LPDDR4 Memory Controller IP Core for Nexus Devices

IP Version:v2.7.0

User Guide

FPGA-IPUG-02127-2.2

June 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
AHB-L	Advanced High-Performance Bus Lite
APB	Advanced Peripheral Bus
AXI4	Advanced eXtensible Interface 4
BL	Burst Length
CA	Command and Address
CDC	Clock Domain Crossing
CS	Chip Select
CSR	Configuration Set Register
CBT	Command Bus Training
DBI	Data Bus Inversion
DDR	Double Data Rate
DDRDLL	Double Data Rate Delay-Locked Loop
DFI	DDR PHY Interface
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
LPDDR4	Low Power Double Data Rate Generation 4
LVSTL	Low Voltage Swing Terminated Logic
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
SSN	Simultaneous Switching Noise
VREF	Voltage Reference
VT	Voltage and Temperature

1. Introduction

The Lattice Semiconductor LPDDR4 Memory Controller for Nexus™ devices offers a turnkey solution that includes a controller, Physical Layer (DDR PHY), and associated clocking and training logic to interface with LPDDR4 SDRAM. The IP Core is implemented in SystemVerilog and is compatible with Lattice Radiant™ software and Synopsys Synplify Pro® synthesis tools. The LPDDR4 Memory Controller simplifies the interface between CertusPro™-NX and MachXO5™-NX devices and external LPDDR4 memory for user applications.

1.1. Quick Facts

Table 1.1 summarizes the LPDDR4 Memory Controller for Nexus Devices.

Table 1.1. Quick Facts

IP Requirements	Supported Devices	CertusPro-NX (LFCPNX-50, LFCPNX-100, LFCPNX-100AUTODIE),MachXO5-NX (LFMXO5-55T, LFMXO5-55TDQ, LFMXO5-65TDQ, LFMXO5-100T), UT24CP.
	IP Changes[1]	For a list of changes to the IP, refer to the LPDDR4 Memory Controller IP Core for Nexus Devices IP Release Notes (FPGA-RN-02057)
Resource Utilization	Supported User Interfaces	AHB-L for data access in IP Core v1.x.x only AXI4 for data access in IP Core v2.x.x and up APB for configuration access in IP Core v1.x.x and up
	Resources	Refer to Table A.1
Design Tool Support	Lattice Implementation[2]	IP Core v2.7.0 – Lattice Radiant software 2026.1
	Synthesis	Synopsys Synplify Pro for Lattice Lattice Synthesis Engine (LSE) is not supported
	Simulation	Questasim
Driver Support	API Reference	Refer to the CertusPro-NX LPDDR4 Memory Controller Driver User Guide (FPGA-TN-02378)

Notes:

1. In some instances, the IP may be updated without changes to the user guide. The user guide may reflect an earlier IP version but remain 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. LPDDR4 Memory Controller IP Support Readiness

Device Family	IP	Rank	DDR Width	Data Rate (Mbps)	Timing Model
CertusPro-NX	LPDDR4	Single	x16	600	Final
				700	
				800	
				1,066	
			x32	600	
				700	
				800	
				1,066	
		Dual	x16	600	
				700	
				800	
			x32	600	
				700	
				800	

Device Family	IP	Rank	DDR Width	Data Rate (Mbps)	Timing Model
MachXO5-NX	LPDDR4	Single	x16/x32	600/700/800/1,066	Preliminary
		Dual	x16/x32	600/700/800	

The example design for the LPDDR4 Memory Controller IP Core allows for simulation and deployment to development boards for testing. For instructions on running the example design on hardware and in simulation, refer to the Designing and Simulating the IP section.

1.3. Features

The LPDDR4 Memory Controller for Nexus Devices supports the following key features:

- LPDDR4 SDRAM protocol, compliant to LPDDR4 JEDEC Standard
  - LPDDR4 SDRAM speeds of:
    - 300 MHz (600 Mbps)
    - 350 MHz (700 Mbps)
    - 400 MHz (800 Mbps)
    - 533 MHz (1,066 Mbps)
- LPDDR4 Memory Controller features:
  - Component support for interface data widths of x16 and x32
  - Up to 16 Gb density support per channel
  - x16 LPDDR4 device support (8:1 DQ:DQS ratio)
  - Burst length of BL16
  - 8:1 gearing mode (LPDDR4:FPGA logic interface clock ratio)
  - Dual-rank support
  - Data Bus Inversion (DBI) is supported on read only. Write DBI is not supported
  - Configurable CAS latencies for Reads and Writes based on target interface speed
  - Configurable Address widths to support various memory densities
- AHB-Lite data interface support (in IP Core v1.x.x only):
  - SINGLE, INCR1, and INCR8 for read/write operations
  - Aligned addressing only
- AXI4 data interface support (in IP Core v2.x.x and up):
  - INCR with AxLEN = 0-255 for read/write operations
  - Unaligned transfer using byte strobes
  - Narrow transfers (in IP Core v2.1.x and up)
  - Narrow AXI widths of 32, 64, and 128
  - Aligned addressing to AxSIZE only
- APB configuration interface support (in IP Core v1.x.x and up):
  - Automatic LPDDR4 SDRAM initialization
  - Dynamic valid window optimization for Read/Write paths
  - DQ-DQS skew optimization for Write training
- Periodic training support featuring:
  - Temperature tracking
  - Adaptive/Derate refresh rate for extended temperature support
  - ZQ calibration (ZQCAL START and LATCH only)
  - Periodic Voltage/Temperature (VT) compensation on I/O delay
- Automatic detection of idle triggering Self-Refresh with Power-down entry
- Polling and Out-of-band interrupt support for error and extended temperature support

Table 1.3. Features Overview

Key Features	LPDDR4 Support Details[1]
Device Format	Component
Data Widths	x16, x32
Data User Interface	AHB-L, AXI4
Configuration Interface	APB
Maximum Command Speed	533 Mbps
Maximum Data Speed	1,066 Mbps
HW Managed Periodic Events	
Refresh	All bank auto refresh
ZQ Calibration	Yes, for ZQ start and latch No, for ZQ reset
Low Power Features	Self-refresh with power-down
Other Features	
Error Correction Code (ECC)	No
Dual rank	Yes (requires dual rank DRAM chip)
Data Bus Inversion (DBI)	Yes for read, No for write
Temperature Tracking	Yes
Refresh Adaptation (derate) to Temperature Variation	Yes
On-Die Termination (ODT)	Yes, for both DQ and CA
Periodic VT compensation on I/O delay	Yes
Training	
Initialization	Yes
Command Training	Yes
Write Leveling	Yes
Read Training	Yes
Write Training	Yes
VREF Training	Yes

Note:

1. *Yes* implies that a configurable option exists to enable or disable the feature. *No* implies that the feature is currently not supported and will not be supported in the future.

1.4. Licensing and Ordering Information

The LPDDR4 Memory Controller IP Core is available with the Lattice Radiant Subscription software. To purchase the Lattice Radiant software subscription license, contact Lattice Sales or go to the Lattice Online Store.

1.5. Minimum Device Requirements

The LPDDR4 Memory Controller for Nexus Devices supports both CertusPro-NX and MachXO5-NX devices. Table 1.4 summarizes the minimum device requirements for the Memory Controller IP Core.

Table 1.4. Minimum Device Requirements

LPDDR4 Interface Speed	LPDDR4 Data Width	Supported Speed Grades	
		CertusPro-NX	MachXO5-NX
300 MHz (600 Mbps)	x16, x32	7,8,9	7,8,9
350 MHz (700 Mbps)	x16, x32	7,8,9	7,8,9
400 MHz (800 Mbps)	x16, x32	7,8,9	7,8,9
533 MHz (1,066 Mbps)	x16, x32	8,9	9

1.6. Naming Conventions

This section defines the nomenclature and signal naming conventions used throughout this document.

1.6.1. Nomenclature

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

1.6.2. Signal Names

Signal names use the following suffix conventions:

- _n Active-low signal
- _i Input signals
- _o Output signals
- _io Bidirectional input/output signals

2. Functional Description

This section describes the LPDDR4 Memory Controller for Nexus devices, including clock and reset handling, available user data and configuration interfaces, the LPDDR4 calibration sequence, and LPDDR4 operation descriptions.

2.1. IP Architecture

The LPDDR4 Memory Controller for Nexus Devices includes three main blocks: the Memory Controller, Physical Layer (PHY), and Training Engine. Figure 2.1 shows the LPDDR4 Memory Controller submodules and their connectivity.

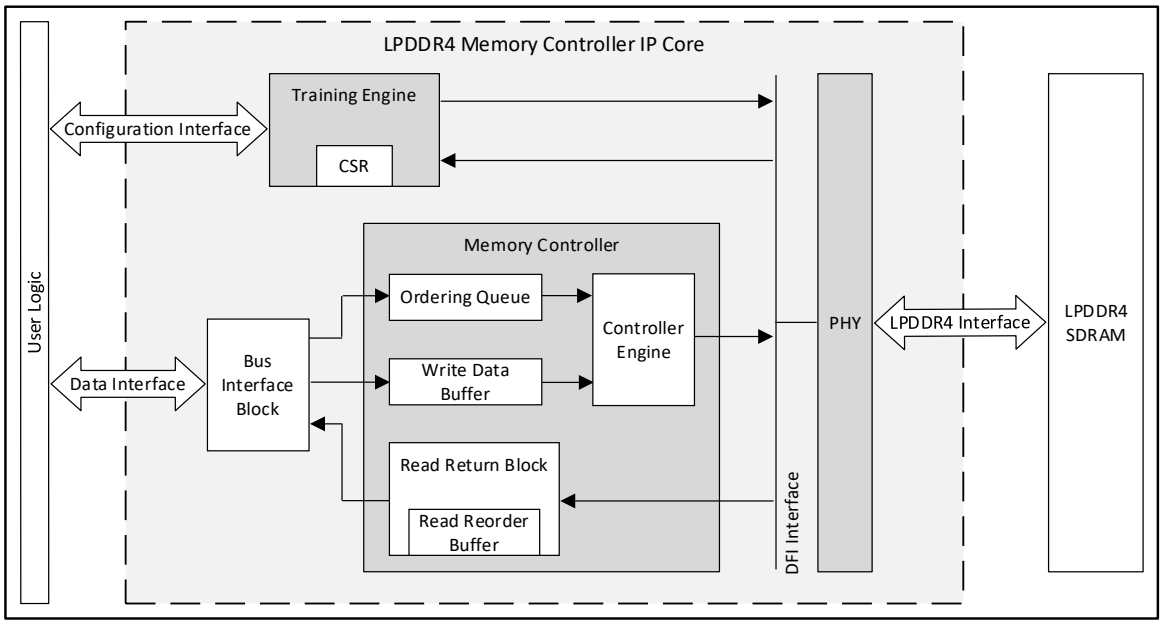


Figure 2.1. Memory Controller IP Core Functional Diagram

The data interface lets you initiate the LPDDR4 command, address, control, and read/write operations to the external LPDDR4 SDRAM. The configuration interface provides access to the Training Engine and the Configuration Set Registers (CSRs), which configure the Memory Controller and perform the LPDDR4 training sequences. The LPDDR4 interface enables the selected Lattice FPGA to communicate with the external LPDDR4 memory. For more information on the data and configuration interfaces, refer to the User Interfaces section of this user guide.

2.1.1. Soft Memory Controller

The Soft Memory Controller consists of the following submodules:

- Bus Interface Block
- Controller Engine
- Read Return Block

2.1.1.1. Bus Interface Block

The Bus Interface Block accepts user-initiated operations on the selected data interface and translates them for processing within the Memory Controller. The data provided to the Controller Engine consists of:

- Address interface size: determined based on the selected LPDDR4 SDRAM device density and LPDDR4 data width.
- Read and Write interface sizes: determined based on the selected LPDDR4 data width.

The ordering queue stores the address, command, and control information from the Bus Interface Block. It prioritizes the execution of commands targeting an already opened bank and maintains correct order of execution when multiple requests target the same bank in LPDDR4 SDRAM. The write data buffer buffers the incoming data for write commands.

2.1.1.2. Controller Engine

The Controller Engine accepts requests from the Bus Interface and translates the address to row, column, rank, and bank format. It supports outstanding writes and reads, which enter a queue and are arbitrated for processing. The following table describes the three types of requests and their processing methods.

Table 2.1. MCE Request Handling

Request type	Prioritization and Definition	Comments
Read	Varies depending on selected data interface protocol. Refer to the Data Interface Protocols section of this user guide for details.	The read request size is equal to the supported LPDDR4 burst length.
Write	Write requests are defined as a write, where the entire data needs to be written for the supported LPDDR4 burst length.	Example: For a 32-bit LPDDR4 data bus width, with burst length 16 (BL16), a full write would be 16×4B = 64B.
Partial write	Partial write requests are defined as a write, where the entire data is not written.	Example: For a 32-bit LPDDR4 data bus width, a partial write would be a write request that is less than 32B. This is equivalent to a masked write.

The system processes all write requests out of order if there is no data dependency. For example, it prioritizes an incoming request tied to an already opened page in LPDDR4 SDRAM if it is not dependent on the requests already in the queue. The system also processes read requests out of order if there is no data dependency, depending on the selected data interface protocol. Refer to the Data Interface Protocols section of this user guide for information regarding outstanding write/read support.

The Controller Engine includes bank management logic to track all the opened and closed pages within each bank, along with timers for each bank and rank. This ensures issued memory commands meet the operational sequence and timing requirements of the LPDDR4 SDRAM. The last layer of logic within the Controller Engine handles handshaking between the PHY and external LPDDR4 memory. The Controller Engine also supports periodic events compliant with the LPDDR4 SDRAM requirements, such as temperature tracking, adaptive/derate refresh rates, and ZQ calibration. For more information on supported periodic features, refer to the LPDDR4 Operation Description section of this user guide.

2.1.1.3. Read Return Block

The Read Return Block responds to the Controller Engine's read requests. It handles the reordering of commands to ensure that the read data is sent to the Bus Interface in order, even if the execution of commands is out of order. The Read Return Block also manages Data Bus Inversion (DBI), an I/O signaling technique that reduces power consumption and improves signal integrity.

2.1.2. Soft PHY

The Soft PHY includes logic in the FPGA fabric to convert SDR to DDR, and vice versa. The DFI interface operates in SDR mode, while the LPDDR4 interface operates in DDR mode. Figure 2.2 shows a high-level block diagram of the Soft PHY and its major submodules. The Training Engine and Memory Controller can access both the Command/Address path and Data Input/Output paths.

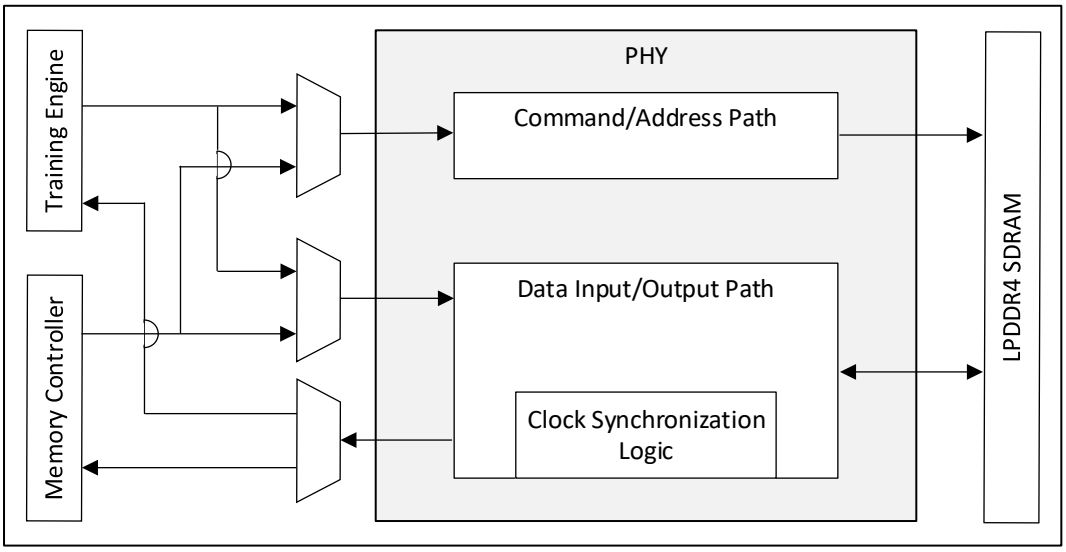


Figure 2.2. Lattice PHY

The PHY uses hardened logic, called I/O modules or hardware primitives, to implement clock synchronization logic and the command/address and data input/output paths. The clock synchronization logic handles Clock Domain Crossing (CDC) between the Edge Clock (ECLK) and System Clock (SCLK). ECLK is an internal clock that clocks the DDR primitives, while SCLK clocks the Memory Controller IP Core. Without proper synchronization, individual bits can become misaligned and the data bus is corrupted.

The LPDDR4 Memory Controller operates in 8:1 gearing mode, meaning ECLK runs at the same frequency as the LPDDR4 interface, while SCLK runs at a quarter of the LPDDR4 memory clock. For a 400 MHz LPDDR4 interface, ECLK runs at 400 MHz and SCLK runs at 100 MHz. This clocking ratio and DDR transfer relationship make the user data bus eight times the width of the LPDDR4 memory data bus. For example, a 32-bit LPDDR4 memory interface requires a 256-bit write and read data bus on the DDR PHY (DFI) side, which is an interface protocol between the Memory Controller and PHY. For more information on the soft PHY, hardware primitives, and clock synchronization logic, refer to the LPDDR4 Memory Interface Module User Guide (FPGA-IPUG-02154).

2.1.3. Soft Training Engine

The Training Engine initializes and trains the external LPDDR4 memory. It provides a configuration interface from user logic to the Configuration Set Registers (CSR). You can issue commands to the Training Engine to perform reads and writes to these registers, allowing you to set desired programmable options, handle interrupts, and obtain details during the LPDDR4 memory training sequence. It includes command and data generators, a comparator, user-accessible registers, delay logic, a command sequencer, and a RISC-V CPU subsystem.

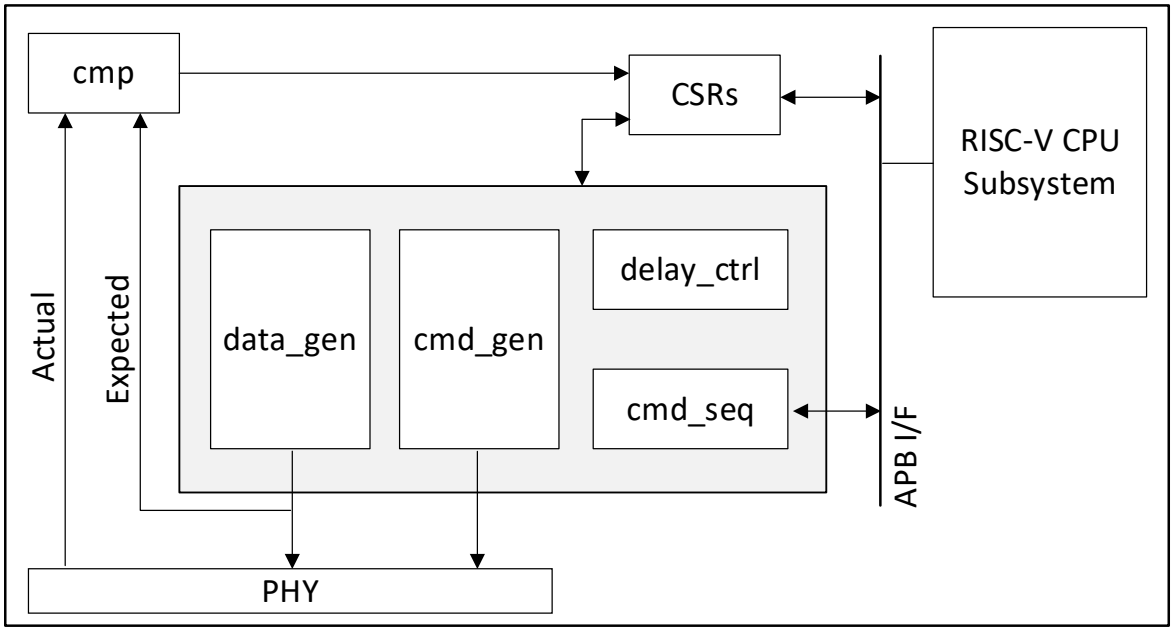


Figure 2.3. Training Engine

The command generator (cmd_gen) generates LPDDR4 commands for all training procedures. The data generator (data_gen) is a pseudo-random (PRBS) number generator that produces the expected data for read training. During write training, it generates both the write data and expected read data. The compare (cmp) block compares the actual read data with the expected read data and sends the result to the registers within the CSR block. The command sequencer (cmd_seq) dictates the command and data generation during the various training stages. The delay control (delay_ctrl) block adjusts the delay of the target LPDDR4 memory signals during training. The CSR block includes various user-accessible registers outlined in the Register Description section of this user guide.

The RISC-V subsystem writes to the CSRs and command sequencer block to initialize and train the LPDDR4 interface. It is also responsible for programming the pattern and seed in the command and data generator blocks prior to the execution of the training procedure.

2.2. Clocking and Reset

2.2.1. General Clocking and Reset

This section describes the required input clocks and resets for the Memory Controller IP, and the reset-assertion combinations to avoid AXI4 bus hang-up.

2.2.1.1. Input Clocks

The Memory Controller IP requires three main input clocks.

1. **Differential reference clock (pll_refclk_i)** is used to generate the clock used for controller core and LPDDR4 interfaces. You can provide this clock from an external source at 100 MHz through dedicated FGPA routing. For a list of valid reference clock frequencies, refer to the IP Parameter Description section. The input clock is then routed through a dedicated PLL within the Memory Controller IP to generate the Edge Clock (ECLK) and System Clock (SCLK). The ECLK signal is used internally by the IP Core to clock the I/O modules, and SCLK is used to clock the IP Core. Both these clocks scale directly with the DDR data rate.
2. **APB clock (pclk_i)** is used for the configuration interfaces within the controller and its training subsystem. This clock is also used as the synchronization logic for SCLK and ECLK. For additional details regarding clock synchronization, refer to the LPDDR4 Memory Interface Module (FPGA-IPUG-02154).
3. **AXI clock (aclk_i)** is used for the AXI bus interface. Refer to the AXI4 section of this user guide for information regarding clock selection for the AXI interface.

2.2.1.2. Input Resets

The Memory Controller IP has four main input resets:

1. **Main controller reset (rst_n_i)**. When asserted, the Memory Controller, its training registers and SDRAM are reset to their default values. The IP core internally synchronizes the reset deassertion to its internal target clock.
2. **APB reset (preset_n_i)** is used to reset the APB interface and the training subsystem. (PHY training registers are preserved). The IP core does not synchronize this reset deassertion to pclk_i.
3. **AXI reset (areset_n_i)** is used to reset the AXI bus interface. The IP core internally synchronizes the reset deassertion of this reset to its internal target clock.
4. **PLL reset (pll_rst_n_i)** is used to reset the internal PLL in the Memory Controller.

2.2.1.3. Reset-Assertion Expectations (to Prevent AXI Bus Hang-Up)

- When you assert rst_n_i
  - If rst_n_i is asserted when AXI4 Bus Idle is not true (AXI is not idle): assert_n_i is expected.
  - If rst_n_i is asserted during AXI4 Bus Idle: assert_n_i is not required.
- When you assert areset_n_i
  - If areset_n_i is asserted when AXI4 Bus Idle is not true (AXI is not idle): rst_n_i is expected.
  - If areset_n_i is asserted during AXI4 Bus Idle: rst_n_i is not required.
- When you assert pll_rst_n_i
  - If pll_rst_n_i is asserted when AXI4 Bus Idle is not true (AXI is not idle): both rst_n_i and areset_n_i are expected.
  - If pll_rst_n_i is asserted during AXI4 Bus Idle: rst_n_i is expected.
- Summary guidance
  - If AXI is not idle, reset the whole path, not only part of it.
  - If AXI is idle, partial resets are safe under the expectations listed above.

Table 2.2. AXI Reset Dependency vs. Bus State

Reset asserted	AXI4 Bus state	Required additional reset	What this implies logically
rst_n_i	Not idle	areset_n_i required	AXI interface must also be reset.
rst_n_i	Idle	areset_n_i **not** required	AXI interface can remain stable.
areset_n_i	Not idle	rst_n_i required	Core must also be reset.
areset_n_i	Idle	rst_n_i **not** required	Core can remain stable.
pll_rst_n_i	Not idle	rst_n and areset_n_i expected	Everything downstream must be reset.
pll_rst_n_i	Idle	rst_n_i expected	Core reset is still required.

Notes: The AXI4 Bus state is *Idle*, meaning there are no outstanding transactions on the controller's AXI interface, where:

- All read data is returned on the Read Data channel.
- All write completions have returned on the Write Response channel.
- There is no ongoing activity on the Write Address, Write Data or Read Address channel.

2.3. User Interfaces

This section describes the supported protocols for data and configuration interfaces available to the user and supported by the LPDDR4 Memory Controller.

2.3.1. Data Interface Protocols

The data interface allows you to initiate read and write operations to external LPDDR4 SDRAM. The LPDDR4 Memory Controller supports two protocols for data interfacing: AHB-Lite or AXI4.

2.3.1.1. AHB-Lite

The AHB-Lite interface is available in IP Core v1.x.x and serves as a single-channel bus intended for high-performance and high-frequency applications. The AHB-Lite interface operates off the sclk_o signal and resets through the rst_n_i signal. Refer to the Clocking and Reset section of this user guide for more details.

The Memory Controller IP supports the following types of AHB-Lite burst transfers:

- Burst Type (HBURST):
  - SINGLE: single burst (1 beat)
  - INCR: incrementing burst that does not wrap at address boundaries (4 or 8 beats)
- Burst Size (HSIZE):
  - [8, 16, 32, 64, 128, 256] Bytes
- Burst Address (HADDR):
  - Unaligned addressing is not supported so all transfers must be aligned to the address boundary
  - SINGLE: address should be aligned to HSIZE
  - INCR4: address should be aligned to (BUS_WIDTH / 8) × 16
  - INCR8: address should be aligned to (BUS_WIDTH / 8) × 32

The AHB-Lite protocol supports only a single outstanding transaction per bus manager. As a result, when configuring AHB-Lite as the user data interface, the Memory Controller supports up to four outstanding writes and only one outstanding read, where the next request cannot start until the current one completes. For AHB-Lite implementation, reads take priority over writes unless there is a dependent write to be serviced. For more information regarding the AHB-Lite protocol, refer to the AMBA AHB Protocol Specification.

Table 2.3. Supported AHB-Lite Transactions

Transaction Type	HBURST	HSIZE[2:0]	Comment
Write	SINGLE	[0, 1, 2, 3, 4, 5] [1]	[1-32] Byte write
Write	INCR4	[0, 1, 2, 3, 4, 5] [1]	[4-128] Byte write
Write	INCR8	[0, 1, 2, 3, 4, 5] [1]	[8-256] Byte write
Read	SINGLE	[0, 1, 2, 3, 4, 5] [1]	[1-32] Byte read
Read	INCR4	[3, 4, 5][1]	[32-128] Byte read
Read	INCR8	[3, 4, 5][1]	[64-256] Byte read

Note:

1. For HSIZE=3, only BUS_WIDTH=16 is supported. For HSIZE=4, only BUS_WIDTH=32 is supported. For HSIZE=5, only BUS_WIDTH=64 is supported.

2.3.1.2. AXI4

The AXI4 interface is available from IP Core v2.x.x and serves high-bandwidth and low-latency operation. The AXI4 interface is a multi-channel bus consisting of five independent channels: write address, read address, write data, read data, and write response channels (read response is sent along with the read data). The AXI4 interface operates off the aclk_i signal when the Enable Local Bus Clock attribute is *checked*. The AXI4 interface resets through the areset_n_i signal, which is an asynchronous active low reset, where you must ensure it is synchronously de-asserted to aclk_i.

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

- Burst Type (AxBURST):
  - INCR: incrementing burst that does not wrap at address boundaries
- Burst Size (AxSIZE):
  - [1, 2, 3, 4, 8, 16, 32, 64] Bytes
- Burst Length (AxLEN):
  - 1-64 beat burst: when IP parameter MAX_BURST_LEN = 64
  - 1-128 beat burst: when IP parameter MAX_BURST_LEN = 128
  - 1-256 beat burst: when IP parameter MAX_BURST_LEN = 256
- Burst Address (AxADDR):
- The system does not support unaligned addressing, so all transfers must align to AxSIZE
- The system does not support unaligned transfer using byte strobes
- For addresses not aligned to BUS_WIDTH x 16 / 8, the maximum allowable AxLEN is as defined below:

DDR Bus Width (BUS_WIDTH)	Maximum Burst Length	Burst Length Offset (BUS_WIDTH x 16) / AXI_DATA_WIDTH	Maximum Allowable AxLEN
16	64	16 x 16 / AXI_DATA_WIDTH	64 - (256 / AXI_DATA_WIDTH) - 1
32	64	32 x 16 / AXI_DATA_WIDTH	64 - (512 / AXI_DATA_WIDTH) - 1
16	128	16 x 16 / AXI_DATA_WIDTH	128 - (256 / AXI_DATA_WIDTH) - 1
32	128	32 x 16 / AXI_DATA_WIDTH	128- (512 / AXI_DATA_WIDTH) - 1
16	256	16 x 16 / AXI_DATA_WIDTH	256 - (256 / AXI_DATA_WIDTH) - 1
32	256	32 x 16 / AXI_DATA_WIDTH	256 - (512 / AXI_DATA_WIDTH) - 1

- Example Use Case: DDR Bus Width is 32 bits and Maximum Burst Length is 64 beats
  - AXI Data Width is 32 bits
  - Maximum Allowable AxLEN: 64 - (512/32) - 1 = 47 beats
- Write Strobes (WSTRB)

The AXI4 protocol supports out-of-order transaction completion and allows multiple outstanding transactions per bus manager. As a result, when configuring AXI4 as the user data interface, the Memory Controller supports up to four outstanding writes and eight outstanding reads, with both reads and writes having the same priority. For more information on the AXI4 protocol, refer to the AMBA AXI Protocol Specification.

Table 2.4. Supported AXI4 Transactions

Transaction Type	AxBURST	AxLEN[7:0]	AxSIZE[2:0][1]	Comment
Write	INCR	[0 – MAX_BURST_LEN-1]	[0, 1, 2, 3, 4, 5]	[1-4096] Byte write
Read	INCR	[0 – MAX_BURST_LEN-1]	[0, 1, 2, 3, 4, 5]	[1-4096] Byte read

Notes:

1. For IP parameter AXI_DATA_WIDTH = 32, AxSIZE = [0, 1, 2] are supported.
   For IP parameter AXI_DATA_WIDTH = 64, AxSIZE = [0, 1, 2, 3] are supported.
   For IP parameter AXI_DATA_WIDTH = 128, AxSIZE = [0, 1, 2, 3, 4] are supported.
   For IP parameter AXI_DATA_WIDTH = 256, AxSIZE = [0, 1, 2, 3, 4, 5] are supported.

2.3.2. Configuration Interface Protocol

The configuration interface allows you to initialize and train the LPDDR4 memory interface. The Memory Controller IP Core uses the APB protocol for its configuration interface.

2.3.2.1. APB

The APB interface is available in both IP Core v1.x.x and v2.x.x and serves as a low-power protocol intended for accessing programmable control registers. It is not pipelined and operates as a synchronous protocol with a single address bus and two data buses: write and read. For more information on the APB protocol, refer to the AMBA APB Protocol Specification.

The Memory Controller uses this protocol to initialize and train the interface between FPGA logic and LPDDR4 SDRAM. The APB interface operates on the pclk_i signal and resets through the preset_n_i signal. Refer to the Clocking and Reset section of this user guide for more details.

The APB interface becomes unavailable when the Enable APB interface attribute is *unchecked*. For details on initializing and training LPDDR4 SDRAM without the APB interface, refer to the Initialization and Training without APB Interface section of this user guide.

2.4. LPDDR4 Calibration

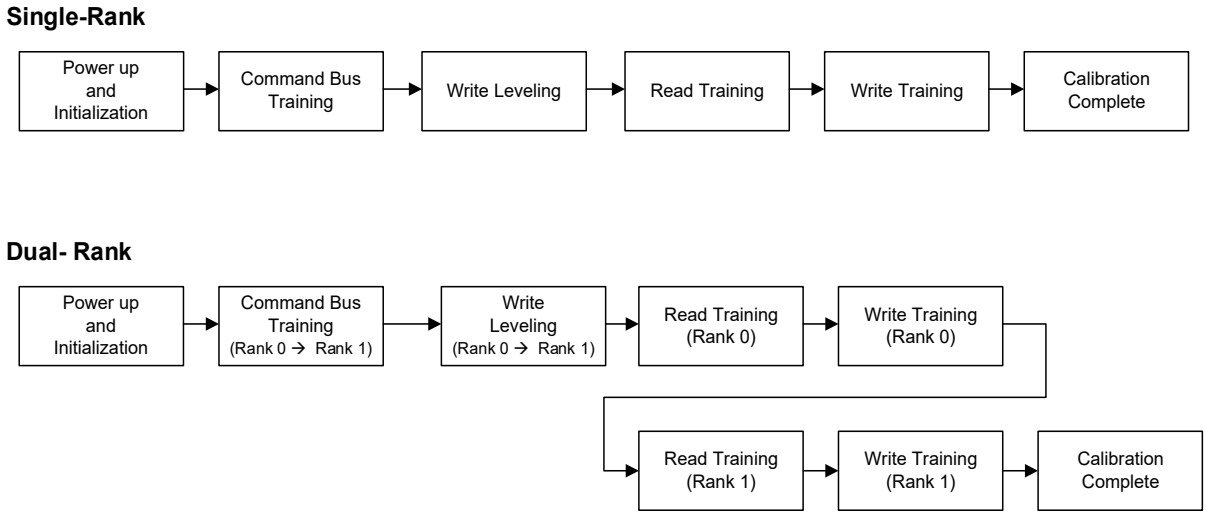
To ensure proper device functionality, initialize and train the external LPDDR4 memory before the Memory Controller performs data accesses. The LPDDR4 JEDEC Standard outlines a detailed explanation of the initialization and training sequence.

Upon device power-up, the soft RISC-V CPU inside the Training Engine remains in reset. To start the initialization and training sequence of the LPDDR4 SDRAM device, execute the following steps through the configuration interface:

- Poll the Status Register (STATUS_REG) until the phy_ready signal asserts to ensure that the PHY is ready.
- Enable initialization and training by writing 8'hFF to the Training Operation Register (TRN_OP_REG). This enables the execution of initialization, command bus training, write leveling, read training, and write training sequences. For simulation purposes, shorten the initialization and training sequences by writing 8'h00 to TRN_OP_REG. For more information, refer to the Register Description section of this user guide.
- Pull the CPU and Training Engine out of reset by writing 2'h3 to the Reset Register (RESET_REG). This starts the initialization and training sequence.
- Wait until initialization and training completes using one of the following methods:
  - Poll the Status Register (STATUS_REG) until the write_trn_done signal asserts (STATUS_REG[4]=1). This indicates that write training is complete, which is the final stage in the initialization and training process.
  - Wait for the trn_done_int signal (INT_STATUS_REG[0]=1) or the trn_error_int signal (INT_STATUS_REG[1]=1) to assert in the Interrupt Status Register (INT_STATUS_REG). This method requires the trn_done_en signal (INT_ENABLE_REG[0]=1) and the trn_err_en signal (INT_ENABLE_REG[1]=1) to assert in the Interrupt Enable Register (INT_ENABLE_REG).

After completing the above steps, the PHY control transfers to the Memory Controller and you can start accessing the LPDDR4 memory through the data interface. Once init_done_o asserts, the RISC-V CPU and Training Engine enter reset to save power. Refer to the Initialization and Training Sequence section of this user guide for more details on the LPDDR4 calibration sequence.

Figure 2.4 compares the LPDDR4 calibration sequence for single-rank and dual-rank modes. In dual-rank mode, if rank 0 encounters a failure, the training routine continues with rank 1.



Notes:
1. Read training consists of four stages: read-gate training, read-DQS training, MC-side DQ_Vref training, and read-deskew training.
2. Write training consists of three stages: initial write training, MC-side DQ_Vref training, and final write training.

Figure 2.4. LPDDR4 Calibration Sequence

2.4.1. Initialization and Training Sequence

The LPDDR4 Memory Controller IP initializes according to the LPDDR4 JEDEC Standard. This section describes the steps in the LPDDR4 initialization and training sequence.

Once the CPU and Controller Engine exit reset, the ddr_reset_n_o signal is held low for 200 µs. The ddr_cke_o signal then asserts low for 2 ms before the LPDDR4 clock (ddr_ck_o) activates and begins toggling. These steps take a long time to simulate, to shorten the simulation time, set the init_en signal low (TRN_OP_REG[0]=0) to reduce the ddr_reset_n_o and ddr_cke_o assertion times to a few clock cycles. Refer to the Simulation Example Design section of this user guide for simulation runtimes of different LPDDR4 configurations.

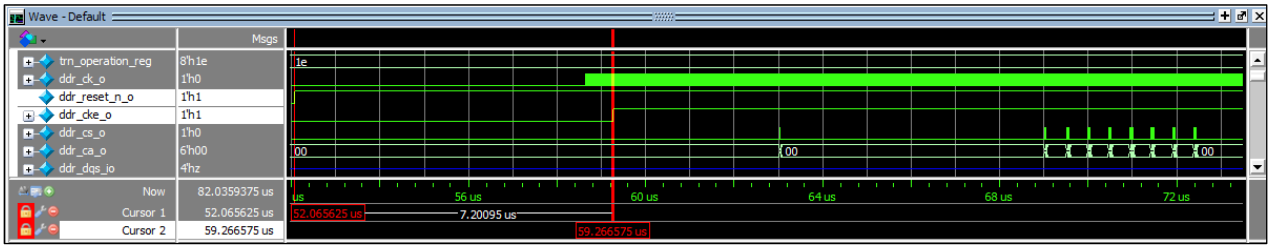


Figure 2.5. Shortened Initialization Sequence Simulation Waveform (TRN_OP_REG[0]=0)

2.4.1.1. Command Bus Training (CBT)

Command and Address (CA) calibration delays the command and address signals as necessary to optimize the CA window. When setting the cbt_en signal high (TRN_OP_REG[1]=1), the Memory Controller performs Command Bus Training (CBT) according to the LPDDR4 JEDEC Standard. This centers the entire CA bus relative to ddr_ck_o by aligning the rising edge of CK to the middle of the CA valid window. The LPDDR4 memory then provides feedback through the DQ line, which the Memory Controller uses to determine if additional adjustment is required.

During this time, the DDR clock, ddr_ck_o, stops before and after CBT. This is expected as the clock switches between low frequency (50 MHz) and high frequency (LPDDR4 interface speed) operation, as outlined in the LPDDR4 JEDEC Standard. Upon successful completion of CBT, ddr_ca_o centers to the eye of ddr_ck_o, ensuring correct behavior during high frequency operation. Refer to the Command Bus Training Debug section of this user guide for additional details on how the LPDDR4 Memory Controller performs CBT.

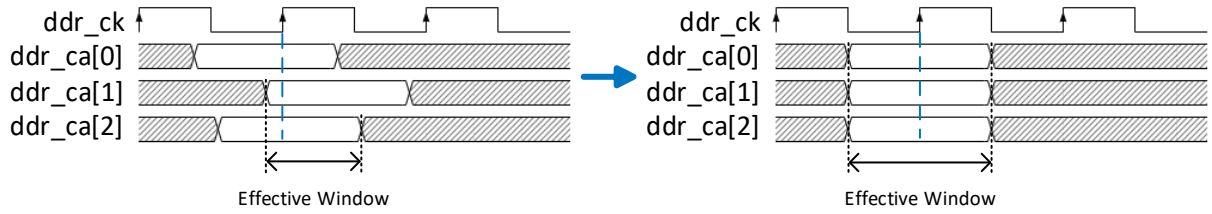


Figure 2.6. Command Bus Training

2.4.1.2. Write Leveling

Write leveling delays each Data Strobe (DQS) relative to the DDR clock during write operations to ensure edge alignment. This addresses the CK to DQS timing skew introduced by adopting Fly-By Topology, which reduces Simultaneous Switching Noise (SSN) by allowing different memory components to receive write commands at different times. Setting the write_lvl_en signal high (TRN_OP_REG[2]=1) enables the Memory Controller to perform write leveling on all available ranks. During this process, the Memory Controller delays the ddr_dqs_o signal until the DQS rising edge of the LPDDR4 SDRAM device. The Memory Controller captures a 0-to-1 transition on ddr_ck_o. The LPDDR4 memory provides feedback of the captured clock signal value through the DQ line, which the Memory Controller uses to determine if additional adjustment is required. Refer to the Write Leveling Debug section of this user guide for additional details on how the LPDDR4 Memory Controller performs write leveling.

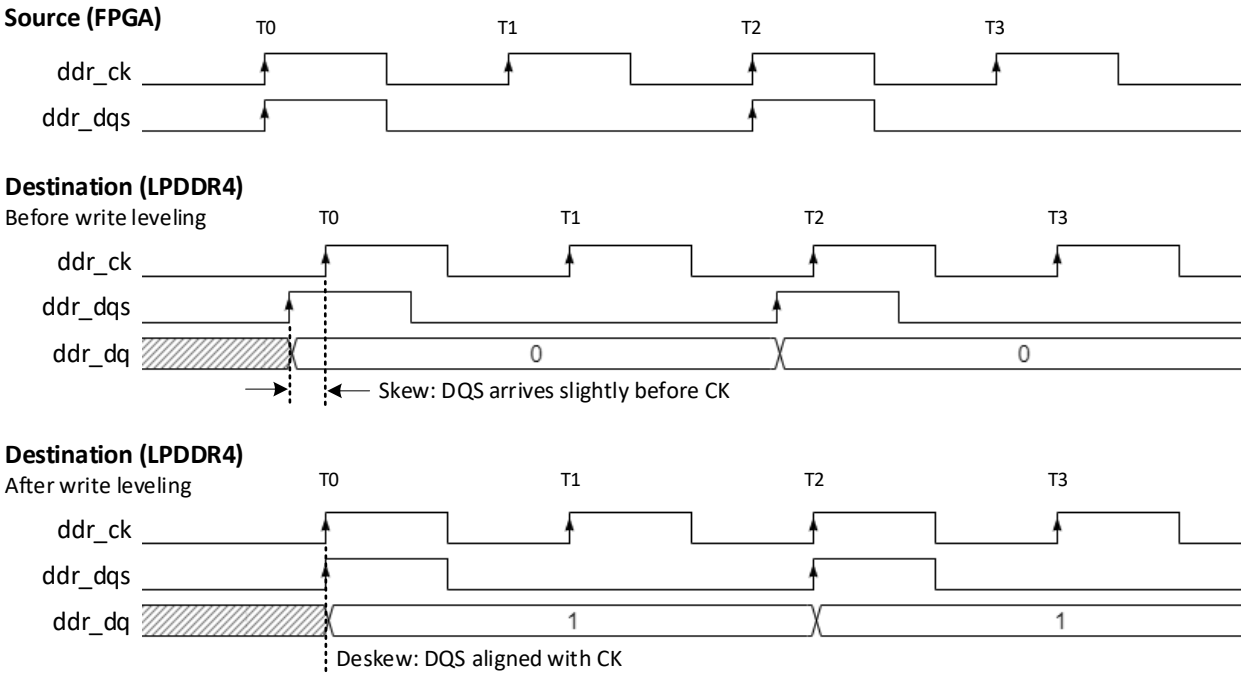


Figure 2.7. Write Leveling

2.4.1.3. Read Training

Read training delays each DQS relative to the DDR clock during read operations to ensure a full burst is captured correctly. Read Training also center-aligns DQS relative to the DQ window. Setting the read_trn_en signal high (TRN_OP_REG[3]=1) enables the Memory Controller to perform read training. During this process, the Memory Controller issues burst reads and delays the ddr_dqs_o signal until a full burst is captured correctly. The read training has four stages:

1. Read gate training - The Memory Controller delays its read gate with a coarse delay adjustment of 45° phase shift per tap to ensure all data sent by the external LPDDR4 SDRAM is received optimally.
2. Read DQS training - The DQS delay is optimized such that the 8 DQS pulses are centered to the read gate using fine delay adjustment of 12.5 ps per tap. This ensures that the DQS is captured properly for the entire read burst.
3. MC-side DQ_Vref training –When TRN_OP_REG.mc_vref_training_en is set, the Memory Controller performs this as part of read training. Refer to VREF Training section for more information.
4. Read deskew training - The Memory Controller performs read per-byte deskew to optimize the read DQ window to its associated DQS. This stage uses the trained DQ_Vref. Refer to the Read Training Debug section of this user guide for additional details on how the LPDDR4 Memory Controller performs read training.

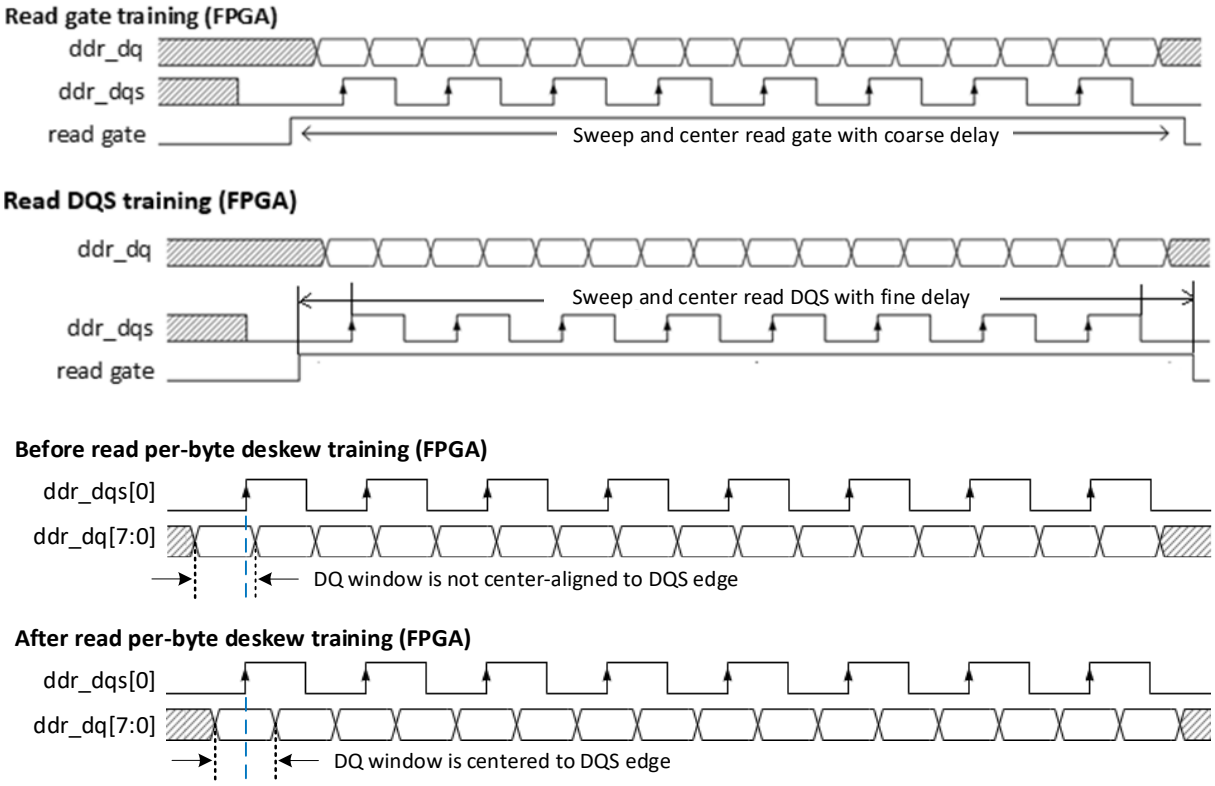


Figure 2.8. Read Training

2.4.1.4. Write Training

Write training delays each DQS relative to DQ during write operations to optimize the data window. Setting the write_trn_en signal high (TRN_OP_REG[4]=1) enables the Memory Controller to perform write training. This entails aligning the rising edge of DQS to the center of the DQ valid window per byte on the LPDDR4 memory side. The write training has three stages:

1. Initial write training – Uses the OUTDELAYA primitive to adjust the ddr_dq delay. The goal is to confirm that the *Initial MC DQ_VREF Value* parameter is good. The next stage will not be performed if this first stage failed.
2. DRAM-side DQ_Vref training – When TRN_OP_REG.mem_vref_training_en is set, the Memory Controller performs this as part of write training. Refer to VREF Training section for more information.
3. Final write training – uses the trained DRAM DQ_Vref. This training stage uses the OUTDELAY primitive for incrementing sweep but uses the DQSBUF.DQSW270 for decrementing sweep.

Refer to the *Data Input/Output Path* section of LPDDR4 Memory Interface Module User Guide (FPGA-IPUG-02154) for more information of the OUTDELAY and DQSBUF. Refer to the Write Training Debug section of this user guide for additional details on how the LPDDR4 Memory Controller performs write training.

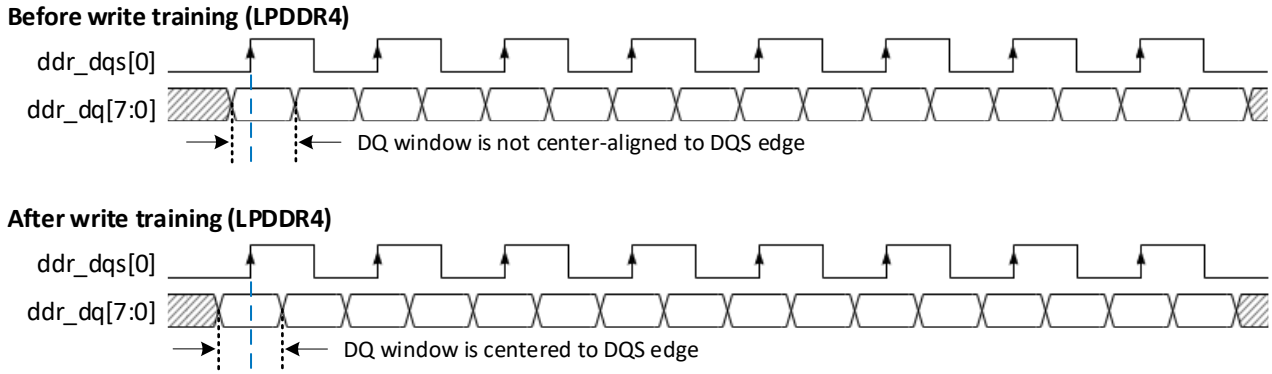


Figure 2.9. Write Training

2.4.1.5. VREF Training

VREF training aims to provide stable and robust memory access by maximizing margins on command, address, and data signals. VREF training is part of CBT, read training, and write training. Setting the ca_vref_training_en (TRN_OP_REG[5]=1), mc_vref_training_en (TRN_OP_REG[6]=1), and mem_vref_training_en (TRN_OP_REG[7]=1) signals high enables VREF training during CBT, read training, and write training.

The voltage reference determines if a signal is registered as high (1) or low (0). PVT variations and ODT drive/impedance matching can cause the internal voltage reference on LPDDR4 to fluctuate, moving the center of the eye. VREF training involves repeating the CBT, read deskew training, and write training steps across different VREF values. The margin for each VREF is recorded and the VREF that produces the maximum margin are chosen. If many VREFs produced the same maximum margin, their center was chosen. This is a two-dimensional VREF training wherein the first dimension is the horizontal delay and the second dimension is the voltage.

Disable CA_VREF training (ca_vref_training_en). It is not required because the CA eye margin is sufficient even at 533 MHz.

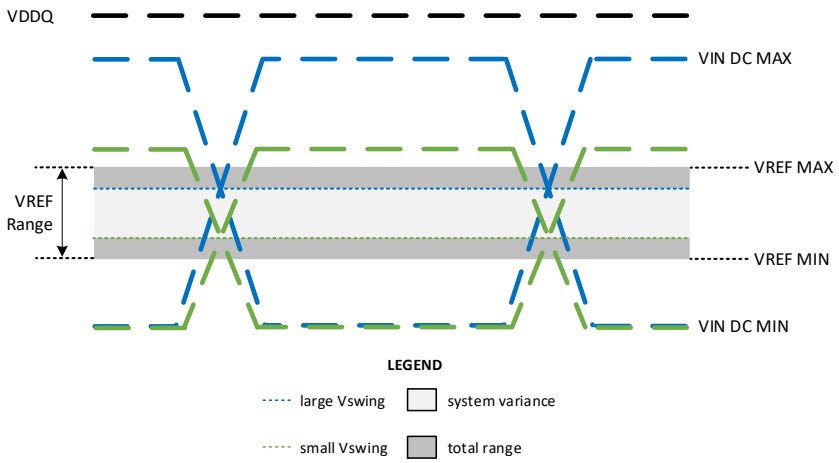


Figure 2.10. VREF Training

2.4.2. Initialization and Training without APB Interface

The APB interface is unavailable when the Enable APB interface attribute is unchecked. In this case, the init_start_i and trn_opr_i signals are available for starting and configuring the LPDDR4 initialization and training. The trn_opr_i sets the value of the Training Operation Register (TRN_OP_REG), controlling which specific training steps to perform based on the asserted bits. For hardware implementation, set trn_opr_i=0xFF to perform command bus training, write leveling, read training, and write training procedures. For simulation purposes, shorten the initialization and training sequences by setting trn_opr_i=0x00. Refer to the Register Description section of this user guide for more details.

To start the initialization and training sequence of the LPDDR4 SDRAM follow these steps:

1. Set init_start_i=0 while the Memory Controller IP is in reset or while pll_lock_o=0.
2. Set init_start_i=1 and maintain this value until init_done_o asserts.
3. Once init_done_o=1, set init_start_i=0 to hold the training CPU in reset to save power.

2.5. LPDDR4 Operation Description

This section provides detailed information on the various operations and features supported by the LPDDR4 Memory Controller IP.

2.5.1. Write and Read Data Access

Once the init_done_o signal asserts, you can initiate write and read accesses. The types of data accesses supported by the Memory Controller IP depend on the selected data interface protocol. Refer to Table 2.2 and Table 2.3 for a list of all supported AHB-Lite and AXI4 data accesses.

The Memory Controller decodes the memory-mapped address to check if the row and bank addresses are already open in the memory device. When a WRITE/READ command is issued to LPDDR4 Memory, the following commands are issued:

- If the target row and bank are not open, the Memory Controller issues an ACTIVATE command to the LPDDR4 SDRAM to open the row, followed by a WRITE/READ command.
- If a row is open in the current bank and the target row address differs from the open row, the Memory Controller issues a PRECHARGE command to close the open row, followed by an ACTIVATE command to open the target row, and then a WRITE/READ command.
- If the target row is already open, the Memory Controller issues only a WRITE/READ command.

The Memory Controller does not immediately close a row after a WRITE/READ command, it issues a separate PRECHARGE command as needed.

2.5.2. Auto Refresh Support

Ideally, issue REFRESH commands every refresh interval, specified by the Refresh Period attribute. To improve efficiency in scheduling and switching between tasks, the LPDDR4 memory allows postponing a maximum of 8 REFRESH commands throughout the LPDDR4 operation. The Memory Controller has an internal auto refresh generator that sends out a set of consecutive AUTO REFRESH commands to the memory when it reaches the time specified in the following calculation: *Refresh Period* × *Number of Outstanding Refresh*.

For high performance applications, set the *Number of Outstanding Refresh* to the maximum value to increase LPDDR4 bus throughput by minimizing Memory Controller intervention.

2.5.3. Power Saving Feature

The Memory Controller supports power-saving when the Enable Power Down attribute is set. The Memory Controller IP tracks the period of inactivity on the local data bus by monitoring the System Clock (SCLK), the main clock used by the Memory Controller IP. When the period of inactivity on the bus reaches the value set in the *Number of SCLK to enter Self-Refresh from no traffic* attribute, the Memory Controller issues SELF REFRESH and Power-Down Entry commands. This puts the memory into Power-Down mode to save power until a new request is received on the local data bus, at which point the Memory Controller issues SELF REFRESH and Power-Down Exit commands, followed by an additional REFRESH command.

2.5.4. Periodic ZQ Calibration

ZQ Calibration calibrates the output driver impedance across PVT. The Multi-Purpose Command (MPC) initiates two ZQ Calibration modes: ZQCAL START and ZQCAL LATCH. The ZQCAL START command initiates the SDRAM's calibration procedure, while the ZQCAL LATCH command captures the result and loads it into the SDRAM's drivers.

The Memory Controller IP periodically performs ZQ Calibration because voltage and temperature may fluctuate during operation. Users can configure the frequency and duration of ZQ Calibration through the *ZQ Calibration Period* and *ZQ Calibration Start to Latch* attributes.

2.5.5. Temperature Tracking and Extended Temperature Support

LPDDR4 SDRAM devices support temperature tracking, where the device's refresh rate changes based on the operational temperature. When the memory device's temperature is within normal operating conditions, the rate is set to 1x refresh, the default case. As the temperature fluctuates below or above the default case, the Memory Controller decreases or increases the frequency of refreshes, respectively. Note that this change takes effect on the next refresh cycle. The required refresh rate according to temperature is specified in MR4 within LPDDR4 SDRAM. When the Temperature Update Flag (TUF) is set in MR4, it indicates that the refresh rate has changed since the last read issued to MR4. LPDDR4 SDRAM devices also support operation at extended temperatures. When the operational temperature increases beyond normal operation conditions, the refresh frequency begins to increase. When the refresh rate in MR4 is set to 3'b110, timing derating is required for certain commands, which slows performance. For more information regarding temperature derating timing requirements, refer to the LPDDR4 JEDEC Standard.

The Memory Controller IP reads MR4 periodically to track the temperature of the LPDDR4 memory device. Based on the refresh rate set in MR4, the Memory Controller IP issues REFRESH commands at the required frequency. Note that when the refresh rate in MR4 is set to either 3'b000 or 3' b111, the LPDDR4 memory may not work as expected due to exceeding of low/high temperature operating limits.
When the refresh rate is set to 3'b110 in MR4, the Memory Controller accommodates the temperature derating timing requirements when issuing commands to the memory. Users can change the frequency at which the memory's MR4 register is read through the *Temperature Check Period* attribute. The Memory Controller starts counting the temperature check period when the training is completed. When reading MR4, the Memory Controller IP sets STATUS_REG[18:16] to the read refresh rate value, and INT_STATUS_REG[4]=1 when TUF is set in MR4. Refer to the Register Description section of this user guide for more details.

2.5.6. Periodic VT Compensation

The LPDDR4 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 LPDDR4 Soft PHY has a DDR Delay-Locked Loop (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°. This code is used by the LPDDR4 I/O Logic to adjust the DQ delay by 90° with respect to the DQS. The Periodic VT Compensation logic performs the following:

1. Update the DLL code such that the new DLL code value is still 45° based on the drifted delay.
2. Modify the write DQ/DMI delay based on the new DLL code. The LPDDR4 DQ/DMI signal has non-DLL-compensated delay, and this function takes care of that.
3. Both steps 1 and 2 above are done during refresh to avoid glitch on the DQ/DMI signals. For dual rank, these steps are performed during Rank 0 refresh and Rank 1 has no active/ongoing DDR access at the PHY layer.

The periodic VT compensation function is shown in Figure 2.11.

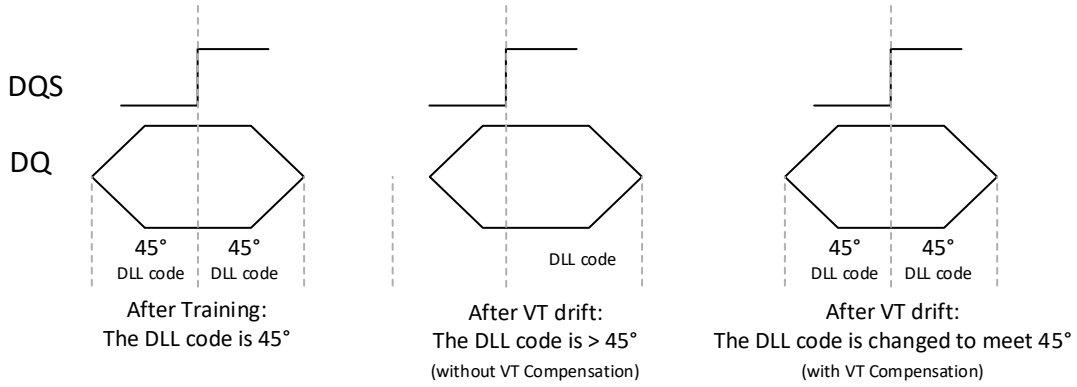


Figure 2.11. Periodic VT compensation

3. IP Parameter Description

The configurable attributes of the LPDDR4 Memory Controller for Nexus Devices are shown and described in Table 3.1. Configure these attributes through the IP Catalog's Module/IP Block Wizard of the Lattice Radiant software. Refer to the Designing and Simulating the IP section of this user guide for information on configuring and generating the Memory Controller IP. Use the **Calculate** button at the bottom to calculate the optimum parameter values for the PLL.

3.1. General

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

Table 3.1. General Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
DDR Interface Type	LPDDR4	LPDDR4	Display only
I/O Buffer Type	LVSTL_I, LVSTL_II	LVSTL_I	—
DDR Command Frequency (MHz)[1]	300, 350, 400, 533	533	—
Gearing Ratio	8:1	8:1	Display only
Enable ECC	Checked, Unchecked	Unchecked	Display only (ECC is not yet supported)
Enable Power Down	Checked, Unchecked	Unchecked	—
Enable DBI	Checked, Unchecked	Unchecked	—
Read Latency	Calculated	N/A	Display only Calculated based on selection for *DDR Command Frequency*
Write Latency	Calculated	N/A	Display only Calculated based on selection for *DDR Command Frequency*
Enable Internal RISC-V CPU	Checked, Unchecked	Checked	Display only (Disabling of Internal RISC-V CPU is not yet supported)

Note:

1. Only devices with speed grades of 8 and 9 support DDR command frequencies up to 533 MHz.

Table 3.2. Clock Settings Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Enable PLL	Checked	Checked	Display only
PLL Reference Clock from Pin	Checked, Unchecked	Checked	—
I/O Standard for Reference Clock	LVSTLD_I, LVSTLD_II	LVSTL_I	Display only Based on selection for *I/O Buffer Type*
DDR Command Actual Frequency (MHz) [6.25 – 800]	—	533	Display only
DDR Command Actual Frequency (MHz)	Calculated	N/A	Display only Based on selection for *DDR Command Frequency*

Table 3.3. Memory Configuration Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
DDR Density (per Channel)	2 Gb, 4 Gb, 8 Gb, 16 Gb	4 Gb	—
DDR Bus Width(BUS_WIDTH)	16, 32	32	—
Number of Ranks[1]	1, 2	1	DDR command frequency
Number of DDR Clocks (CK_WIDTH)	1, 2	1	—
Number of Chip Selects (CS_WIDTH)	1, 2	1	Display only Calculated based on selection for *Number of Ranks*

Note:

1. Dual-Rank supports DDR command frequencies lower than 533 MHz only.

Table 3.4. Local Data Bus Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Local Data Bus Type	AHBL	AHB-Lite	Display only For IP Core v1.x.x
	AXI4	AXI4	For IP Core v2.x.x
Address Width	Calculated	N/A	Display only Calculated based on selection for *DDR Density*
Data Width (AHBL_DATA_WIDTH or AXI_DATA_WIDTH)	Calculated	N/A	Calculated based on selection for *Local Data Bus Type*
ID Width	1, 2, 3, 4, 5, 6, 7, 8	4	For IP Core v2.x.x
Maximum Burst Length	64, 128, 256	64	For IP Core v2.x.x
Number of Outstanding Writes	1, 2, 3, 4	4	For IP Core v1.x.x
Write Ordering Queues	1, 2, 3, 4	2	For IP Core v2.x.x
Number of Outstanding Reads	1	1	Display only For IP Core v1.x.x
Read Ordering Queues	1, 2, 3, 4	2	For IP Core v2.x.x
Enable Local Bus Clock	Checked/Unchecked	Unchecked	Display only For IP Core v1.x.x
		Checked	For IP Core v2.x.x
Enable APB interface	Checked/Unchecked	Checked	For IP Core v2.x.x

Table 3.5. General Definitions

Attribute	Description
General Group	
DDR Interface Type	Specifies the SDRAM Memory interface: LPDDR4.
I/O Buffer Type	I/O Standard for the memory interface signals.
DDR Command Frequency (MHz)	Speed at which the memory controller will issue commands to the memory device.
Gearing Ratio	Specifies the ratio relationship between the DDR data speed and the memory controller speed.
Enable ECC	Enables error-correction code (ECC) for single-bit error correction and double-bit error detection (not yet supported).
Enable Power Down	Enables the memory controller automatically place the memory device into power-down mode after a specified number of idle controller clock cycles.
Enable DBI	Enables data bus inversion (DBI) for better signal integrity and read/write margins.
Read Latency	Specifies the delay from issuing of a read command to receiving read data from SDRAM.
Write Latency	Specifies the delay from issuing of a write command to providing of write data to SDRAM.

Attribute	Description
Enable Internal RISC-V CPU	Enables RISC-V subsystem to support initialization and training of the memory device (disabling is not yet supported).
Clock Settings Group	
Enable PLL	Enables PLL.
PLL Reference Clock from Pin	Indicates if provided PLL reference clock originates from a pin.
I/O Standard for Reference Clock	Specifies the I/O standard for the PLL reference clock.
Ref Clock (MHz)	Indicates the PLL reference clock speed (100 MHz).
DDR Command Actual Frequency (MHz)	Specifies the actual operating frequency of the memory interface (calculated by the PLL – includes tolerance).
Memory Configuration Group	
DDR Density (per Channel)	Density of SDRAM (No. of chips on memory module). Only DDR Density with power of 2 is supported.
DDR Bus Width	Total number of data pins in memory interface.
Number of Ranks	Specifies number of ranks in memory interface.
Number of DDR Clocks	Specifies the number of CK/CK# clock pairs to be driven to SDRAM.
Number of Chip Selects	Specifies the number of chip select (CS) signals to be driven to SDRAM, dependent on the number of ranks selected.
Local Data Bus Group	
Local Data Bus Type	Indicates bus for local data interface: IP Core v1.x.x: AHB-Lite IP Core v2.x.x: AXI4
Address Width	Specifies number of address pins in memory interface, dependent on the SDRAM density.
Data Width	Indicates data width for local data bus: BUS_WIDTH × 4 if using AHB-Lite BUS_WIDTH × 8 if using AXI4
ID Width	Indicates bit width of AXI4 ID. Only for IP Core v2.x.x.
Maximum Burst Length	Indicates maximum number of data transfers that occur in a burst.
Number of Outstanding Writes / Write Ordering Queues	The number of outstanding writes/reads and the write/read ordering queues process write and read data requests. The higher the value, the better bandwidth on the interface since gaps between accesses are lessened at the cost of consuming more LUTs.
Number of Outstanding Reads / Read Ordering Queues	
Enable Local Bus Clock	Enables clock domain crossing logic for the local data bus. IP Core v1.x.x: disabled by default (fixed). IP Core v2.x.x: enabled by default (configurable).
Enable APB interface	Enables APB interface. Should be enabled if target application includes a CPU. When disabled, initialization and training of SDRAM should be handled through init_start_i and trn_opr_i signals. Refer to the Initialization and Training without APB Interface section of this user guide for more information.

3.2. Memory Device Timing

This 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 varies for each DDR Command Frequency value and follow the JESD209-4C SDRAM standard. You may need to manually adjust these values based on the selected LPDDR4 device.

Table 3.6. Memory Device Timing Setting Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Manual Timing Adjust Enable	Checked/Unchecked	Unchecked	—
TRCD (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRAS (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRPPB (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TWR (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRTP (tCLK)	8 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TCCD (tCLK)	8 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
MWR2MWR (tCLK)	32 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRRD (tCLK)	6 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRFC (tCLK)	24 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TFAW (tCLK)	40 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TZQLAT (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TMRR (tCLK)	8 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TMRD (tCLK)	10 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRPAB (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TRTW (tCLK)	21 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TDQSS (tCLK)	Integer	1	Enabled when *Manually Adjust* is Checked
TRD2PRE (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TWR2PRE (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TXP (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked
TXPSR (tCLK)	4 – 65,536	Calculated	Enabled when *Manually Adjust* is Checked

Table 3.7. Periodic Event Setting Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
Number of SCLK to enter Self-Refresh from no traffic	Integer	5,000	Enabled when *Enable Power Down* is Checked.
Refresh Period (tSCLK)	Calculated	Calculated	Default value is calculated based on selection for *DDR Command Frequency.*
Number of Outstanding Refresh	1	1	Display only For IP Core v1.x.x
	1, 2, 3, 4, 5, 6, 7	7	For IP Core v2.x.x
ZQ Calibration Period (sec)	Integer	32	—
ZQ Calibration Start to Latch (usec) (TZQCAL)	Integer	1	—
Temperature Check Period (sec)	Integer	32	—

Table 3.8. Memory Device Timing Definitions

Attribute	Description
Memory Device Timing Setting Group[1]	
Manual Timing Adjust Enable	Lets you manually set any of the memory timing parameters.
TRCD	Indicates the delay between the ACTIVATE command (RAS) and the internal access to data (CAS).
TRAS	Indicates how long memory must wait after an ACTIVATE command before a PRECHARGE command can be issued to close the row.
TRPPB	Row PRECHARGE time for a single bank.
TWR	Specifies the amount of clock cycles needed to complete a WRITE before a PRECHARGE command can be issued.
TRTP	Internal READ to PRECHARGE delay.
TCCD	Minimum time between two READ/WRITE (CAS) commands (burst length / 2).
MWR2MWR	Masked WRITE to masked WRITE.
TRRD	Minimum time interval between two ACTIVATE commands to different banks.
TRFC	Indicates how long memory must wait after a REFRESH command before an ACTIVATE command can be accepted by memory.
TFAW	Specifies the period duration during which only four banks can be active.
TZQCAL	ZQ calibration time from START to LATCH command.
TMRR	Mode register READ command period.
TMRD	Minimum time between two MRS commands.
TRPAB	Row PRECHARGE time for all banks.
TRTW	READ to WRITE command delay.
TDQSS	Describes skew between the output data strobe with respect to the memory clock for writes (DQS to CK).
TRD2PRE	READ to PRECHARGE time.
TWR2PRE	WRITE to PRECHARGE time.
TXP	Power-down exit latency to next valid command.
TXPSR	Power-down exit latency to next self-refresh.
Periodic Event Setting Group	
Number of SCLK to enter Self-Refresh from no traffic[2]	The memory controller puts the memory in self-refresh when there is no traffic for the specified number of SCLK cycles.
Refresh Period	Specifies the number of SCLK cycles between refresh commands.
Number of Outstanding Refresh	Specifies the maximum number of outstanding refresh commands. Refer to the Auto Refresh Support section of this User Guide for more information.
ZQ Calibration Period	Indicates period for performing ZQ Calibration in seconds.
ZQ Calibration Start to Latch	Indicates time from start to latch during ZQ Calibration in microseconds.
Temperature Check Period[2]	Indicates period for reading the temperature register (MR4) in LPDDR4 memory in seconds.

Notes:

1. The Memory Device Timing tab lists the memory device timing parameters according to the JESD209-4C SDRAM standard. Refer to the memory device data sheet for detailed descriptions and allowed values for these parameters.
2. The number of SCLK cycles required to enter Self-Refresh from a no-traffic state must be shorter than the configured Temperature Check Period when the Power Down feature is enabled.

3.3. Training Settings

This section describes the parameters available in the Training Settings tab of the IP parameter editor, which applies only to IP Core v2.x.x.

Table 3.9. Training Settings Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
DDR Clock Delay Value	0 – 127	50	Default value is calculated based on selection for *DDR Command Frequency.*
Initial MC DQ_VREF Value	0 – 127	49	—
Initial Memory CA_VREF Value	0 – 127	30	—
Initial Memory DQ_VREF Value	0 – 127	25	—

Table 3.10. MC I/O Settings Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
CK/CS Slew Rate	Fast, Med	Fast	Default value is calculated based on selection for *DDR Command Frequency.*
Command Address Slew Rate	Fast, Med	Fast	—
DQS Slew Rate	Fast, Med	Fast	Default value is calculated based on selection for *DDR Command Frequency.*
DQ/DMI Slew Rate	Fast, Med	Fast	Default value is calculated based on selection for *DDR Command Frequency.*
MC CA Drive Impedance	120 Ω, 60 Ω, 48 Ω, 40 Ω, 34 Ω	40 Ω	—
MC DQ Drive Impedance	120 Ω, 60 Ω, 48 Ω, 40 Ω, 34 Ω	40 Ω	—
MC ODT Value	120 Ω, 80 Ω, 60 Ω, 48 Ω, 40 Ω	60 Ω	—

Table 3.11. Memory ODT Settings Attributes

Attribute	Selectable Values	Default	Dependency on Other Attributes
CA_ODT Value	Disable, RZQ/1, RZQ/2, RZQ/3, RZQ/4, RZQ/5, RZQ/6	RZQ/2	—
DQ_ODT Value	Disable, RZQ/1, RZQ/2, RZQ/3, RZQ/4, RZQ/5, RZQ/6	RZQ/2	—
SOC_ODT Value	Disable, RZQ/1, RZQ/2, RZQ/3, RZQ/4, RZQ/5, RZQ/6	RZQ/6	—
CK_ODT Override Enable	Checked, Unchecked	Unchecked	—
CS_ODT Override Enable	Checked, Unchecked	Unchecked	—
CA_ODT Override Disable	Checked, Unchecked	Unchecked	—
ODT-CS/CA/CLK Disable	Checked, Unchecked	Unchecked	—
Pull-Down Drive Strength	Disable, RZQ/1, RZQ/2, RZQ/3, RZQ/4, RZQ/5, RZQ/6	RZQ/6	—

Table 3.12. Trained Values Attributes

Attributes	Selectable Values	Default	Dependency on Other Attributes
DQS<0,1> Trained Write Leveling Delay	0 – 127	50	Default value is calculated based on selection for *DDR Command Frequency.*
DQS<2,3> Trained Write Leveling Delay	0 – 127	50	Default value is calculated based on selection for *DDR Command Frequency.* Visible only when *DDR Bus Width* >= 32
DQS<0,1> Trained RDCLKSEL	0 – 15	9	Default value is calculated based on selection for *DDR Command Frequency*.
DQS<2,3> Trained RDCLKSEL	0 – 15	9	Default value is calculated based on selection for *DDR Command Frequency.* Visible only when *DDR Bus Width* >= 32
Trained CS Delay	0 – 127	50	Default value is calculated based on selection for *DDR Command Frequency.*
Trained CA Delay	0 – 127	50	Default value is calculated based on selection for *DDR Command Frequency.*
Trained DQSBUF Read Delay	0 – 127	24	—
Trained DQSBUF Read Sign	0, 1	0	—
Trained Write DQ/DBI delay	0 – 127	56	Default value is calculated based on selection for *DDR Command Frequency.*
Trained Write Latency	0 – 18	4[1]	—
Trained Read Latency	0 – 40	9[2]	Default value is calculated based on selection for *DDR Command Frequency* and *Enable DBI.*

Notes:

1. For Lattice Radiant version 2025.1.1 and later, the default value is 4. For earlier revisions, the default value is 7.
2. For Lattice Radiant version 2025.1.1 and later, the default value is 9 or 8, depending on whether the DDR Command Frequency is set to 11 or 10 with DBI enabled. Values of 12 or 11 are retained in earlier revisions.

Table 3.13. Training Settings Definition

Attribute	Description
Training Settings Group[1]	
DDR Clock 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.
Initial MC DQ_VREF Value	Initial DQ VREF value for the FPGA I/O. This value is translated to an internal reference voltage where the input DQ/DBI signal will be compared to determine a 0 and 1 value.
Initial Memory CA_VREF Value	Initial CA VREF value that is written to MR12 in LPDDR4 memory.
Initial Memory DQ_VREF Value	Initial DQ VREF value that is written to MR14 in LPDDR4 memory.
MC I/O Settings	
CK/CS Slew Rate[2]	Specifies the CK/CS driver strength.
Command Address Slew Rate[2]	Specifies the CA driver strength.
DQS Slew Rate[2]	Specifies the DQS driver strength.
DQ/DMI Slew Rate[2]	Specifies the DQ/DMI driver strength.
MC CA Drive Impedance	Memory controller CA I/O impedance to reach VDDQ threshold.
MC DQ Drive Impedance	Memory controller DQ I/O impedance to reach VDDQ threshold.
MC ODT Value	Memory controller termination resistance value at FPGA.
Memory ODT Settings Group	
CA_ODT Value	Memory side CA ODT resistance value.
DQ_ODT Value	Memory side DQ ODT resistance value.
SoC_ODT Value	Memory controller ODT resistance value correlating to ZQ calibration.
CK_ODT Override Enable	Overrides the default CK ODT value for the non-terminating rank when option is checked. Refer to MR22 in LPDDR4 device datasheet for more details.

Attribute	Description
CS_ODT Override Enable	Overrides the default CS ODT value for the non-terminating rank when option is checked. Refer to MR22 in LPDDR4 device datasheet for more details.
CA_ODT Override Disable	Disables termination for CA when option is checked. Refer to MR22 in LPDDR4 device datasheet for more details.
ODT-CS/CA/CLK Disable	Disable ODT for CS/CA/CLK when option is checked.
Pull-Down Drive Strength	Default pull-down drive.
Trained Values Group[3]	
DQS<0,1,2,3,4,5,6,7> Trained Write Leveling Delay	Specifies delay value to be programmed to the PHY when TRN_OP_REG[2] = 0. Each step adjusts the write DQS delay by 12.5 ps.
DQS<0,1,2,3,4,5,6,7> Trained RDCLKSEL	Specifies the delay value to be programmed to the PHY when TRN_OP_REG[3] = 0. Each step adjusts the internal read DQS delay by a 450-phase shift from the incoming DQS.
Trained CS Delay	Specifies the CS delay. It is recommended to use the default value and to only adjust when CBT fails after already adjusting the Initial MC DQ_VREF Value. Each step adjusts the CS delay by 12.5 ps.
Trained CA Delay	Specifies the delay value to be programmed to the PHY when TRN_OP_REG[1] = 0. Each step adjusts the CA delay by 12.5 ps.
Trained DQSBUF Read Delay	Specifies the delay value to be programmed to the PHY when TRN_OP_REG[3] = 0. Each step adjusts the internal read DQS delay by a small delay based on the DDR clock frequency.
Trained DQSBUF Read Sign	Specifies the read delay value for DQSBUF to either increment (0) or decrement (1).
Trained Write DQ/DBI Delay	Specifies the delay value to be programmed to the PHY when TRN_OP_REG[4] = 0. Each step adjusts the write DQ/DBI delay by 12.5 ps.
Trained Write Latency	Specifies the write latency setting of the PHY that passes write training. Ideally, this is equal to *Write Latency* + 1.
Trained Read Latency	Specifies the read latency setting of the PHY that passes read training. Ideally, this is equal to *Read Latency* + 1.

Notes:

1. Modify these attributes only if you encounter an error during LPDDR4 memory training.
2. It is recommended to use Fast Slew Rate to meet JEDEC's slew rate requirement. The Med Slew rate is only provided when power is a serious concern and not meeting JEDEC requirement is acceptable to you.
3. Use these attributes only if you skip LPDDR4 training during simulation. Although you can skip training on hardware when set correctly, it is not recommended because the delays and VREFs will not be adjusted according to environmental temperature variations.

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 on the constraints file, which applies only to IP Core v2.x.x

Table 3.14. Example Design

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

4. Signal Description

The section covers the input and output signals of the LPDDR4 Memory Controller for Nexus devices. The available signals depend on the selected configuration of the LPDDR4 Memory Controller IP.

4.1. Clock and Reset

This section describes the interface ports for clock and reset in the LPDDR4 Memory Controller IP.

Table 4.1. Clock and Reset Port Definitions

Port Name	I/O	Width	Description
pll_refclk_i	In	1	PLL reference clock input.
pll_rst_n_i	In	1	PLL reset active low.
pclk_i	In	1	Clock for APB interface, training CPU and control logic of the internal PLL. This clock is independent of sclk_o, since sclk_o stops during clock frequency changes. Supported frequency range: 50 MHz to 125 MHz.
preset_n_i	In	1	Asynchronous active low reset for APB interface. This reset must be de-asserted synchronous to pclk_i.
pll_lock_o	Out	1	PLL lock output indicating when PLL is locked.
sclk_o	Out	1	System clock. This is ¼ of the DDR clock frequency and is the main clock of the LPDDR4 Memory Controller IP.
rst_n_i	In	1	Asynchronous active low reset. When asserted, output ports and registers are forced to their reset values. The LPDDR4 Memory Controller IP implements logic to de-assert the internal reset synchronous to the internal clocks after rst_n_i de-asserts. After rst_n_i deassertion, wait for a minimum of 5 pclk_i before accessing the registers.
aclk_i	In	1	AXI4 interface clock. Available in IP Core v2.x.x. This is only available when *Enable Local Bus Clock* attribute is checked.
areset_n_i	In	1	Asynchronous active low reset for AXI4 interface. Available in IP Core v2.x.x. This reset must be de-asserted synchronous to aclk_i. This is only available when *Enable Local Bus Clock* attribute is checked.

4.2. Interrupts and Initialization/Training

This section describes the interface ports for interrupts and initialization/training control in the LPDDR4 Memory Controller IP.

Table 4.2. Interrupts and Initialization/Training Port Definitions

Port Name	I/O	Width	Description
irq_o	Out	1	Interrupt signal. Reset value is 1'b0.
init_start_i	In	1	Starts the memory initialization and training according to trn_opr_i. This signal is available when *Enable APB interface* attribute is unchecked.
init_done_o	Out	1	Indicates completion of initialization and training and the memory is available for access through the data interface.
trn_opr_i	In	8	Sets the TRN_OP_REG, which specifies the training steps to perform. Refer to the Training Operation Register (TRN_OP_REG) (0x20) section of this user guide for more information. This signal is available when *Enable APB interface* attribute is unchecked.
trn_err_o	Out	1	Indicates failure in training.

4.3. AHB-Lite Data Interface

This section describes the data interface port when you set the Local Data Bus Type attribute to AHB-Lite in the LPDDR4 Memory Controller IP. These ports are available only when using IP Core v1.x.x.

Refer to the AMBA AHB Protocol Specification for a description of these signals. Table 2.3 describes the transactions allowed on the AHB-Lite interface to the Memory Controller.

Table 4.3. AHB-Lite Interface Port Definitions

Port Name	I/O	Width	Description
ahbl_hsel_i	In	1	Start transaction.
ahbl_hready_i	In	1	Ready signal from AHB-Lite interconnect. When connecting to an AHB-Lite manager directly, connect this to ahbl_hreadyout_o or set to 1.
ahbl_haddr_i[1]	In	AHBL_ADDR_WIDTH	Request address.
ahbl_hburst_i	In	3	Burst type.
ahbl_hsize_i	In	3	Indicates the size of the transfer.
ahbl_hmastlock_i	In	1	AHB-Lite HMASTLOCK signal – not used for this IP.
ahbl_hprot_i	In	4	AHB-Lite HPROT signal – not used for this IP.
ahbl_htrans_i	In	2	Transfer type IDLE, BUSY, NONSEQ, SEQ.
ahbl_hwrite_i	In	1	1 = Write, 0 = Read
ahbl_hwdata_i[1]	In	AHBL_DATA_WIDTH	The write data for write transactions.
ahbl_hreadyout_o	Out	1	Read response valid.
ahbl_hrdata_o[1]	Out	AHBL_DATA_WIDTH	Read data response.
ahbl_hresp_o	Out	1	Read response state.

Note:

1. Calculate the bit width of ahbl_haddr_i/ahbl_hwdata_i/ahbl_hrdata_o based on the *DDR density* and *DDR Bus Width* attributes in Table 3.3. Refer to Table 5.4 for the address mapping.

4.4. AXI4 Data Interface

This section describes the data interface port when the Local Data Bus Type attribute is set to AXI4 in the LPDDR4 Memory Controller IP. These ports are available only when using IP Core v2.x.x.

Refer to the AMBA AXI Protocol Specification for a description of these signals. The transactions allowed on the AXI4 interface to the Memory Controller are described in Table 2.4.

Table 4.4. AXI4 Interface Port Definitions

Port Name	I/O	Width	Description
axi_arid_i	In	AXI_ID_WIDTH	AXI4 read address channel: Read address ID signal
axi_araddr_i[1]	In	AXI_ADDR_WIDTH	AXI4 read address channel: Read address signal
axi_arlen_i	In	8	AXI4 read address channel: Burst length signal Supports up to burst length 64 only, it is prohibited to issue more than this
axi_arsize_i	In	3	AXI4 read address channel: Burst size signal
axi_arburst_i	In	2	AXI4 read address channel: Burst type signal Only INCR is supported
axi_arqos_i	In	4	AXI4 read address channel: Quality of service signal This signal is currently unused
axi_arvalid_i	In	1	AXI4 read address channel: Read address valid signal
axi_arready_o	Out	1	AXI4 read address channel: Read address ready signal
axi_rid_o	Out	AXI_ID_WIDTH	AXI4 read data channel: Read ID tag signal
axi_rdata_o[1]	Out	AXI_DATA_WIDTH	AXI4 read data channel: Read data signal
axi_rresp_o	Out	2	AXI4 read data channel: Read response signal

Port Name	I/O	Width	Description
axi_rlast_o	Out	1	AXI4 read data channel: Read last signal
axi_rvalid_o	Out	1	AXI4 read data channel: Read valid signal
axi_rready_i	In	1	AXI4 read data channel: Read ready signal
axi_awid_i	In	AXI_ID_WIDTH	AXI4 write address channel: Write address ID signal
axi_awaddr_i[1]	In	AXI_ADDR_WIDTH	AXI4 write address channel: Write address signal
axi_awlen_i	In	8	AXI4 write address channel: Burst length signal
axi_awsize_i	In	3	AXI4 write address channel: Burst size signal
axi_awburst_i	In	2	AXI4 write address channel: Burst type signal
axi_awqos_i	In	4	AXI4 write address channel: Quality of service signal
axi_awvalid_i	In	1	AXI4 write address channel: Write address valid signal
axi_awready_o	Out	1	AXI4 write address channel: Write address ready signal
axi_wdata_i[1]	In	AXI_DATA_WIDTH	AXI4 write data channel: Write data signal
axi_wstrb_i	In	AXI_STRB_WIDTH	AXI4 write data channel: Write strobe signal
axi_wlast_i	In	1	AXI4 write data channel: Write last signal
axi_wvalid_i	In	1	AXI4 write data channel: Write valid signal
axi_wready_o	Out	1	AXI4 write data channel: Write ready signal
axi_bid_o	Out	AXI_ID_WIDTH	AXI4 write response channel: Response ID tag signal
axi_bresp_o	Out	2	AXI4 write response channel: Write response signal
axi_bvalid_o	Out	1	AXI4 write response channel: Write response valid signal
axi_bready_i	In	1	AXI4 write response channel: Response ready signal

Note:

1. The bit width of axi_araddr_i/axi_rdata_o/axi_awaddr_i/axi_wdata_i is calculated based on the *DDR density* and *DDR Bus Width* attributes in Table 3.3. Refer to Table 5.4 for the address mapping.

4.5. APB Register Interface

This section describes the configuration interface port when the *Enable APB interface* attribute is checked in the LPDDR4 Memory Controller IP. Refer to the AMBA APB Protocol Specification for a description of these signals.

Table 4.5. APB Interface Port Definitions

Port Name	I/O	Width	Description
apb_psel_i	In	1	APB Select signal. Indicates that the subordinate device is selected, and a data transfer is required.
apb_paddr_i	In	12	APB Address signal.
apb_pwdata_i	In	32	APB Write data signal. Bits [31:8] are not used.
apb_pwrite_i	In	1	APB Direction signal. 1 = Write, 0 = Read
apb_penable_i	In	1	APB Enable signal. Indicates the second and subsequent cycles of an APB transfer.
apb_pready_o	Out	1	APB Ready signal. Indicates transfer completion. Subordinate uses this signal to extend an APB transfer. Reset value is 1'b0.
apb_pslverr_o	Out	1	APB Error signal. Indicates a transfer failure. This signal is tied to 1'b0.
apb_prdata_o	Out	32	APB Read data signal.

4.6. LPDDR4 Memory Interface

This section describes the interface ports for LPDDR4 SDRAM.

Table 4.6. LPDDR4 Interface Port Definitions

Port Name	I/O	Width	Description
ddr_ck_o[1]	Out	CK_WIDTH	LPDDR4 CK signal.
ddr_cke_o[1]	Out	CS_WIDTH	LPDDR4 CKE signal.
ddr_cs_o[1]	Out	CS_WIDTH	LPDDR4 CS signal.
ddr_ca_o	Out	6	LPDDR4 CA signal.
ddr_reset_n_o	Out	1	Memory reset signal.
ddr_dq_io[1]	In/Out	BUS_WIDTH	LPDDR4 DQ signal.
ddr_dqs_io[1]	In/Out	DQS_WIDTH	LPDDR4 DQS signal.
ddr_dmi_io[1]	In/Out	DQS_WIDTH	LPDDR4 DMI signal.

Note:

1. The bit width of the SDRAM Memory Interface signals is defined based on the attributes listed in Table 3.3.

5. Register Description

This section describes the user-accessible registers in the LPDDR4 Memory Controller IP. Some registers are reserved for internal CPU usage and should not be written to. Writing to these registers may cause failures during initialization and training operations.

Table 5.1. Summary of LPDDR4 Memory Controller IP Registers

Offset	Register Name	Access Type	Description
0x00	FEATURE_CTRL_REG	RW	Feature Control Register
0x04	RESET_REG	RW	Reset Register
0x08	SETTINGS_REG	RW	Settings Register
0x0C	Reserved	—	Reserved
0x10	INT_STATUS_REG	RW1C	Interrupt Status Register
0x14	INT_ENABLE_REG	RW	Interrupt Enable Register
0x18	INT_SET_REG	WO	Interrupt Set Register
0x1C	CLK_CHANGE_REG	—	Reserved for use by the internal CPU
0x20	TRN_OP_REG	RW	Training Operation Register
0x24	STATUS_REG	RW	Status Register
0x28 to 0xA4	For internal use	—	Reserved for use by the internal CPU
0xA8 onwards	Reserved	—	Reserved

Table 5.2. Register Access Type Definitions

Access Type	Behavior on Read Access	Behavior on Write Access
RO	Returns register value	Ignores write access
WO	Returns 0	Updates register value
RW	Returns register value	Updates register value
RW1C	Returns register value	Writing 1’b1 on register bit clears the bit to 1’b0 Writing 1’b0 on register bit is ignored
RSVD	Returns 0	Ignores write access

5.1. Feature Control Register (FEATURE_CTRL_REG) (0x00)

The Feature Control Register reflects the modes of operation specified by the attributes selected during IP configuration. You set these attributes during IP configuration and cannot modify them at runtime. The CPU reads this register to identify the modes of operation.

Table 5.3. Feature Control Register

Field	Name	Access	Width	Reset
[31:17]	reserved	RSVD	15	—
[16]	num_ranks	RO	1	Number of Ranks
[15:12]	ddr_width	RO	4	DDR Bus Width
[11:8]	ddr_type	RO	4	DDR Interface Type
[7:4]	addr_translation	RO	4	0
[3]	gear_ratio	RO	1	Gearing Ratio
[2]	power_down_en	RO	1	Enable Power Down
[1]	dbi_en	RO	1	Enable DBI
[0]	reserved	RSVD	1	—

num_ranks

- The num_ranks specifies the number of ranks for the LPDDR4 interface:
  - 0 – Single-rank
  - 1 – Dual-rank

ddr_width

- The ddr_width specifies the bit width of the DDR data bus as follows:
  - 0 – DDR Bus Width = 8 bits
  - 1 – DDR Bus Width = 16 bits
  - 2 – reserved
  - 3 – DDR Bus Width = 32 bits
  - 4 – reserved
  - 5 – reserved
  - 6 – reserved

ddr_type

- The ddr_type is fixed at value 1, corresponding to LPDDR4 as the standard implemented by the Memory Controller IP Core.

addr_translation

- The address translation specifies the mapping of the address bits of the local memory-mapped address and the memory address in terms of row, column, bank, and rank. Only one address translation scheme (addr_translation=0) is currently supported; refer to Table 5.4 and Table 5.5 for details.

gear_ratio

- The gear_ratio is fixed at value 1, corresponding to 8:1 gearing, and specifies the ratio of the DDR clock domain to the system clock domain as ddr_ck_o frequency = 4 × sclk_o frequency.

power_down_en

- The pwr_down_en enables the power saving mode by putting the memory in self-refresh when there is no traffic for sclk_o cycles as specified in the *Number of SCLK to enter Self-Refresh from no traffic* attribute.

dbi_en

- The dbi_en enables the Data Bus Inversion function to reduce the toggling of DDR data signals, thus improving the signal integrity and reducing dynamic power consumption as specified in the *Enable DBI* attribute.

Table 5.4. Address Mapping for addr_translation=0

Memory Address	Size	Local Address Map	
Rank	NUM_RANKS = 1	Local address mapping within each rank remains the same.	
Row	ROWW = refer to memory device data sheet	ROW_H = ROW_L + ROWW-1 ROW_L = BANK_H + 1	Addr[ROW_H: ROW_L]
Bank	BANKW = 3	BANK_H = COL_H +BANKW -1 BANK_L = COL_H + 1	Addr[BANK_H: BANK _L]
Column	COLW = 10	COL_H = COL_L + COLW – 1 COL_L = OFFSETW	Addr[COL_H: COL_L]
Offset	If *DDR Bus Width* == 64: OFFSETW = 3 If *DDR Bus Width* == 32: OFFSETW = 2 If *DDR Bus Width* == 16: OFFSETW = 1 If *DDR Bus Width* == 8: OFFSETW = 0	N/A	

Table 5.5. Address Mapping Example for single rank

Memory Address	Example User Value	Actual Line Size	Local Address Map
Offset	DDR Bus Width = 32	2	*addr_i[1:0]
Column Width (COLW)	10 (Fixed for LPDDR4)	10	*addr_i[11:2]
Bank Width (BANKW)	3 (Fixed for LPDDR4)	3	*addr_i[14:12]
Row Width (ROWW)	DDR Density = 4	15	*addr_i[29:15]
Rank Width (RANKW)	Number of Ranks = 1	0	--
Total Local Address Line Size		30	*addr_i[29:0]

5.2. Reset Register (RESET_REG) (0x04)

The Reset Register controls the reset of the internal CPU and Training Engine, both of which reset at power-on. The host should de-assert the reset to the internal CPU and Training Engine to begin memory initialization and training. Upon training completion, the internal CPU sets the trn_eng_rst_n signal low (RESET_REG[0]=0) to place the Training Engine in reset to save power. When the *Enable APB interface* attribute is unchecked, the init_start_i signal controls the internal CPU reset (RESET_REG[1]). This register does not reset the Configuration Set Registers (CSRs).

Table 5.6. Reset Register

Field	Name	Access	Width	Reset
[31:2]	reserved	RSVD	30	—
[1]	cpu_reset_n	RW	1	0
[0]	trn_eng_rst_n	RW	1	0

5.3. Settings Register (SETTINGS_REG) (0x08)

The Settings Register controls the DDR write and read latencies according to the attributes selected during IP configuration.

Table 5.7. Settings Register

Field	Name	Access	Width	Reset
[31:28]	reserved	RSVD	4	—
[27:16]	clk_freq	RO	12	1
[15:12]	reserved	RSVD	4	—
[11:8]	read_latency	RW	4	Read Latency
[7:4]	reserved	RSVD	4	—
[3:0]	write_latency	RW	4	Write Latency

clk_freq

- The clk_freq specifies the DDR command clock frequency in MHz.

read_latency

- The read_latency specifies the number of DDR clock cycles from read command to the first read data.

write_latency

- The write_latency specifies the number of DDR clock cycles from write command to the first write data.

5.4. Interrupt Status Register (INT_STATUS_REG) (0x10)

Table 5.8 lists all the pending interrupts supported in the Memory Controller IP. When an interrupt bit asserts, it remains asserted until the host clears it by writing 1'b1 to the corresponding bit.

The interrupt status bits are independent of the interrupt enable bits. In other words, status bits may indicate pending interrupts, even though those interrupts are disabled in the Interrupt Enable Register. User logic that handles interrupts should mask (bitwise and logic) the contents of INT_STATUS_REG and INT_ENABLE_REG to determine which interrupts to service. The irq_o interrupt signal asserts whenever both an interrupt status bit and the corresponding interrupt enable bits are set.

Table 5.8. Interrupt Status Register

Field	Name	Access	Width	Reset
[31:5]	reserved	RSVD	27	—
[4]	temp_change_int	RW1C	1	0
[3:2]	reserved	RSVD	2	—
[1]	trn_err_int	RW1C	1	0
[0]	trn_done_int	RW1C	1	0

temp_change_int

- Temperature Change Interrupt. The Memory Controller periodically reads MR4 of the LPDDR4 memory device according to the *Temperature Check Period* attribute. This interrupt bit asserts when MR4 indicates a temperature change.
  - 0 – No interrupt
  - 1 – Interrupt pending

trn_err_int

- Training Error Interrupt. This Interrupt bit asserts when the Training Engine encounters an error during training. You should read STATUS_REG to determine the specific error.
  - 0 – No interrupt
  - 1 – Interrupt pending

trn_done_int

- Training Done Interrupt. This Interrupt bit asserts when initialization and training is completed successfully.
  - 0 – No interrupt
  - 1 – Interrupt pending

5.5. Interrupt Enable Register (INT_ENABLE_REG) (0x14)

The Interrupt Enable Register lists all configurable interrupts within the Memory Controller IP.

Table 5.9. Interrupt Enable Register

Field	Name	Access	Width	Reset
[31:5]	reserved	RSVD	27	—
[4]	temp_change_en	RW	1	0
[3:2]	reserved	RSVD	2	—
[1]	trn_err_en	RW	1	0
[0]	trn_done_en	RW	1	0

temp_change_en

- Temperature Change Interrupt Enable.
  - 0 – Interrupt disabled
  - 1 – Interrupt enabled

trn_err_en

- Training Error Interrupt Enable.
  - 0 – Interrupt disabled
  - 1 – Interrupt enabled

trn_done_en

- Training Done Interrupt Enable.
  - 0 – Interrupt disabled
  - 1 – Interrupt enabled

5.6. Interrupt Set Register (INT_SET_REG) (0x18)

Table 5.10 shows a summary of the Interrupt Set Register. Writing 1'b1 to a register bit in this register asserts the corresponding interrupt status bits in INT_STATUS_REG. Writing 1'b0 is ignored.

Table 5.10. Interrupt Set Register

Field	Name	Access	Width	Reset
[31:5]	reserved	RSVD	27	—
[4]	temp_change_set	WO	1	0
[3:2]	reserved	RSVD	2	—
[1]	trn_err_set	WO	1	0
[0]	trn_done_set	WO	1	0

temp_change_set

- Temperature Change Interrupt Set.
  - 0 – No Action
  - 1 – Assert the temp_change_int signal (INT_STATUS_REG[4]=1)

trn_err_set

- Training Error Interrupt Set.
  - 0 – No Action
  - 1 – Assert the trn_err_int signal (INT_STATUS_REG[1]=1)

trn_done_set

- Training Done Interrupt Set.
  - 0 – No Action
  - 1 – Assert the trn_done_int signal (INT_STATUS_REG[0]=1)

5.7. Clock Register (CLK_CHANGE_REG) (0x1C)

Table 5.11. CLK_Change Register

Field	Name	Access	Width	Reset
[31:4]	reserved	RSVD	28	—
[3]	pll_lock	RO	1	—
[2]	clk_update	WO	1	0
[1]	dll_update_en	WO	1	0
[0]	ddr_clk_sel	WO	1	0

pll_lock

- Asserts when PLL is locked.

clk_update

- Initiates clock frequency change when asserted. Used by the Training Engine. Do not write 1 to this register.

dll_update_en

- Re-calibrates DLL after clock frequency change when asserted. Used by the Training Engine. Do not write 1 to this register.

ddr_clk_sel

- Indicates target DDR clock frequency for clock frequency change. Used by the Training Engine. Do not write 1 to this register.

5.8. Training Operation Register (TRN_OP_REG) (0x20)

Table 5.12 summarizes the Training Operation Register. This register controls memory initialization and training. Set these register bits to 1'b0 during simulation to shorten the initialization and training procedure. When *Enable APB interface* is unchecked, TRN_OP_REG[7:0] matches the value of the trn_opr_i input signal.

Table 5.12. Training Operation Register

Field	Name	Access	Width	Reset
[31:8]	reserved	RSVD	24	—
[7]	mem_vref_training_en	RW	1	1
[6]	mc_vref_training_en	RW	1	1
[5]	ca_vref_training_en	RW	1	1
[4]	write_trn_en	RW	1	1
[3]	read_trn_en	RW	1	1
[2]	write_lvl_en	RW	1	1
[1]	cbt_en	RW	1	1
[0]	init_en	RW	1	1

mem_vref_training_en

- Enables VREF training on the SDRAM memory as part of write training. VREF training is embedded in the training routine where the VREF values in the GUI are read in as starting values.
  - 0 – VREF training is not included in write training
  - 1 – The Training Engine performs memory controller VREF Training as part of write training

mc_vref_training_en

- Enables VREF training on the memory controller as part of read training. VREF training is embedded in the training routine where the VREF values in the GUI are read in as starting values.
  - 0 – disable VREF training during read training
  - 1 – The Training Engine performs memory controller VREF Training as part of read training

ca_vref_training_en

- Enables VREF training on command and address bus as part of command bus training. VREF training is embedded in the CBT routine; the GUI values are used as starting values.
  - 0 – VREF training is not included in the CBT routine
  - 1 – The Training Engine performs command and address VREF Training as part of command bus training

write_trn_en

- Enables write training during initialization and training. Write training optimizes the write DQ delay with respect to the write DQS to improve the data valid window for write operations. If write training fails for a certain latency, the Training Engine will increase the latency setting by 1 and retry write training.
  - 0 – The Training Engine programs the *Trained Write DQ/DBI delay* and *Trained Write Latency to PHY* attributes, and checks that write and read FIFO access is equal.
  - 1 – The Training Engine performs Write Training.

read_trn_en

- Enables read training during initialization and training. Read training tunes the PHY to capture the incoming read DQS burst according to the read latency setting. If read training fails for a certain latency setting, the Training Engine will increase the latency setting by one and retry read training. Once the read DQS burst is captured properly, the read DQS-DQ timing is trained to improve the read data valid window.
  - 0 – The Training Engine programs the *DQS<0,1,2,3,4,5,6,7> Trained RDCLKSEL, Trained DQSBUF Read Delay/Sign*, and *Trained Read Latency* attributes to the PHY and checks that read access will be successful.
  - 1 – The Training Engine performs Read Training.
    Note: When dual-rank operation is enabled, the read gate/DQS training parts of the read training are performed only on rank 0, while the read deskew training is done on both ranks.

write_lvl_en

- Enables write leveling during initialization and training. Write leveling compensates for CK-DQS timing skews. The Training Engine performs write leveling according to the JEDEC standard.
  - 0 – The Training Engine programs the *DQS<0,1,2,3,4,5,6,7> Trained Write Leveling Delay* attributes to the PHY and checks that the DQ feedback after DQS pulse is high.
  - 1 – The Training Engine performs Write Leveling.

cbt_en

- Enables Command Bus Training (CBT) during initialization and training. CBT performs CA_VREF programming and aligns the CS/CA and CK for high frequency operation.
  - 0 – The Training Engine will shorten the command bus training. Instead of iterating through the different CA delays to find the optimal delay value, it will only program the *Trained CA Delay* attribute to PHY.
  - 1 – The Training Engine performs Command Bus Training (CBT) to find the optimal CA delay value.

init_en

- Provides ability to shorten initialization for simulation purposes.
  - 0 – Initialization is greatly reduced. For example, reset time and CKE low time is greatly shortened.
  - 1 – Initialization is performed according to the LPDDR4 JEDEC Standard.

5.9. Status Register (STATUS_REG) (0x24)

Table 5.13 shows a summary of the Status Register. The internal CPU writes to this register to communicate the status to the host CPU.

Table 5.13. Status Register

Field	Name	Access	Width	Reset
[31:21]	reserved	RSVD	11	—
[20]	rank1_done	RW	1	—
[19]	rank0_done	RW	1	—
[18:16]	refresh_rate	RO	3	—
[15:14]	reserved	RSVD	2	—
[13:12]	error_on_rank	RW	2	0
[11]	write_trn_err	RW	1	0
[10]	read_trn_err	RW	1	0
[9]	write_lvl_err	RW	1	0
[8]	cbt_err	RW	1	0
[7]	reserved	RSVD	1	—
[6:5]	in_self_refresh	RO	2	0
[4]	write_trn_done	RW	1	0
[3]	read_trn_done	RW	1	0
[2]	write_lvl_done	RW	1	0
[1]	cbt_done	RW	1	0
[0]	phy_ready	RO	1	0

rank1_done

- Asserts when training completes for rank 1.

rank0_done

- Asserts when training completes for rank 0.

refresh_rate

- Reflects the value of the refresh rate set in the MR4 within LPDDR4 memory. It specifies the required refresh period based on the temperature of the LPDDR4 device.
  - 3'b000: SDRAM Low temperature operating limit exceeded
  - 3'b001: 4x refresh
  - 3'b010: 2x refresh
  - 3'b011: 1x refresh (default)
  - 3'b100: 0.5x refresh
  - 3'b101: 0.25x refresh, no derating
  - 3'b110: 0.25x refresh, with derating
  - 3'b111: SDRAM High temperature operating limit exceeded

error_on_rank

- Indicates which rank has encountered an error.
  - 2'bx1: training error on rank 0
  - 2'b1x: training error on rank 1

write_trn_err

- Asserts when a failure occurs during write training.

read_trn_err

- Asserts when a failure occurs during read training. If the read_trn_done signal is low (STATUS_REG[3]=0), the training fails to find a setting that properly captures the read DQS burst. If the read_trn_done signal is high (STATUS_REG[3]=1), the training fails to find an optimal read DQS-DQ delay for capturing the read data.

write_lvl_err

- Asserts when a failure occurs during write leveling.

cbt_err

- Asserts when a failure occurs during Command Bus Training.

in_self_refresh

- Asserts when the memory is in self-refresh.
  - 2'bx1: rank 0 is in self-refresh
  - 2'b1x: rank 1 is in self-refresh

write_trn_done

- Asserts when write training completes.

read_trn_done

- Asserts when read training completes.

write_lvl_done

- Asserts when write leveling completes.

cbt_done

- Asserts when a Command Bus Training completes.

phy_ready

- 1'b0: PHY initialization is ongoing. APB writes are ignored to prevent training from starting while the PHY is not ready. APB read accesses return 0.
- 1'b1: PHY initialization is complete and the PHY is ready for operation.

6. LPDDR4 Memory Controller Example Design

This section describes the LPDDR4 Memory Controller Example Design that is available to you for synthesis and simulation after successful IP generation.

6.1. Overview

Table 6.1 summarizes the IP parameter configurations reflected in the LPDDR4 Memory Controller Example Design. For steps on generating the Memory Controller IP Core, refer to the Designing and Simulating the IP section of this user guide.

Table 6.1. Supported Example Design Configurations

Attribute[1]	Supported Setting
I/O Buffer Type	LVSTL_I, LVSTL_II
DDR Command Frequency	300, 350, 400, 533
Enable Power Down	Checked, Unchecked
Enable DBI	Checked, Unchecked
PLL Reference Clock from Pin	Checked, Unchecked
DDR Bus Width	16, 32
Number of Ranks	1, 2
Local Data Bus Type	AHBL, AXI4
ID Width	All
Number of Outstanding Writes/Reads	All
Write/Read Ordering Queues	All
Enable Local Bus Clock	Checked, Unchecked
Enable APB interface	Checked, Unchecked
Memory Device Timing Tab	All
Training Settings Tab	All

Note:

1. The LPDDR4 Memory Controller Example Design supports all configurable attributes.

6.2. Synthesis Example Design

After successfully generating the LPDDR4 Memory Controller for Nexus Devices, you receive a synthesizable example design. This design includes a test program that lets you evaluate the Memory Controller IP on hardware.

Figure 6.1 shows a block diagram of the LPDDR4 Memory Controller Example Design.

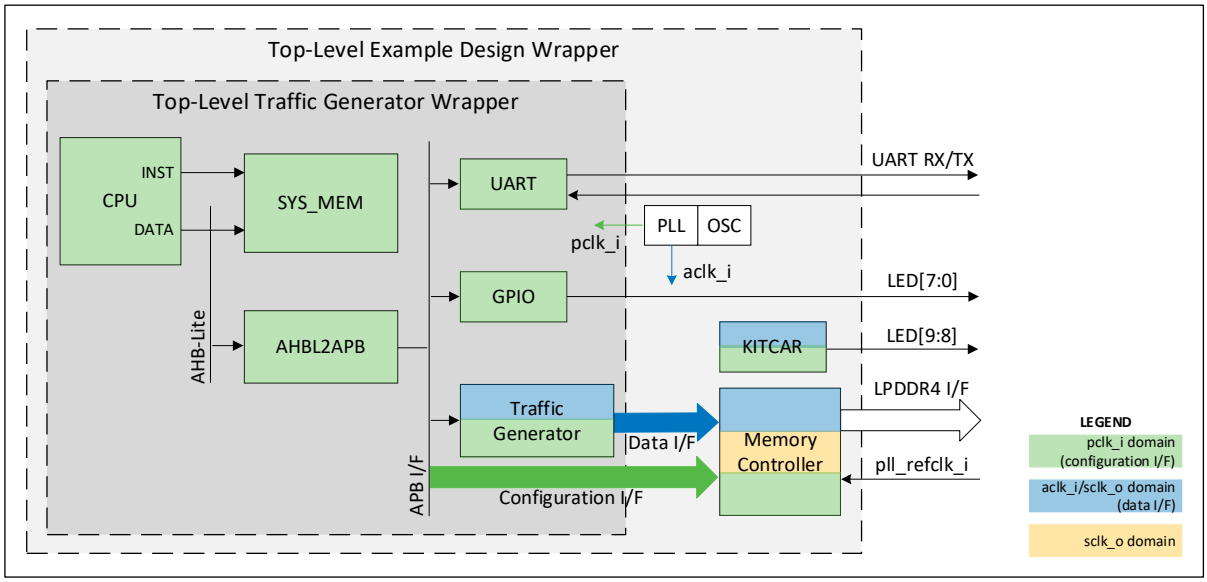


Figure 6.1. Memory Controller Example Design Functional Diagram

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

- RISC-V CPU subsystem – initializes and trains the LPDDR4 interface and performs data access checks.
- System Memory (SYS_MEM) – stores the instruction code for the example design test program.
- AHBL2APB bridge – converts the CPU data interface (AHB-Lite) to the configuration interface (APB).
- UART block – interfaces with the test program and prints out results through a serial terminal connection.
- GPIO block – verifies the result of LPDDR4 training sequences and the functionality of pclk_i and aclk_i clock signals.
- Traffic generator block – implements a series of pseudo-random writes and reads, compares the read data to the expected data, and displays the result over a UART connection.
- LPDDR4 Memory Controller IP – interfaces with external LPDDR4 memory to issue reads and writes.
- Oscillator (OSC) – generates a 90 MHz clock for the configuration interface (pclk_i).
- PLL – included only in IP Core v2.x.x, takes the oscillator output as an input reference clock to the PLL. The PLL generates both the configuration interface clock (pclk_i) and the user data interface AXI clock (aclk_i).

The synthesizable example design consists of a test program stored in system memory. The CPU instruction port fetches this program. The CPU data port accesses system memory and connects to the following:

- Configuration Set Registers (CSRs)
- UART
- GPIO
- Traffic generator
- LPDDR4 Memory Controller

The test program associated with the example design performs the following:

- Wait for user input over a serial terminal connection when the reset signal: rstn_i is asserted.
- Configure the Memory Controller to perform a complete reset, initialization, and training (TRN_OP_REG=0x5F) of the LPDDR4 interface.
- Perform a series of loop-back data access checks.
- Calculate performance of the LPDDR4 interface.

The top-level example design wrapper file, eval_top.sv, provides ports for a PLL reference clock, LPDDR4 interface, UART interface, and a 10-bit LED output. LED[7:0] corresponds to bits STATUS_REG[8:1], whereas toggling on LED[8] signifies aclk_i is functioning and toggling on LED[9] signifies pclk_i is functioning.

For information on the example design test program and instructions on how to generate and perform hardware evaluation of the LPDDR4 Memory Controller 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
- Replaces the external LPDDR4 memory with an LPDDR4 memory model

The simulation disables the UART connection because simulating the UART takes a long time. The LPDDR4 memory model allows simulation of user-initiated operations to LPDDR4 SDRAM. For instructions on how to simulate the LPDDR4 Memory Controller, including the example design, refer to the Designing and Simulating the IP section of this user guide.

Table 6.2 summarizes the simulation runtime of the LPDDR4 Memory Controller for various configurations.

Table 6.2. Simulation Runtime Summary

LPDDR4 Configuration	Typical Initialization Time	Typical Training Time
x16, 533 MHz (single-rank)	2251 µs	328 µs
x32, 533 MHz (single-rank)	2251 µs	439 µs
x16, 400 MHz (single-rank)	2251 µs	430 µs
x32, 350 MHz (single-rank)	2251 µs	536 µs
x16, 533 MHz (dual-rank)	1.36 ms	1.02 ms
x32, 533 MHz (dual-rank)	1.36 ms	1.18 ms

The initialization time is measured from the moment the CPU and Controller Engine pull out of reset (RESET_REG) to the assertion of the LPDDR4 Clock Enable signal (ddr_cke_o). The training time is measured from ddr_cke_o assertion to the assertion of init_done_o, which signifies the completion of the initialization and training sequence. For more details on the initialization and training sequences, refer to the LPDDR4 Calibration section of this user guide.

7. Designing and Simulating the IP

This section describes the steps required within Lattice Radiant software to configure and generate the LPDDR4 Memory Controller for Nexus Devices. It 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 to create, configure, and generate an instance of the LPDDR4 Memory Controller for Nexus Devices.

7.1.1. Creating a Lattice Radiant Project

To generate an instance of the Memory Controller IP, first create a Lattice Radiant project.

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

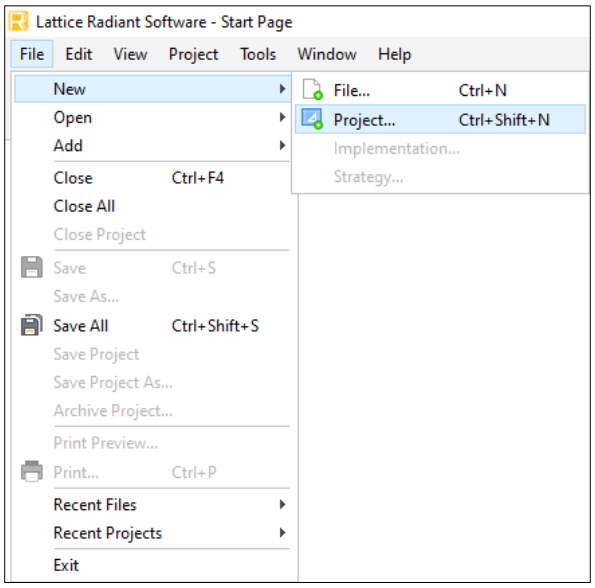


Figure 7.1. Creating a New Lattice Radiant Project

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

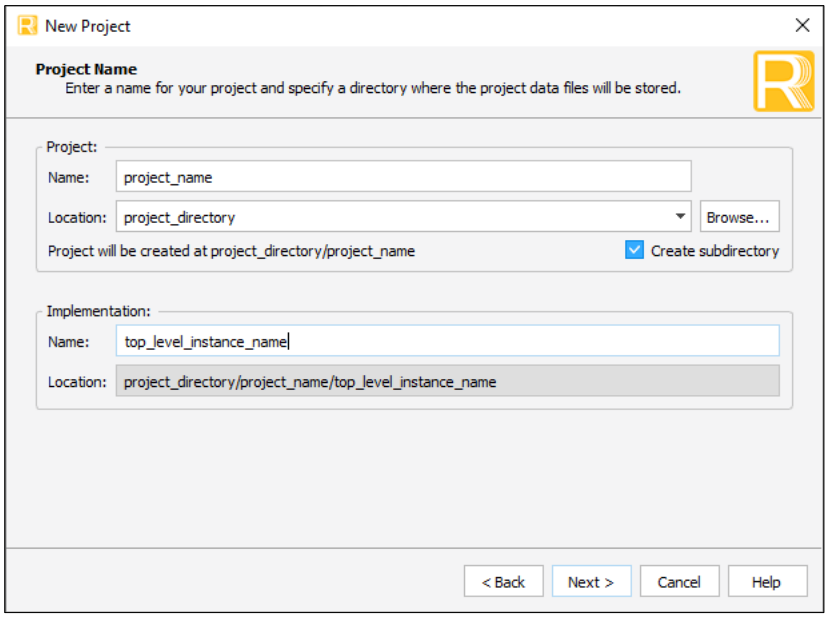


Figure 7.2. New Project Settings

3. Under **Family**, select **CertusPro-NX** or **MachXO5-NX**. For **Device, Package, Operating Condition, and Performance Grade**, make the appropriate selections that represent the selected device part number. Click **Next**.

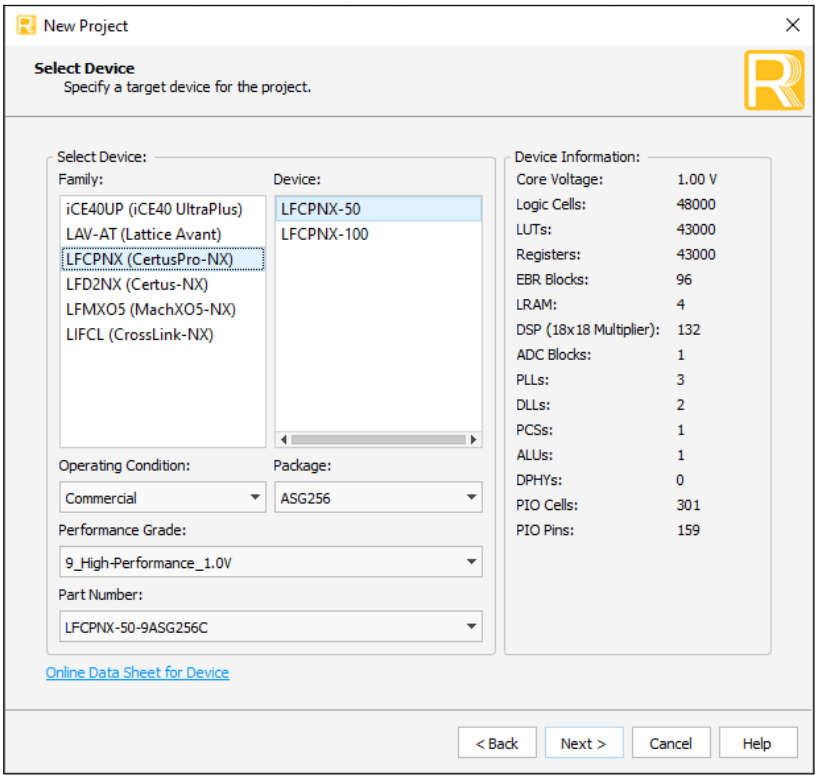


Figure 7.3. Project Device Settings

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

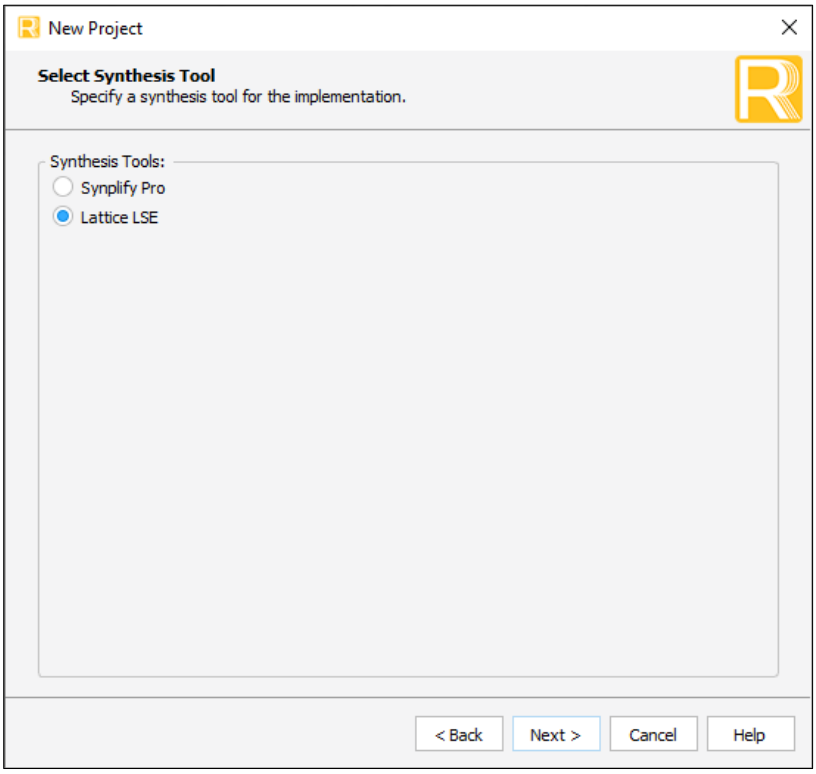


Figure 7.4. Project Synthesis Tool Selection

7.1.2. Configuring and Generating the IP

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

1. Under the **IP Catalog** (Tools > IP Catalog), locate and double-click on the desired Memory Controller IP listed under **IP > Processors_Controllers_and_Peripherals**.
   a. If no Memory Controller IPs are installed on the system, select the **IP on Server** tab within the IP Catalog.
   b. Click on the **Download from Lattice IP Server** icon next to the desired Memory Controller IP for installation. This action opens an **IP License Agreement** dialog box.
   c. Click **Accept** and then click on the **Refresh IP Catalog** icon.
   d. Locate the installed Memory Controller IP under the **IP on Local** tab within the IP Catalog and double-click.
      - IP Core v1.x.x = Memory Controller
      - IP Core v2.0.1 = Memory Controller for CertusPro-NX
      - IP Core v2.x.x = LPDDR4 Memory Controller for Nexus
2. The **Module/IP Block Wizard** dialog box opens. 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**.

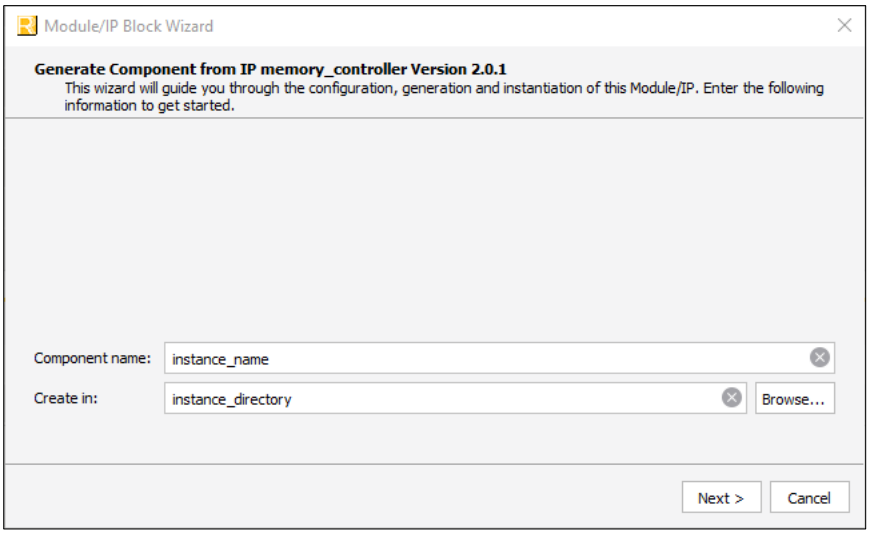


Figure 7.5. IP Instance Settings

3. The Memory Controller IP editor contains multiple tabs that need to be configured according to the desired LPDDR4 memory interface implementation. Table 7.1 provides high-level guidance for configuring the tabs in the Memory Controller IP editor. For detailed information on the individual attributes, refer to the IP Parameter 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 LPDDR4 memory device to ensure that you set the channel density (*DDR Density per Channel*) correctly.
Memory Device Timing	Refer to the datasheet for the selected LPDDR4 memory device to modify the default timing parameters as needed.
Training Settings (IP Core v2.x.x)	• Perform signal integrity analysis with system IBIS model, using a tool such as HyperLynx, to determine the optimal I/O and ODT settings for custom boards. • Modify the MC DQS VREF only if an error occurs during read training and/or data access. • Modify the Memory CA VREF only if an error occurs during command bus training. • Modify the Memory DQ VREF only if an error occurs during write training and/or data access.
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**.

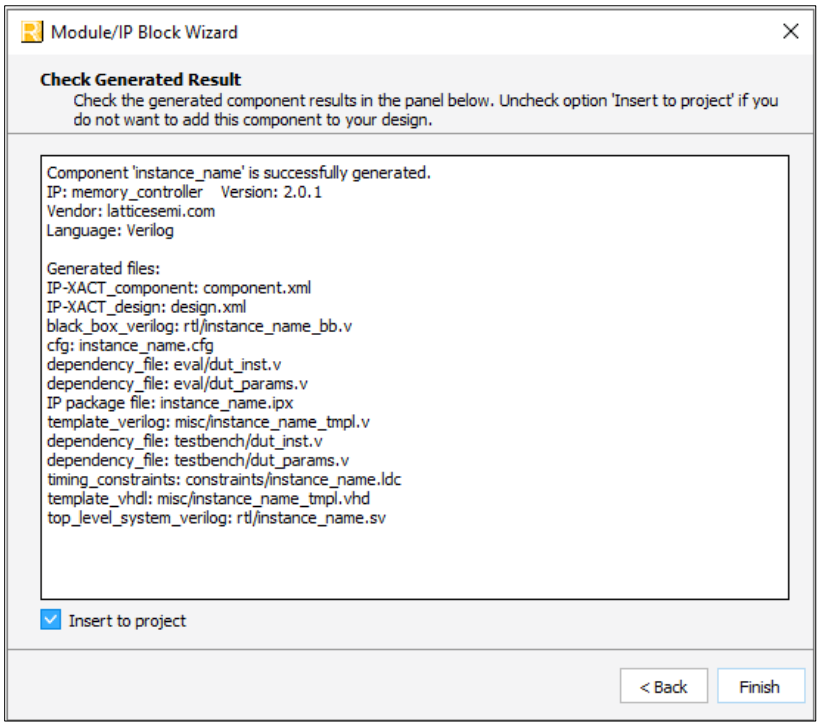


Figure 7.6. IP Generation Result

5. All the generated files are placed in the <instance_directory>/<instance_name> directory. Table 7.2 lists the generated files under this directory.

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 or changed parameter values set during IP configuration.
<instance_name>.ipx	Lists the files associated with IP generation.
constraints/constraint.sdc	Pre-Synthesis and Pre-Map constraints for the IP.
constraints/<instance_name>.ldc	Defines the I/O standard for LPDDR4 memory interface signals.
eval/apb2init.sv	Implements handshaking between APB accesses and the internal RISC-V CPU for initializing LPDDR4 memory in the LPDDR4 Memory Controller Example Design.
eval/clock_constraint.sdc	Pre-synthesis constraint for setting the PLL reference clock frequency of the LPDDR4 Memory Controller Example Design.
eval/constraint.pdc	Post-synthesis clock uncertainty constraints and LPDDR4 Memory Controller Example Design constraints.
eval/dut_inst.v	Instantiates the generated IP core in eval_top.sv for the LPDDR4 Memory Controller Example Design.
eval/dut_params.v	Defines local parameters for eval_top based on parameter values set during IP configuration for the LPDDR4 Memory Controller Example Design.
eval/eval_top.sv	Top-level RTL file for the LPDDR4 Memory Controller Example Design.
eval/kitcar.v	Counter to drive LEDs indicating that the internal clocks (pclk and aclk) are active in the LPDDR4 Memory Controller Example Design.
eval/pll_aclk_pclk.v	PLL responsible for generating APB clock (pclk) and AXI4 (aclk) in LPDDR4 Memory Controller Example Design.

File Name	Description
eval/select_protocol.py	Script that defines LPDDR4 protocol for eval_top and tb_top files in LPDDR4 Memory Controller Example Design.
eval/axi_bridge/	Contains RTL for AXI bus interface to Native interface.
eval/traffic_gen/	Contains RTL files for the LPDDR4 Memory 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/debug_c_code.sv	Used to debug training c code during simulation. For internal use only.
testbench/delay_1dir.sv	Used for modeling delay on each Command, Control and Address signal.
testbench/dqs_grp_delay.sv	Used for modeling delay on each DQ, DM, and DQS signal.
testbench/dut_inst.v	Template instance files.
testbench/dut_params.v	Lists of parameters based on user IP configurations.
testbench/tb_top.sv	Top level testbench file.
testbench/lpddr4/	Contains LPDDR4 memory model and instances for simulation.

7.2. Design Implementation

This section describes the steps required to properly run an LPDDR4 Memory 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 CertusPro-NX and MachXO5-NX 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 three to four HPIO DQS groups, 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. Refer to the High-Speed I/O User Guide and Pinout files located on the CertusPro-NX and MachXO5-NX web pages for more information.

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, place those banks adjacent to one another to minimize timing and routing impact. 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. Place the DQS_P signal 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) in the same HPIO DQS group. Typically, an LPDDR4 DQS group consists of eight DQ signals, one DM signal, and one DQS/DQSN pair. This means a HPIO DQS group needs 11 pins to support an LPDDR4 DQS group.

- Assign the input reference clock to the PLL to use dedicated clock routing (GPLL or PCLK). 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 placement ensures better performance by minimizing jitter and routing.

Always test proposed pinouts in Lattice Radiant software with 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 your LPDDR4 Memory Controller for Nexus IP project.

Table 7.3. Project Constraints

File Name	Description	Action Required
Memory Controller IP SDC file: constraints/constraint.ldc	Pre-synthesis and pre-map IP timing constraints	No. These constraints are automatically propagated.
Memory Controller IP LDC file: constraints/<instance_name>.ldc	Sets the I/O type for each of the ports necessary to interface with LPDDR4 SDRAM	No. These constraints are automatically propagated.
Clock Constraint SDC file: eval/clock_constraint.sdc	Contains an example constraint for the input PLL reference clock	Yes – you need to include a create_clock constraint based on the frequency of the input PLL reference clock. Place this constraint in a user-created SDC file.
Memory Controller Example Design PDC file: eval/constraint.pdc	Contains generated clock uncertainty constraints and example design constraints based on IP configuration	Yes – you need to copy the clock uncertainty constraints listed in this file directly into your top-level PDC file. Also, copy the example design constraints to run the LPDDR4 Memory Controller for Nexus Devices Example Design.

The Memory Controller IP constraints do not define the input and output delay on the DDR I/O since the routing from the I/O to DDR primitives is a dedicated path. Additionally, the Memory Controller calibration sequence compensates for controller-to-DRAM end-to-end skew.

7.2.2.1. Reference Clock SDC

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 Lattice Radiant Project SDC (not in PDC file) so that the synthesis tool can optimize the logic for the target clock.

7.2.2.2. Example Design PDC

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

- Clock Uncertainty constraints – defines the clock name for sclk_o and specifies its clock uncertainty based on the measured clock jitter during hardware validation.
- Eval constraints – specific to the Memory Controller Example Design.

If you are implementing your own Memory Controller design, only the clock uncertainty constraints need to be copied into the top-level user PDC file.

Copy the eval constraints, which are composed of set_false_path, set_max_delay, and ldc_create_group constraints, only if you are running the provided Memory Controller Example Design. These eval constraints are located 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
```

Refer to the Lattice Radiant Timing Constraints Methodology User Guide (FPGA-AN-02059) for details regarding the implementation of constraints.

7.3. Example Design Hardware Evaluation

After successfully configuring and generating the Memory Controller IP Core, the included synthesis example design can be used for hardware evaluation of the LPDDR4 Memory Controller. For a detailed description of the LPDDR4 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 are described in the 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 through 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	The 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 configuring and generating the Memory 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 Memory Controller Example Design project and generate the associated bitstream.

1. After generating the Memory Controller IP Core, the Lattice Radiant project contains the <instance_name>.ipx under the project's input files. If the file is not present, right-click on **Input Files** and select **Add > Existing File** under the **File List** tab in the lower-left corner of the Lattice 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**.

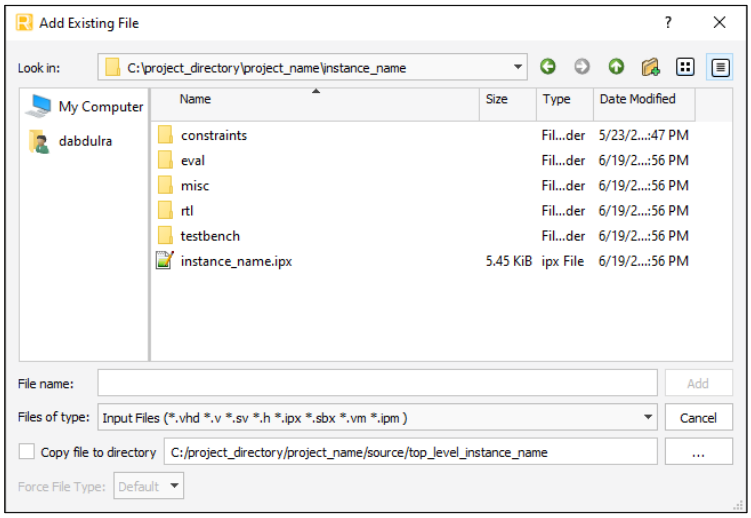


Figure 7.7. Add Existing File Dialog Box

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. This action ensures that any user modifications are not overwritten when the Memory Controller IP Core is regenerated. 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 file to add the pin assignment for the CertusPro-NX or MachXO5T™-NX board. You can accomplish this by either modifying the constraint.pdc file directly or first synthesizing the Lattice Radiant project by clicking on the Synthesize Design button, and then adding pinouts through the Device Constraint Editor under **Tools > Device Constraint Editor**. Note that when assigning the UART pins, the UART TX signal connects to the UART RX on the FPGA side, and that the UART RX signal connects to the UART TX signal on the FPGA side.
6. Before running 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*.

Sometimes you encounter timing failures during *Place and Route* due to unconstrained paths specific to their design. For details on implementing constraints, refer to the Constraints section of this user guide.

7.3.2. Running on Hardware

The LPDDR4 Memory Controller IP Core is hardware tested on CertusPro-NX Versa board. To perform hardware evaluation of the LPDDR4 Memory Controller Example Design, you need the following:

- CertusPro-NX or MachXO5T-NX 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, a bitstream file is required. 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.

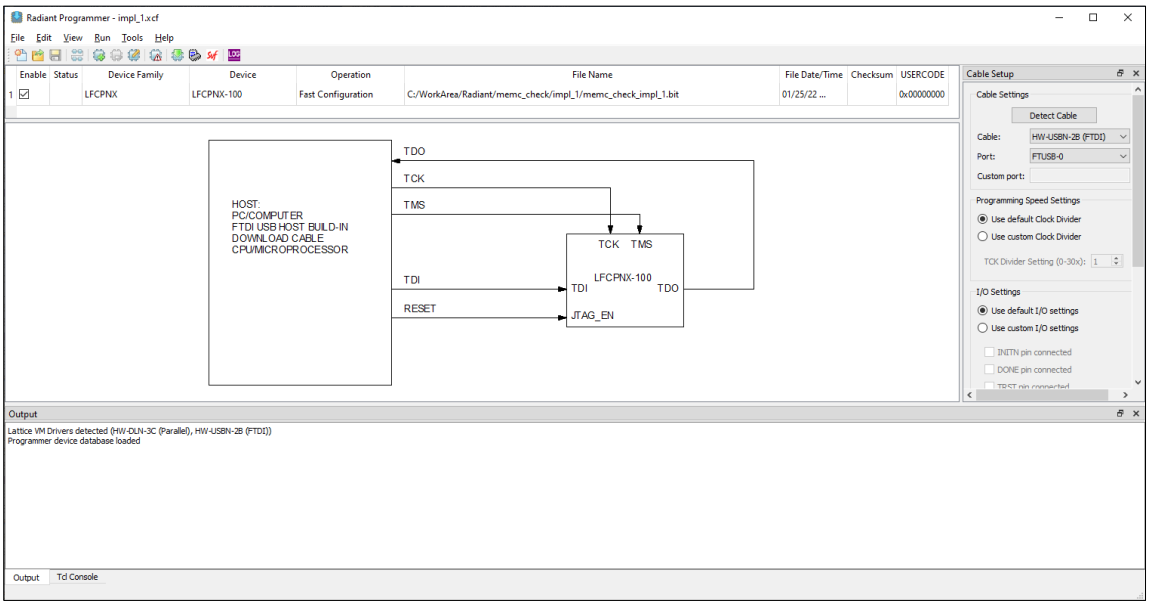


Figure 7.8. Lattice Radiant Programmer

3. Click under the **File Name** field and 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* in the Preparing the Bitstream section of this user guide 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 displays the following message:
   - After programming the Lattice FPGA with the example design bitstream, launch a serial terminal. If you want to use your own serial communication terminal, proceed to Step 7. 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.

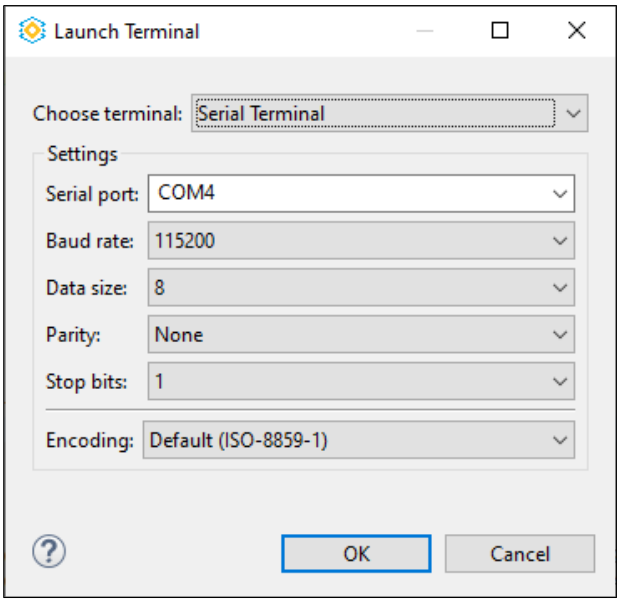


Figure 7.9. Serial Terminal Settings

To run the example design, assert the rstn_i signal. This 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.

```
================================================================
LPDDR4 Memory Controller for Nexus Devices Example Design
================================================================
Select Test to Run (key + Enter; Enter alone = option 1 then exit menu)
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
x = Exit
```

Press **0** then **Enter** prints a series of definitions correlating to the memory training and data access test pattern.

```
--------------------------------------------------
Memory Training Stage Definitions
--------------------------------------------------
A sequence of letters will print out to indicate which training stages were successfully
completed.
I = Initialization
C = Command Bus Training
L = Write Leveling
R = Read Training
W = Write Training
--------------------------------------------------
Memory Data Access Pattern Definitions
--------------------------------------------------
A sequence of numbers will print out to indicate which data access patterns were
successful.
0 = 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 = INCR64: 64-beat incrementing burst write followed by burst read with no delay
between transactions
8 = RAND2: 2-beat random address burst write followed by burst read with no delay
between transactions
p = INCR64 Performance Test: 64-beat incrementing parallel burst write and burst read
with no delay between transactions
```

Press **1** then **Enter** starts the training sequence, followed by eight different data access checks:

```
--------------------------------------------------
Starting Memory Training
--------------------------------------------------
I
 C
  L
   R
    W
RESULT: Training PASSED with STATUS_REG_ADDR value: b001f (hex)
DDR Clock Delay Value is not Optimal.
  Please update the IP GUI with this value : 74
VREF has been trained to the following values:
  MC DQ_Vref Value     = 53 <-- Please update the IP GUI with this value.
  Memory CA_Vref Value = 30
  Memory DQ_Vref Value = 21 <-- Please update the IP GUI with this value.
--------------------------------------------------
Starting Data Access Check
--------------------------------------------------
0
 1
  2
   3
    4
     5
      8
       p
--------------------------------------------------
Starting Performance check
--------------------------------------------------
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 % : 81
Performance in Mbps : 13978
Ran : 20000 transactions across sequential addresses
```

Press **2** then **Enter** starts memory training and data access in a loop with reset. You can input the number of reset.

```
iteration: 0 of Iterations: 1000
Starting Memory Training
----------------------------------------------------
I
 C
  L
   R
    W
RESULT: Training PASSED with STATUS_REG_ADDR value: b001f (hex)
VREF has been trained to the following values:
MC DQS Grp Vref = 48, Memory CA Vref = 30, Memory DQ Vref = 20
----------------------------------------------------
Starting Data Access Check
----------------------------------------------------
0
 1
  2
   3
    4
     5
      8
       p
----------------------------------------------------
Starting Performance check
----------------------------------------------------
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 % : 81
Performance in Mbps : 13978
Ran : 20000 transactions across sequential addresses

RESULT: All transaction types have PASSED.

iteration: 1 Starting Memory Training
----------------------------------------------------
I
 C
  L
   R
    W
RESULT: Training PASSED with STATUS_REG_ADDR value: b001f (hex)
VREF has been trained to the following values:
MC DQS Grp Vref = 48, Memory CA Vref = 30, Memory DQ Vref = 20
```

```
----------------------------------------------------
Starting Data Access Check
----------------------------------------------------
0
 1
  2
   3
    4
     5
      8
       p
----------------------------------------------------
Starting Performance check
----------------------------------------------------
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 % : 81
Performance in Mbps : 13978
Ran : 20000 transactions across sequential addresses
RESULT: All transaction types have PASSED.
```

Press **Enter** starts the training sequence, followed by eight different data access checks and exit the menu.

```
----------------------------------------------------
Starting Memory Training
----------------------------------------------------
I
 C
  L
   R
    W
RESULT: Training PASSED with STATUS_REG_ADDR value: b001f (hex)
DDR Clock Delay Value is not Optimal.
  Please update the IP GUI with this value : 74
VREF has been trained to the following values:
  MC DQ_Vref Value     = 53 <-- Please update the IP GUI with this value.
  Memory CA_Vref Value = 30
  Memory DQ_Vref Value = 21 <-- Please update the IP GUI with this value.
----------------------------------------------------
Starting Data Access Check
----------------------------------------------------
0
 1
  2
   3
    4
     5
      8
       p
----------------------------------------------------
Starting Performance check
----------------------------------------------------
```

```
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 % : 81
Performance in Mbps : 13978
Ran : 20000 transactions across sequential addresses

RESULT: All transaction types have PASSED.
HARDWARE VALIDATION PASS.
```

Before any test is executed, press **x** followed by **Enter** exits the menu.

```
Exiting.
```

After a test is executed, press **x** followed by **Enter** exits the menu and reports the hardware validation result as *Pass* or *Fail*.

```
Exiting.
        HARDWARE VALIDATION PASS.
```

The results from the data access checks and performance measurement are printed over the serial connection, press **Backspace** to edit your selection before pressing **Enter** to execute it.

The bus efficiency value represents the efficiency percentage of the LPDDR4 bus utilization, whereas the performance value represents the bandwidth of the LPDDR4 interface. The following formulas are used to calculate the efficiency and bandwidth:
Efficiency = (Total number of bytes transferred) / (Number of DDR clock cycles × (DDR_WIDTH / 8) × gearing ratio)
Bandwidth = (Total number of bytes transferred) / (Number of DDR clock cycles × DDR clock period)

You can also calculate the bandwidth using the following equation: LPDDR4 Data Width × LPDDR4 Data Rate × (LPDDR4 Bus Efficiency / 100), where the LPDDR4 Data Rate is in Mbps.

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

```
-----------------------------------------------------
Starting Memory Training
-----------------------------------------------------

I
Command Bus Training Failed
```

7.4. Example Design Simulation

After successfully configuring and generating the Memory Controller IP Core, you can use the included example design to simulate the LPDDR4 Memory 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 simulating the example design, complete steps 1-2 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 check the **Copy file to directory** option is unchecked. Click **Add**.
2. Before creating the simulation environment, set the SIM parameter in eval_top.sv to one. This parameter shortens the initialization sequence of the LPDDR4 interface and disables the UART interface for simulation. Do not modify the SIM parameter for hardware implementation.

When SIM is set to 1, it programs 0x1E to the Training Operation Register (TRN_OP_REG) to speed up the simulation runtime by reducing the reset and CKE initialization time. You can configure the simulation of reset and the initialization and training sequences by forcing the value of TRN_OP_REG by locating the following comment in tb_top.sv:

```
// This force shortens the initialization, vref trainings are also skipped

// force
tb_top.u_eval_top.u_lp4mc_0.lscc_lpddr4_mc_inst.u_trn_eng.i_csr.trn_operation_reg =
8'h1E;
```

Note that the memory model in the example design simulation requires a minimum simulation timescale of 100 fs as defined by the memory manufacturer. For proper functionality, ensure that the PLL reference clock is generated with a minimum timescale of 100 fs. Skip the training sequences by writing 8'h00 to TRN_OP_REG/trn_opr_i. This action reduces simulation time by programming the verified trained values to the PHY. Refer to the Simulation Example Design section of this user guide for more details.

To create a simulation environment for the LPDDR4 Memory Controller Example Design, follow these steps:

1. Launch the simulation wizard in the Lattice Radiant software 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**.

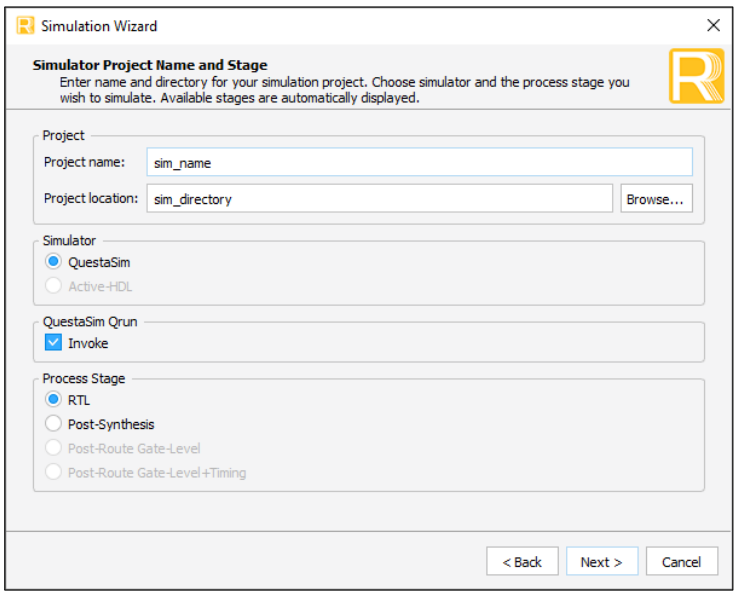


Figure 7.10. Simulation Wizard

2. In the **Add and Reorder Source** window, notice that the **Source Files** only contain the top-level evaluation (eval_top.sv) and top-level testbench (tb_top.sv) files. Click **Next**.

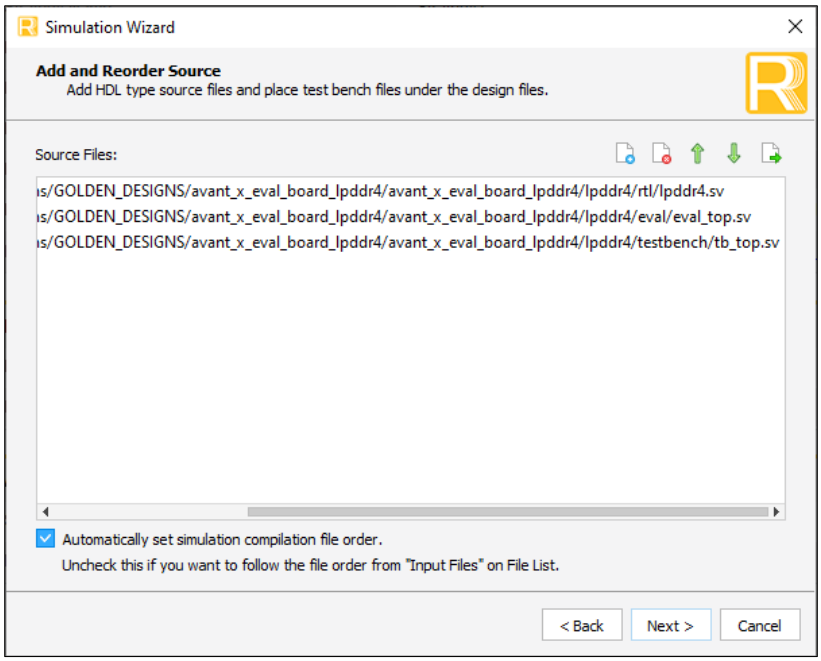


Figure 7.11. Adding and Reordering Simulation Source Files

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

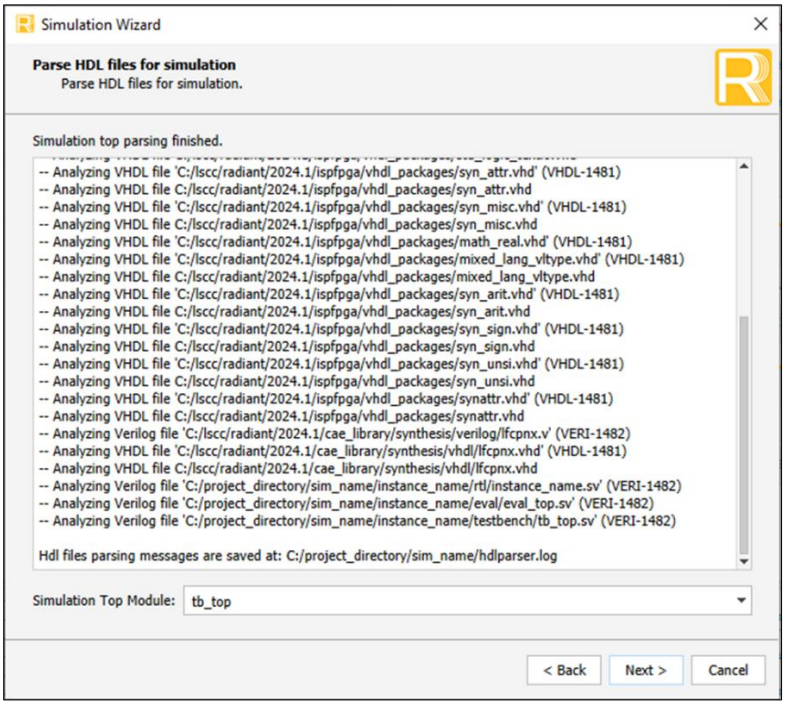


Figure 7.12. Parsing Simulation HDL Files

4. This action opens the **Summary** window.

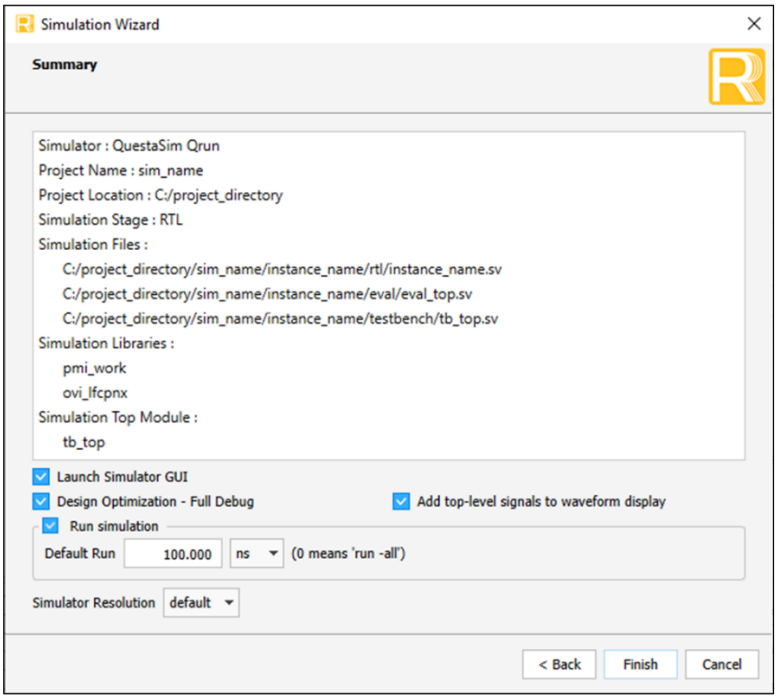


Figure 7.13. Simulation Summary

The results of the Memory Controller IP simulation design are shown in Figure 7.14.

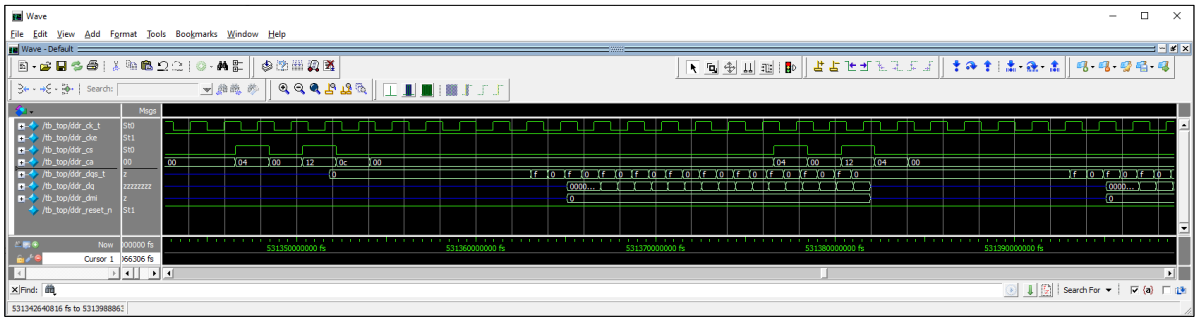


Figure 7.14. Simulation Result Waveform

The following error messages are expected in the QuestaSim simulation log and should be disregarded. These messages occur due to the violation of timing requirements regarding the LPDDR4 simulation model because of the shortening of reset and CKE initialization.

```
# tb_top.LP4MEM_00.mem_x16_00.ins_1ch.<protected> 26083125000 : ###  lpddr4_debug
RESET_n high input

# tb_top.LP4MEM_01.mem_x16_01.ins_1ch.<protected> 26083125000 : ###  lpddr4_debug
RESET_n high input

# Error: tb_top.LP4MEM_01.mem_x16_01.ins_1ch.<protected>.<protected> 26083125000  tINIT1
Error.

# Error: tb_top.LP4MEM_00.mem_x16_00.ins_1ch.<protected>.<protected> 26083125000  tINIT1
Error.

# tb_top.LP4MEM_00.mem_x16_00.ins_1ch.<protected> 32804075000 : ###  lpddr4_debug CKE
high input

# tb_top.LP4MEM_01.mem_x16_01.ins_1ch.<protected> 32804075000 : ###  lpddr4_debug CKE
high input

# Error: tb_top.LP4MEM_01.mem_x16_01.ins_1ch.<protected>.<protected> 32804075000  tINIT3
Error.

# Error: tb_top.LP4MEM_00.mem_x16_00.ins_1ch.<protected>.<protected> 32804075000  tINIT3
Error.
```

By default, simulation runs without DATA MASK assertion. To enable DATA MASK toggling during simulation, edit the *eval_top.sv* file and uncomment the ENABLE_SIM_DATA_MASK macro. This will increase simulation time from a maximum of 10 minutes to approximately 1 hour and 30 minutes.

```
//Uncomment this define to enable Data Mask toggling during simulation.
//This will increase simulation time.
//`define ENABLE_SIM_DATA_MASK
```

Figure 7.15 Macro to enable DATA MASK toggling

8. Debugging

This section discusses tools and strategies available to you for debugging your LPDDR4 memory interface.

8.1. Debug with the Example Design

The provided LPDDR4 example design can serve to help debug functional issues regarding the training of external LPDDR4 memory or data accesses issued by the Memory Controller. For a description of the example design and instructions on how to run the test program for hardware evaluation, refer to the Synthesis Example Design and Running on Hardware sections of this user guide.

The LPDDR4 example design prints the optimal settings after the training and before the data access check as shown in the Running on Hardware section of this user guide; a copy is reproduce below for quick reference.

```
...
RESULT: Training PASSED with STATUS_REG_ADDR value: b001f (hex)
DDR Clock Delay Value is not Optimal.
  Please update the IP GUI with this value : 74
VREF has been trained to the following values:
  MC DQ_Vref Value     = 53 <-- Please update the IP GUI with this value.
  Memory CA_Vref Value = 30
  Memory DQ_Vref Value = 21 <-- Please update the IP GUI with this value.
-------------------------------------------------
Starting Data Access Check
-------------------------------------------------
...
```

It is recommended to set these in the IP GUI's **Training Settings Tab** for faster training time and better training result. Using the optimal *DDR Clock Delay Value*, will result in better CBT results and allows the write leveling to pass on first sweep run. Note that when write leveling fails, the training engine will find the optimal DDR clock delay and retries the write leveling.

Using the optimal *Initial MC DQ_Vref Value* allows it is also used during write leveling. Though the write leveling can tolerate bad Vref value, it is still better to use the optimal value. This is also used as starting point of the MC-side DQ_Vref training so it will be easier to find the passing vref window. Similarly, the *Initial Memory DQ_Vref Value* is also used as the starting point of the DRAM-side DQ_Vref training.

8.2. Debug with Reveal Analyzer

The LPDDR4 Memory Controller for Nexus Devices introduces reveal signals to help you debug issues with the training sequence of the LPDDR4 memory interface using Reveal Analyzer. This feature was introduced in IP Core v2.1.x and exists within the rtl/<instance_name>.sv file following successful IP generation. When the user data interface is set to the AHB-Lite protocol, you can locate the reveal signals under the following comment in the generated IP RTL file:

```
// For reveal debugging
```

When the user data interface is set to the AXI4 protocol, the reveal signals can be located under the following comment in the generated IP RTL file:

```
// Reveal tap points for AXI INTERFACE
```

Refer to the Debugging with Reveal Usage Guidelines and Tips Application Note (FPGA-AN-02060), for information regarding the setup and usage of Reveal Analyzer. Table 8.1 covers the list of reveal signals available to you for debug.

Table 8.1. Reveal Analyzer Signal Definitions

Signal Name	Description
rvl_cs_r	Chip select
rvl_cbt_en_r	Asserted high during Command Bus Training (CBT)
rvl_ca_adj_cout_r	CA delay per bit adjustment cout signal
rvl_ca_adj_dir_r	CA delay per bit adjustment direction signal
rvl_ca_adj_load_n_r	CA delay per bit adjustment load signal
rvl_ca_adj_move_r	CA delay per bit adjustment signal
rvl_cbt_dq_fbak_r	DQ[13:8] output pins to feedback captured value to the Memory Controller during CBT
rvl_wrlvl_en_r	Asserted high during Write Leveling
rvl_wrlvl_load_en_r	Asynchronously resets the delay code to the default value for CK-DQS skew compensation during Write Leveling
rvl_wrlvl_move_r	At rising edge, it changes the delay code (±1) according to the direction set by wrlvl_dir_i for CK-DQS skew compensation during Write Leveling
rvl_wrlvl_dir_r	Controls the direction of delay code change for CK-DQS skew compensation during Write Leveling (0 = increase delay, 1 = decrease delay)
rvl_wrlvl_cout_r	Margin test output flag to indicate the under-flow or over-flow for CK-DQS skew compensation during Write Leveling
rvl_dqwl_r2	Data output of Write Leveling
rvl_rd_trn_en_r	Specifies current stage of read training (0=inactive, 1=read gate training (DQS), 2=read DQS training, 3=read deskew (DQ) training
rvl_burst_det_sclk_r	Clock generated using burst_det_o
rvl_rd_clksel_r	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_read_dqs_ie_r	Read enable signal for capturing the incoming read BL16. Each bit captures the DQ/DMI for the corresponding ddr_ck_o cycle.
rvl_pause_r	Set to 1 to stop the DQSBUF-generated internal clocks when updating rvl_rd_clksel_r and the delay codes. This is to avoid metastability.
rvl_rd_load_n_r	DQS input delay adjustment load signal
rvl_rd_move_r	DQS input delay adjustment move signal
rvl_rd_dir_r	DQS input delay adjustment direction signal
rvl_rd_cout_r	DQS input delay adjustment cout signal
rvl_rd_comp_done_r	Asserted high when compared data is done.
rvl_rd_comp_result_r	Results of compared data for each DQS group: [0] = 1: DQS group 0 comparison passes [1] = 1: DQS group 1 comparison passes [2:7] = 1: DQS group 2/3/4/5/6/7 comparison passes This is only valid when rvl_rd_comp_done_r is asserted.
rvl_dqdmi_i_load_n_r	DQ/DMI input delay per bit adjustment load signal
rvl_dqdmi_i_move_r	DQ/DMI input delay per bit adjustment move signal
rvl_dqdmi_i_dir_r	DQ/DMI input delay per bit adjustment direction signal
rvl_dqdmi_i_cout_r	DQ/DMI input delay per bit adjustment cout signal
rvl_wr_trn_en_r	Specifies the current rank in training (1=1st Rank, 2=2nd Rank)
rvl_wr_load_n_r	Output DQ/DMI clock delay per DQS group adjustment load signal
rvl_wr_move_r	Output DQ/DMI clock delay per DQS group adjustment move signal
rvl_wr_dir_r	Output DQ/DMI clock delay per DQS group adjustment dir signal
rvl_wr_cout_r	Output DQ/DMI clock delay per DQS group adjustment cout signal
rvl_dqdmi_o_load_n_r	DQ/DMI output delay per bit adjustment load signal

Signal Name	Description
rvl_dqdmi_o_move_r	DQ/DMI output delay per bit adjustment move signal
rvl_dqdmi_o_dir_r	DQ/DMI output delay per bit adjustment direction signal
rvl_dqdmi_o_cout_r	DQ/DMI output delay per bit adjustment cout signal
rvl_trn_stat_done	{write_trn_done, read_trn_done, write_lvl_done, cbt_done, phy_ready}
rvl_trn_stat_err	{write_trn_err, read_trn_err, write_lvl_err, cbt_err}
rvl_scratch_0_r	Registers used by the training CPU to write debug information
rvl_scratch_1_r	Registers used by the training CPU to write debug information

For general purpose debugging, use the rvl_trn_stat_done and rvl_trn_stat_err signals to determine if the training of the LPDDR4 interface has passed or failed. The rvl_trn_stat_done signal indicates a particular stage of training has completed. The expected result is that all bits are set to 1, indicating that the following stages have completed from MSB to LSB: write training, read training, write leveling, command bus training, initialization of PHY.

The rvl_trn_stat_err signal indicates an error has occurred during a particular stage of training. The expected result is that all bits are set to 0, indicating that the following stages have completed successfully from MSB to LSB: write training, read training, write leveling, command bus training. Examples are included in Figure 8.1 and Figure 8.2.

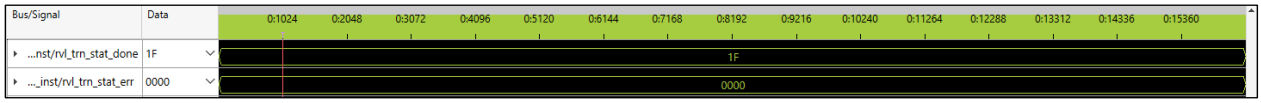


Figure 8.1. Reveal Example: LPDDR4 Training Passes

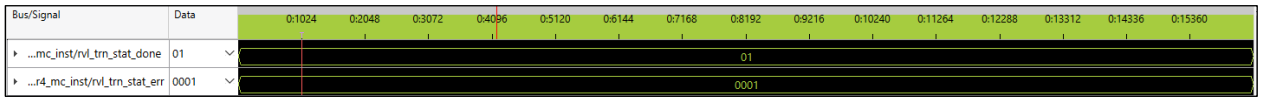


Figure 8.2. Reveal Example: Command Bus Training Failure

8.2.1. Command Bus Training (CBT)

The Memory Controller begins Command Bus Training (CBT) by sending CA pattern 0x19 to the LPDDR4 memory device, delaying the CA bus, and using DQ[13:8] to provide feedback on the captured value. CBT occurs when the rvl_cbt_en_r signal is asserted high. The upper byte of the rvl_dqwl_r2 signal acts as the feedback mechanism, and the lower byte correlates to the CA VREF level and range setting. DQ[13:8] maps to the rvl_cbt_dq_fbak_r signal, which notifies the Memory Controller whether CBT has passed for the current delay setting. Once 0x19 is read on the rvl_cbt_dq_fbak_r signal, it indicates CBT has passed. This sequence repeats until the optimal setting, providing the best margin, is found. In dual-rank configurations, the rvl_cs_r signal indicates the rank (0 = rank 0, 1 = rank 1). For more information on CBT, refer to the Command Bus Training (CBT) Description section of this user guide.

Figure 8.3. Command Bus Training Simulation Waveform

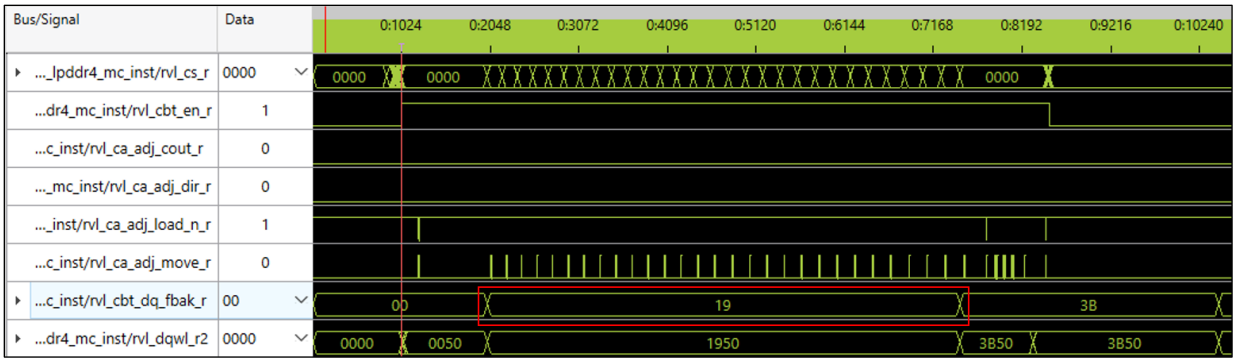


Figure 8.4. Command Bus Training Reveal Capture

The CBT is expected to pass with wide margin. If the CBT failed, the cause is likely an issue with the MC-to-DRAM connection. Check the following:

- The pin assignments are correct
- The CK width and CS width match the schematic.
- The DRAM's ODT(ca)_A/B pins are enabled on the board.
- *CK/CS Slew Rate* and *Command Address Slew Rate* – use the default, Fast

The combination of following Training Settings IP parameters improve the CBT result; it is recommended to set these based on the SI/PI simulation with your board design:

- *Initial Memory CA_VREF Value* – this is also the final Vref setting since Vref training on CBT is recommended to be disabled.
- MC CA Drive Impedance
- CA_ODT Value

8.2.2. Write Leveling

Write leveling begins with the LPDDR4 memory device sampling the DDR clock with the rising edge of DQS, and asynchronously providing feedback to the Memory Controller through the DQ pins. Write leveling occurs when rvl_wrlvl_en_r is asserted high. The rvl_dqwl_r2 signal acts as the feedback mechanism to indicate whether write leveling has passed for the current delay setting. Write leveling starts with a coarse delay adjustment for all DQS groups, followed by a fine delay adjustment for each of DQS groups. When all bits in the rvl_dqwl_r2 signal are set high, it indicates write leveling has passed. In dual-rank configurations, the rvl_cs_r signal indicates the rank (0 = rank 0, 1 = rank 1). Note that if the first sample of the rvl_dqwl_r2 signal has all bits set high, increase the setting for the *DDR Clock Delay Value* attribute. For more information on write leveling, refer to the Write Leveling Description section of this user guide.

Figure 8.5. Write Leveling Simulation Waveform

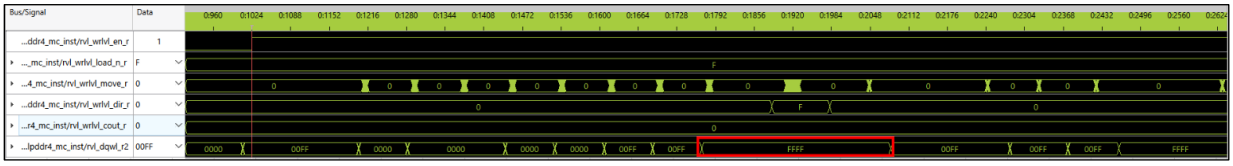


Figure 8.6. Write Leveling Reveal Capture

The initial write leveling feedback should be rvl_dqwl_r2==0x0000 for the normal write leveling process to be successful. If this is not the case, the correct DDR clock and DQS alignment cannot be achieved because the DQS is already ahead of the DDR clock even at 0 delay. To solve this, the write leveling increases the *DDR Clock Delay Value* to satisfy the condition and proceed with the normal write leveling process. It is recommended to run the example design on your board to find the optimal value and set it in the IP GUI. See the Example Design Hardware Evaluation and Debug with the Example Design sections for more information.

8.2.3. Read Training

Read training consists of four stages: read gate training (DQS), read DQS training, MC-side DQ Vref training, and read deskew (DQ) training.

Read gate training adjusts the read gate with incoming DQS burst until a full burst is captured. The read gate is the DQS capture window shown in the top waveform in Figure 2.8. The read_dqs_ie_i signal adjusts the delay of the read gate by 1 DDR clock cycle, while the rd_clksel_i signal adjusts it by 45° phase shift. Once the read preamble and eight DQS pulses are captured, it indicates read gate training is successful. This is indicated when the burst_det_sclk signal is set high for four occurrences in a row. The final setting for the rd_clksel_i signal is calculated as the average of the burst_det_sclk assertion window. The start of the read gate training is marked by the reveal debug signal rvl_rd_trn_en_r=1 in both Figure 8.7 and Figure 8.8. Also, in Figure 8.8, the first read gate training is done at read latency = 9 (rvl_read_lat_r=9) but it reaches the maximum rvl_rd_clksel_r taps of 0xF. It was rerun at rvl_read_lat_r =10 and gets a better passing window.

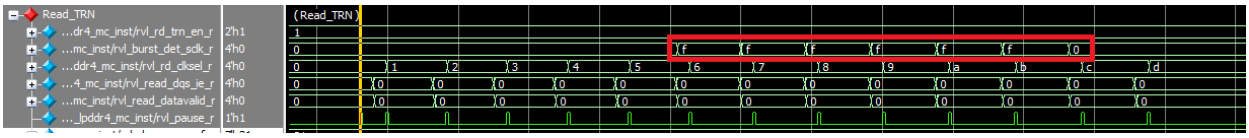


Figure 8.7. Read Gate Training Simulation Waveform

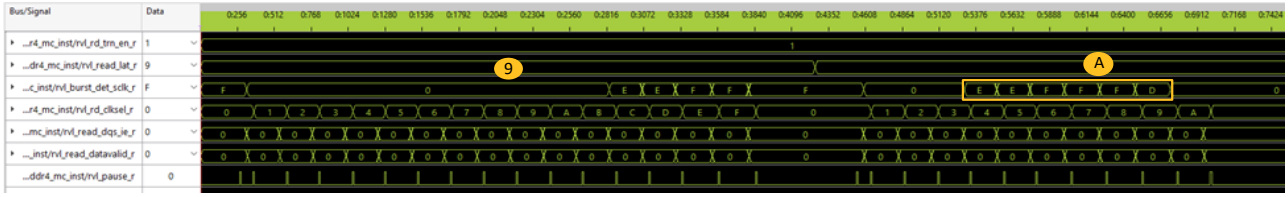


Figure 8.8. Read Gate Training Reveal Capture

The read gate training uses a coarse delay with a 45° phase shift per tap. An additional read DQS training was added to train the DQS delay using a fine delay of 12.5 ps per tap. This centers the read DQS burst within the read gate. The goal is to improve the stability across PVT – the read gate-to-read DQS margin is big enough to absorb the delay change of the read DQS due to VT drift. The start of the read DQS training is marked by the reveal debug signal rvl_rd_trn_en_r=2. It starts by moving the read dqs delay to the minimum, followed by the actual read DQS delay sweep. The minimum and maximum read DQS delays that produced a burst detect assertion are taken and their average is the final trained value. This is shown in Figure 8.9 and Figure 8.10 but the final programing of the trained value per DQS group is not shown here.

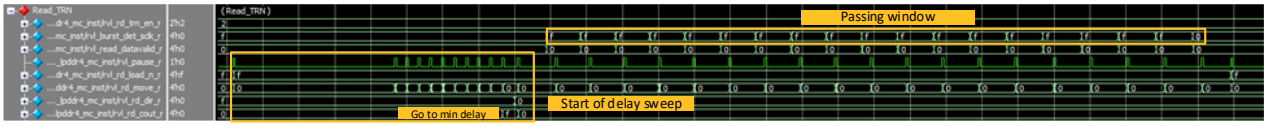


Figure 8.9. Read DQS Training Simulation Waveform

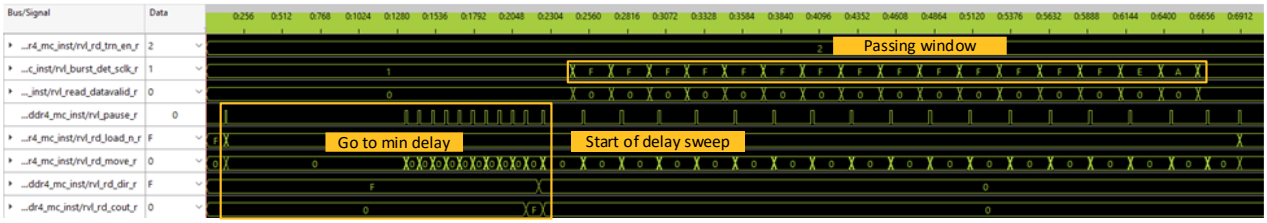


Figure 8.10. Read DQS Training Reveal Capture

Read deskew training start is marked by the reveal debug signal rvl_rd_trn_en_r=3. It issues Mode Register Write (MRW) commands to MR32 and MR40 to set the read training data and then issues Multi-Purpose Command: MPC[READ DQ CALIBRATION] to see if the content read back on DQ is correct. If the DQ signal does not capture the correct data, it is delayed, and the process is repeated. Once the read data is correct, the rvl_rd_comp_result_r signal is asserted high, each bit corresponds to the DQS group. The final DQ delay value is calculated after a passing window is achieved. It is then programmed per DQS group at the end, this is shown in Figure 8.11 simulation waveform but not shown in Figure 8.12 due to not enough reveal samples. In dual-rank configurations, the rvl_rank_sel_r signal indicates the rank (0 = rank 0, 1 = rank 1). For more information on read training, refer to the Read Training Description section of this user guide.

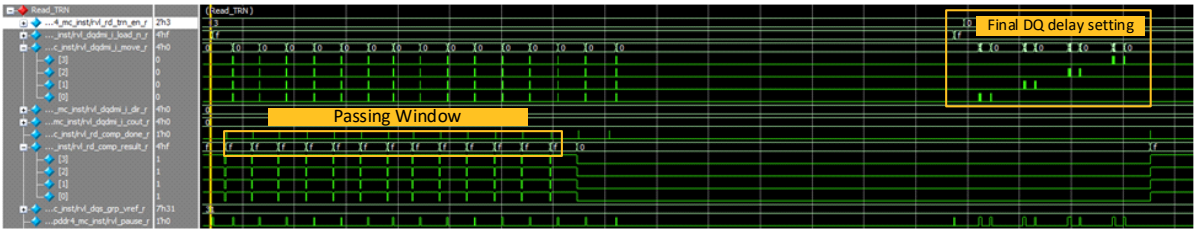


Figure 8.11. Read Deskew Training Simulation Waveform

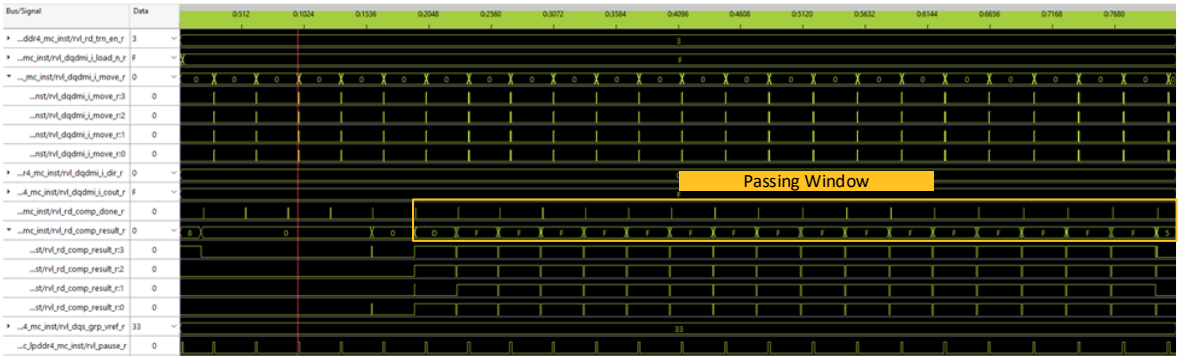


Figure 8.12. Read Deskew Training Reveal Capture

The MC-side Vref training is performed after the read DQS training and before the read deskew training. It performs similar routine as the read deskew training and is repeated across multiple DQ Vref values. The read deskew training shown in Figure 8.12 uses the trained MC-side DQ Vref. It is not recommended to debug it with reveal since process is very long, a better debug feature is needed. This will be addressed in future enhancement.

The read training or the MC-side Vref training may fail if the read DQ/DQS/DM signal quality is bad. The combination of following Training Settings IP parameters improve the read DQ signal quality (hence the read training as well), it is recommended to set these based on the SI/PI simulation with your board design:

- MC ODT Value
- SOC_ODT Value
- Pull-Down Drive Strength

8.2.4. Write Training

Write training consists of three stages: initial write training, DRAM-side DQ Vref training, and final write training. The goal of the initial write training is to prepare for the DRAM-side DQ_Vref training. The DQ/DMI clock delay (controlled by rvl_wr_*_r signals) was reduced so that the DQ/DMI output delay (controlled by rvl_dqdmi_o_*_r signals) can find both edges of the valid window. The DRAM-side Vref training is performed afterwards to get the optimal Vref. The final write training uses the optimal DRAM DQ_Vref and uses both the DQ/DMI clock delay and DQ/DMI output delay to find the optimal DQ/DMI output delay.

The initial write training starts when rvl_wr_trn_en_r=1. It issues a set of five MPC[WRITE-FIFO] commands and five MPC[READ-FIFO] commands. Once data is written to the external LPDDR4 SDRAM, the system reads the data back on the DQ signal and compares it with the “expected” data to determine if the read data is correct. If the DQ signal does not capture the correct data, the system delays the signal and repeats the process. Once the read data is correct, the system asserts the rvl_rd_comp_result_r signal high. The system calculates the DQ passing window as shown in Figure 8.13 and Figure 8.14. If the passing window is 0, the training engine increases the write latency value and retries. The training engine retries up to four times before asserting the error interrupt.

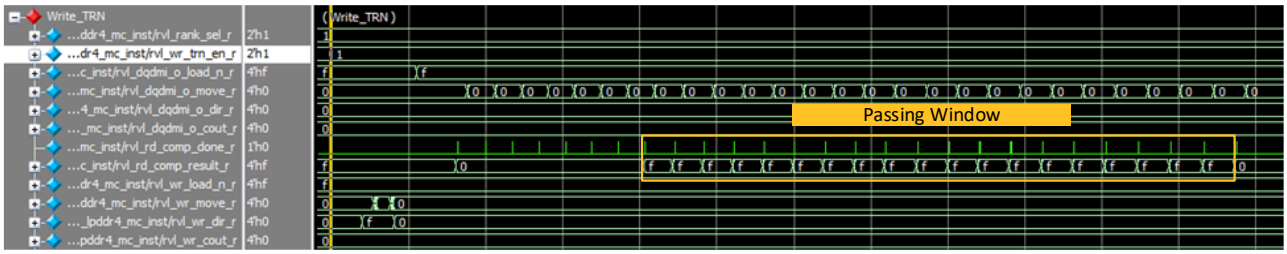


Figure 8.13. Initial Write Training Simulation Waveform

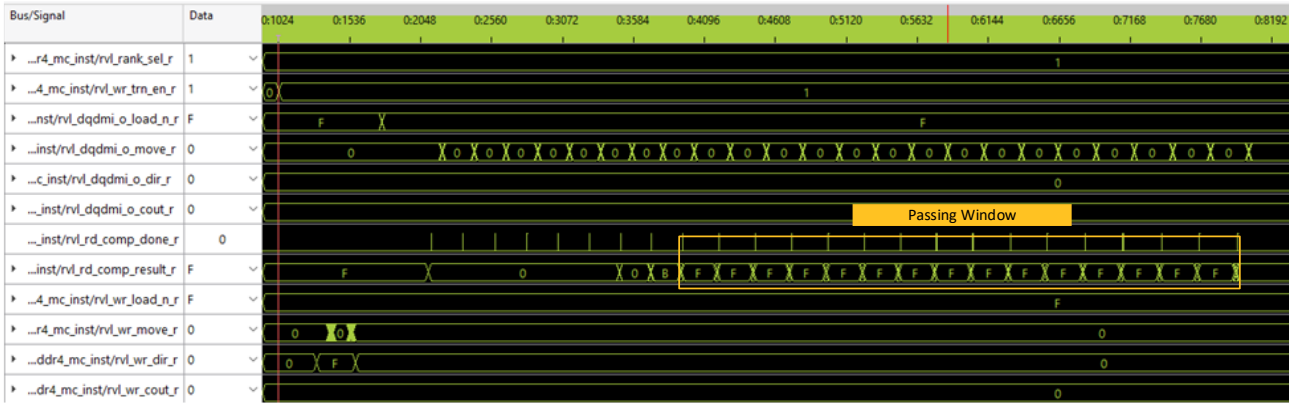


Figure 8.14. Initial Write Training Reveal Capture

The final write training starts when rvl_wr_trn_en_r=2. It uses the trained DRAM DQ_Vref value and performs similar routine with the initial Vref value with the following modifications:

- Starts with the DQ/DMI clock delay = 0.
- Use DQ/DMI output delay for incrementing sweep (performed first) and use DQ/DMI clock delay for decrementing sweep. Note that only the later can have negative delay value (the 0 delay is internally equal to the DLL code).
- The final delay setting is the average of the maximum and minimum passing delays and it is expected to be positive. Thus, for the final delay setting, the DQ/DMI clock delay is 0 and the DQ/DMI output delay is set to the calculated value.

Figure 8.15 and Figure 8.16 shows the incrementing sweep using the DQ/DMI output delay, followed by delay code reset and then by the decrementing sweep using the DQ/DMI clock delay. Note that the DQ/DMI clock delay sweep may be omitted if both the maximum and minimum edges of the valid window has been found during the DQ/DMI output delay sweep that is, when the first sample(s) of the DQ/DMI output delay sweep already fail (indicating the minimum edge has been identified). In dual-rank configurations, the rvl_cs_r signal indicates the rank (0 = rank 0, 1 = rank 1). For more information on write training, refer to the Write Training Description section of this user guide.

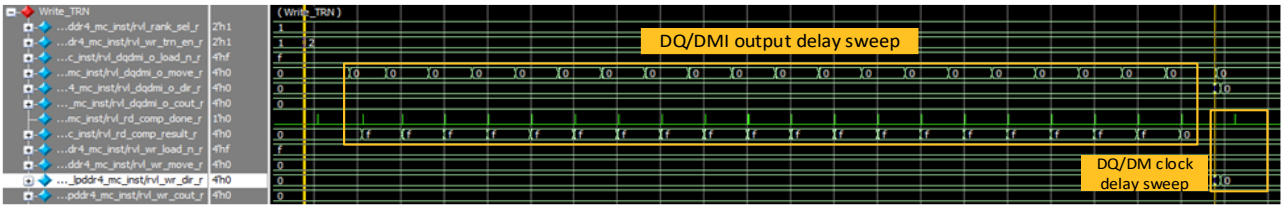


Figure 8.15. Final Write Training Simulation Waveform

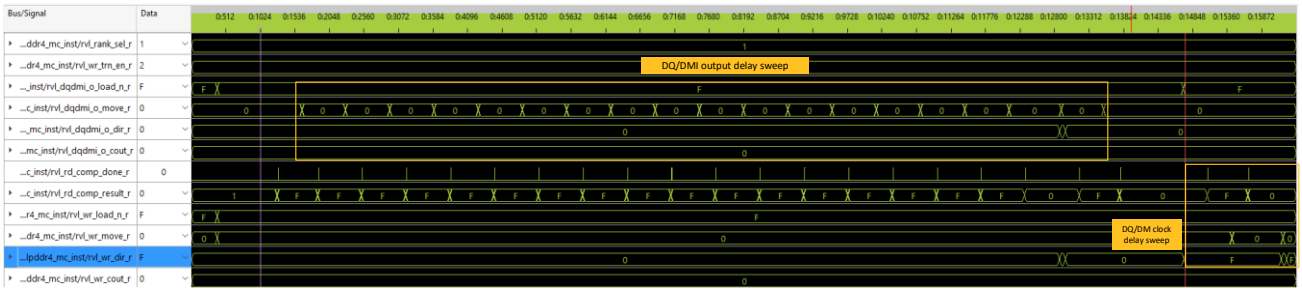


Figure 8.16. Final Write Training Reveal Capture

The DRAM-side Vref training performs similar DQ/DMI output delay sweep routine as the initial write training and is repeated across multiple DQ Vref values. It is not recommended to debug it with reveal since process is very long, a better debug feature is needed. This will be addressed in future enhancement.

The initial write training may fail if the write DQ/DQS/DM signal quality is bad. The combination of following Training Settings IP parameters improve the write DQ signal quality (hence the write training as well), it is recommended to set these based on the SI/PI simulation with your board design:

- Initial Memory DQ_VREF Value – the initial value during bring-up may be the default value or from the SI/PI simulation. However, if the example design recommends a different Memory DQ_Vref value, use it.
- MC DQ Drive Impedance
- DQ_ODT Value

Appendix A. Resource Utilization

Note: Resource utilization values in this section are provided for reference only and may change based on the compilation strategy and selected tool options.

Table A.1 shows the configuration and resource utilization for IP Core v2.7.0 implemented for LFCPNX-100-9LFG672C using Synplify Pro of Lattice Radiant software 2026.1.

Table A.1. Resource Utilization for IP Core v2.7.0

Configuration	sclk_o Fmax[1] (MHz)	Registers	LUTs[2]	EBR	IDDR/ODDR/TDDR
DDR Bus Width = 16, Others = Default	133.25	7,279	10,367	25	65 (16+29+20)
Default	133.25	9,510	12,347	33	121 (32+49+40)
DDR Bus Width = 16, Number of Ranks = 2, Others = Default	100	7,671	11,907	25	67 (16+31+20)
DDR Bus Width = 32, Number of Ranks = 2, Others = Default	100	9,782	13,782	33	123 (32+51+40)

Notes:

1. The sclk_o is derived from DDR Command Frequency divided by 4 (Gearing Ratio 8:1). When the Number of Ranks is 1, DDR Command Frequency reaches up to 533 MHz. When the Number of Ranks is 2, DDR Command Frequency reaches up to 400 MHz.
2. The resource usage number in the LUTs column represents the total LUT4 usage, including logic, distributed RAM and ripple logic.

Appendix B. Known Issue

Enabling the power-down feature for dual-rank mode may cause intermittent hardware failures. This issue will be addressed in a future release.

References

- LPDDR4 Memory Controller IP Core for Nexus Devices IP Release Notes (FPGA-RN-02057)
- CertusPro-NX LPDDR4 Memory Controller Driver User Guide (FPGA-TN-02378)
- LPDDR4 Memory Interface Module User Guide (FPGA-IPUG-02154)
- Lattice Radiant Timing Constraints Methodology User Guide (FPGA-AN-02059)
- Debugging with Reveal Usage Guidelines and Tips Application Note (FPGA-AN-02060)
- CertusPro-NX web page
- MachXO5-NX web page
- MachXO5T-NX Development Board web page
- AMBA AHB Protocol Specification
- AMBA AXI Protocol Specification
- AMBA APB Protocol Specification
- LPDDR4 JEDEC Standard
- Lattice Radiant FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, 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 2.2, IP v2.7.0, June 2026

Section	Change Summary
All	Updated the IP version to *2.7.0* and minor editorial fixes.
Abbreviations in This Document	• Added CDC, CSR, DDRDLL, and VT in this section. • Updated the definition of DFI to *DDR PHY Interface*.

Introduction	• Updated the following in Table 1.1. Quick Facts:
	  • Lattice Implementation: *IP Core v2.7.0, Lattice Radiant Software 2026.1*.
	  • Added supported devices: *LFMXO5-55TDQ, LFMXO5-65TDQ, and UT24CP.*
	• Updated the Timing Model status of CertusPro-NX to *Final* in Table 1.2. LPDDR4 Memory Controller IP Support Readiness.
	• Updated the ECC entry to *No* in Table 1.3. Features Overview.

Functional Description	• Reworked the Clocking and Reset section content.
	  • Added the General Clocking and Reset section with Input Clocks, Input Resets, and Reset-Assertion Expectations (to Prevent AXI Bus Hang-Up) subsections.
	  • Added Table 2.2. AXI Reset Dependency vs. Bus State.
	• Added a step: *Poll STATUS_REG until phy_ready*; to start the initialization and training sequence of the LPDDR4 SDRAM device in the LPDDR4 Calibration section.
	• Removed the *SETTINGS_REG[16] short/long description* from the Periodic ZQ Calibration section.

IP Parameter Description	• Updated Table 3.6. Memory Device Timing Setting Attributes.
	  • Updated the TRFC (tCLK) selectable values to *24 – 65,536.*
	  • Changed *TZQCAL (tCLK)* to *TZQLAT (tCLK)*.
	• Appended *TZQCAL* in ZQ Calibration Start to Latch (usec) attribute in Table 3.7. Periodic Event Setting Attributes.
	• Updated the *MC ODT Value* selectable values to *120 Ω, 80 Ω, 60 Ω, 48 Ω, 40 Ω* in Table 3.10. MC I/O Settings Attributes.

Signal Description	• Corrected the ahbl_htrans_i width to *2* in Table 4.3. AHB-Lite Interface Port Definitions. • Changed the ddr_cke_o width to *CS_WIDTH* in Table 4.6. LPDDR4 Interface Port Definitions.
Register Description	• Renamed the Field [2] name to *power_down_en* in Table 5.3. Feature Control Register. • Renamed the Field [1] name to *cpu_reset_n* in the Table 5.6. Reset Register. • Removed the duplicate *refresh_rate heading* in the Status Register (STATUS_REG) (0x24) section. • Updated the *phy_ready* field description to both logic states and APB behavior during initialization in the Status Register (STATUS_REG) (0x24) section.

Designing and Simulating the IP	• Updated the LPDDR4 Memory Controller Example Design GUI in the Running on Hardware section.
	  • Added the *x = Exit* option to the serial terminal menu.
	  • Added the instruction that pressing *Enter* alone selects *option 1* and exits the menu.
	  • Added the *RAND2 (2-beat random address burst)* data access pattern as pattern 8 in the Memory Data Access Pattern Definitions.
	  • Included *pattern 8* in the Data Access Check terminal output sequence.
	  • Added the *HARDWARE VALIDATION PASS* and *Exiting messages* to the example design terminal output.

Appendix A. Resource Utilization	• Added a disclaimer note regarding Resource utilization values. • Updated values in Table A.1. Resource Utilization for IP Core v2.7.0.

Section	Change Summary
Appendix B. Known Issue	• Removed the fixed issue regarding LPDDR4 SDRAM operating at 300 MHz. • Documented the new known issue that *enabling the power-down feature in dual-rank mode may cause intermittent hardware failures*.

Revision 2.1, IP v2.6.4, March 2026

Section	Change Summary
All	Updated IP version to *2.6.4*.
Introduction	• Added LSE information in Table 1.1. Quick Facts. • Removed *hardware validation* and *note 1* in Table 1.2. LPDDR4 Memory Controller IP Support Readiness. • Added a write-up that the *LPDDR4 Memory Controller IP Core example design supports simulation and deployment on development boards* before the Features section. • Removed the *Hardware Support* section. • Added Periodic VT compensation on I/O delay in Features section.
Functional Description	• Updated the Read Training section. • Updated Figure 2.8. Read Training. • Updated VREF Training section. • Added Periodic VT Compensation section.
IP Parameter Description	• Added *note 2* regarding *Self-Refresh period* in Table 3.8. Memory Device Timing Definitions. • Updated *Command Address Slew Rate* default value in Table 3.10. MC I/O Settings Attributes. • Added *note 2* for *slew rate settings* in Table 3.13. Training Settings Definition.
Register Description	Added a note regarding the *Read Training operation* in the read_trn_en subsection.
Designing and Simulating the IP	• Added a statement in the Running on Hardware section indicating that the *LPDDR4 Memory Controller IP core has been hardware tested on the CertusPro-NX Versa board*. • Updated the *debug prints* in the Running on Hardware section.

Debugging	• Added guidance on using the *debug prints* in the Debug with the Example Design section.
	• Added a write-up on how to troubleshoot failures in all training subsections in the Debug with Reveal Analyzer section.
	  • Command Bus Training (CBT)
	  • Read Training
	  • Write Training

Appendix A. Resource Utilization	• Removed *LSE table* – formerly Table A.1. Resource Utilization for IP Core v1.3.0. • Updated the values of the Synplify Pro table –Table A.2. Resource Utilization for IP core v2.6.4, and renamed it to Table A.1. Resource Utilization for IP Core .
Appendix B. Known Issue	Reworked section content.

Revision 2.0, IP v2.6.3, December 2025

Section	Change Summary
All	Updated IP version to *2.6.3.*

Introduction	• Updated Table 1.1. Quick Facts.
	  • Added *Driver Support*.
	  • Added notes *1* and *2*.
	• Reworked the Licensing and Ordering Information section.

Functional Description	Updated the Clocking and Reset section. • Added write-up regarding implementation of the *reset domain crossing*.
IP Parameter Description	• Updated Number of Ranks attribute in Table 3.3. Memory Configuration Attributes. • Updated Local Data Bus Type attribute in Table 3.4. Local Data Bus Attributes.

Section	Change Summary
Register Description	Updated the Interrupt Status Register (INT_STATUS_REG) (0x10) section. • Removed the refresh rate of *not equal to 1x refresh* under the Temperature Change Interrupt (temp_change_int) register.
Designing and Simulating the IP	• Added *Example Design* in Table 7.1. Memory Controller Attribute Guidelines. • Added write-up regarding *DATA MASK* assertion immediately following the procedure for creating a simulation environment to simulate the LPDDR4 Memory Controller Example Design. • Added Figure 7.15 Macro to enable DATA MASK toggling.
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 1.9, IP v2.6.2, October 2025

Section	Change Summary
All	• Updated IP version to *2.6.2*. • Minor editorial fixes
IP Parameter Description	• Added *table note 1* and *table note 2* in Table 3.12. Trained Values Attributes. • Added the Example Design section.
Appendix A. Resource Utilization	Added *table note 2* in Table A.1. Resource Utilization for IP Core .
Appendix B. Known Issues	Documented known issue, *LPDDR4 SDRAM operating at 300 MHz experiences intermittent hardware failures. This issue will be addressed in future release*.

Revision 1.8, IP v2.6.0, June 2025

Section	Change Summary
All	• Updated IP version from *2.5.0* to *2.6.0*. • Minor editorial fixes.

Introduction	• Updated Table 1.1. Quick Facts.
	  • Removed Targeted Device
	  • Changed *Supported FPGA Family* to *Supported Devices*.
	• Updated Table 1.4. Minimum Device Requirements.
	  • Added the MachXO5-NX device.

Functional Description	• Updated the Burst Length (AxLEN) under the AXI4 section. • Updated the maximum allowable AxLEN table under the AXI4 section.

IP Parameter Description	• Added, *The Calculate button at the bottom is used to calculate the optimum parameter value for the PLL* in the introduction part of this section.
	• Updated Table 3.2. Clock Settings Attributes.
	  • Added DDR Command Actual Frequency (MHz) (6.25 – 800).
	• Updated Table 3.4. Local Data Bus Attributes.
	  • Maximum Burst Length: added 128 and 256
	• Updated Table 3.7. Periodic Event Setting Attributes.
	  • Changed Default value of Number of Outstanding Refresh to 7.
	• Updated Table 3.9. Training Settings Attributes.
	  • Updated the DDR Clock Delay Default value and the Dependency on Other Attributes column.
	  • Updated the Initial Memory DQ_VREF Default value.
	• Updated Table 3.10. MC I/O Settings Attributes.
	  • Updated the Dependency on Other Attributes of the CK/CS Slew Rate and DQA Slew Rate.
	  • Updated the MC ODT Default value.
	• Updated Table 3.11. Memory ODT Settings Attributes.

Section	Change Summary
	• Updated the DQ_ODT Default value. • Updated contents of Table 3.12. Trained Values Attributes.
Signal Description	• Updated Table 4.1. Clock and Reset Port Definitions. • Added supported frequency range to pclk_i.
Appendix A. Resource Utilization	Updated contents of Table A.1. Resource Utilization for IP Core .

Revision 1.7, IP v2.5.0, March 2025

Section	Change Summary
All	• Updated IP version from *2.4.0* to *2.5.0* • Removed all *x64 width* in this document. • Minor editorial fixes.
Abbreviations in This Document	Replaced *Acronyms* with *Abbreviations*.
Introduction	• Updated the LPDDR4 Memory Controller features section. • Deleted *BL32 and On The Fly (OTF)* from the statement *Burst length of BL16*. • Added IP Support Summary section. • Added Hardware Support section. • Deleted Table 1.4. IP Validation Summary.
Functional Description	• Added the statement, *This also means that there will be two clock cycle of latency upon reset deassertion for the IP Core to be fully out of reset* into the Clocking and Reset section. • Updated the AXI4 section. • Added a bullet for the definition of AxLEN. • Added a bullet for the Example Use Case. • Changed the no. of outstanding writes the Memory Controller can support from *8 outstanding writes* and 8 outstanding reads to *4 outstanding writes* and 8 outstanding reads. • Deleted the statement, *Only the first and last beats of a burst can have incomplete byte strobes (WSTRB cannot go low in middle of burst).* • Changed *8'1h* to *8'hFF* in the LPDDR4 Calibration section. • Updated Figure 2.5. Shortened Initialization Sequence Simulation Waveform (TRN_OP_REG[0]=0). • Changed *trn_opr_i=0x1F* to *trn_opr_0xFF* in the Initialization and Training without APB Interface section. • Added the statement, *The Memory Controller starts counting the temperature check period when the training is completed* into the Temperature Tracking and Extended Temperature Support section.
IP Parameter Description	• Updated Table 3.4. Local Data Bus Attributes. • Updated the Number of Outstanding Reads from *1 for AHBL and 8 for AXI4* to *1*. • Added a Selectable Value of *1* to the *ID Width* attribute.
Register Description	Changed pll_lock access from *RD* to *RO* in Table 5.11. CLK_Change Register.
LPDDR4 Memory Controller Example Design	Changed TRN_OP_REG=*0x1F* to *0x5F* in the Synthesis Example Design section.
Designing and Simulating the IP	• Reworked the Constraints section. • Updated Table 7.3. Project Constraints. • Updated the *Action Required* column of the Memory Controller IP SDC file and Memory Controller IP LDC file, to *No. These constraints are automatically propagated*. • Added Option 2 = *Run Memory Training and Data Access In a loop With Reset* into the GUI of the LPDDR4 Memory Controller of Nexus Devices Example Design, under Running on Hardware section.
Appendix B. Known Issues	Added this section.

Section	Change Summary
References	Added the following to this section: • LPDDR4 Memory Controller IP Core for Nexus Devices IP Release Notes (FPGA-RN-02057) • CertusPro-NX LPDDR4 Memory Controller Driver User Guide (FPGA-TN-023780)

Revision 1.6, June 2024

Section	Change Summary
All	Minor editorial fixes. Replaced ModelSim with *QuestaSim*.
Disclaimers.	Updated this section.
Inclusive Language	Added this section.

Introduction	• Updated IP Core v2.x.x only to *IP Core v2.x.x and up*.
	• Updated IP Core v1.x.x and v2.x.x to *IP Core v1.x.x and up*.
	• Updated IP Core v2.1.x to *IP Core v2.1.x and up*.
	• Table 1.1.
	  • Updated from Lattice Radiant software 2023.1 to *Lattice Radiant software 2024.1*.
	• Updated the Features subsection
	  • Added Dual-rank support.
	  • Updated INCR with AxLEN = 0-63 to *INCR with AxLEN = 0-255.*
	• Updated Table 1.3. Features Overview
	  • Added table note 1.
	  • Updated dual-rank support to *Yes*.
	  • Updated On-Die Termination (ODT) to *Yes for both DQ and CA.*
	  • Updated VREF Training support to *Yes*.
	• Updated Table 1.5. IP Validation Summary.
	  • Added table note 2.
	  • Added CertusPro-NX (LPDDR4 dual-rank mode)
	  • Added MachXO5T-NX (LPDDR4 dual-rank mode)
	  • Added single-rank mode to both CertusPro-NX and MachXO5T-NX.

Functional Description	• Updated the Controller Engine subsection by adding the word *RANK* in the statement, *the Controller Engine accepts the received requests from the Bus Interface and is responsible for translating the address to row, column, rank, and bank format*.
	• Updated the Clocking and Reset subsection by changing the following:
	  • *LVSTL input* to *Differential input*
	  • *pll_rst_n_i* to *pll_refclk_i*
	• Updated the AXI4 subsection
	  • Updated IP Core v2.0.0 to *IP Core v2.x.x*.
	  • Updated the following Burst Length (AxLEN) to:
	  • *1-64 beat burst: when IP parameter MAX_BURST_LEN = 64*
	  • *1-128 beat burst: when IP parameter MAX_BURST_LEN = 128*
	  • *1-256 beat burst: when IP parameter MAX_BURST_LEN = 256*
	• Updated Table 2.3. Supported AXI4 Transactions.
	  • Updated *AxLEN[3:0]* to *AxLEN[7:0]*
	  • Updated *AxLEN[7:0]* Write to *[– MAX_BURST_LEN-1]*
	  • Updated *AxLEN[7:0]* Read to *[– MAX_BURST_LEN-1]*
	  • Updated table note 1 to:
	  • *For IP parameter AXI_DATA_WIDTH = 32, AxSIZE = [0, 1, 2] are supported.*
	  • *For IP parameter AXI_DATA_WIDTH = 64, AxSIZE = [0, 1, 2, 3] are supported.*
	  • *For IP parameter AXI_DATA_WIDTH = 128, AxSIZE = [0, 1, 2, 3, 4] are supported.*
	  • *For IP parameter AXI_DATA_WIDTH = 256, AxSIZE = [0, 1, 2, 3, 4, 5] are*

Section	Change Summary
	supported. • *For IP parameter AXI_DATA_WIDTH = 512, AxSIZE = [0, 1, 2, 3, 4, 5, 6] are supported.* • Updated LPDDR4 Calibration subsection. • Added the statement, *the following figure compares the LPDDR4 calibration sequence for single-rank and dual-rank modes. In dual-rank mode, if rank 0 were to encounter a failure, the training routine will continue with rank 1*. • Added Figure 2.4. LPDDR4 Calibration Sequence. • Added VREF Training subsection.
IP Parameter Description	• Updated the I/O Standard for Reference Clock attributes in Table 3.2. Clock Settings Attributes. • From LVSTL_I, LVSTL_II to *LVSTLD_I, LVSTLD_II.* • From equal to selection for I/O Buffer Type to *based off selection for I/O Buffer Type.* • Updated Table 3.3. Memory Configuration Attributes. • Updated number of ranks attributes. • Updated from 1 to *1,2*. • Updated from Display only (Only single rank is supported) to "—". • Updated Number of DDR Clocks (CK_WIDTH) attributes. • Updated from 1 to *1,2*. • Updated from Display only to "—". • Updated Number of Chip Selects (CS_WIDTH) attributes, • Updated from 1 to *1,2*. • Updated from Display only to *Display only. Calculated based off selection for Number of Ranks*. • Updated Table 3.4. Local Data Bus Attributes. • Added Maximum Burst Length attribute. • Updated the default value of Write Ordering Queue and Read Ordering Queue from 4 to *2*. • Updated Table 3.5. General Definitions. • Updated the description of Number of Chip Selects to *Specifies the number of chip select (CS) signals to be driven to SDRAM, dependent on the number of ranks selected.* • Added Maximum Burst Length attribute. • Updated the default value of *No of SCLK to enter Self-Refresh from no traffic* to *5000* in Table 3.7. Periodic Event Setting Attributes. • Updated Table 3.9. Training Settings Attributes. • Changed the default value of DDR Clock Delay Value to *90*. • Changed the default value of Initial MC DQ_VREF Value to *49*. • Changed the default value of Initial Memory CA_VREF Value to *30*. • Changed the default value of Initial Memory DQ_VREF Value to *20*. • Added Table 3.10. MC I/O Settings Attributes. • Added Table 3.11. Memory ODT Settings Attributes. • Reworked Table 3.12. Trained Values Attributes. • Reworked Table 3.13. Training Settings Definition.
Register Description	• Updated the register name of 0x1C to *CLK_CHANGE_REG* in Table 5.1. Summary of LPDDR4 Memory Controller IP Registers. • Updated the definition of *num_ranks* in the Features Control Register subsection. • Added *RANK* memory address in Table 5.4. Address Mapping for addr_translation=0. • Changed the title of Table 5.5 to Address Mapping Example for single rank. • Updated the Settings Register (SETTINGS_REG) (0x08) subsection. • Changed zq_cal_sel to *clk_freq*. • Updated the definition of *clk_freq*.

Section	Change Summary

	• Updated Table 5.7. Settings Register.
	  • Updated field settings register [31:17] to *[31:28]* and updated the width from 16 to *4*.
	  • Updated field settings register [16] to *[27:16].* Updated the name to *clk_freq.* Updated the access to *RO* and updated the width to *12*.
	• Added Clock Register (CLK_CHANGE_REG)(0x1C) subsection.
	• Updated the Training Operation Register (TRN_OP_REG) (0x20) subsection by adding the *mem_vref_training_en, mc_vref_training_en* and *ca_vref_training_en* registers.
	• Updated Table 5.12. Training Operation Register.
	  • Added *mem_vref_training_en, mc_vref_training_en* and *ca_vref_training_en* registers.
	  • Removed the reserved training operation register.
	• Updated the Status Register (STATUS_REG) (0x24) subsection.
	  • Added *rank1_done* register.
	  • Added *refresh_rate* register.
	  • Updated the definition of *error_on_rank* and *in_self_refresh* registers.
	  • Updated Table 5.13. Status Register.
	  • Added *rank1_done* status register.
	  • Added *rank0_done* status register.
	  • Updated the field of in_self_refresh to *[6:5]*.
	  • Updated the field of reserved to *[32:21]* and updated the width to *11*.
	  • Updated the field of reserved to *[7]* and updated the width to *1*.

LPDDR4 Memory Controller Example Design	• Updated Table 6.1. Supported Example Design Configurations.
	  • Added number of ranks attribute.
	  • Updated local data bus type to *AHBL, AXI4*.
	• Updated Table 6.2. Simulation Runtime Summary.
	  • Added *single-rank* to all LPDDR4 configuration.
	  • Added *x16, 533 MHz (dual-rank)* and *x32, 533 MHz (dual-rank).*

Designing and Simulating the IP	• Updated Table 7.1. Memory Controller Attribute Guidelines by adding the following:
	  • *Refer to the datasheet for the selected LPDDR4 memory device to ensure the channel density (DDR Density per Channel) is set correctly, in the GENERAL Memory Controller IP tab*.
	  • *Users are advised to perform signal integrity analysis with system IBIS model, using a tool such as HyperLynx, to determine the optimal I/O and ODT settings for custom boards to the Training Setting Memory Controller IP tab*.
	• Updated Table 7.2. Generated File List.
	  • Added the *constraints/constraint.sdc, testbench/delay_1dir.sv* and *testbench/dqs_grp_delay.sv* file names.
	  • Updated the definition of *eval/constraint.pdc and testbench/debug_c_code.sv* file names.
	• Updated the Design Implementation subsection.
	  • Updated the guidelines when placing pins for external memory interfaces:
	  • Removed the statement, *At least one VREF pin per HPIO bank that is used to implement an external memory interface, must be available and used as a reference voltage input*.
	  • Reworked list bullet no. 5 to, *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/routing*.
	• Updated Table 7.3. Project Constraints.
	  • Added the Memory Controller IP SDC file: *constraints/constrain.ldc* file name.
	  • Reworked the description and the action required column of the Memory

Section	Change Summary

	Controller IP PDC file: *eval/constraint.pdc* file name.
	• Reworked the Memory Controller IP subsection.
	• Updated Preparing the Bitstream subsection.
	  • Updated the steps to illustrate how to prepare the Memory Controller Example Design project and generate the associated bitstream, by adding the statement, *Make sure the Copy file to directory option is unchecked in steps no. 1 and 2*.
	• Updated Running on Hardware subsection.
	  • Updated the steps to illustrate how to program the FPGA board with the example design.
	  • Updated STEP NO.3 to, *Navigate to the bitstream file generated in Step 6 in* Preparing the Bitstream *section under this user guide and click Open*.
	  • Updated STEP NO.4 from to, *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*.
	  • Added the *LPDDR4 Memory Controller for Nexus Example Design* and removed the old user interface.
	• Updated the Example Design Simulation subsection.
	  • Added the statement*, Note that the memory model in the example design simulation requires a minimum simulation timescale of 100 fs as defined by the memory manufacturer. For proper functionality, ensure that the PLL reference clock is generated with a minimum timescale of 100 fs*.
	  • Updated the steps to illustrate how to prepare the Memory Controller Example Design project for simulation.
	  • Updated STEP NO.1 to, *Before simulating the example design, steps 1-2*.
	  • Added the statement, *In the Add Existing File dialog box, make sure the check the Copy file to directory option is unchecked in STEP NO.1*.
	  • Removed in STEP NO.4 the statement, *By default the simulation will run for 100 µs, which allows users to configure the waveform to log signals of interest in the QuestaSim simulator before continuing. Users can then enter the following TCL command in QuestaSim to run the simulation until completion: run -all. Alternatively, if users wish to run the simulation with the default top-level signals, users can change the 100 µs value to 0 µs in order to execute the simulation completely*.
	  • Updated the following figures.
	  • Figure 7.10. Simulation Wizard
	  • Figure 7.11. Adding and Reordering Simulation Source Files
	  • Figure 7.12. Parsing Simulation HDL Files
	  • Figure 7.13. Simulation Summary
	  • Updated the Simulation Top Model, *set to tb_top* in STEP No.3 to align to Figure 7.12. Parsing Simulation HDL Files.

Debugging	• Reworked the Debug with the Example Design subsection. • Reworked the Debug with Reveal Analyzer subsection by updating this paragraph. • *For general purpose debug, the rvl_trn_stat_done and rvl_trn_stat_err signals can be used to determine if the training of the LPDDR4 interface has passed or failed. The rvl_trn_stat_done signal indicates a particular stage of training has completed. The expected result is that all bits are set to 1, which would indicate that the following stages have completed from MSB to LSB: write training, read training, write leveling, command bus training, initialization of PHY. The rvl_trn_stat_err signal indicates an error has occurred during a particular stage of training. The expected result is that all bits are set to 0, which would indicate that the following stages have completed successfully from MSB to LSB: write training, read training, write leveling, command bus training. Examples are included in Figure 8.1 and Figure 8.2.* • Added the statement, *In dual-rank configurations, the rvl_cs_r signal indicates the rank (0 = rank 0, 1 = rank 1)*, in the following subsections: • Command Bus Training (CBT)

Section	Change Summary
	• Write Leveling • Read Training • Write Training
Appendix A	• Updated IP Core v2.1.0 to *IP Core v2.2.0*. • Updated Lattice Radiant software 2023.1. to *Lattice Radiant software 2024.1*. • Reworked Table A.1. Resource Utilization for IP Core .

Revision 1.5, September 2023

Section	Change Summary
All	• Changed the document title from *Memory Controller IP Core - Lattice Radiant Software* to *LPDDR4 Memory Controller for Nexus Devices*. • Reworked document content and structure for clarity.
Acronyms in this document	• Reworked section contents.
Introduction	• Reworked section contents. • Reworked *section 4 Ordering Part Number* and renamed *to subsection 1.3 Licensing and Ordering Information*. • Added IP Validation Summary and Minimum Device Requirements subsection. • Reworked *subsection 1.3 Conventions* and renamed to *subsection 1.6 Naming Conventions*.
Functional Description	• Reworked *subsection 2.5. Submodules description* and renamed *to subsection 2.1 IP Architecture*. • Added subsection 2.2 Clocking and Reset. • Reworked *subsection 2.2.1 AHB-Lite Interface* and moved under *subsection* 2.3 User Interfaces. • Reworked *subsection 2.2.4 AXI4 Interface* and moved under added *subsection* 2.3 User Interfaces. • Reworked *subsection 2.6. Initialization and Training* and renamed to subsection 2.4 LPDDR4 Calibration. • Reworked *subsection 2.7. Operations Details* and renamed to *subsection* 2.5 LPDDR4 Operation Description.
IP Parameter Description	Reworked *subsection 2.3 Attributes Summary* and moved this under IP Parameter Description section.
Signal Description	Reworked *subsection 2.2 Signal Description* and converted it to Signal Description section.
Register Description	Reworked *subsection 2.4 Register Description* and converted it to Register Description section.
LPDDR4 Memory Controller Example Design	Added this section.
Designing and Simulating the IP	Reworked *section 3 IP Generation, Simulation, and Verification* and renamed to this main section.
Debugging	Added this section.
Appendix A. Resource Utilization	Reworked section contents.
References	Reworked section contents.
Technical Support Assistance	Added FAQ link in this section.

Revision 1.4, December 2022

Section	Change Summary
Acronyms in This Document	Added AXI4 and revised CBI.

Section	Change Summary

Introduction	• Added AXI4 interface.
	• In Table 1.1:
	  • Removed Performance Grade
	  • Updated Supported User Interfaces and Resources in
	• In Features section:
	  • Updated Supported Transactions and Command Frequency.
	  • Indicated future enhancements.
	  • Removed Dynamic On-Die Termination (ODT) controls.
	  • Added AXI4 I/F in Table 1.2.

Functional Description	• In Overview section:
	  • Removed *ODT for generating ODT*.
	  • Added AXI4 I/F.
	• In Table 2.1:
	  • Added AXI4 I/F.
	  • Updated Clock and Reset group and Other Signals group.
	  • Corrected ahbl_htrans_i I/O.
	• Updated AHB-Lite Interface and Native Interface sections to indicate availability in IP Core v1.x.x.
	• Added AXI4 Interface section.
	• Added new attributes for IP Core v2.x.x in Table 2.6 and Table 2.7.
	• Removed RefClock (MHz) Selectable Values from Table 2.6.
	• In Register Description section:
	• Improve description of addr_translation field in FEATURE_CTRL_REG. Also indicated that num_ranks 1-Dual Rank is not supported.
	  • Updated COLW Local Address map in Table 2.12.
	  • Updated description of RESET_REG and TRN_OP_REG due to Enable APB I/F attribute.
	  • Updated subsection headings.
	• Indicated future enhancements in the Training Operation Register (TRN_OP_REG) (0x20) section.
	• Removed the On-Die Termination Control section.
	• Added Initialization and Training without APB I/F section.
	• Updated interface in the Write and Read Data Access section. Also added reference to Table 2.5.

IP Generation, Simulation, and Validation	• Updated the outline of this section. • Updated Figure 3.1, Figure 3.2, Figure 3.3, Figure 3.4, and Figure 3.5 for IP Core v 2.0.0. • Added Constraining the IP section. • Updated Hardware Validation section for IP Core v 2.0.0. Updated step 8 in the procedure for running hardware evaluation and added information on VREF training support.
Appendix A. Resource Utilization	Added Resource Utilization for IP Core v 2.0.0.
References	Added link to Lattice Radiant Software User Guide.
Technical Support Assistance	Added reference to the Lattice Answer Database on the Lattice website.
All	• Replaced *slave* with *subordinate* and *master* with *manager* when appropriate. • Minor adjustments in formatting and style.

Revision 1.3, January 2022

Section	Change Summary
All	Removed DDR3 features across the document.
Introduction	• Updated Lattice Radiant software version in Table 1.1. • Updated Features to add Native I/F, Memory DQ_VREF Training, Memory Controller

Section	Change Summary
	DQ_VREF training, and updated Command frequency.

Functional Description	• Updated Table 2.1 to remove hclk_i and hreset_n_i, add Native Interface, and update table note.
	• Added Native Interface and Native Interface to AXI4 Bridge sections.
	• Updated Error Log Register (ERROR_LOG_REG), Training Operation Register (TRN_OP_REG), and Status Register (STATUS_REG) section.
	• Updated the following in Table 2.5:
	  • DDR Command Frequency (MHz)
	  • Local Data Bus Type
	  • Data Width
	  • Periodic Event Setting Group
	• Updated Local Interface group in Table 2.6.
	• Separated Initialization and Training section from Operation Details section.
	• Split REFRESH Support section into Auto Refresh Support and Power Saving Feature.
	• Added Periodic ZQ Calibration and Temperature Tracking and Extended Temperature Support sections.

Core Generation, Simulation, and Validation	• Updated Figure 3.1, Figure 3.2, and Figure 3.3. • Updated Table 3.1. • Updated Running Functional Simulation section to add the simulation support. • Replaced Hardware Evaluation section with Hardware Validation.
Ordering Part Number	Updated content to add DDR4 part number with one year subscription license and remove part numbers for DDR3.
Appendix A. Resource Utilization	• Updated Lattice Radiant software version to 3.1. • Updated values in Table A.1.

Revision 1.2, June 2021

Section	Change Summary
All	Minor adjustments in formatting.
Introduction	Updated Features section content to correct acronym from INC4 and INC8 to *INCR4* and *INCR8*.
Functional Description	Improved description of rst_n_i in Table 2.1.
Core Generation, Simulation, and Validation	Updated Figure 3.1, Figure 3.2, and Figure 3.3.
Ordering Part Number	Added this section.
Appendix A. Resource Utilization	Updated Lattice Radiant software version to 3.0.
References	Updates section content to add CertusPro-NX webpage.

Revision 1.1, March 2021

Section	Change Summary
Introduction	Updated Features section content to correct acronym from INC4 and INC8 to *INCR4* and *INCR8*.
Functional Description	• Updated Table 2.1 to add hreset and preset port name. • Updated Table 2.2 to correct acronym from INC4 and INC8 to INCR4 and INCR8. • Updated Table 2.3 to correct Selectable Values, Default, and Dependency on Other Attributes. • Updated Feature Control Register (FEATURE_CTRL_REG) section to correct num_rank bullet and [31:7] and [16] in Table 2.7. • Updated Interrupt Status Register (INT_STATUS_REG) section to correct temp_change_int bullet and [4] in Table 2.12. • Updated Interrupt Enable Register (INT_ENABLE_REG) section to correct

Section	Change Summary
	temp_change_en bullet and [4] in Table 2.13. • Updated Interrupt Set Register (INT_SET_REG) section to correct temp_change_set bullet and [4] in Table 2.14.
Core Generation, Simulation, and Validation	Updated Figure 3.1 and Figure 3.2.
Appendix A. Resource Utilization	Updated values in Table A.1.

Revision 1.0, February 2021

Section	Change Summary
All	Initial release

www.latticesemi.com