burst transfer.
axi_awsize_i	in	3	Burst Size. This signal specifies the size of the transfer in terms of number of bytes.
axi_awburst_i	in	2	Burst Type. This signal specifies whether it is FIXED, INCR or WRAP mode. Only INCR is supported.
axi_awlock_i	in	1	Lock Type. Not supported.
axi_awcache_i	in	4	Memory Type. Not supported.
axi_awprot_i	in	3	Protection Type. Not supported.
axi_awqos_i	in	4	Quality of Service. Not supported.
axi_awregion_i	in	4	Region Identifier. Not supported.
axi_awuser_i	in	1	User Signal. Not supported.
axi_awready_o	out	1	Write Address Ready. This signal indicates that the subordinate is ready to accept the write address and control information.
axi_wvalid_i	in	1	Write Valid. This signal indicates the write data and strobes are valid.
axi_wdata_i	in	AXI_DATA_WIDTH	Write Data.
axi_wstrb_i	in	AXI_DATA_WIDTH/4	Write Strobe. This signal indicates valid byte lanes.
axi_wlast_i	in	1	Write Last. This signal indicates the last beat of the burst.
axi_wuser_i	in	1	User Signal. Not supported.
axi_wready_o	out	1	Write Ready. This signal indicates that the subordinate is ready to accept the write data and strobe.
axi_bready_i	in	1	Response Ready. This signal indicates that the manager is ready to accept the write response.
axi_bvalid_o	out	1	Response Valid. This signal indicates that the write response is valid.

Port Name	I/O	Width	Description
axi_bid_o	out	AXI_ID_WIDTH	Response ID. This signal provides an identification tag for the write transaction.
axi_bresp_o	out	2	Write Response. This signal indicates the write transaction status.
axi_buser_o	out	1	User Signal. Not supported.
axi_arvalid_i	in	1	Read Address Valid. This signal indicates that the read address and control information are valid.
axi_arid_i	in	AXI_ID_WIDTH	Read Address ID. This signal provides an identification tag for the read address and control information.
axi_araddr_i	in	AXI_ADDR_WIDTH	Read Address. This signal provides the address of the first beat in a burst transfer.
axi_arlen_i	in	8	Burst Length. This signal specifies the number of beats in a burst transfer.
axi_arsize_i	in	3	Burst Size. This signal specifies the size of the transfer in terms of number of bytes.
axi_arburst_i	in	2	Burst Type. This signal specifies whether it is FIXED, INCR or WRAP mode. Only INCR is supported.
axi_arlock_i	in	1	Lock Type. Not supported.
axi_arcache_i	in	4	Memory Type. Not supported.
axi_arprot_i	in	3	Protection Type. Not supported.
axi_arqos_i	in	4	Quality of Service. Not supported.
axi_arregion_i	in	4	Region Identifier. Not supported.
axi_aruser_i	in	1	User Signal. Not supported.
axi_arready_o	out	1	Read Address Ready. This signal indicates that the subordinate is ready to accept the read address and control information.
axi_rready_i	in	1	Read Ready. This signal indicates that the manager is ready to accept the read response.
axi_rvalid_o	out	1	Read Valid. This signal indicates that the read response is valid.
axi_rid_o	out	AXI_ID_WIDTH	Read ID. This signal provides an identification tag for the read transaction.
axi_rdata_o	out	AXI_DATA_WIDTH	Read Data.
axi_rresp_o	out	2	Read Response. This signal indicates the read transaction status.
axi_rlast_o	out	1	Read Last. This signal indicates the last beat of the read response.
axi_ruser_o	out	1	User Signal. Not supported.

4.4. Native Interface

Table 4.4. Native Interface Port Definitions

Port Name	I/O	Width	Description
init_start_i	In	1	Initialization start request. Assert this signal to initiate memory initialization either right after the power-on reset or before sending the first user command to the memory controller.
init_done_o	Out	1	Initialization done output. This signal is asserted for one clock period after the core completes memory initialization and write leveling. When sampled high, you must immediately deassert the input signal init_start_i at the same edge of the sampling clock.
cmd_valid_i	In	1	Command and address valid input. When you assert this signal, the addr_i, cmd_i and cmd_burst_cnt_i inputs are considered valid.

Port Name	I/O	Width	Description
cmd_rdy_o	Out	1	Command ready output. When you assert this signal, it indicates that the core is ready to accept the next command and the corresponding address. This signal is active for one clock period.
cmd_i	In	4	This signal provides user command input to the memory controller.
cmd_burst_cnt_i	In	5	Command burst count input. This signal indicates the number of times a given read or write command is to be repeated by the controller automatically. The controller generates the address for each repeated command sequentially as per burst length of the command. The burst range is from 1 to 32, with *0* indicates 32 repetitions.
ofly_burst_len_i	In	1	On-the-fly burst length for current command. 0 = BC4 1 = BL8 This input is valid only when Mode Reg0 is set for on-the-fly mode. When set, the system samples this input when cmd_valid_i and cmd_rdy_o are high.
addr_i	In	ADDR_WIDTH	User read or write address input to the memory controller. Refer to Table 2.3 for more details.
datain_rdy_o	Out	1	Data ready output. When asserted, this signal indicates that the core is ready to receive the write data.
write_data_i	In	DSIZE	Write data input from user logic to the memory controller. The user-side write data width is four times the width of the memory data bus.
data_mask_i	In	DSIZE/8	Data mask input for write data. Each bit masks a corresponding byte of the local write data.
read_data_o	Out	DSIZE	Read data output from the memory controller to the user logic.
read_data_valid_o	Out	1	Read data valid output. When asserted, this signal indicates that the data on the read_data_o bus is valid.
ext_auto_ref_i	In	1	Refresh user request. This signal is available only when you select the *External Auto Refresh Port* attribute in the Module/IP Block Wizard.
ext_auto_ref_ack_o	Out	1	Completion of memory refresh in response to ext_auto_ref_i signal assertion. This signal is available only when you check the *External Auto Refresh Port* in the Module/IP Block Wizard.
wl_start_i	In	1	Write levelling start. This signal is available only when you check the *Enable External PLL Reset* attribute in the Module/IP Block Wizard.
wl_done_o	Out	1	Completion of write levelling in response to wl_start_i signal assertion. This signal is available only when you check the *Enable External PLL Reset* in the Module/IP Block Wizard.
wl_err_o	Out	1	Write levelling error. This signal indicates a failure in Write leveling. The controller does not work properly if there is a write leveling error. Check this signal when the init_done_o signal is asserted.
rt_err_o	Out	1	Read Training error. This signal indicates a failure in Read Training process. The controller does not work properly if there is a Read Training error. Check this signal when the init_done signal is asserted.

4.5. DDR3 Memory Interface

The following section describes the interface ports for DDR3 SDRAM.

Table 4.5. DDR3 Interface Port Definitions

Port Name	I/O	Width	Description
em_ddr_clk_t_o	Out	CLKO_WIDTH	Positive side of the differential memory clock generated by the controller.
em_ddr_clk_c_o	Out	CLKO_WIDTH	Negative side of the differential memory clock generated by the controller.
em_ddr_cke_o	Out	CKE_WIDTH	Memory clock-enable generated by the controller.
em_ddr_addr_o	Out	ROW_WIDTH	Memory address bus – multiplexed row and column address for the memory.
em_ddr_ba_o	Out	3	Memory bank address.
em_ddr_data_io	In/Out	DATA_WIDTH	Memory bi-directional data bus.
em_ddr_dm_o	Out	DATA_WIDTH/8	DDR3 memory write data mask – to mask the byte lanes for byte-level write.
em_ddr_dqs_io	In/ Out	DQS_WIDTH	Memory bi-directional data strobe.
em_ddr_cs_n_o	Out	CS_WIDTH	Memory chip select.
em_ddr_cas_n_o	Out	1	Memory column address strobe.
em_ddr_ras_n_o	Out	1	Memory row address strobe.
em_ddr_we_n_o	Out	1	Memory write enable.
em_ddr_odt_o	Out	CS_WIDTH	High Output Memory on-die termination control.

5. Register Description

The register is available only when the APB Interface is enabled.

Table 5.1. APB Interface Register Map

Register	Access	Offset	Name	Bits	Description
Config Reg	Read/Write	0x00	Enable APB Config	3	Write a '1' to this bit to enable configuration access mode. This bit is sticky. To return to data access mode, write a '0' to this bit.
			Start Write Levelling	2	Write a '1' to this bit to begin write leveling. This is a strobe bit.
			Reserved	1	This is a reserved bit.
			Initialization	0	Write a '1' to this bit to begin initialization. This is a strobe bit.
Status Reg	Read Only	0x04	Power Down Mode	8	1: In power down mode 0: Out of power down mode
			Write DQ/DM-DQS Phase Adjustment Error	0	This is part of the extended training capability. When enabled, Write DQ/DM to DQS phase is adjusted to the optimal position When extended training is disabled, this flag remains 0. 1: Write DQ/DM-DQS phase adjustment fail 0: Write DQ/DM-DQS phase adjustment pass
			Read DQ-DQS Phase Adjustment Error	0	This is part of the extended training capability. When enabled, Read DQ to DQS phase is adjusted to the optimal position When extended training is disabled, this flag remains 0. 1: Read DQ-DQS phase adjustment fail 0: Read DQ-DQS phase adjustment pass
			MC CK/Command Delay Value Actual	undefined	During write leveling, if the value for the parameter "MC CK/Command Delay Value" is insufficient, the internal training engine will auto adjust this delay until it is sufficient for a successful write leveling. The final value will be captured here as the actual value. If the value for the parameter "MC CK/Command Delay Value" is sufficient, then this value gets reflected here as the actual.
			Self-refresh mode	7	1: Self refresh mode 0: Out of self-refresh mode
			PLL Lock	6	1: PLL Lock asserted 0: PLL Lock not asserted
			Command Ready	5	1: Ready to accept commands 0: Not ready to accept commands
			Clock Good	4	1: Clock is ready 0: Clock not ready
			Read Training Error	3	1: Read training failed 0: Read training successful
			Write Levelling Error	2	1: Write leveling failed 0: Write leveling successful
			Write Level Done	1	1: Write leveling done 0: Write leveling not done
			Initialization Done	0	1: Initialization done 0: Initialization not done

Register	Access	Offset	Name	Bits	Description
User Command Reg	Read/Write[1]	0x08	User Command	[3:0]	Supported user commands: • Power down Entry • Self-Refresh Entry • Self-Refresh Exit • Power down Entry • ZQ Calibration Long • ZQ Calibration Short Refer to the User Commands section for the complete list of supported user commands via Native interface.

Note:

1. Write access to the user command register is available only when you set the *Enable APB Config* bit to '1'.

6. DDR3 SDRAM Controller Example Design

This section and its associated subsections describe the DDR3 SDRAM Controller Example Design, which is available to users for synthesis and simulation after successful IP generation.

6.1. Overview

Table 6.1 summarizes the IP parameter configurations reflected in the Nexus LPDDR4 Memory Controller Example Design. To generate the DDR3 SDRAM Controller IP Core, refer to the Designing and Simulating the IP section of this user guide.

Table 6.1. Supported Example Design Configurations

Attribute	Supported Setting
Gearing Ratio	4:1, 8:1
I/O Buffer Type	SSTL15_I, SSTL15_II
Select Memory	Micron DDR3 1Gb-187E, Micron DDR3 2Gb-187E, Micron DDR3 4Gb-187E
Enable PLL	Checked/Unchecked
Enable External PLL Reset	Checked/Unchecked
PLL Reference Clock from Pin	Checked/Unchecked
RefClock (MHz)	100 (111 when MemClock is 333)
MemClock (MHz)	300, 333, 400, 533 (Only when Gearing Ratio is 8:1)
Memory Data Bus Width	8, 16, 24, 32
Configuration	x8, x16
Bus Interface Type	AMBA, NATIVE
AXI ID Width	1, 2, 3, 4, 5, 6, 7, 8
Data Ready to Write Data Delay	1,2
Write Levelling	Checked/Unchecked
Controller Reset to Memory	Checked/Unchecked
MC CK/Command Delay Value	0 to 127
MC DQS/DQ ODT Value	OFF, 40, 50, 60, 75
Write DQ/DM Delay Direction	Increment, Decrement
Write DQ/DM Delay Value	0 to 63
Read DQ Delay Value	0 to 63
Memory Device Settings Tab	All
Memory Device Timing Tab	All

6.2. Synthesis Example Design

Figure 6.1 represents a block diagram of the DDR3 SDRAM Controller synthesis example design.

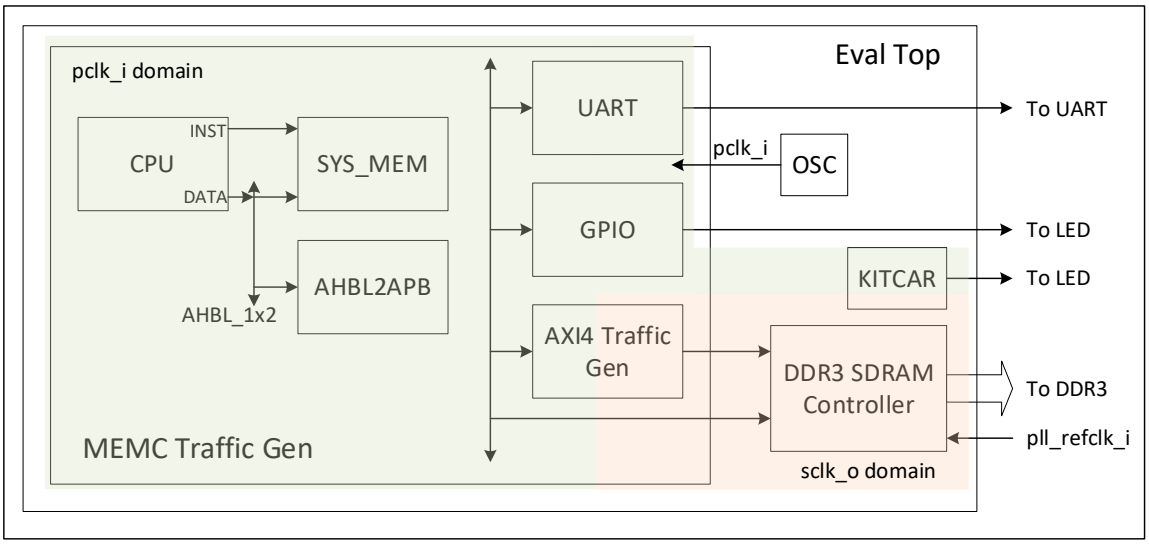

Figure 6.1. Memory Controller IP Core Functional Diagram

The main blocks that make up the synthesizable example design include the following:

- RISC-V CPU subsystem, which enables local UART for users to initialize and train the DDR3 interface and perform data access checks.
- System memory, which holds the RISC-V test program.
- GPIO block, allowing users to monitor signals to verify the functionality of the clocks and data access checks.
- Traffic generator, which implements a series of pseudo-random (PRBS) reads and writes and compares the result to display over UART.
- DDR3 SDRAM Controller IP Core, which provides an interface to external DDR3 memory to issue reads and writes.
- Oscillator, which generates a 90 MHz clock APB interface clock (pclk_i).

The synthesizable example design consists of a test program stored in system memory and fetched by the CPU instruction port. The CPU data port accesses system memory and connects to the following: Configuration Set Registers (CSRs), UART, GPIO, traffic generator, and DDR3 SDRAM Controller. The test program associated with the example design contains the following features:

- Waits for user keypress over a serial terminal connection upon the assertion of the reset signal rstn_i.
- Configures the DDR3 SDRAM Controller to perform a complete reset, initialization, and training of the DDR3 interface.
- Performs a series of loop-back data access checks.
- Calculates performance of the DDR3 interface.

When accessing the CSR, the selected user data interface protocol (AHB-Lite or AXI4) is converted to APB via a bridge. The UART connection allows users to run selected features within the test program and displays the result over a serial terminal. The GPIO block allows users to easily monitor signals for debugging purposes. In other words, if a failure or error is encountered while running the test program, the GPIO output can be used to determine which part of the test program was last executed successfully. The traffic generator runs a series of six different data access patterns:

- Test 0: single write followed by single read
- Test 1: burst write with address incrementing by 2 followed by a burst read
- Test 2: burst write with address incrementing by 4 followed by a burst read
- Test 3: burst write with address incrementing by 8 followed by a burst read
- Test 4: burst write with address incrementing by 8 followed by a delay before issuing burst read
- Test 5: burst write with address incrementing by 32 followed by a burst read

For information on how to generate and perform hardware evaluation of the synthesis example design, refer to the Designing and Simulating the IP section of this user guide.

6.3. Simulation Example Design

The simulation example design is similar to the synthesizable example design, with the following changes:

- Disables the UART connection.
- External DDR3 memory is replaced with a DDR3 memory model.
- The IP Core is configured with the *Simulation Mode Enable* checked.

The UART connection is disabled in simulation because it takes a long time to simulate, and the DDR3 memory model allows simulation of user-initiated operations to DDR3 SDRAM.

7. Designing and Simulating the IP

This section describes the steps required within Lattice Radiant software to configure and generate the DDR3 SDRAM Controller for Nexus Devices. This section also provides information regarding design implementation and hardware evaluation of the synthesizable example design. The screenshots provided are for reference only. The details may vary depending on the version of the IP or software being used.

7.1. Generating the IP

This section describes the steps required to create, configure, and generate an instance of the DDR3 SDRAM Controller for Nexus Devices.

7.1.1. Creating a Radiant Project

To generate an instance of the DDR3 SDRAM Controller IP, first create a Lattice Radiant project:

1. Launch the Lattice Radiant software and select **File > New > Project**. This opens the **New Project** dialog box. Click **Next**.

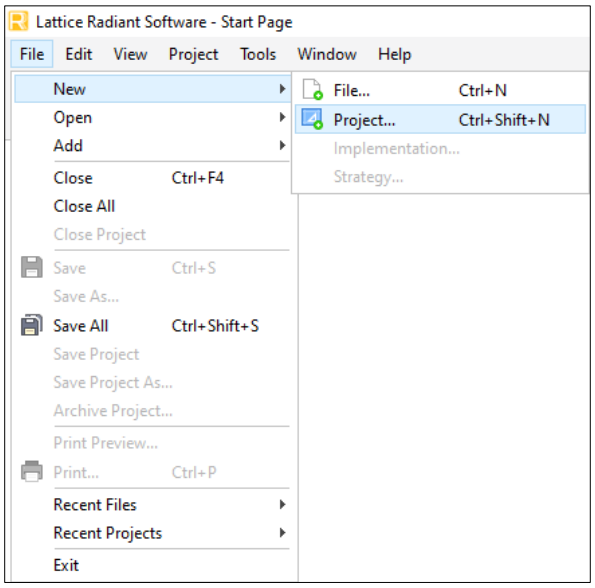

Figure 7.1. Creating a New Radiant Project

2. Specify a name (<project_name>) for the Lattice Radiant project, a directory (<project_directory>) to store the project files, and a top-level design implementation name (<top_level_instance_name>). Click **Next** twice.

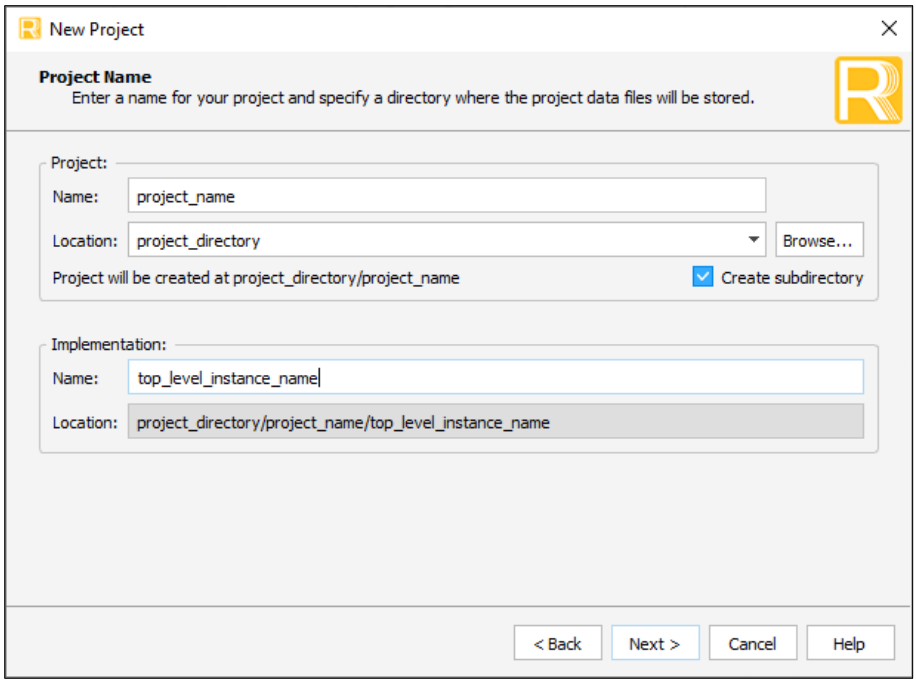

Figure 7.2. New Project Settings

3. Under **Family**, select Certus-NX. Under **Device, Package, Operating Condition, and Performance Grade**, make the appropriate selections representative of the selected device part number. Click **Next**.

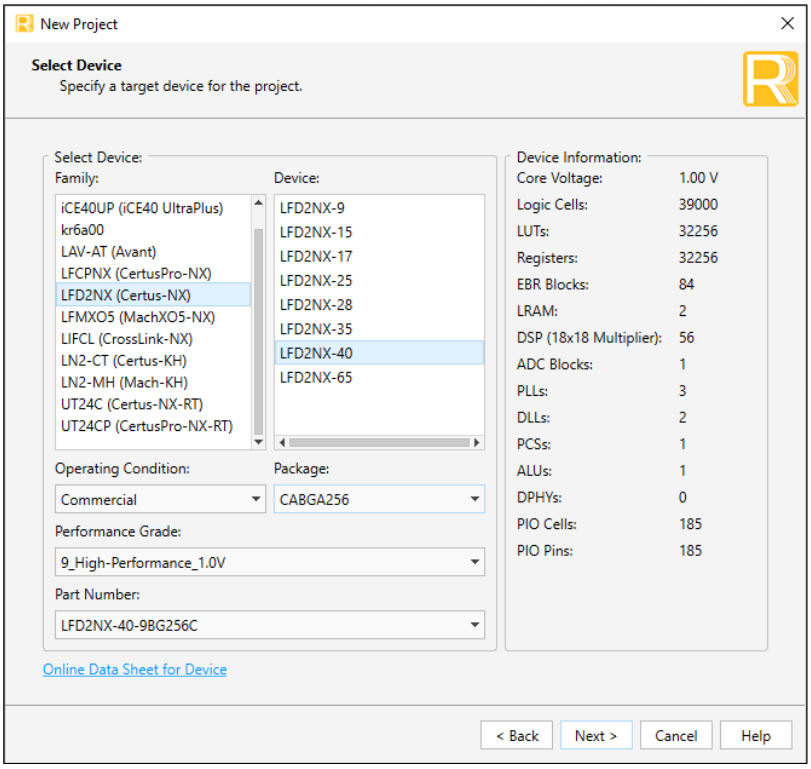

Figure 7.3. Project Device Settings

4. Specify the desired synthesis tool for implementation the Lattice Radiant project. Click **Next** and **Finish**.

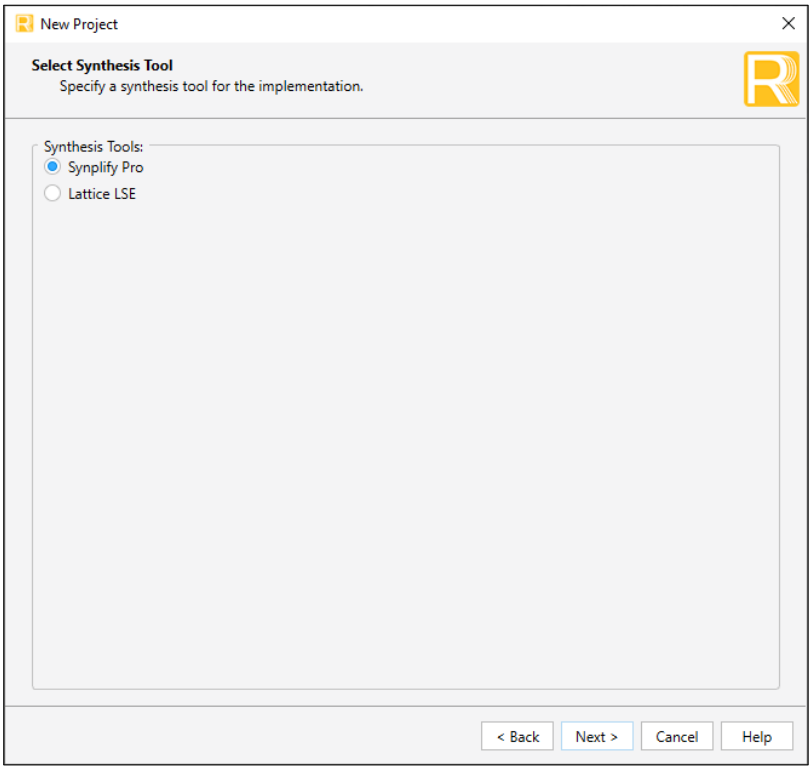

Figure 7.4. Project Synthesis Tool Selection

7.1.2. Configuring and Generating the IP

The following steps illustrate how to generate the **DDR3** SDRAM Controller IP Core in Lattice Radiant software.

1. In the **IP Catalog** (Tools > IP Catalog), locate and double-click on the DDR3 SDRAM Controller IP listed under **IP > Processors_Controllers_and_Peripherals**.
   a. If the DDR3 SDRAM Controller IP is not installed on the system, select the IP on Server tab within the IP Catalog.
   b. Click on **Download from Lattice IP Server** icon next to the DDR3 SDRAM Controller IP for installation. This will open an **IP License Agreement** dialog box.
   c. Click **Accept** and then click on the **Refresh IP Catalog** icon.
   d. Locate the installed DDR3 SDRAM Controller IP under the **IP on Local** tab within the IP Catalog and double-click.
2. The **Module/IP Block Wizard** dialog box will open. Provide a name (<instance_name>) and directory (<instance_directory>) for the Memory Controller IP, where the default directory is set to <project_directory>/<project_name>. Click **Next**.

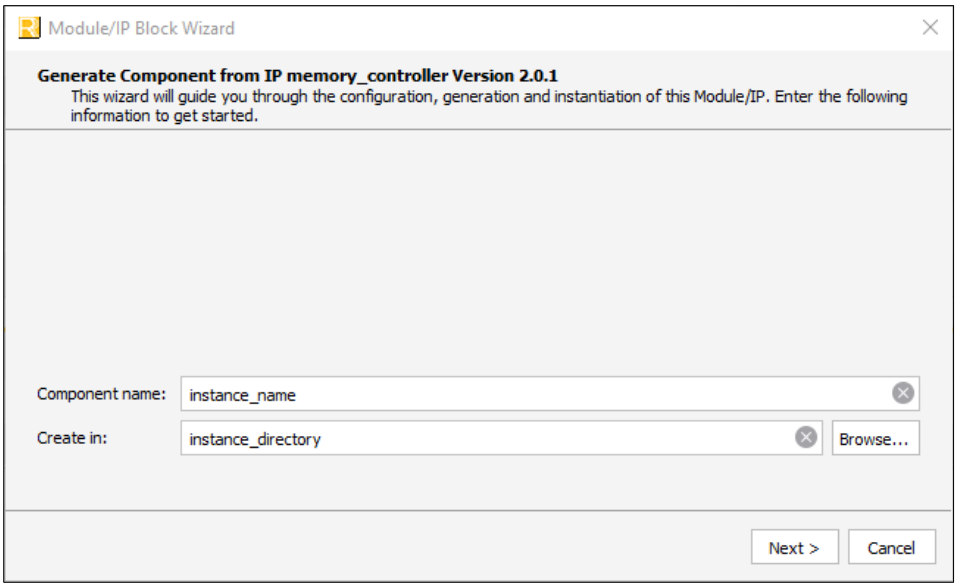

Figure 7.5. IP Instance Settings

3. The DDR3 SDRAM Controller IP editor contains multiple tabs that need to be configured according to the desired LPDDR4 memory interface implementation. The following table provides high-level guidance for configuring the tabs in the DDR3 SDRAM Controller IP editor. For detailed information on the individual attributes, refer to the DDR3 Operation Description section of this user guide.

Table 7.1. Memory Controller Attribute Guidelines

Memory Controller IP Attribute Tab	Guidelines
General	• Ensure that you enter the Memory clock frequency (*DDR Command Frequency*) correctly. • Refer to the datasheet for the selected DDR3 memory device to ensure the channel density (*DDR Density per Channel*) is set correctly. • Ensure that you check the Simulation Mode Enable option only if simulation is intended. Otherwise, keep it unchecked.
Memory Device Setting	Refer to the datasheet for the selected DDR3 memory device to modify the initial setting.
Memory Device Timing	Refer to the datasheet for the selected DDR3 memory device to modify the default timing parameters as needed.
Example Design	Select the type of evaluation board for this IP.

4. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in Figure 7.6. Click **Finish**.

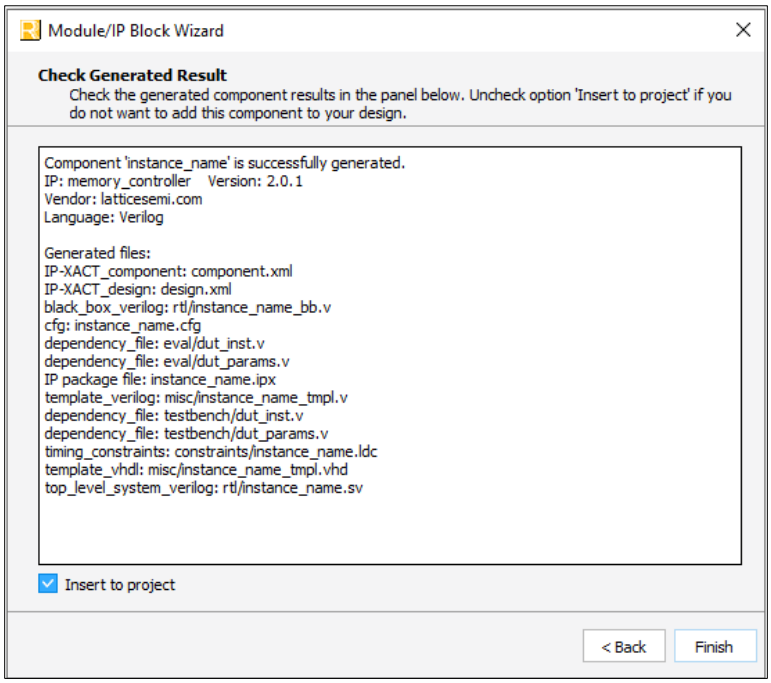

Figure 7.6. IP Generation Result

5. All the generated files are placed under the <instance_directory>/<instance_name> directory path. The generated files under the <instance_directory>/<instance_name> directory are listed in the following table.

Table 7.2. Generated File List

File Name	Description
component.xml	Contains the ipxact:component information of the IP.
design.xml	Lists the set parameters of the IP in IP-XACT 2014 format.
<instance_name>.cfg	Lists only the configured/changed parameter values set during IP configuration.
<instance_name>.ipx	Lists the files associated with IP generation.
constraints/<instance_name>.sdc	Pre-Synthesis and Pre-Map constraints for the IP.
constraints/<instance_name>.ldc	Defines the I/O standard for DDR3 memory interface signals.
eval/clock_constraint.sdc	Pre-synthesis constraint for setting the PLL reference clock frequency of the DDR3 SDRAM Controller Example Design.
eval/constraint.pdc	Post-synthesis constraints for the DDR3 SDRAM Controller Example Design.
eval/dut_inst.v	Instantiation of generated IP core in eval_top.sv for DDR3 Memory Controller Example Design.
eval/dut_params.v	Defines local parameters for eval_top based on parameter values set during IP configuration for DDR3 Memory Controller Example Design.
eval/eval_top.sv	Top-level RTL file for the DDR3 SDRAM Controller Example Design.
eval/kitcar.v	Counter that drives LEDs to indicate that internal clocks (pclk and aclk) are active in DDR3 SDRAM Controller Example Design.
eval/lscc_pll.v	PLL responsible for generating eclk_i and sync_clk_i to the IP when the Enable PLL attribute is disabled.
eval/ddr_clks_src.v	Wrapper for instantiating lscc_pll.v.
eval/pll_instance.vh	Instantiation of ddr_clks_src.v in eval_top.sv.
eval/amba_bridge/	Contains RTL for AXI/APB bus interface to Native I/F.

File Name	Description
eval/traffic_gen/	Contains RTL files for the DDR3 SDRAM Controller Example Design.
misc/<instance_name>_tmpl.v misc/<instance_name>_tmpl.vhd	These files provide instance templates for the IP core.
rtl/<instance_name>.sv	Example RTL top-level file that instantiates the IP core.
rtl/<instance_name>_bb.v	Example synthesizable RTL black box file that instantiates the IP core.
testbench/dut_inst.v	Template instance files.
testbench/dut_params.v	List of parameters based on user IP configurations.
testbench/tb_top.sv	Top level testbench file.
testbench/<ddr_density>_ddr3_parameters.vh	Holds the timing parameters for the different densities.
testbench/ddr3.v	DDR3 SDRAM simulation model.
testbench/ddr3_dimm_<ddr_width>.vh	DIMM simulation model.
testbench/ddr3_parameters.vh	Timing parameters for the configured ddr density.
testbench/ddr3_sdram_mem_params.vh	Macros definition used in the simulation model.
testbench/dqs_grp_delay.v	Delay models.
testbench/monitor.v	Module for monitoring the DDR3 Ios.
testbench/odt_watchdog.v.	Module for checking the odt timing.
testbench/tb_config_params.vh	Macros definition used in the testbench.

7.2. Design Implementation

This section describes the steps required to properly run a DDR3 SDRAM Controller for Nexus IP design on hardware.

7.2.1. Pin Placement

Typically, all external memory interfaces require the following FPGA resources:

- Data, data mask, and data strobe signals
- Command, address, and control signals
- PLL and clock network signals
- RZQ and VREF signals
- Other FPGA resources

In Lattice Certus-NX FPGA devices, external memory interfaces are supported in the High Performance I/O (HPIO) banks located at the bottom of the device (banks 3, 4, 5). These banks are labeled as HIGHSPEED in the device pinout tables. Each of these banks consists of 3-4 HPIO DQS groups, but depending on the device package, the number of HPIO DQS groups within each bank could be fewer. These HPIO DQS groups are labeled as DQx and DQSx/DQSNx in the device pinout tables, where 'x' represents the assigned HPIO DQS group number. Dedicated clock routing within HPIO banks is represented as GPLL or PCLK, and dedicated reference voltage pins are represented as VREF, in the device pinout tables. For more information, refer to the Certus-NX High-Speed I/O Interface User Guide (FPGA-TN-02216) and Pinout files located on the Certus-NX web page.

Observe the following guidelines when placing pins for external memory interfaces:

- Ensure that the pins for external memory interfaces reside within HPIO banks at the bottom of the device.
- An external memory interface can occupy one or more banks. When an interface occupies multiple banks, it is recommended to use adjacent banks to minimize timing and routing issues. Banks 5 and 3 are adjacent to bank 4, but bank 5 is not adjacent to bank 3.
- The DQS signals are fixed to specific locations (DQS/DQSN) within each HPIO bank. The DQS_P signal must be placed at these locations within the HPIO DQS group. The DQS_N signal will be auto placed and should not be manually assigned.

- All associated Data (DQ) and Data Mask (DM) signals belonging to a Data Strobe (DQS) must be placed in the same HPIO DQS group. Typically, a DDR3 DQS group consists of 8 DQ signals, 1 DM signal, and 1 DQS/DQSN pair. This means a HPIO DQS group needs 11 pins to support a DDR3 DQS group.
- The input reference clock to the PLL must be assigned to use dedicated clock routing (GPLL or PCLK). It is strongly recommended to place the input reference clock on the pin closest to the dedicated PLLs located in the lower left or right corner of the device (labeled as PB2A or PB156A under Pin/Ball in device pinout tables). This will ensure better performance by minimizing jitter and routing issues.

Always test proposed pinouts in the Lattice Radiant software with the correct I/O standards before finalizing.

7.2.2. Constraints

To ensure proper design coverage and hardware functionality, you must include the following necessary constraints in the DDR3 SDRAM Controller for Nexus IP project.

Table 7.3. Project Constraints

File Name	Description	Action Required
Memory Controller IP SDC file: Constraints/constraint.sdc	Pre-map IP timing constraints	No. These constraints automatically propagate and are only applicable to Radiant 2024.1 and above.
Memory Controller IP LDC file: constraints/<instance_name>.ldc	Sets the I/O type for each of the ports necessary to interface with DDR3 SDRAM.	No. These constraints automatically propagate.
Clock Constraint SDC file: eval/clock_constraint.sdc	Contains an example constraint for the input PLL reference clock	Yes – include a create_clock constraint based on the frequency of the input PLL reference clock. Place this in a user-created SDC file.
Memory Controller IP PDC file: eval/constraint.pdc	Contains generated constraints based on IP configuration.	Yes – copy the clock uncertainty, max skew and false path on I/O constraints listed in this file directly into the top-level PDC file. For Radiant versions before 2024.1, copy all the constraints listed in this file. Copy the example design constraints to run the DDR3 Memory Controller for Nexus Devices Example Design.

7.2.2.1. Memory Controller IP

The provided constraint.pdc file contains two sets of constraints:

- **IP constraints** – specific to the DDR3 SDRAM Controller IP Core. These cover only the clock uncertainty constraints, max skew, and false path pertaining to the I/O ports. All other IP-specific constraints are handled in the constraint.sdc. This applies to Radiant version 2024.1 and above. For Radiant version before 2024.1, the constraints cover all related IP specific constraints.
- **Eval constraints** – specific to the DDR3 SDRAM Controller Example Design.

To implement the generated DDR3 SDRAM Controller Example Design, copy the IP and Eval constraints into your top-level user PDC file. If you implement your own Memory Controller design, copy only the IP constraints into the top-level user PDC file. It is recommended to copy these constraints into your own PDC file because the provided constraint.pdc file overwrites every time the DDR3 SDRAM Controller IP Core regenerates.

The IP constraints include create_generated_clock, set_false_path, and set_max_delay constraints. For Radiant version 2024.1 and above, only the create_generated_clock, set_max_skew and set_false_paths for I/O ports are included. The eval constraints include set_false_path, set_max_delay, and ldc_create_group constraints. Copy the eval constraints only when running the provided DDR3 SDRAM Controller Example Design. Find the eval constraints below the following comment in the provided constraint.pdc file:

```
##############################################################################################
##
# Below are the constraints for eval design, you don’t need these if you are not using
the eval
```

The provided clock_constraint.sdc file contains a single create_clock constraint for the PLL reference clock (pll_refclk_i). This constraint is needed by the PLL to generate the constraint for its output clock. Copy this constraint into the Radiant Project SDC (not in PDC file) so that the synthesis tool can optimize the logic for the target clock. Refer to the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details regarding the implementation of constraints.

7.3. Example Design Hardware Evaluation

After you successfully configure and generate the DDR3 SDRAM Controller IP Core, you can use the included synthesis example design for hardware evaluation of the DDR3 SDRAM Memory Controller. For a detailed description of the DDR3 Memory Controller Example Design, refer to the Synthesis Example Design section of this user guide. The traffic generator file set is located under the eval/traffic_gen directory and is described in Table 7.4.

Table 7.4. Contents of eval/traffic_gen

File List	Description
ahbl0.v	AHBL_1x2. It routes the CPU data access to SYS_MEM or AHBL2APB.
ahbl2apb0.v	AHBL to APB bridge.
apb0.v	APB_1x4. It routes the CPU data access via AHBL2APB going to each module’s CSR.
cpu0.v	RISC-V CPU.
gpio0.v	GPIO module.
ahbl_tragen.v	RTL files for AHB-Lite traffic generator (IP Core v1.x.x).
lscc_ahbl_traffic_gen.sv,	
lscc_ahb_master.sv	
lscc_traffic_gen_csr.sv	
lscc_lfsr.v	
memc_traffic_gen.v	
ctrl_fifo.v	RTL files for AXI4 traffic generator (IP Core v2.x.x).
lscc_axi4_traffic_gen.sv	
lscc_axi4_m_csr.sv	
lscc_axi4_m_rd.sv	
lscc_axi4_m_wr.sv	
lscc_axi4_perf_calc.sv	
mc_axi4_traffic_gen.v	
lscc_osc.v	RTL files for OSC module.
osc0.v	
lscc_ram_dp_true.v	Copy of Lattice Radiant RAM_DP_True Foundational IP (needed by SYS_MEM).
memc_apb.v	RTL file for APB configuration interface.
sysmem0.v	The SYS_MEM for hardware validation, enabled when eval_top.SIM=0 (Implementation).
sysmem0_sim.v	The SYS_MEM for RTL simulation, enabled when eval_top.SIM=1 (Simulation).
uart0.v	The UART module.

7.3.1. Preparing the Bitstream

After configure and generate the DDR3 SDRAM Controller IP Core, all associated example design files are created under the eval directory. Refer to Table 7.2 for more details. The following steps illustrate how to prepare the DDR3 SDRAM Controller Example Design project and generate the associated bitstream.

1. After you generate the DDR3 SDRAM Controller IP Core, the Radiant project should contain the <instance_name>.ipx under the project's input files. If not, right-click on **Input Files** and select **Add > Existing File** under the **File List** tab in the lower-left corner of the Radiant window. This action opens an **Add Existing File** dialog box. Navigate to the <instance_directory>/<instance_name> directory and select the <instance_name>.ipx file. Ensure the **Copy file to directory** option is unchecked. Click **Add**.
2. To add the top-level example design file to the project, right-click on **Input Files** and select **Add > Existing File**. This action opens an **Add Existing File** dialog box. Navigate to the eval directory and select the eval_top.sv file. Ensure the **Copy file to directory** option is unchecked. Click **Add**.

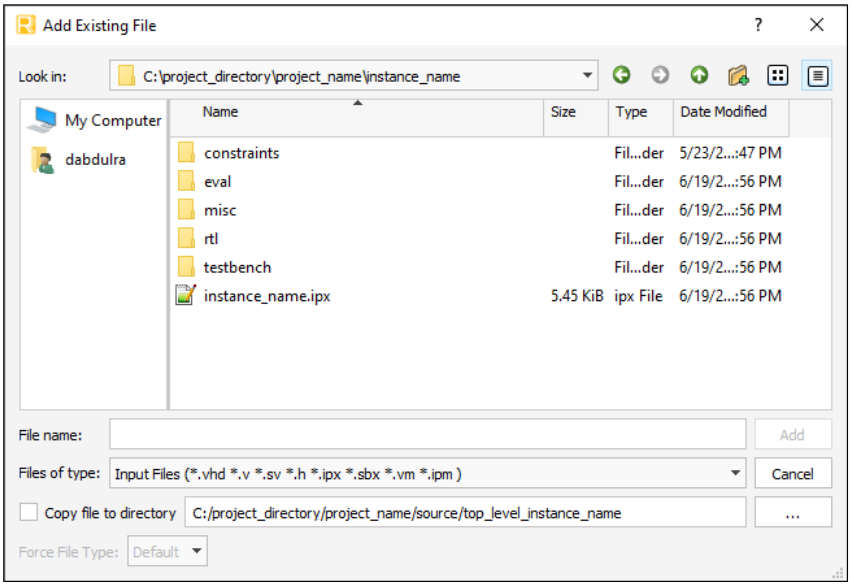

Figure 7.7. Add Existing File Dialog Box

3. To add the pre-synthesis constraint file to the project, right-click on **Pre-Synthesis Constraint Files** and select **Add > Existing File**. In the **Add Existing File** dialog box, check the **Copy file to directory** option. Navigate to the eval directory and select the clock_constraint.sdc file. Click **Add.**
4. To add the post-synthesis constraint file to the project, right-click on **Post-Synthesis Constraint Files** and select **Add > Existing File**. In the **Add Existing File** dialog box, check the **Copy file to directory** option. Navigate to the eval directory and select the constraint.pdc file. Click **Add**.
5. Modify the constraint.pdc to add the pin assignments for the Certus-NX board. You can accomplish this by either modifying constraint.pdc directly or synthesizing the Radiant project by clicking the Synthesize Design button, then adding pinouts via *Device Constraint Editor* **(Tools > Device Constraint Editor)** or through IP configuration GUI under the *Example Design* tab. Refer to the Example Design section for more details.

   Note that when assigning UART pins, UART TX connects to UART RX on the FPGA side, and the UART RX connects to UART TX on the FPGA side.
6. Before you run the example design on hardware, generate a bitstream by clicking on the button. This action generates a <project_name>_<top_level_instance_name>.bit file under the <project_directory>/<project_name>/<top_level_instance_name> directory specified in Step 2.

You may sometimes encounter timing failures during Place & Route due to unconstrained paths specific to your design. For details on implementing constraints, refer to the Constraints section of this user guide.

7.3.2. Running on Hardware

The DDR3 SDRAM Controller IP core is hardware tested on the Certus-NX Versa Evaluation board. To perform hardware evaluation of the DDR3 SDRAM Controller Example Design, you need the following:

- Certus-NX FPGA board with UART connection.
- Associated power supply and programming cable.
- Personal computer running Lattice Radiant software 2023.1 or later.
- Lattice Propel™ software 1.0 or later, or any terminal that supports serial communication.

To run the example design on hardware, you need a bitstream file. To generate the .bit file, refer to the Preparing the Bitstream section of this user guide. The following steps illustrate how to program the FPGA board with the example design.

1. Connect the FPGA board to the computer and power on the board.
2. Run the Lattice Radiant Programmer by clicking on the button. This action launches the Lattice Radiant Programmer, which scans for devices and configures the programmer automatically.

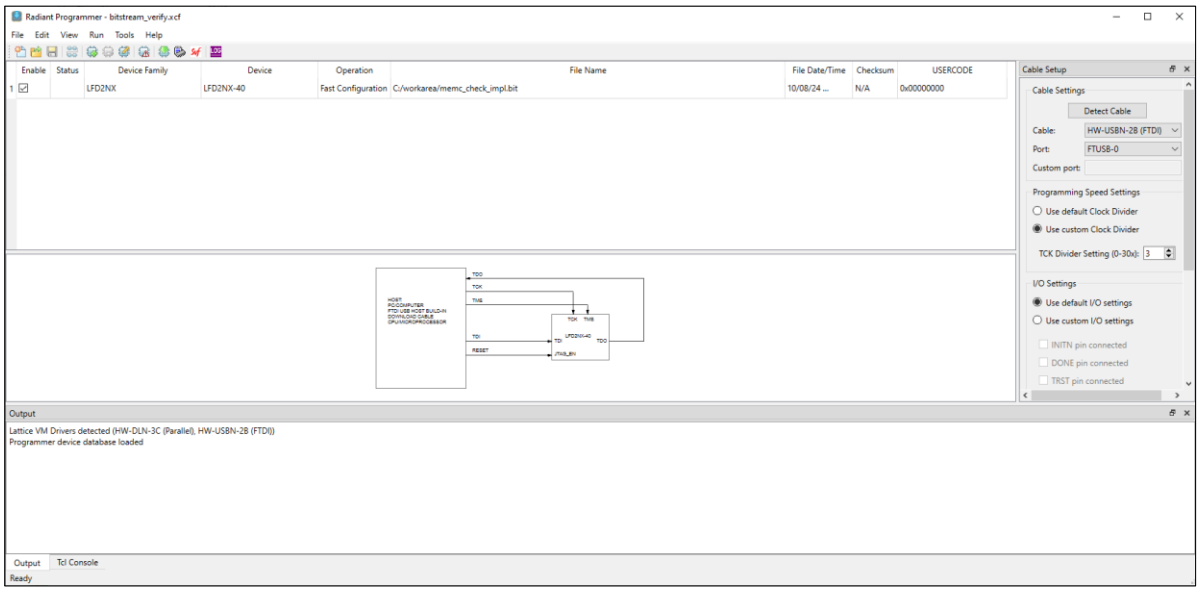

Figure 7.8. Radiant Programmer

3. Click under the **File Name** field, then click on **…** to the right of the field to launch the **Open File** dialog box. Navigate to the bitstream file generated in Step 6 and click **Open**.
4. Program the Lattice FPGA device by clicking on the button. Upon successful programming, the **Output** pane at the bottom of the Programmer window will display the following message:
   - After programming the Lattice FPGA with the example design bitstream, a serial terminal needs to be launched. For users wishing to use their own serial communication terminal, skip to Step 7. For users wishing to use the Lattice Propel terminal, continue with Step 5.
5. Launch the Lattice Propel software and select **Launch**. This action opens the default workspace.
6. To open the terminal, click on the button.
7. Configure the terminal settings to be a **Serial Terminal** with the appropriate **Baud rate, Data size, Parity, Stop bits, and Encoding**. Click **OK** to launch the **Terminal** pane at the bottom of the Propel window. Note that the **Serial port** varies depending on the computer setup.

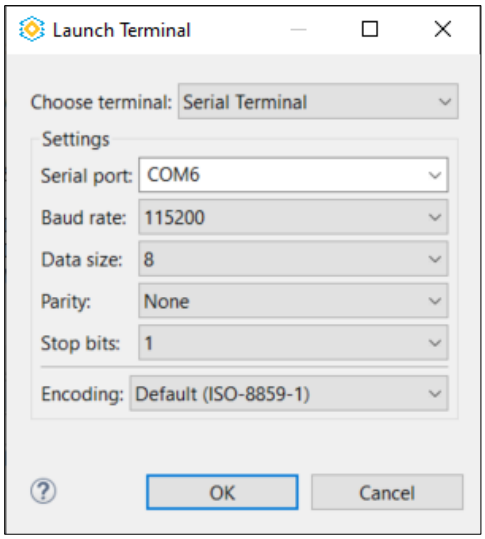

Figure 7.9. Serial Terminal Settings

To run the example design, assert rstn_i. This action prompts the following message to appear in the terminal. If no message is received, close the current serial terminal and open a new one with a different **Serial port**.

```
================================================================
DDR3 Memory Controller for Nexus Devices Example Design
================================================================
Select Test to Run press corresponding key)
0 = Print Memory Training Stages and Data Access Pattern Definitions
1 = Run Memory Training and Data Access Checks
2 = Run Memory Training and Data Access In a loop With Reset
```

Pressing 0 prints a series of definitions correlating to the memory training and data access test pattern:

```
-------------------------------------------
Memory Training Stage Definitions
-------------------------------------------
There are two training sequences to be executed, Write Leveling followed by Read
Training.

-------------------------------------------
Memory Data Access Pattern Definitions
-------------------------------------------
A sequence of numbers will print out to indicate which data access patterns were
successful.
0 = INCR2: 2-beat incrementing single write followed by single read
1 = INCR2: 2-beat incrementing burst write followed by burst read with no delay between
transactions
2 = INCR4: 4-beat incrementing burst write followed by burst read with no delay between
transactions
3 = INCR8: 8-beat incrementing burst write followed by burst read with no delay between
transactions
4 = INCR8: 8-beat incrementing burst write followed by a delay before issuing burst read
5 = INCR32: 32-beat incrementing burst write followed by burst read with no delay
between transactions
p = INCR32 Performance Test: 32-beat incrementing parallel burst write and burst read
with no delay between transactions
```

Pressing 1 starts the training sequence, followed by 7 different data access checks:

```
Begin Initialization. Training will proceed after initialization is done, if enabled.
-------------------------------------------
Initialization and Core Training Stage
-------------------------------------------

C
Actual MC CK/Command Delay Value: 31

-------------------------------------------
Starting Data Access Check
-------------------------------------------
0
 1
  2
   3
    4
     5
      p

-------------------------------------------
Starting Performance Test
-------------------------------------------
The performance test measures the throughput of the MC and calculates efficiency via the
calculation:
        Total number of wr_rd transactions  /  Duration of the MC clock (sclk)

BUS EFFICIENCY in %: 84
Performance in Mbps : 14423
Ran : 20000 transactions across sequential addresses

RESULT : All transaction types have PASSED.
```

Pressing 2 starts memory training and data access in a loop with Reset. User is able to input the number of Reset.

```
Enter Number of Iterations: 50
iteration: 0
Begin Initialization. Training will proceed after initialization is done, if enabled.

Write Leveling and Read Training has completed successfully!

-------------------------------------------
Starting Data Access Check
-------------------------------------------
0
 1
  2
   3
    4
     5
      p
```

```
-------------------------------------------
Starting Performance Test
-------------------------------------------
The performance test measures the throughput of the MC and calculates efficiency via the
calculation:
        Total number of wr_rd transactions  /  Duration of the MC clock (sclk)
BUS EFFICIENCY in %: 44
Performance in Mbps : 5709
Ran : 20000 transactions across sequential addresses

RESULT : All transaction types have PASSED.
polling for clocking_good_o to be 1
Done polling for clocking_good_o to be 1
iteration: 1
Begin Initialization. Training will proceed after initialization is done, if enabled.

Write Leveling and Read Training has completed successfully!

-------------------------------------------
Starting Data Access Check
-------------------------------------------
0
 1
  2
   3
    4
     5
      p

-------------------------------------------
Starting Performance Test
-------------------------------------------
The performance test measures the throughput of the MC and calculates efficiency via the
calculation:
        Total number of wr_rd transactions  /  Duration of the MC clock (sclk)
BUS EFFICIENCY in %: 44
Performance in Mbps : 5709
Ran : 20000 transactions across sequential addresses

RESULT : All transaction types have PASSED.
```

RESULT : All transaction types have PASSED. The Bus_efficiency value represents the efficiency percentage of the DDR3 bus utilization, while the Perf_MBps value represents the bandwidth of the DDR3 interface. Use the following formulas to calculate *efficiency* and *bandwidth*:

Efficiency formula

$$\text{Efficiency} = \frac{\text{Total bytes transferred}}{\text{Number of DDR clock cycles} \times \left[\frac{\text{DDR\_WIDTH}}{8}\right] \times \text{Gearing ratio}}$$

Bandwidth formula

Method 1: Based on transfer and clock period.

$$\text{Bandwidth} = \frac{\text{Total bytes transferred}}{\text{Number of DDR clock cycles} \times \text{DDR clock period}}$$

Method 2: Based on DDR3 parameters.

$$\underset{\text{DDR3 data rate in Mbps}}{\text{Bandwidth}} = \text{DDR3 data width} \times \text{DDR3 data rate} \times \left(\frac{\text{DDR3 bus efficiency}}{100} \right)$$

If you encounter a failure during training, a message is sent over the serial connection notifying you of the particular stage that has failed. This action also aborts the loop-back data access checks.

```
Write Levelling FAILED.
Read Training FAILED.
                    HARDWARE VALIDATION FAIL.
ABORTING Data Access Check…
```

7.4. Example Design Simulation

After you successfully configure and generate the DDR3 SDRAM Controller IP Core, you can use the included example design to simulate the DDR3 SDRAM Controller. All associated simulation files are located under the testbench directory. Refer to Table 7.2 for more details. The following steps illustrate how to prepare the Memory Controller Example Design project for simulation.

1. Before you simulate the example design, complete steps 1 to 2 under the Preparing the Bitstream section of this user guide. To add the top-level testbench file to the project, select **File > Add > Existing Simulation File**. This action opens an **Add Existing Simulation File** dialog box. Navigate to the testbench directory and select the tb_top.sv file. In the **Add Existing File** dialog box, ensure the **Copy file to directory** option is unchecked. Click **Add**.
2. Ensure that the *Simulation Mode Enable* option is checked when generating the IP. This option shortens the initialization sequence of the DDR3 interface and disables the UART interface for simulation. Leave this option unchecked for hardware implementation.

Refer to the Simulation Example Design section of this user guide for more details.

To create the simulation environment for the DDR3 SDRAM Controller Example Design, take the following steps:

1. Launch the simulation wizard in Radiant by selecting **Tools > Simulation Wizard**. This action opens the **Simulation Wizard** dialog box. Click **Next** and provide a name (<sim_name>) and directory (<sim_directory>) for the simulation project, where the default directory is set to <project_directory>/<project_name>. Click **Next**.

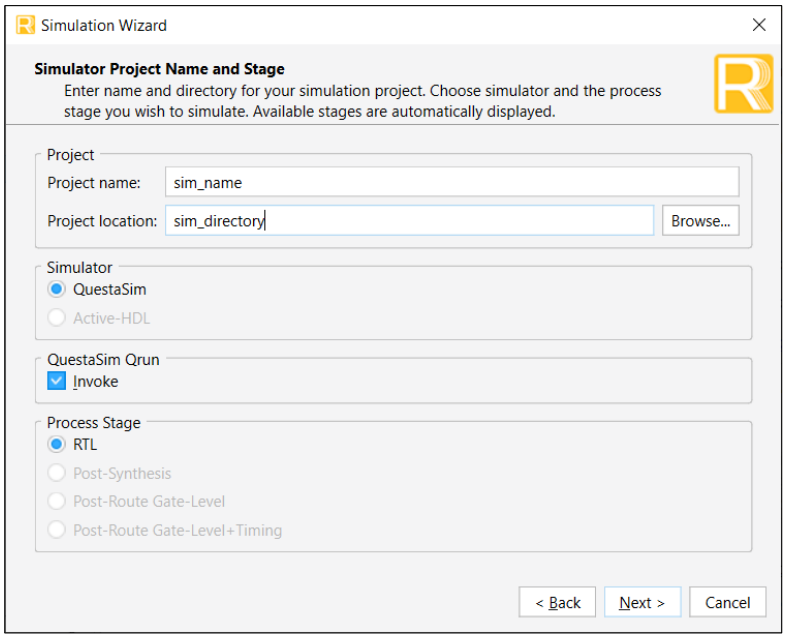

Figure 7.10. Simulation Wizard

2. Under the **Add and Reorder Source** window, notice that the **Source Files** only contains the DDR3 SDRAM Controller IP instance (instance_name.sv), top-level evaluation (eval_top.sv) and top-level testbench (tb_top.sv) files. Click **Next**.

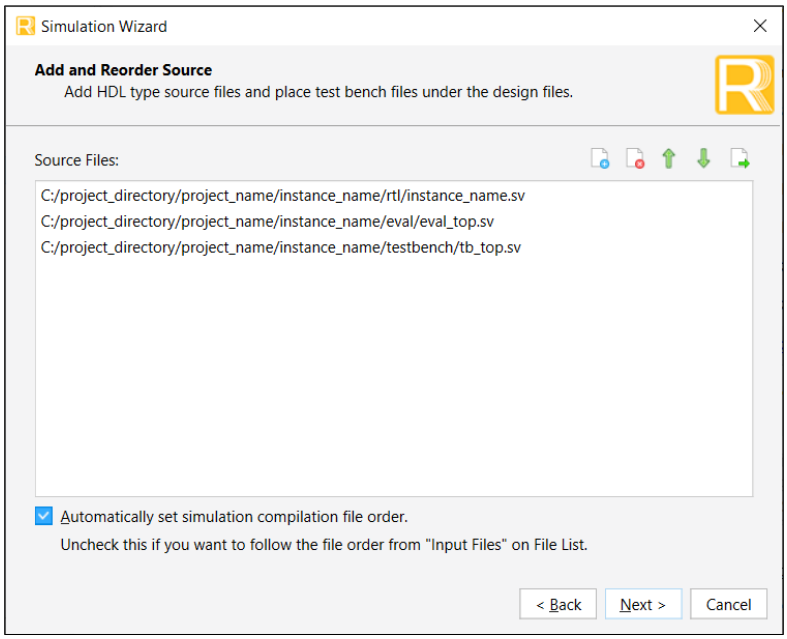

Figure 7.11. Adding and Reordering Simulation Source Files

3. Under the **Parse HDL files for simulation** window, notice that the **Simulation Top Module** is set to tb_top. Click **Next**.

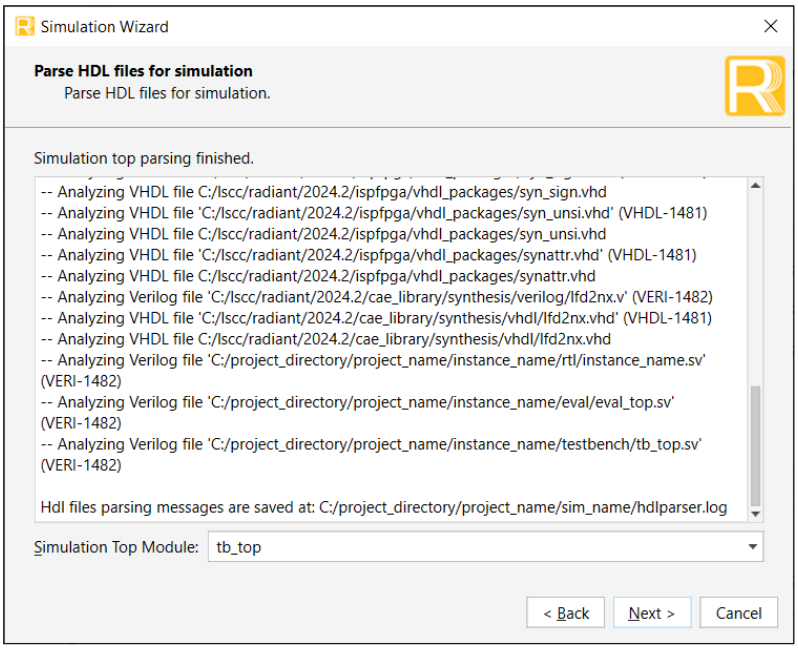

Figure 7.12. Parsing Simulation HDL Files

4. This action opens the **Summary** window. By default, the simulation runs for 100 µs, allowing you to configure the waveform to log signals of interest in the Questasim simulator before continuing. You can then enter the following TCL command in Questasim to run the simulation until completion: **run -all**. Alternatively, if you wish to run the simulation with the default top-level signals, change the 100 µs value to 0 µs to execute the simulation completely.

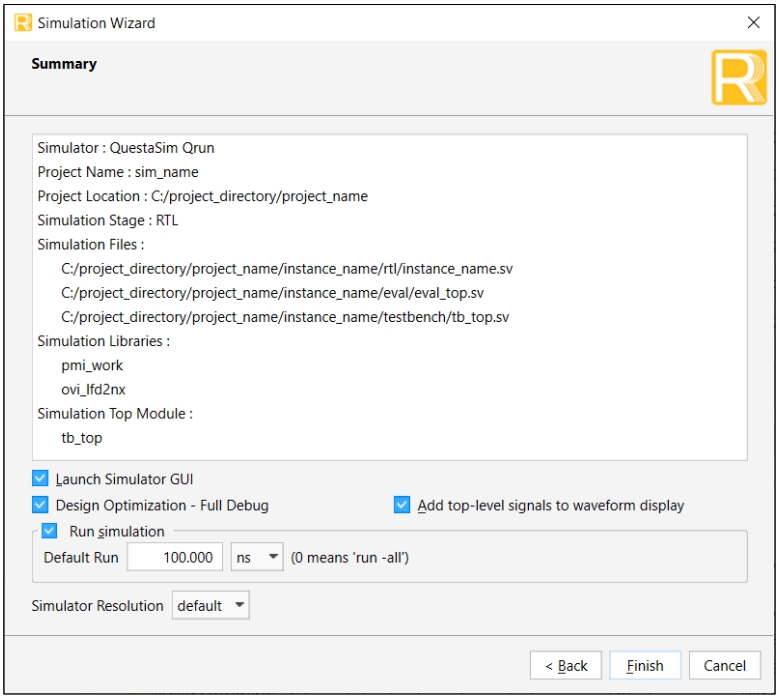

Figure 7.13. Simulation Summary

Figure 7.14 shows the results of the DDR3 SDRAM Controller IP simulation design.

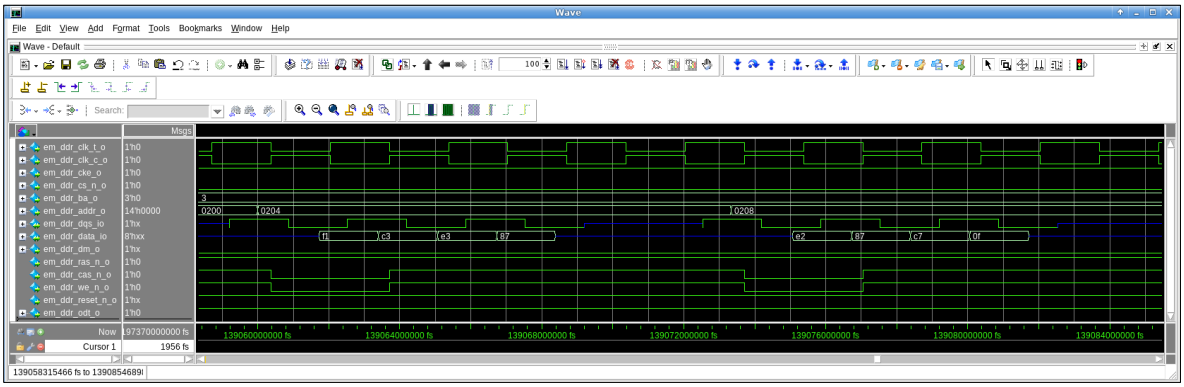

Figure 7.14. Simulation Result Waveform

8. Debugging

This section discusses tools and strategies available to assist you with debugging your DDR3 memory interface.

8.1. Debug with the Example Design

The provided DDR3 example design helps debug functional issues related to the training of external DDR3 memory or data accesses issued by the Memory Controller. For a description of the example design and instructions on running the test program for hardware evaluation, refer to the Synthesis Example Design and Running on Hardware sections of this user guide.

8.2. Debug with Reveal Analyzer

The DDR3 Memory Controller for Nexus Devices introduces reveal signals to assist you in debugging issues related to the training sequence of the DDR3 memory interface using Reveal Analyzer. This feature was introduced in IP Core v2.2.x and exists within the rtl/<instance_name>.sv file following successful IP generation. You can locate the reveal signals (with rvl_ prefix) under the following comment in the generated RTL file:

```
//Added for reveal
```

Refer to the Debugging with Reveal Usage Guidelines and Tips (FPGA-AN-02060), for information regarding the setup and usage of Reveal Analyzer.

Table 8.1. Reveal Analyzer Signal Definitions

Signal Name	Description
rvl_init_clk_ready	Clock is stable
rvl_init_start	Start Initialization Trigger
rvl_init_complete	Initialization Complete Flag
rvl_init_phy_init_act	Initialization or Training in progress
rvl_init_wl_start	Write Leveling started
rvl_init_wl_act	Write Levelling in progress
rvl_init_wl_err	Write Leveling error
rvl_init_wl_done	Write Leveling complete
rvl_init_rt_act	Read Training in progress
rvl_init_rt_err	Read Training error
rvl_init_rt_done	Read Training complete
rvl_cmd_cs_p0	Chip Select
rvl_cmd_cs_p1	Chip Select
rvl_cmd_ba_p0	Bank Address
rvl_cmd_ba_p1	Bank Address
rvl_cmd_addr_p0	Address
rvl_cmd_addr_p1	Address
rvl_cmd_casn_p0	Column Address Strobe
rvl_cmd_casn_p1	Column Address Strobe
rvl_cmd_rasn_p0	Row Address Strobe
rvl_cmd_rasn_p1	Row Address Strobe
rvl_cmd_wen_p0	Write Enable
rvl_cmd_wen_p1	Write Enable
rvl_wrlvl_move	At rising edge, it changes the delay code (±1) according to the direction set by wrlvl_dir for CK-DQS skew compensation during Write Leveling
rvl_wrlvl_dir	Controls the direction of delay code change for CK-DQS skew compensation during Write Leveling (0 = increase delay, 1 = decrease delay)

Signal Name	Description
rvl_wrlvl_loadn	Asynchronously resets the delay code to the default value for CK-DQS skew compensation during Write Leveling
rvl_wrlvl_cout	Margin test output flag to indicate the under-flow or over-flow for CK-DQS skew compensation during Write Leveling
rvl_wrlvl_dq_fback	Data output for Write Leveling
rvl_wrlvl_bitfback	Data Output LSB for Write Leveling
rvl_wrlvl_act_cntl	Write Leveling DQS Control
rvl_wrlvl_dqspuls_p0	Current DQS number in a rank
rvl_wrlvl_dqspuls_p1	Current DQS number in a rank
rvl_dqsbuf_pause	Set to 1 to stop the DQSBUF-generated internal clocks when updating rdclksel and the delay codes. This is to avoid metastability.
rvl_rt_burstdet	Asserts when DQSI has been detected
rvl_rt_rdclksel	Used to select read clock source and polarity control: [1:0]: sets phase shift from DQS write delay cell to 0, 45, 90, or 135 degrees (2'b00 to 2'b11) [2] = 0: use inverted clock [2] = 1: use non-inverted clock (adds 180-degree phase shift) [3] = 0: bypasses the register in the read enable path [3] = 1: selects the register in the read enable path
rvl_rt_datavalid	Indicates that the read data is valid during Read Training
rvl_rt_dqs_read	Read data during Read Training. Valid when rt_datavalid asserts
rvl_rt_data_match	Asserts when the read data matches the expected data pattern during Read Training
rvl_dfi_address	DFI Interface – Address
rvl_dfi_bank	DFI Interface – Bank Address
rvl_dfi_cke	DFI Interface – Clock Enable
rvl_dfi_ras_n	DFI Interface – Row Address Strobe
rvl_dfi_we_n	DFI Interface – Write Enable
rvl_dfi_cas_n	DFI Interface – Column Address Strobe
rvl_dfi_cs_n	DFI Interface – Chip Select
rvl_dfi_wrdata_en	DFI Interface – Write Data Enable
rvl_dfi_wrdata	DFI Interface – Write Data
rvl_dfi_wrdata_mask	DFI Interface – Write Data Mask
rvl_dfi_rddata_valid	DFI Interface – Read Data Valid
rvl_dfi_rddata	DFI Interface – Read Data
rvl_pause	Pause signal that asserts before any delay adjustment is applied
rvl_code	The internal DLL delay code that corresponds to a 90^0 phase shift
rvl_read_clk_sel_ext_trn	The read clock source selection from the extended training engine
rvl_read_clk_sel_dqs_gate	The read clock source selection from the Read Training (DQS Gate)
rvl_read_clk_sel_sel	The read clock source selection MUX selector to switch between the extended training copy or the Read Training (DQS Gate) copy of read clock source selection
rvl_wr_loadn_dqs	Asynchronously resets the delay code to the default value for DQ-DQS skew compensation during Write Training
rvl_wr_move_dqs	At rising edge, it changes the delay code (±1) according to the direction set by wr_dir for DQ-DQS skew compensation during Write Training
rvl_wr_dir_dqs	Controls the direction of delay code change for DQ-DQS skew compensation during Write Training (0 = increase delay, 1 = decrease delay)
rvl_wr_cout_dqs	Margin test output flag to indicate the under-flow or over-flow for DQ-DQS skew compensation during Write Training
rvl_rd_loadn_dqs	Asynchronously resets the delay code to the default value for DQ-DQS sampling skew compensation during Read Data Eye Training

Signal Name	Description
rvl_rd_move_dqs	At rising edge, it changes the delay code (±1) according to the direction set by rd_dir for DQ-DQS sampling skew compensation during Read Data Eye Training
rvl_rd_dir_dqs	Controls the direction of delay code change for DQ-DQS sampling skew compensation during Read Data Eye Training (0 = increase delay, 1 = decrease delay)
rvl_rd_cout_dqs	Margin test output flag to indicate the under-flow or over-flow for DQ-DQS sampling skew compensation during Read Data Eye Training
rvl_dq_dqs_in_adj_load_n	Asynchronously resets the delay code to the default value for DQ-DQS skew compensation during Read Data Eye Training
rvl_dq_dqs_in_adj_move	At rising edge, it changes the delay code (±1) according to the direction set by dq_dqs_in_adj_dir for DQ-DQS skew compensation during Read Data Eye Training
rvl_dq_dqs_in_adj_dir	Controls the direction of delay code change for DQ-DQS skew compensation during Read Data Eye Training (0 = increase delay, 1 = decrease delay)
rvl_dq_dqs_n_adj_cout	Margin test output flag to indicate the under-flow or over-flow for DQ-DQS skew compensation during Read Data Eye Training

There are Reveal debug signals for the AXI4 interface and APB Scratch Registers as well. This section focuses on debugging the training sequences.

For general-purpose debugging, use the rvl_init_start and rvl_init_complete signals to determine the start and completion of the training sequence. The rvl_init_wl_err and rvl_init_rt_err are signals indicate whether the training sequences passed or failed. For better granularity, use rvl_init_wl_done and rvl_init_rt_done signals to indicate the stages of completion of the training sequence. Write Leveling completes before Read Training starts. The dbg_wl_err signal indicates a write leveling failure (unable to detect a rising edge of the clock), while dbg_rt_err signal indicates a read training failure (read data does not match the expected pattern). Examples below demonstrate good training (Figure 8.1) and bad training (Figure 8.2) scenarios.

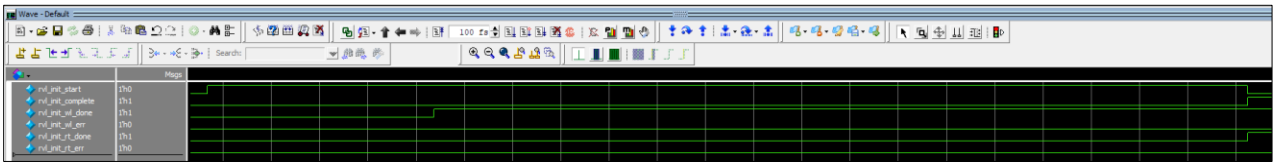

Figure 8.1. DDR3 Training Passes

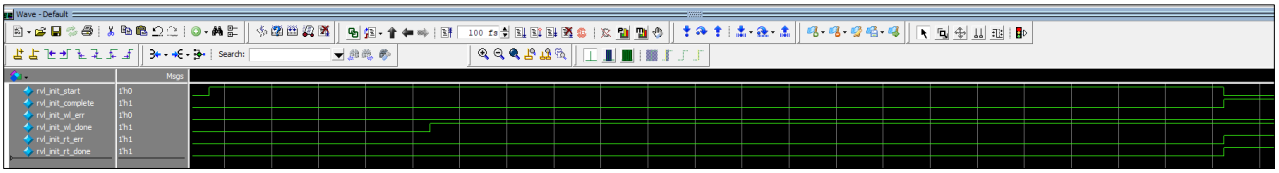

Figure 8.2. DDR3 Read Training Failure

8.2.1. Write Leveling

Write leveling starts when the DDR3 memory device samples the DDR clock with the rising edge of DQS. We intentionally delay the DDR clock edge so that the initial rising edge of DQS samples the DDR clock when it is deasserted. The Memory Controller then adjusts DQS until it samples a '1'. This occurs when the DQS rising edge aligns with the DDR Clock rising edge, and it samples the DDR clock when it Is asserted. We perform write leveling on each DQS sequentially. Refer to Figure 8.3 for the sampled DDR clock on the rising edge of DQS. These signals are returned as wrlvl_bitfback. The wrlvl_bitfback[0] signal is the write leveling data output for DQS0. Refer to the Write Leveling section for more details.

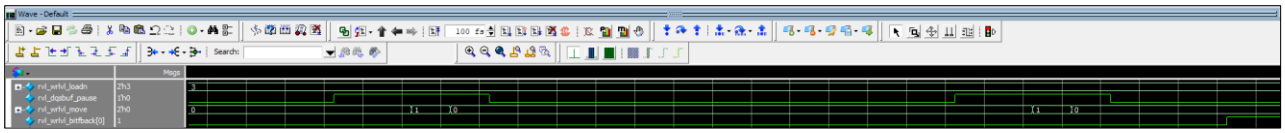

Figure 8.3. DDR3 Write Leveling output for DQS0

8.2.2. Read Training (DQS Gate)

Read training is good when there are no failures on the burst detect for 9 consecutive reads. The rt_data_match signal must assert when burst detect asserts to be considered a good burst detect. Refer to the Read Training (DQS Gate) section for more details. Figure 8.4 illustrates the 9 consecutive burst detects.

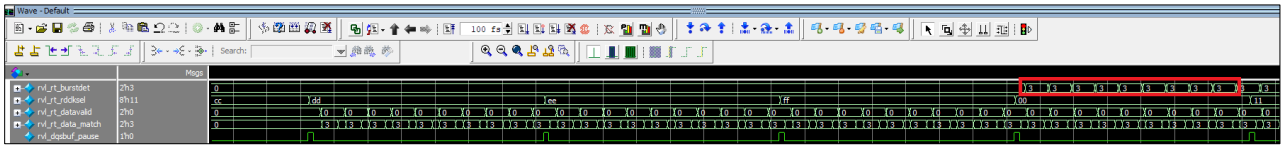

Figure 8.4. DDR3 Read Training (DQS Gate)

8.2.3. Write Training

Write training is the first step of the extended training stage. This stage is intended to search for the valid DQ-DQS phase relationship and determine the optimal point when performing a write operation. Refer to the Write Training section for more details. Figure 8.5 illustrates a typical write training behavior. The *rvl_wr_cout_dqs* asserts, indicating that the sweep has reached the maximum or minimum boundary. Two criteria are used to determine if the boundary has been reached. Either cout asserts, indicating overflow, or there is a data mismatch between the read-back data and the data written.

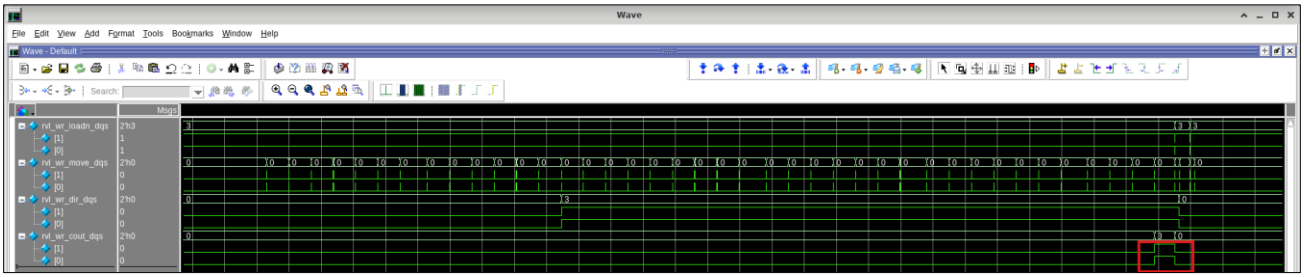

Figure 8.5. DDR3 Write Training

Figure 8.6 shows the coarse adjustment that takes place before the actual tuning of the DQ-DQS phase. During the coarse adjustment stage, both the read DQ-DQS and write DQ-DQS are adjusted to achieve reliable read and write operations, allowing the fine-tuning stage to proceed successfully.

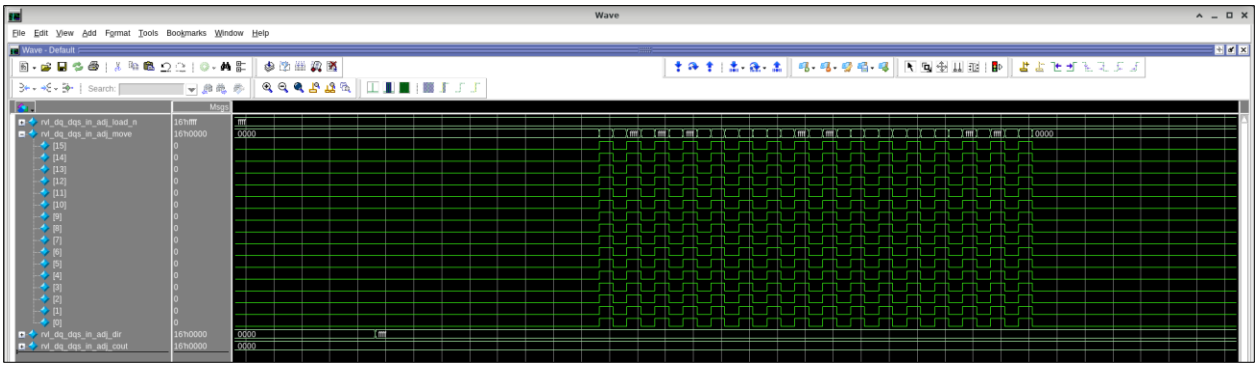

Figure 8.6. DDR3 DQ-DQS Course Adjustment

8.2.4. Read Data Eye Training

Read data eye training is the second step of the extended training stage. This stage is intended to search for the valid DQ-DQS phase relationship and determine the optimal point when performing a read operation. Refer to the Read Data Eye Training section for more details. Figure 8.7 illustrates a typical read data-eye training behavior. In this diagram, *rvl_dq_dqs_in_adj_cout* does not assert, indicating that the maximum boundary is determined based on the data mismatch between read and write.

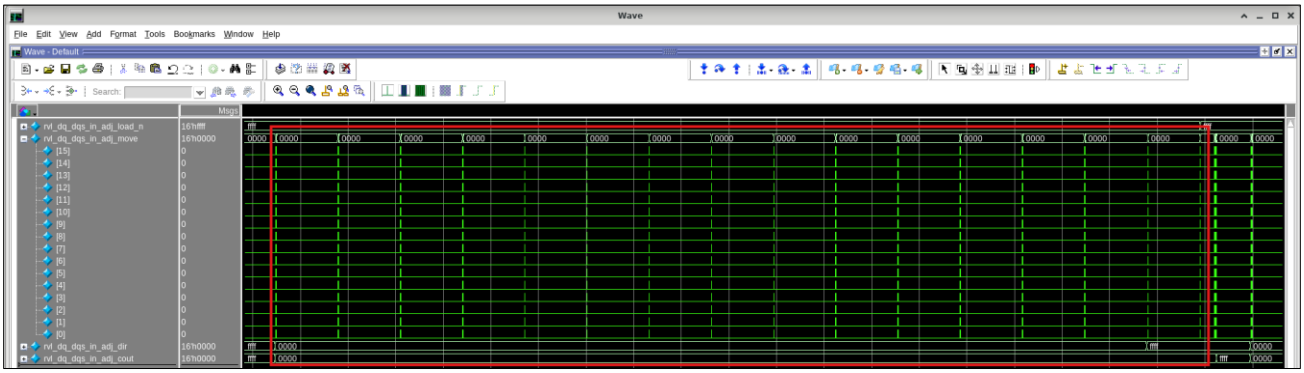

Figure 8.7. DDR3 Read Data Eye Training

Appendix A. Resource Utilization

Table A.1 displays the configuration and resource utilization for IP Core v2.3.1 DDR3 SDRAM Controller IP implemented for LFD2NX-40-8BG256C using Synplify Pro of Lattice Radiant software 2025.2.1.

Table A.1 Resource Utilization for IP Core v2.3.1

Configuration	sclk_o[1] (MHz)	Registers	LUTs	EBR	IDDR/ODDR/TDDR
Interface Type = NATIVE, DDR Width = 8 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	1,622	2,448	0	52 (8+35+9)
Interface Type =AMBA, DDR Width = 8 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,757	3,726	3	52 (8+35+9)
Interface Type = NATIVE, DDR Width = 16 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	1,892	2,662	0	78 (16+44+18)
Interface Type = AMBA, DDR Width = 16 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	3,133	3,947	5	78 (16+44+18)
Interface Type = NATIVE, DDR Width = 24 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,159	2,775	0	106 (24+55+27)
Interface Type = NATIVE, DDR Width = 32 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,436	2,955	0	132 (32+64+36)
Interface Type = AMBA, DDR Width = 32 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,893	4,264	9	132 (32+64+36)

Note:

1. The IP Core generates sclk_o based on the DDR Clock divided by the gearing ratio. Sclk_o = DDR Clock/Gearing Ratio. The DDR Clock is 333 MHz, and the gearing ratio is 2:1, so sclk_o = 333/2 = 166.5 MHz.

Table A.2 displays the configuration and resource utilization for IP Core v2.3.1 implemented for LFD2NX-40-8BG256C using LSE of Lattice Radiant software 2025.2.1.

Table A.2. Resource Utilization for IP Core v2.3.1

Configuration	sclk_o[1] (MHz)	Registers	LUTs	EBR	IDDR/ODDR/TDDR
Interface Type = NATIVE, DDR Width = 8 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	1,676	2,618	0	52 (8+35+9)
Interface Type =AMBA, DDR Width = 8 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,777	3,798	3	52 (8+35+9)
Interface Type = NATIVE, DDR Width = 16 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	1,950	2,806	0	78 (16+44+18)
Interface Type = AMBA, DDR Width = 16 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	3,158	4,045	5	78 (16+44+18)
Interface Type = NATIVE, DDR Width = 24 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,222	2,964	0	106 (24+55+27)
Interface Type = NATIVE, DDR Width = 32 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	2,499	3,162	0	132 (32+64+36)
Interface Type = AMBA, DDR Width = 32 bits, DDR Clock = 333 MHz, Gearing Ratio = 4:1, Others = Default	166.5	3,923	4,419	9	132 (32+64+36)

Note:

1. The IP Core generates sclk_o based on the DDR Clock divided by the gearing ratio. Sclk_o = DDR Clock/Gearing Ratio. The DDR Clock is 333 MHz, and the gearing ratio is 2:1, so sclk_o = 333/2 = 166.5 MHz.

Appendix B. Known Issue

The DDR3 SDRAM Controller IP Core may fail simulation due to timing differences between the simulated library module and actual silicon delays. The provided testbench (tb_top.sv) includes a workaround to compensate for latency variations in simulations for the Lattice Radiant version 2025.1 and earlier. This issue is resolved in Lattice Radiant version 2025.1.1. No workaround is needed when running tb_top.sv with Lattice Radiant version 2025.1.1 or later. The workaround is dynamically applied based on the Lattice Radiant version used to generate the IP.

References

- DDR3 SDRAM Controller IP Release Notes (FPGA-RN-02032)
- Nexus DDR3 Memory Controller Driver API Reference (FPGA-TN-02401)
- Certus-NX High-Speed I/O Interface User Guide (FPGA-TN-02216)
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- MachXO5-NX web page
- AMBA AXI Protocol Specification
- AMBA AXI4 Protocol Specification
- AMBA APB Protocol Specification
- DDR3 JEDEC Standard
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Debugging with Reveal Usage Guidelines and Tips (FPGA-AN-02060)
- Lattice Radiant FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans
- www.jedec.org

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 2.3, IP v2.3.1, April 2026

Section	Change Summary
All	Updated the IP Core version to *2.3.1*.
Introduction	• Removed *hardware validation* and *note 1* in Table 1.1. Quick Facts. • Removed *TDQS* in the Features section. • Removed the *Hardware Support* section • Added statement that the *DDR3 SDRAM Controller IP Core example design supports simulation and deployment on development boards* before the Features section.
Functional Description	• Removed the DDR3 SDRAM PHY IP reference and ported over the *DDR3 PHY IP Core Block Diagram* and *DDR3 PHY IP components* in the Soft PHY and Soft Training Engine section. • Added a new PHY feature – *Periodic VT Compensation* in the Soft PHY and Soft Training Engine section. • Updated Write Training section.
Parameter Description	• Updated default values and description in Table 3.5. Additional Configuration Group Attributes. • Added Simulation Mode Enable in Table 3.1. Device Information Attributes and Table 3.6. General Definitions.
Designing and Simulating the IP	Added statement in the Running on Hardware section indicating that the *DDR3 SDRAM Controller IP core has been hardware-tested on the Certus-NX Versa Evaluation board*.

Revision 2.2, IP v2.3.0, December 2025

Section	Change Summary
All	Updated IP Core version to *2.3.0.*

Introduction	• Updated Table 1.1. Quick Facts.
	  • Added notes *1* and *2.*
	  • Updated the Design Tool Support Simulation to *QuestaSim.*
	  • Added *Driver Support* row.
	• Enabled Write Training support in Table 1.3. Features Overview.
	• Reworked the Licensing and Ordering Information section.
	• Added *bullet no.4* in the Limitations section.

Functional Description	• Updated Figure 2.1. Memory Controller IP Core Functional Diagram. • Removed *Write Leveling* and *External Auto Refresh Port* in the Clocking and Reset section. • Removed the *Reference Description* section reference after Table 2.3. AXI4 to Memory Address Mapping Example. • Updated the Write Leveling section. • Updated the section title from *Read Training* to *Read Training (DQS Gate)*. • Added the Extended Training section. • Added the Write Training section. • Added the Read Data Eye Training section. • Removed *Write Leveling* in the Initialization Control section, • Added *bullet no.4* note in Table 2.6. Defined User Commands. • Updated the REFRESH Support section.
IP Parameter Description	• Updated Table 3.4. Local Interface Attributes. • Added *AXI Maximum Burst Length* and *AXI Outstanding Command Support* attributes.

Section	Change Summary

	• Updated Table 3.5. Additional Configuration Group Attributes.
	  • Added Enable Extended Training attribute.
	  • Updated the *Write DQ/DM Delay Direction* and *Write DQ/DM Delay Value* attributes.
	  • Added the *Read DQ Phase Adjustment Direction* and *Read DQ Phase Adjustment Value* attributes.
	  • Added *note 1*.
	• Added the *AXI Maximum Burst Length*, *AXI Outstanding Command Support*, *Read DQ Phase Adjustment Direction* and *Read DQ Phase Adjustment Value* attributes in Table 3.6. General Definitions.
	• Updated the *Row Size* attribute in Table 3.7. Address Attributes.
	• Added *note 1* in Table 3.8. Auto Refresh Control Attributes.
	• Updated default value of the *CAS Latency* attribute in Table 3.9. Mode Register Initial Setting Attributes.
	• Added the Example Design section.

Signal Description	• Updated the description of *rst_n_i* in Table 4.1. Clock and Reset Port Definitions.
	• Updated Table 4.5. DDR3 Interface Port Definitions.
	  • Updated the description of *em_ddr_clk_t_o.*
	  • Added *em_ddr_clk_c_o*.

Register Description	Added the following under Table 5.1. APB Interface Register Map. • Config Reg – *Reserved* • Status Reg - *Write DQ/DM-DQS Phase Adjustment Error*, *Read DQ-DQS Phase Adjustment Error*, and *MC CK/Command Delay Value Actual.*
Designing and Simulating the IP	• Added *Example Design* in Table 7.1. Memory Controller Attribute Guidelines. • Removed ldc_create_group in the Memory Controller IP section. • Updated the steps in Preparing the Bitstream. • Updated the DDR3 Memory Controller for Nexus Devices Example Design GUI. • Updated the format for the *Efficiency* and *Bandwidth* formula before the Example Design Simulation section.
Debugging	• Added the following signals under Table 8.1. Reveal Analyzer Signal Definitions, *rvl_pause*, *rvl_code, rvl_read_clk_sel_ext_trn, rvl_read_clk_sel_dqs_gate, rvl_read_clk_sel_sel, rvl_wr_loadn_dqs, rvl_wr_move_dqs, rvl_wr_dir_dqs, rvl_wr_cout_dqs, rvl_rd_loadn_dqs, rvl_rd_move_dqs, rvl_rd_dir_dqs, rvl_rd_cout_dqs, rvl_dq_dqs_in_adj_load_n, rvl_dq_dqs_in_adj_move, rvl_dq_dqs_in_adj_dir*, and *rvl_dq_dqs_n_adj_cout*. • Added *APB Scratch Registers* under the Debug with Reveal Analyzer section. • Updated the section title from *Read Training* to *Read Training (DQS Gate)*. • Added the Write Training section. • Added the Read Data Eye Training section.
Appendix A	Updated the *Register* values*, LUT* values, and *Gearing ratio* in Table A.1 Resource Utilization for IP Core v2.3.1 and Table A.2. Resource Utilization for IP Core v2.3.1.
Appendix B	• Moved the Known Issues section from *Section 9* to *Appendix B*. • Removed Known issue no. 2, *According to the JEDEC specification, the refresh period (TREFI) is 7.8 µs. However, the internal auto- refresh timer implemented in the DDR3 SDRAM Controller IP Core is based off a value less than 7.8 µs. This ensures the number of postponed refreshes never exceeds JEDEC limits, but may reduce performance*.
Revision History	Added a 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* to this section.

Revision 2.1, IP v2.2.1, October 2025

Section	Change Summary
All	• Updated IP Core version to *2.2.1*. • Minor editorial fixes.
Introduction	Updated the Lattice Radiant Software version to *2025.1.1* in Table 1.1. Quick Facts.
Known Issues	• Reworked item no.1 known issue. • Added, item no. 2 known issue - *According to the JEDEC specification, the refresh period (TREFI) is 7.8 µs. However, the internal auto- refresh timer implemented in the DDR3 SDRAM Controller IP Core is based off a value less than 7.8 µs. This ensures the number of postponed refreshes never exceeds JEDEC limits, but may reduce performance*.

Revision 2.0, IP v2.2.0, June 2025

Section	Change Summary
All	• Updated the IP version from *2.1.0* to *v2.2.0*. • Minor editorial fixes.

Introduction	• Updated Table 1.1. Quick Facts.
	  • Changed *Supported FGA Family* to *Supported Devices*.
	  • Removed *Targeted Devices*.
	  • Removed *Resource Utilization*.
	• Updated the Features section.
	  • Added the *Power Down mode with no DRAM data retention*.
	• Updated Table 1.4. Minimum Device Requirements.
	  • Separated Speed grades for Certus-NX, CrossLink-NX and CertusPro-NX, MachXO5-NX.
	• Updated the Limitations section.
	  • Added, *The Lattice LSE synthesis engine causes difficulty in closing timing, especially for the eval_top's RISC-V and its interfacing modules. The DDR3 SDRAM Controller IP Core closes timing correctly*, in this section.
	  • Removed, *The AXI data bus width conversion is not supported. The AXI data bus width must be the same as the memory controller data bus width*, in this section.

Functional Description	• Updated Table 2.1. Supported AXI4 Transactions.

IP Parameter Description	• Added, *The Calculate button at the bottom is used to calculate the optimum parameter values for the PLL*, after the Memory Device Timing Tab paragraph.
	• Updated Table 3.1. Device Information Attributes.
	  • Added MemClock
	• Updated Table 3.2. Clock Settings Attributes.
	  • Updated *MemClock* to *MemClock Actual Frequency*.
	• Updated Table 3.8. Auto Refresh Control Attributes.
	  • Updated the External Auto Refresh Port.
	• Updated Table 3.9. Mode Register Initial Setting Attributes.
	  • Updated the CAS Latency.
	  • Removed Additive Latency.
	• Updated Table 3.10. Memory Device Setting.
	  • Removed Additive Latency.

Signal Description	• Updated Table 4.1. Clock and Reset Port Definitions. • Added srst_n_o port name.
Designing and Simulating the IP	• Updated Table 7.2. Generated File List. • Added constraint/<instance_name>.sdc, eval/clock_constraint.sdc, eval/lscc_pll.v, eval/ddr_clks_src.v, and eval/pll_instance.vh. • Updated the contents of the Constraints sections. • Updated the DDR3 Memory Controller Example Design.
Debugging	Added this section.

Section	Change Summary

Known Issues	• Removed the following from this section.
	  • *Timing closure would be difficult when using the DDR3 SDRAM Controller IP Core on either the CertusPro-NX or MachXO5-NX device family with a gearing ratio of 4:1*.
	  • *Generating bitstream with Radiant 2024.2 on a Linux environment may produce a bitstream that could cause the UART on the Certus-NX Versa Evaluation Board to hang*.

References	• Added Certus-NX High-Speed I/O Interface User Guide (FPGA-TN-02216). • Added AMBA AXI4 Protocol Specification.

Revision 1.9, IP v2.1.0, March 2025

Section	Change Summary
All	Updated the IP version from *v2.0.1* to *v2.1.0*.
References	Added the Certus-NX DDR3 Memory Controller Driver User Guide (FPGA-TN-02401) in this section.

Revision 1.8, IP v2.0.1, December 2024

Section	Change Summary
All	Minor editorial fixes.
Abbreviations in This Document	Changed *Acronyms* to *Abbreviations*.
Introduction	• Added IP Support Summary section. • Moved Licensing and Ordering Information to this section. • Added Hardware Support section. • Added Minimum Device Requirement section. • Added Limitations section.
Functional Description	• Reworked this section. • Renamed *PLL External Reset* to *Clocking and Reset*. • Removed Signal Description in this section.
IP Parameter Description	Reworked the old section *Core Generation, Simulation and Validation* into *IP Parameter Description* section.
Signal Description	Added this section.
Register Description	Added this section.
DDR3 SDRAM Controller Example Design	Added this section.
Designing and Simulating the IP	Added this section.
Appendix A. Resource Utilization	• Updated Table A.1 Resource Utilization for IP Core v2.. • Added Table A.2. Resource Utilization for IP Core v2..
Known Issues	• Added this section.
Ordering Part Number	• Removed this section and transferred to the Introduction section.
References	Added DDR3 SDRAM Controller IP Release Notes (FPGA-RN-02032) in this section.

Revision 1.7, December 2021

Section	Change Summary
Functional Description	• Updated Figure 2.1 to add missing signals. • Added Figure 2.8 and PLL External Reset section.
Signal Description	Added pll_rst_n_i, wl_start_i and wl_done_o in Table 2.1.

Section	Change Summary
Attribute Summary	• Added *Enable External PLL Reset* in Table 2.2. • Updated default values for TRCD, TRP, TRAS, TCKE, TPD, TXPDLL and TWLDQSEN, and added note for *Manually Adjust* at the end of Table 2.2. • Added description for *Enable External PLL Reset* in Table 2.3.
Generation and Synthesis	Updated Figure 3.1, Figure 3.2, Figure 3.3, and Figure 3.5 for IP Core v1.4.2.
Appendix A. Resource Utilization	Updated this section.

Revision 1.6, October 2021

Section	Change Summary
Functional Description	• Updated clock signals in Table 2.1 due to change in Enable PLL option. • Added Clock Settings Group and updated *Select Memory* options in Table 2.2. • Updated Table 2.3 to add descriptions for the following attributes: *Gearing Ratio*, *I/O Buffer Type*, *Enable PLL*, *PLL Reference Clock from Pin*, *I/O Standard for Reference Clock*
Core Generation, Simulation, and Validation	• Updated Figure 3.1, Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5 for IP Core v1.4.0. • Updated Table 3.1 to add eval/lscc_pll.v. • Updated steps for Running Functional Simulation.
Appendix B. Limitation	Removed this section.

Revision 1.5, June 2021

Section	Change Summary
Introduction	• Updated Table 1.1 to add CertusPro-NX support and update IP Core – Lattice Radiant Design version.
Ordering Part Number	• Added part number for CertusPro-NX.
References	Updated this section to add CertusPro-NX web page.

Revision 1.4, October 2020

Section	Change Summary
Introduction	Updated reference to the Lattice Radiant software user guide.
Core Generation, Simulation, and Validation	Updated reference to the Lattice Radiant software user guide.
Appendix B. Limitation	Added this section.
References	Updated this section.

Revision 1.3, August 2020

Section	Change Summary
Appendix A. Resource Utilization	Updated Table A.1 to add EBR and Register columns, modified values for LUTs and IDDR/ODDR/TDDR columns, and added table note 2.

Revision 1.2, June 2020

Section	Change Summary
Introduction	• Updated Table 1.1 to add Certus-NX as supported FPGA family, LFD2NX-40 as targeted device. Added new entry for Lattice Implementation and changed Synplify version to Pro for Lattice. • Updated Features section to include selectable gearing ratio: 4:1, 8:1.

Section	Change Summary
Functional Description	• Updated note 3 in Table 2.1 for signal bit width change due to gearing ratio feature and added note 4 to indicate the change in signal names. • Updated Table 2.2 for gearing ratio feature and improved attribute dependency checks.
Core Generation, Simulation, and Validation	• Updated Figure 3.1, Figure 3.2 and Figure 3.3 for IP Core v1.0.2. • Updated Table 3.1 for the added eval folder. • Added Required Post-Synthesis Constraints section.
Ordering Part Number	Updated this section.
Appendix A. Resource Utilization	Updated this section.
References	Updated this section.

Revision 1.1, February 2020

Section	Change Summary
Introduction	Updated Table 1.1 to add LIFCL-17 as targeted device.
All	Minor editorial changes.

Revision 1.0, December 2019

Section	Change Summary
All	Initial release

www.latticesemi.com