DDR3 SDRAM Controller IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02086-1.4

October 2020

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 and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
DDR3	Double Data Rate Type 3
JEDEC	Joint Electron Device Engineering Council
SDRAM	Synchronous Dynamic Random Access Memory
FPGA	Field Programmable Gate Array
RTL	Register Transfer Level

1. Introduction

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

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

This design is implemented in Verilog. It can be targeted to CrossLink™-NX and Certus™-NX FPGA devices and implemented using the Lattice Radiant® software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.1. Quick Facts

Table 1.1 presents a summary of the LIFCL DDR3 SDRAM Controller IP Core.

Table 1.1. Quick Facts

IP Requirements	Supported FPGA Family	CrossLink-NX, Certus-NX
Resource Utilization	Targeted Devices	LIFCL-40, LIFCL-17, LFD2NX-40
	Supported User Interface	Native
	Resources	See Table A.1
Design Tool Support	Lattice Implementation	IP Core v1.0.x – Lattice Radiant Design Suite 2.0
		IP Core v1.1.x – Lattice Radiant Design Suite 2.1
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software User Guide.

1.2. Features

The key features of DDR3 SDRAM Controller IP Core include:

- Memory data path widths of 8, 16, 24, 32 bits
- Selectable gearing ratios: 4:1, 8:1
- x8 and x16 device configurations
- Programmable burst lengths of 8 (fixed), chopped 4 or 8 (on-the-fly), or chopped 4 (fixed)
- Programmable read and write CAS latency set
- Read burst type of nibble sequential or interleave
- Automatic DDR3 SDRAM initialization and refresh
- Automatic write levelling for each DQS
- Automatic read training for each DQS
- Power Down mode
- Dynamic On-Die Termination (ODT) controls
- Termination Data Strobe (TDQS) for x8 widths only
- I/O primitives manage read skews (read levelling equivalent)
- Automatic programmable interval refresh or user-initiated refresh
- Option for controlling memory reset outside the controller

The DDR3 SDRAM Controller IP Core supports the following devices:

- All LIFCL FPGA Family devices
- Interfaces to industry standard DDR3 SDRAM components and modules compliant with JESD79-3C, DDR3 SDRAM Standard
- Interfaces to DDR3 SDRAM at speeds up to 400 MHz/800 Mbps

1.3. Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

Signal Names that end with:

- _*n* are active low
- _*i* are input signals
- _*o* are output signals
- _*io* are bi-directional input/output signals

1.3.3. Attribute

The names of attributes in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Description

2.1. Overview

The DDR3 memory controller consists of three submodules: Memory Controller (MC) module, Physical Interface (PHY) module, and the Phase-Locked Loop (PLL) instance. The Submodules Description section briefly describes the operation of each of these submodules. Figure 2.1 provides a high-level block diagram illustrating the main functional blocks used to implement the DDR3 SDRAM Controller IP Core functions.

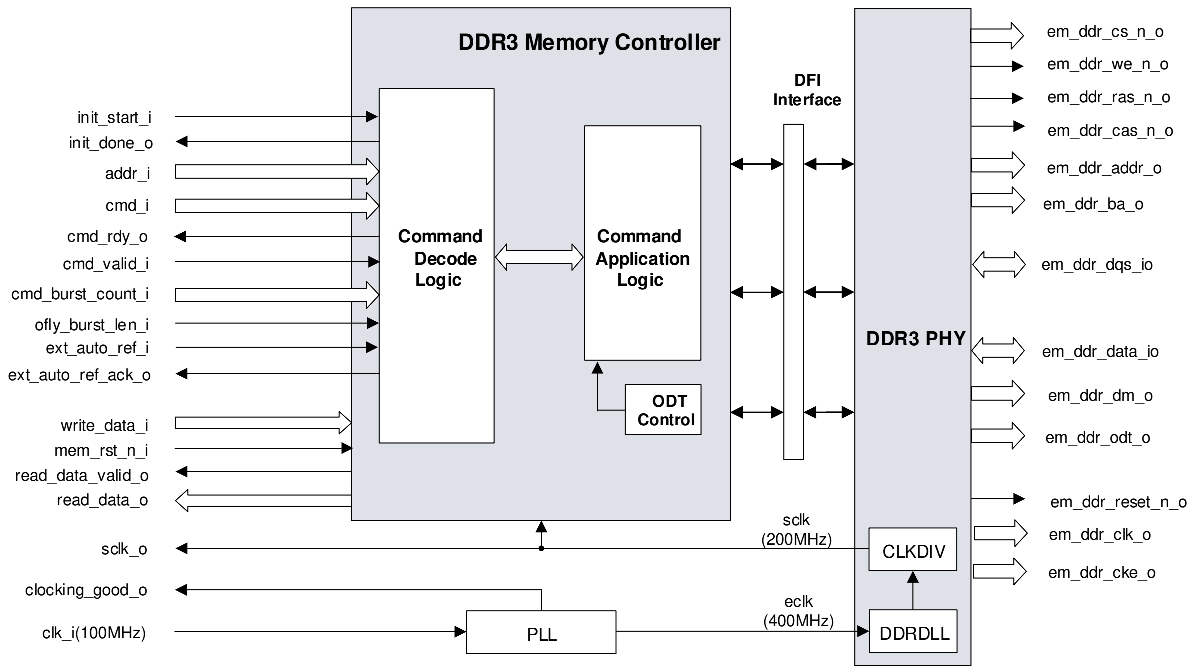

Figure 2.1. DDR3 SDRAM Controller IP Core Functional Diagram

2.2. Signal Description

Table 2.1 lists the input and output signals for DDR3 SDRAM Controller IP Core.

Table 2.1. DDR3 SDRAM Controller IP Core Signal Description

Port Name	I/O	Width	Description
Clock and Reset			
clk_i[4]	In	1	Reference clock to the PLL
clocking_good_o	Out	1	Signal from PLL indicating stable clock condition
sclk_o[4]	Out	1	System clock used by controller's core module You may use this clock for DDR3 controller interface logic.
rst_n_i	In	1	Asynchronous reset By default setting, this signal resets the entire IP core and the DDR3 memory when asserted.
mem_rst_n_i	In	1	Asynchronous reset when *Controller Reset to Memory* is checked from Module/IP Block Wizard Allows you to reset the memory device only. This signal does not reset the memory controller.
em_ddr_reset_n_o	Out	1	Asynchronous reset signal from controller to the memory device Asserted by the controller for the duration of power on reset, or active rst_n_i, or active mem_rst_n_i.
DDR3 SDRAM Memory Interface[1]			
em_ddr_clk_o	Out	*CLKO_WIDTH*	Memory clock generated by the controller
em_ddr_cke_o	Out	*CKE_WIDTH*	Memory clock enable generated by the controller
em_ddr_addr_o	Out	*ROW_WIDTH*	Memory address bus – multiplexed row and column address for the memory
em_ddr_ba_o	Out	3	Memory bank address
em_ddr_data_io	In/Out	*DATA_WIDTH*	Memory bi-directional data bus
em_ddr_dm_o	Out	*DATA_WIDTH*/8	DDR3 memory write data mask – to mask the byte lanes for byte-level write
em_ddr_dqs_io	In/ Out	*DQS_WIDTH*	Memory bi-directional data strobe
em_ddr_cs_n_o	Out	*CS_WIDTH*	Memory chip select
em_ddr_cas_n_o	Out	1	Memory column address strobe
em_ddr_ras_n_o	Out	1	Memory row address strobe
em_ddr_we_n_o	Out	1	Memory write enabl
em_ddr_odt_o	Out	*CS_WIDTH*	High Output Memory on-die termination control
Native Interface			
init_start_i	In	1	Initialization start request Should be asserted to initiate memory initialization either right after the power-on reset or before sending the first user command to the memory controller.
init_done_o	Out	1	Initialization done output Asserted for one clock period after the core completes memory initialization and write levelling. When sampled high, the input signal init_start_i must be immediately deasserted at the same edge of the sampling clock.
cmd_valid_i	In	1	Command and address valid input When asserted, the addr_i, cmd_i and cmd_burst_cnt_i inputs are considered valid.
cmd_rdy_o	Out	1	Command ready output When asserted, indicates that the core is ready to accept the next command and the corresponding address. This signal is active for one clock period.
cmd_i	In	4	User command input to the memory controller

Port Name	I/O	Width	Description
cmd_burst_cnt_i	In	5	Command burst count input Indicates the number of times a given read or write command is to be repeated by the controller automatically. Controller also generates the address for each repeated command sequentially as per the burst length of the command. Burst range is from 1 to 32 and *0* indicates 32 repetitions.
ofly_burst_len_i	In	1	On-the-fly burst length for current command 0 = BC4 1 = BL8 This input is valid only if Mode Reg0 is set for on-the-fly mode. If set, this input is sampled when cmd_valid_i and cmd_rdy_o are high.
addr_i[2]	In	ADDR_WIDTH	User read or write address input to the memory controller Refer the section Local-to-Memory Address Mapping for further details.
datain_rdy_o	Out	1	Data ready output When asserted, indicates the core is ready to receive the write data.
write_data_i[3]	In	*DSIZE*	Write data input from user logic to the memory controller The user side write data width is four times the memory data bus.
data_mask_i[3]	In	*DSIZE*/8	Data mask input for write data Each bit masks a corresponding byte of local write data.
read_data_o[3]	Out	*DSIZE*	Read data output from memory controller to the user logic.
read_data_valid_o	Out	1	Read data valid output When asserted, indicates the data on the read_data_o bus is valid.
ext_auto_ref_i	In	1	Refresh user request This signal is available only when the *External Auto Refresh Port* attribute is selected in the Module/IP Block Wizard.
ext_auto_ref_ack_o	Out	1	Completion of memory refresh in response to ext_auto_ref_i signal assertion. This pin is available only when the *External Auto Refresh Port* is checked in the Module/IP Block Wizard.
wl_err_o	Out	1	Write levelling error Indicates failure in write levelling. The controller does not work properly if there is a write levelling error. This signal should be checked when init_done_o signal is asserted.
rt_err_o	Out	1	Read Training error Indicates failure in Read Training process. The controller does not work properly if there is a Read Training error. This signal should be checked when init_done signal is asserted.

Notes:

1. The bit width of some DDR3 SDRAM Memory Interface signals are set by the attributes. Refer to Table 2.3 for the description of these attributes.
2. The bit width of addr_i is set by ADDR_WIDTH which is defined in Local-to-Memory Address Mapping section.
3. The bit width of write_data_i, data_mask_i, and read_data_o are set by DSIZE which is 4 * DATA_WIDTH for 4:1 gearing ratio and 8 * DATA_WIDTH for 8:1 gearing ratio.
4. The clk_i and sclk_o signals are named clk_in_i and sclk_out_o respectively in the IP Core v1.0.1 or earlier.

2.3. Attributes Summary

The configurable attributes of the DDR3 SDRAM Controller IP Core are shown in Table 2.2 and are described in Table 2.3. The attributes can be configured through the IP Catalog's Module/IP Block Wizard of the Lattice Radiant software. The attributes are arranged into tabs and related attributes are collected into groups. The 3 tabs are as follows:

- General Tab
  The General tab contains the attributes for configuring the target memory device and the IP Core features. These attributes are static; they can only be set in the Module/IP Block Wizard. The DDR3 SDRAM Controller IP Core must be regenerated to change the features set by these attributes.
- Memory Device Setting Tab
  The Memory Device Setting Tab contains the attributes for configuring the target memory device/module. The attributes under Mode Register Initial Setting Group are dynamic, which means, reset values are set from Module/IP Block Wizard and are dynamically changeable using LOAD_MR user commands. Refer to JESD79-3, DDR3 SDRAM Standard, for allowed values.
- Memory Device Timing Tab
  The attribute default displayed in this tab are the default values of the Micron DDR3 1Gb-25E memory module. These attributes can be modified by checking the Manual Adjust attribute. It is important that the attribute values in this tab are adjusted to the timing parameters of the memory device for the target application. The DDR3 SDRAM Controller IP Core also uses these timing parameters when generating memory commands.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
General Tab			
Device Information Group			
Interface Type	DDR3, DDR3L	DDR3	—
Gearing Ratio	4:1, 8:1	8:1	—
I/O Buffer Type	SSTL15_I, SSTL15_II	SSTL15_I	If Interface Type == DDR3
	SSTL135_I, SSTL135_II	SSTL135_I	If Interface Type == DDR3L
Select Memory	Micron DDR3 1Gb-25E, Micron DDR3 2Gb-25E, Micron DDR3 4Gb-25E, Custom	Micron DDR3 1Gb-25E	If *Gearing Ratio* == 4:1
	Micron DDR3 1Gb-187E, Micron DDR3 2Gb-187E, Micron DDR3 4Gb-187E, Custom	Micron DDR3 1Gb-187E	If *Gearing Ratio* == 8:1
Clock Settings Group			
Enable PLL	Checked	Checked	Display only
PLL Reference Clock from Pin	Checked, Unchecked	Checked	—
I/O Standard for Reference Clock	LVDS, SUBLVDS, SLVS, HSTL15_I, HSTL15D_I, LVTTL33, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS18H, HSTL15D_I, LVCMOS15, LVCMOS15H, LVCMOS12, LVCMOS12H, LVCMOS10H, LVCMOS10, LVCMOS10R	SLVS	—
RefClock (MHz)	25, 50, 75, 100, 111, 125, 150	100	—
MemClock (MHz)	300, 333, 400, 533	400	If *Gearing Ratio* == 8:1, 533 is available

Attribute	Selectable Values	Default	Dependency on Other Attributes
Memory Configuration Group			
Memory Type	On-board Memory	On-board Memory	—
Memory Data Bus Size	8, 16, 24, 32	32	Selectable values may change depending on the target device.
Configuration	X8, X16	X8	x16 is available when *Memory Data Bus Size* is 16 or 32
Rank Size	Single, Dual	Single	If *Memory Data Bus Size* == 32, fixed to Single due to limited I/O resource
Clock Width	1, 2	1	If *Rank Size* == Single: [1, 2]
	[2,4]	2	If *Rank Size* == Dual: [2, 4]
CKE Width	Calculated based on *Rank Size*	1	If *Rank Size* == Single: 1
		2	If *Rank Size* == Dual: 2
Local Interface			
Local Bus Type	Native	Native	Display only
Additional Configuration Group			
Data ready to Write Data delay	1, 2	1	Display only
2T Mode	Checked/Unchecked	Unchecked	*Rank Size* == 2
Write levelling	Checked/Unchecked	Checked	—
Controller reset to Memory	Checked/Unchecked	Unchecked	—
Memory Device Setting Tab			
Address Group			
Row Size	13, 14, 15, 16	14	Default value is set by *Select Memory*: • Micron DDR3 1Gb-<25E,187E>:14 • Micron DDR3 2Gb-<25E,187E>:15 • Micron DDR3 4Gb-<25E,187E>:16 • Custom: 13 Editable when *Select Memory* = Custom
Column Size	10, 11, 12	10	Editable when *Select Memory* = Custom
Auto Refresh Control Group			
Auto Refresh Burst Count	1, 2, 3, 4, 5, 6, 7, 8	8	—
External Auto Refresh Port	Checked/Unchecked	Unchecked	—
Mode Register Initial Setting Group			
Burst Length	Fixed 4 (BC4), Fixed 8 (BL8), On the fly	Fixed 8 (BL8)	—
CAS Latency	5, 6, 7, 8	5	Selectable values and default are updated based on *Select Memory* and *MemClock*.
Burst Type	Sequential/Interleave	Sequential	—
Write Recovery	5, 6, 7, 8, 10, 12	6	If *MemClock* == 533, minimum value and default value are both 8
DLL Control for PD	Slow Exit/Fast Exit	Slow Exit	—
ODI Control	RZQ/6, RZQ/7	RZQ/6	—
RTT_Nom (Ohm)	RZQ/2, RZQ/4, RZQ/6, RZQ/8, RZQ/12, Disabled	RZQ/6	—
Additive Latency	0, CL-1, CL-2	0	—
CAS Write Latency	5, 6, 7, 8, 10, 12	5	Selectable values and default are updated based on *Select Memory* and *MemClock*.
RTT_WR	RZQ/2, RZQ/4, Off	RZQ/4	—

Attribute	Selectable Values	Default	Dependency on Other Attributes
Memory Device Timing Tab			
Command and Address Timing Group			
Manually Adjust	Checked/Unchecked	Unchecked	—
TRTP (tCLK)	4–65536	4	Enabled when *Manually Adjust* is Checked
TWTR (tCLK)	4–65536	4	Enabled when *Manually Adjust* is Checked
TMRD(tCLK)	4–65536	4	Enabled when *Manually Adjust* is Checked
TMOD (tCLK)	12–65536	12	Enabled when *Manually Adjust* is Checked
TRCD (tCLK)	4–65536	6	Enabled when *Manually Adjust* is Checked
TRP (tCLK)	4–65536	6	Enabled when *Manually Adjust* is Checked
TRC (tCLK)	15–65536	20	Enabled when *Manually Adjust* is Checked
TRAS (tCLK)	12–65536	15	Enabled when *Manually Adjust* is Checked
TFAW (tCLK)	12–65536	16	Enabled when *Manually Adjust* is Checked
TRRD (tCLK)	4–65536	4	Enabled when *Manually Adjust* is Checked
Calibration Timing Group			
TZQINIT(tCLK)	512–65536	512	Enabled when *Manually Adjust* is Checked
TZQCS (tCLK)	64–65536	64	Enabled when *Manually Adjust* is Checked
TZQOPER (tCLK)	256–65536	256	Enabled when *Manually Adjust* is Checked
Refresh, Reset and Power Down Timing Group			
TCKE (tCLK)	3–65536	3	Enabled when *Manually Adjust* is Checked
TRFC (tCLK)	44–28080	44	Enabled when *Manually Adjust* is Checked Default value is set by *Select Memory*: • Micron DDR3 1Gb-25E:44 • Micron DDR3 2Gb-25E:64 • Micron DDR3 4Gb-25E:104 Custom: 140
TCKESR (tCLK)	4–65536	4	Enabled when *Manually Adjust* is Checked
TPD (tCLK)	3–65536	3	Enabled when *Manually Adjust* is Checked
TXPDLL (tCLK)	10–65536	10	Enabled when *Manually Adjust* is Checked
TXPR (tCLK)	48–65536	48	Enabled when *Manually Adjust* is Checked Default value is set by *Select Memory*: • Micron DDR3 1Gb-25E:48 • Micron DDR3 2Gb-25E:68 • Micron DDR3 4Gb-25E:108 Custom: 144
TREFI (tCLK)	44–4160	3120	Enabled when *Manually Adjust* is Checked
Write levelling and ODT Timing Group			
TWLMRD (tCLK)	40–65536	40	Enabled when *Manually Adjust* is Checked
TWLDQSEN (tCLK)	25–65536	25	Enabled when *Manually Adjust* is Checked
TWLO (ns)	0–9	4	Enabled when *Manually Adjust* is Checked
ODTH4 (tCLK)	4	4	Display only
ODTH8 (tCLK)	6	6	Display only

Table 2.3. Attributes Descriptions

Attribute	Description
General Tab	
Device Information Group	
Interface Type	Specifies the DDR3 Memory interface: DDR3 or DDR3L.
Select Memory	Some attribute default values are dependent on this attribute. The Micron DDR3 1GB -25E is provided as the default DDR3 memory device, the timing parameters of this memory device are listed in the Memory Device Timing tab as default values. The other available options are: Micron DDR3 2Gb-25E and Micron DDR3 4Gb-25E.
RefClock (MHz)	Specifies the reference input clock to PLL which generates the system clock (sclk_o) and memory clock (em_ddr_clk_o).
MemClock (MHz)	Specifies the frequency of the memory clock to memory device. The allowed values are 300 MHz, 333 MHz and 400 MHz. The default value is linked to the speed grade of Lattice device selected. This is the PLL output frequency which depends on the corresponding value of RefClock. For example, for MemClock value of 333 MHz the PLL RefClock should be set to 111 MHz.
Memory Configuration Group	
Memory Type	This attribute is for information only. Only On-board Memory type is supported.
Memory Data Bus Size (DATA_WIDTH)	Specifies the bit width of DDR3 data bus (em_ddr_data_io). If the memory module has a wider data bus than required, only the required data width should be selected.
Configuration	Selects the device configuration of the on-board memory. The memory controller supports device configurations x8, and x16.
Rank Size (CS_WIDTH)	Select the number of Chip selects (em_ddr_cs_n_o) required - Single or Dual. This also specifies the bit width of em_ddr_odt_o.
Clock Width (CLKO_WIDTH)	Specifies the number of clocks signals (em_ddr_clk_o) with which the IP Core drives the memory. The clocks signals are converted to differential pair in the FPGA pins. Please note that the differential pair signals are not shown in simulation.
CKE Width (CKE_WIDTH)	Specifies the number of Clock Enable (CKE) signals (em_ddr_cke_o) with which the IP Core drives the memory.
Local Interface	
Local Bus Type	Specifies the user interface in FPGA fabric side. Only Native Interface is currently supported.
Additional Configuration Group	
Data ready to Write Data delay	This option is for information only. User logic is allowed to send the write data to the controller after a one-clock cycle delay with respect to datain_rdy_o signal.
2T Mode	Enables or disables the 2T timing for command signals when *Rank Size* = 2 (Dual Rank DIMM or 2 Chip select) is selected.
Write levelling	This option allows you to enable or disable the Write Leveling operation of the DDR3 SDRAM Controller IP Core.
Controller reset to Memory	When this option is disabled (unchecked), the reset signals mem_rst_n_i and em_ddr_reset_n_o is no longer available and external logic should take care of the DDR3 memory reset. If the option is enabled, the IP Core responds to mem_rst_n_i signal. When it is 1’b0, it asserts m_ddr_reset_n_o signal for minimum 200 µs as per the DDR3 Memory requirement.

Attribute	Description
Memory Device Setting Tab	
Address Group	
Row Size (ROW_WIDTH)	Indicates the default Row Address size used in the selected memory configuration.
Column Size	Indicates the default Column Address size used in the selected memory configuration.
Auto Refresh Control Group	
Auto Refresh Burst Count	Indicates the number of Auto Refresh commands that the DDR3 SDRAM Controller IP Core is set to send in a single burst. Refer to REFRESH Support for more details.
External Auto Refresh Port	Specifies the generation of refresh commands to the memory. If Unchecked: the controller automatically generates refresh commands to the memory at the interval defined by the *Auto Refresh Burst Count* and memory refresh timing requirement. If Checked: The user logic is allowed to generate a Refresh request to the controller via ext_auto_ref_i signal. Refer to REFRESH Support for more details.
Mode Register Initial Setting Group	
Burst Length	Sets the Burst length value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
CAS Latency	Sets the CAS Latency value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
Burst Type	Sets the Burst Type value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
Write Recovery	Sets the Write Recovery value in Mode Register 0 during initialization. The value is in terms of Memory clock. This value remains until you write a different value to the Mode Register.
DLL Control for PD	Sets the DLL Control for Precharge PD value in Mode Register 0 during initialization. This value remains until you write a different value to the Mode Register.
ODI Control	Sets the Output Driver Impedance Control value in Mode Register 1 during initialization. This value remains until you write a different value to the Mode Register.
RTT_Nom (Ohm)	Sets the nominal termination, Rtt_Nom, value in Mode Register 1 during initialization. This value remains until you write a different value to the Mode Register.
Additive Latency	Sets the Additive latency, AL, value in Mode Register 1 during initialization. This value remains until you write a different value to the Mode Register
CAS Write Latency	Sets the CAS Write Latency, CWL, value in Mode Register 2 during initialization. This value remains until you write a different value to the Mode Register.
RTT_WR	Sets the Dynamic ODT termination, Rtt_WR, value in Mode Register 2 during initialization. This value remains until you write a different value to the Mode Register.
Memory Device Timing Tab*	
Command and Address Timing Group	
Manually Adjust	Checking this box allows you to manually set any of the memory timing parameters. If you need to change any of the default values, the Manual Adjust checkbox must be checked. This selection enables you to modify the memory timing parameters.
TRTP (tCLK)	Internal READ Command to PRECHARGE Command delay
TWTR (tCLK)	Delay from start of internal write transaction to internal read command
TMRD(tCLK)	Mode Register Set command cycle time
TMOD (tCLK)	Mode Register Set command update delay
TRCD (tCLK)	ACT to internal read or write delay time
TRP (tCLK)	PRE command period
TRC (tCLK)	ACT to ACT or REF command period
TRAS (tCLK)	ACTIVE to PRECHARGE command period
TFAW (tCLK)	Four activate window for 1KB/2KB page size
TRRD (tCLK)	ACTIVE to ACTIVE command period for 1KB/2KB page size

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

*Note: The Memory Device Timing parameters listed in this tab are standard parameters as defined in JESD79-3C, DDR3 SDRAM Standard. Refer to the memory device data sheet for detailed descriptions and allowed values of these parameters.

2.4. Submodules Description

2.4.1. DDR3 Memory Controller Module

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

2.4.1.1. Command Decode Logic

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

2.4.1.2. Command Application Logic

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

2.4.1.3. On-Die Termination Control

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

2.4.2. DDR3 PHY Module

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

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

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

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

2.4.2.1. Initialization

The Initialization block performs the DDR3 memory initialization sequence as defined by JEDEC protocol. After power on or a normal reset of the DDR3 controller, memory must be initialized before sending any command to the controller. It is your responsibility to assert the init_start input to the DDR3 controller to start the memory initialization sequence. The completion of initialization is indicated by the init_done_o signal.

2.4.2.2. Write Leveling

The write leveling block adjusts the DQS-to-CLK relationship for each memory device, using the write level mode of the DDR3 SDRAM when the fly-by topology is implemented. Write leveling is always done immediately after a memory initialization sequence if *Write levelling* attribute is enabled. When the init_done_o signal is asserted after the initialization process, it also indicates the completion of write leveling. Along with the assertion of init_done_o, the signal wl_err_o is also asserted if the write leveling process is not successful.

DDR3 memory module adopted fly-by topology for the address, command, control and clock signals for better signal integrity. This reduces the number of stubs and length but it causes flight time skew between the DQS and CLK. Therefore, DDR3 needs to allow the controller to compensate for the skew of the DQS signal delays to those signals at the DDR3 DRAM side through its write leveling capability. When *Write leveling* attribute is checked in the Module/IP Block Wizard, the PCB for the on-board memory application must be routed using the fly-by topology. Otherwise, write leveling failures may occur due to the lack of guaranteed DQS to CLK edge relationship at the beginning of write level training. Due to this reason, the write leveling option must be disabled if the PCB does not utilize fly-by routing for write leveling.

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

2.4.2.3. Read Training

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

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

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

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

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

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

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

2.4.2.4. Data Path Logic

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

2.5. Operations Details

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

Table 2.4. Native Interface Functional Groups

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

2.5.1. Initialization Control

DDR3 memory devices must be initialized before the memory controller can access them. The memory controller starts the memory initialization sequence when the init_start_i signal is asserted from the Native Interface. Once asserted, the init_start_i signal needs to be held high until the initialization process is completed. The output signal init_done_o is asserted High for one clock cycle indicating that the core has completed the initialization sequence and is now ready to access the memory. The init_start_i signal must be negated as soon as init_done_o is sampled high at the rising edge of sclk_o. If the init_start_i is left high at the next rising edge of sclk_o, the memory controller takes it as another request for initialization and starts the initialization process again. Memory initialization is required only once, immediately after the system reset.

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

- During Power-up initialization: 200 ns
- During Reset Initialization with Stable Power: 100 ns

Currently, it is your responsibility to ensure that the above minimum reset assert duration is met.

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

If *Write levelling* attribute is checked, the IP Core performs write levelling for all available ranks and stores the write level delay values.

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

2.5.2. Command and Address

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

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

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

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

When cmd_burst_cnt_i and ofly_burst_len_i are 0, the controller performs 32 commands (reads or writes). The cmd_burst_cnt_i input is sampled the same way as cmd signal. The timing diagram of Command and Address group signals is shown in Figure 2.2. When cmd_burst_cnt_i is sampled with a value greater than *00001*, and the command queue becomes full, the cmd_rdy_o signal is not asserted, and the memory address is automatically increased by the core until the current command burst cycle is completed.

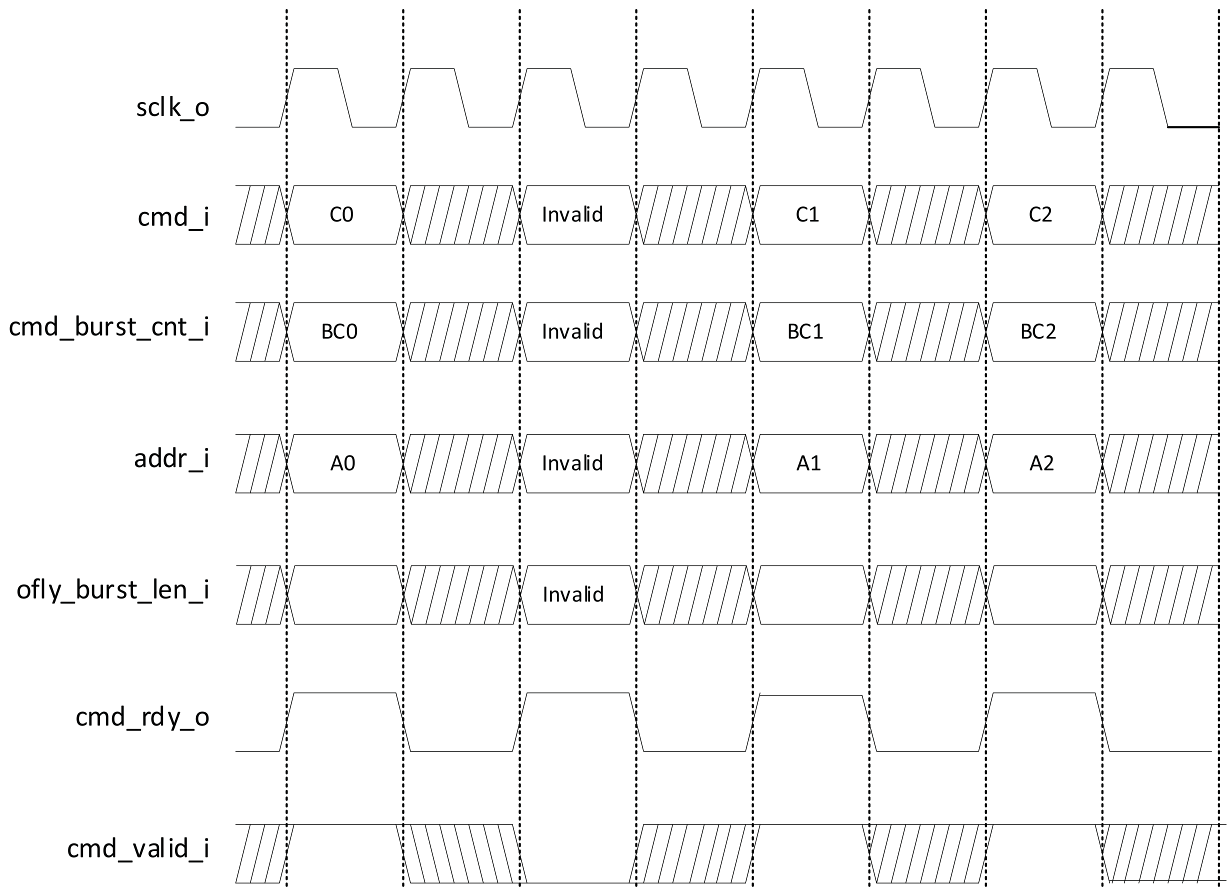

Figure 2.2. Timing of Command and Address

2.5.3. User Commands

You can initiate a request to the memory controller by loading a specific command code in cmd input along with other information such as memory address. The command on the cmd bus must be a valid command. Lattice defines a set of valid memory commands as shown in Table 2.5. All other values are reserved and considered invalid.

Table 2.5. Defined User Commands

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

Notes:

- The controller accepts only the command codes listed above as legal commands. Any other command code is discarded as invalid command.
- The controller discards Self Refresh Entry or Power Down Entry command if the memory is already in Self Refresh mode or Power Down mode respectively.
- The controller discards Self Refresh Exit or Power Down Exit command if the memory is already not in Self Refresh mode or Power Down mode respectively.

2.5.4. WRITE

You can initiate a memory write operation by asserting cmd_valid_i along with the WRITE or WRITEA command and the address. After the WRITE command is accepted, the memory controller core asserts the datain_rdy_o signal when it is ready to receive the write data from the user logic to write into the memory. Since the duration from the time a write command is accepted to the time the datain_rdy_o signal is asserted is not fixed, the user logic needs to monitor the datain_rdy_o signal. Once datain_rdy_o is asserted, the core expects valid data on the write_data bus one or two clock cycles after the datain_rdy signal is asserted. You can program the write data delay by setting the value for *Data ready to Write Data delay* attribute, providing flexible backend application support. For example, setting the value to 2 ensures that the core takes the write data in proper time when the local user interface of the core is connected to a synchronous FIFO module inside the user logic. Figure 2.3 shows two examples of the local user interface data write timing. Both cases are in BL8 (Burst length 8) mode. The upper diagram shows the case of one clock cycle delay of write data, while the lower one displays a two clock-cycle delay case. The memory controller considers D0, DM0 through D5, DM5 valid write data.

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

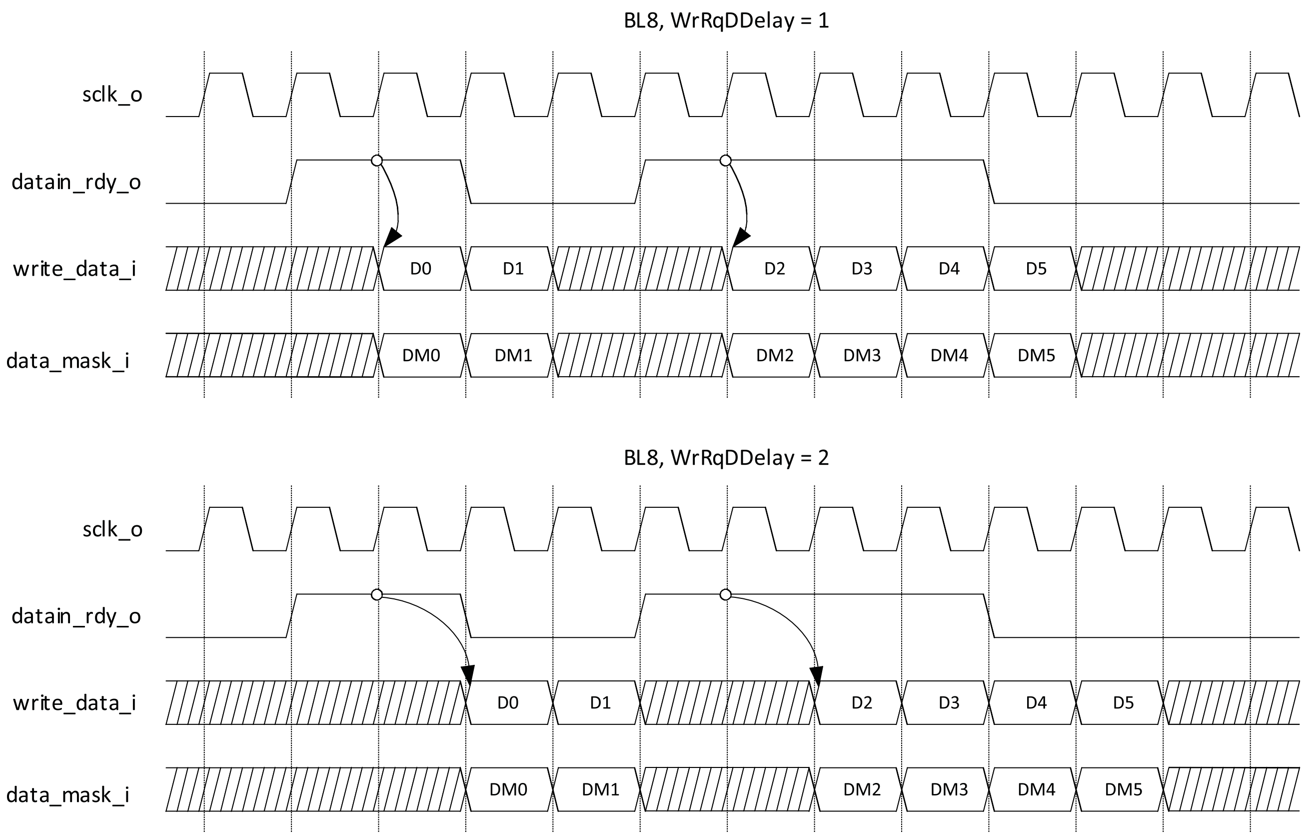

Figure 2.3. One-Clock vs. Two-Clock Write Data Delay

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

2.5.5. WRITEA

WRITEA is treated the same way as WRITE command, except that the IP Core issues a Write with Auto Precharge command to the memory, instead of just a Write command. This causes the memory to automatically close the current row upon completing the write operation.

2.5.6. READ

When the READ command is accepted, the memory controller core accesses the memory to read the addressed data and brings the data back to the local user interface. Once the read data is available on the local user interface, the memory controller core asserts the read_data_valid_o signal to tell the user logic that the valid read data is on the read_data_o bus. The read data timing on the local user interface is shown in Figure 2.4. Read operation follows the same row status checking scheme as mentioned in write operation. Depending on current row status, the memory controller generates ACTIVE and PRECHARGE commands, as required (please refer to the description mentioned in Write operation for more details).

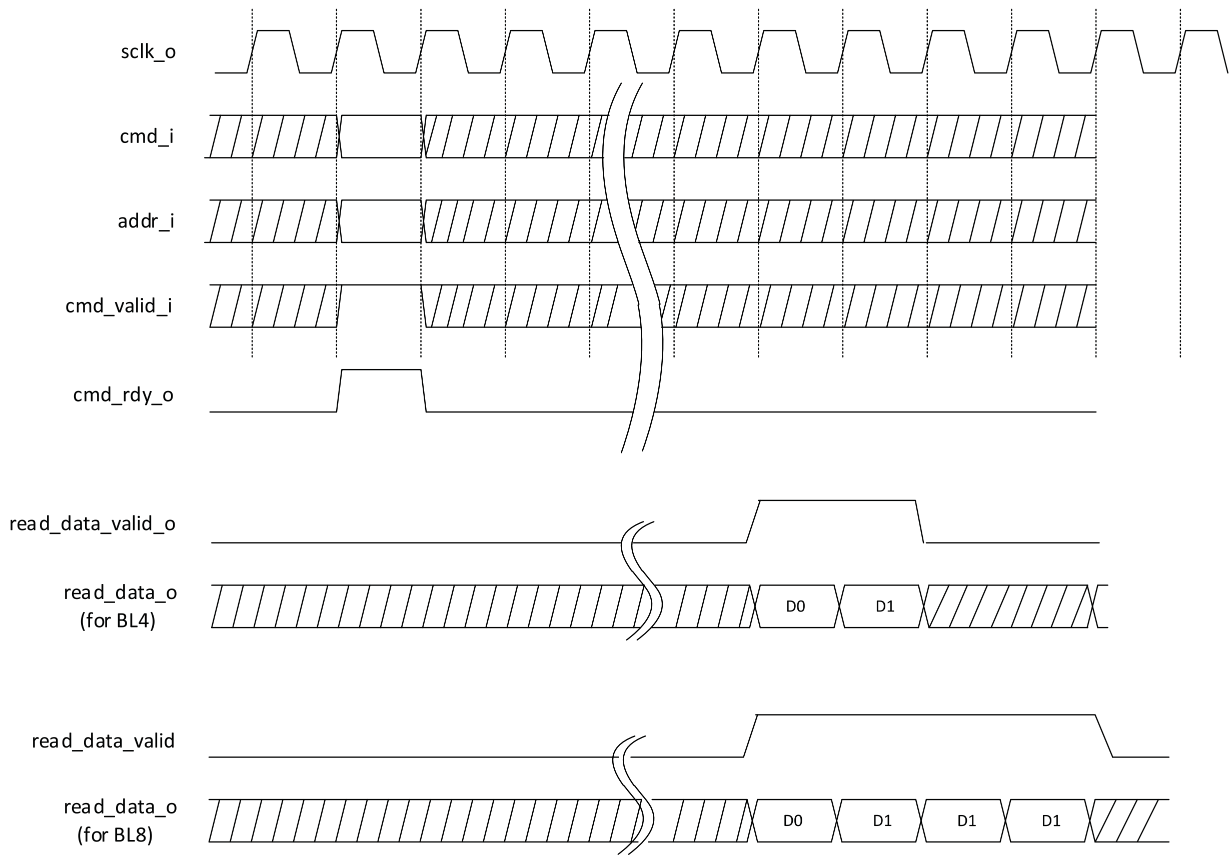

Figure 2.4. User-Side Read Operation

2.5.7. READA

READA is treated in the same way as READ command except for the difference that the IP Core issues a Read with Auto Precharge command to the memory instead of Read command. This makes the memory automatically close the current row after completing the read operation.

2.5.8. REFRESH Support

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

It is recommended that the maximum number be used if the DDR3 interface throughput is a major concern of the system. If it is set to 8, for example, the core sends a set of eight consecutive Auto Refresh commands to the memory at once when it reaches the time period of the eight refresh intervals (*TREFI* x 8). Bursting refresh cycles increases the DDR3 bus throughput because it helps keep core intervention to a minimum. When a refresh burst is used, the controller issues a Precharge command only for the first Refresh command and the subsequent Refresh commands of the burst are issued without the associated Precharge commands. This is to improve the DDR3 throughput.

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

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

2.6. Local-to-Memory Address Mapping

Mapping local addresses to memory addresses is an important part of a system design when a memory controller function is implemented. You must know how the local address lines from the memory controller connect to those address lines from the memory because proper local-to-memory address mapping is crucial to meet the system requirements in applications such as a video frame buffer controller. Even for other applications, careful address mapping is generally necessary to optimize the system performance. In the memory side, the address (A), bank address (BA) and chip select (CS) inputs are used for addressing a memory device. You can obtain this information from a given data sheet. Figure 2.5 shows the local-to-memory address mapping of the Lattice DDR3 memory controller cores.

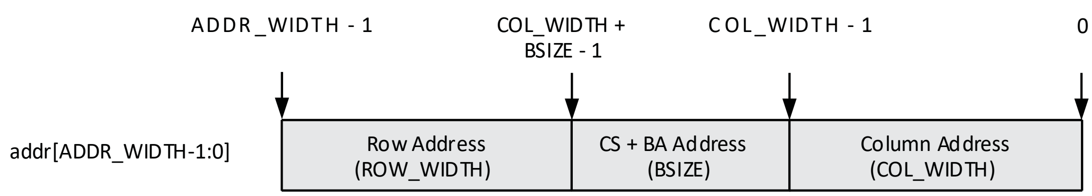

Figure 2.5. Local-to-Memory Address Mapping for Memory Access

ADDR_WIDTH is calculated by the sum of COL_WIDTH, ROW_WIDTH and BSIZE. BSIZE is determined by the sum of the BANK_WIDTH and CS_WIDTH. For DDR3 devices, the bank address size is always 3. When the number of chip select is 1, 2, or 4, the chip select address size becomes 0, 1, or 2, respectively. An example of a typical address mapping is shown in Table 2.6 and Figure 2.6.

Table 2.6. Address Mapping Example

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

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

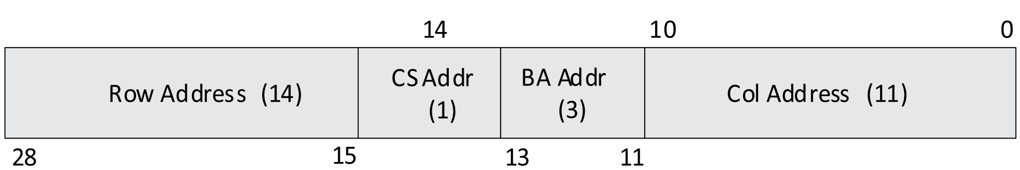

Figure 2.6. Mapped Address for the Example

2.7. Mode register Programming

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

The Lattice DDR3 SDRAM Controller IP Core uses the local address bus, addr_i, to program these registers. The core accepts a user command, LOAD_MR, to initiate the programming of mode registers. When LOAD_MR is applied on the cmd_i signal, the user logic must provide the information for the targeted mode register and the programming data on the addr_i signal. When the target mode register is programmed, the memory controller core is also configured to support the new memory setting. Figure 2.7 shows how the local address lines are allocated for the programming of memory registers.

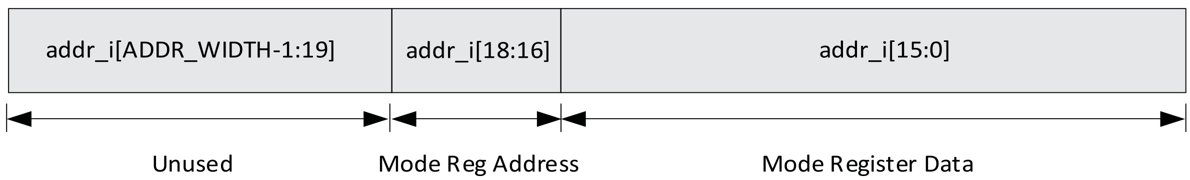

Figure 2.7. User-to-Memory Address Mapping for MR Programming

The register programming data is provided through the lower side of the addr_i starting from the bit 0 for LSB. The programming data requires 16 bits of the local address lines. Three more bits are needed to choose a target register as listed in Table 2.7. All other upper address lines are unused during LOAD_MR command.

Table 2.7. Transmit MAC Statistics Vector

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

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

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

Table 2.8. Initialization Default Values for Mode Register Setting

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

3. Core Generation, Simulation, and Validation

This section provides information on how to generate the DDR3 SDRAM Controller IP Core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide.

3.1. Licensing the IP

An IP core-specific license string is required enable full use of the DDR3 SDRAM Controller IP Core in a complete, top-level design. You can fully evaluate the IP core through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core, which operate in hardware for a limited time (approximately four hours) without requiring an IP license string. See Hardware Evaluation section for further details. However, a license string is required to enable timing simulation and to generate bitstream file that does not include the hardware evaluation timeout limitation.

3.2. Generation and Synthesis

The Lattice Radiant software allows you to customize and generate modules and IPs and integrate them into the device's architecture. The procedure for generating the DDR3 SDRAM Controller IP Core in Lattice Radiant software is described below.

To generate the DDR3 SDRAM Controller IP Core:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click on **DDR3_SDRAM_Controller** under **IP, Processors_Controllers_and_Peripherals** category. The **Module/IP Block Wizard** opens as shown in Figure 3.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.

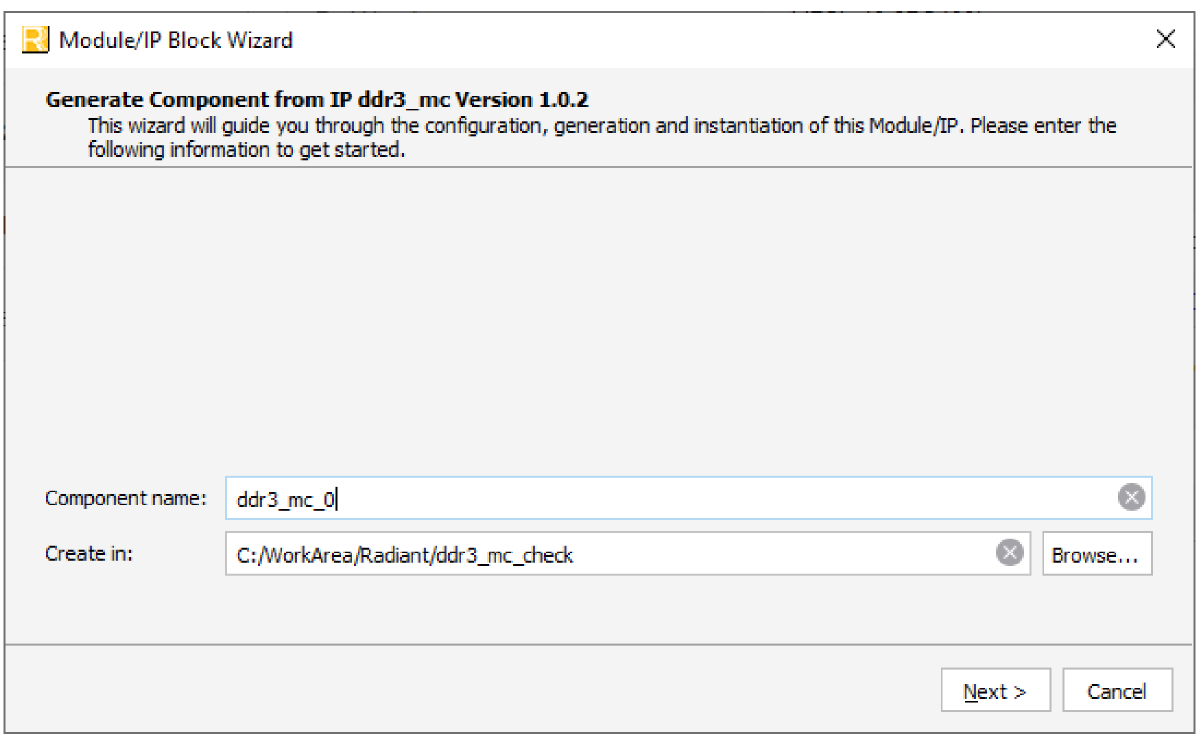

Figure 3.1. Module/IP Block Wizard

3. In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected DDR3 SDRAM Controller IP Core using drop-down menus and check boxes. As a sample configuration, see Figure 3.2. For configuration options, see the Attributes Summary section.

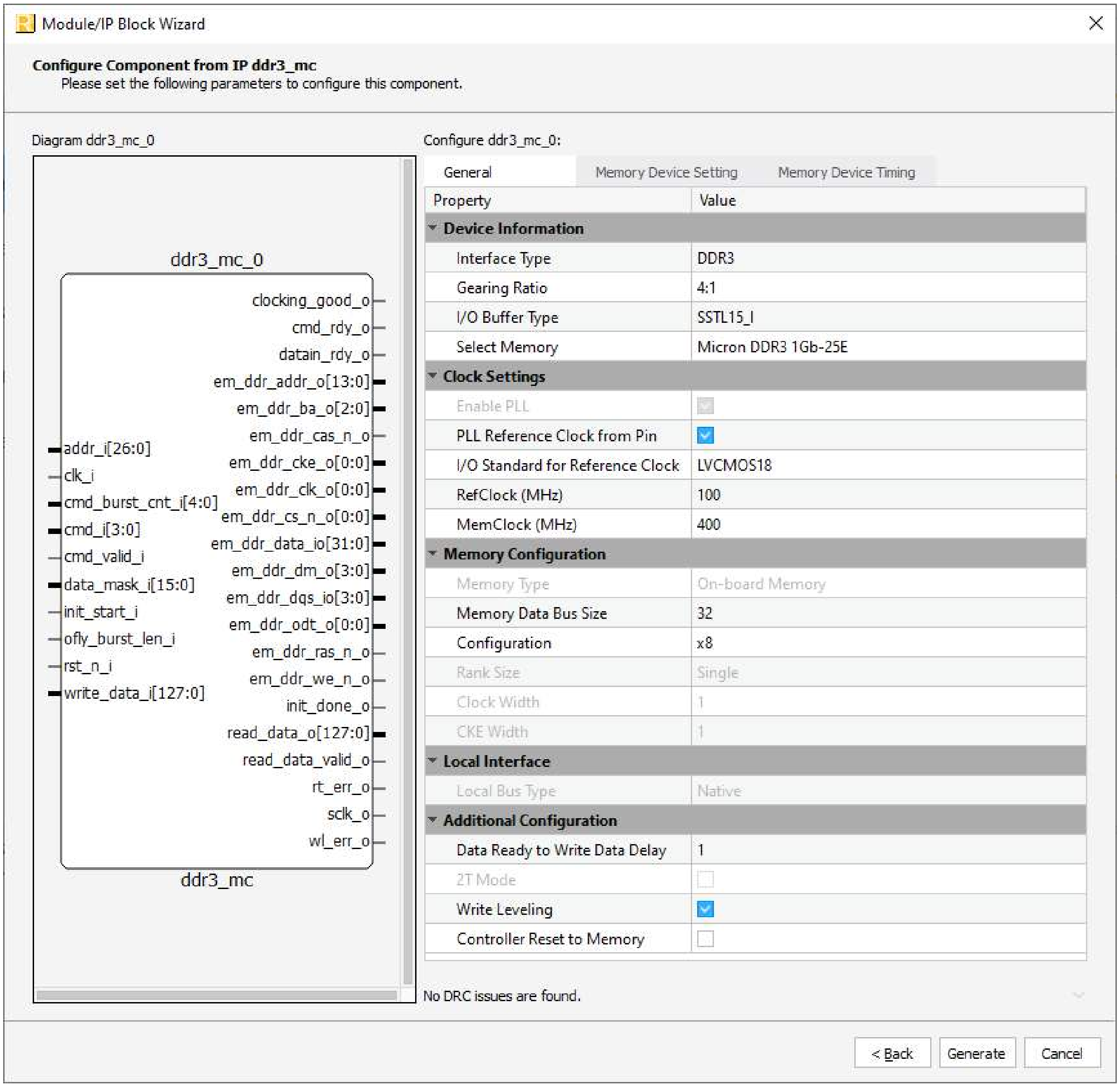

Figure 3.2. Module/IP Block Wizard of DDR3 SDRAM Controller IP Core

4. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in Figure 3.3.

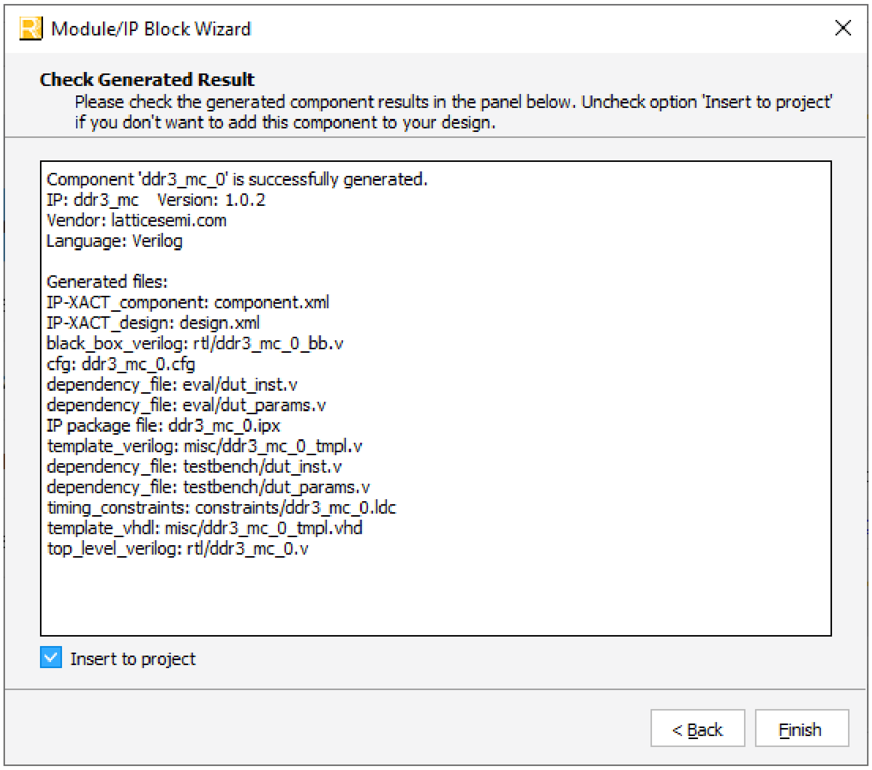

Figure 3.3. Check Generating Result

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

The generated DDR3 SDRAM Controller IP Core package includes the black box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the IP core is also provided. You may also use this top-level reference as the starting template for the top-level for their complete design. The generated files are listed in Table 3.1.

Table 3.1. Generated File List

Attribute	Description
<Component name>.ipx	This file contains the information on the files associated to the generated IP.
<Component name>.cfg	This file contains the parameter values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Component name>.v	This file provides an example RTL top file that instantiates the IP core.
rtl/<Component name>_bb.v	This file provides the synthesis black box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the IP core.
eval/top_constraint.pdc	Guide for setting clock constraint in the post-synthesis constraint Files
eval/create_top_constraint.py	The script that generates top_constraint.pdc based on user settings.
eval/eval_top.v	Top level RTL files that may be used for running Lattice Radiant software flow check (synthesis to export) on the generated IP. Without this, the Radiant software map process fails due to not enough I/O. This is mainly used for checking resource utilization and fmax for the selected IP configuration, this is not for implementation.
eval/lscc_simple_lfsr.v	A simple linear-feedback shift register.
eval/dut_inst.v	A sample instantiation of the generated IP. This is included by the eval_top.v.
eval/dut_params.v	Lists the equivalent localparams of the user settings. This is included by the eval_top.v.

3.2.1. Required Post-Synthesis Constraints

The DDR3 SDRAM Controller IP Core has one PLL reference clock input and three internally-generated clocks. You need to constrain these clocks in the post-synthesis constraint file of your Lattice Radiant project.

The eval/top_constraint.pdc file described in Table 3.1 is a constraint file that is generated based on the user settings. You need to copy the contents of this file to your post synthesis constraint file for your Lattice Radiant project.

3.3. Running Functional Simulation

The reset and initialization of DDR3 memories take around 200 µs (when *Controller Reset to Memory* attribute is checked) and 500 µs respectively. This takes a long time to simulate. After checking that reset and initialization works, it is usually desired to bypass (reduce to few clock cycles) these process in the succeeding simulation to save time. Bypassing these requires manual update on the generated IP RTL file and testbench file as follows:

- Modify the <generated_ip_path>/rtl/<generated_ip_name>.v and set SIM parameter to 1. For example:

```
    ddr3_mc_0_ipgen_lscc_ddr3_mc #(.INTERFACE_TYPE("DDR3"),
            .SIM(1),  // Set SIM=1 for simulation only
            .GEAR(4),
```

- The file <generated_ip_path>/testbench/dut_params.v contains the user-defined parameters. This is used for configuring the test. You should set the localparam SIM to 1 so that the test knows that you bypass reset and initialization.

```
localparam SIM = 1;
```

You should ensure that the generated IP that you used for bitstream generation has SIM parameter set to 0 so that the IP Core can geneate correct reset and initialization period which is necessary for proper DDR3 SDRAM operation.

Below are the steps for running simulation.

1. Add the top level testbench file, tb_top.v in the project as a simulation file, click **File** Tab, then select **Add** in the drop down menu, then click **Existing Simulation File.** Then select *<Component name>/testbench/tb_top.v*.
2. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 3.4.

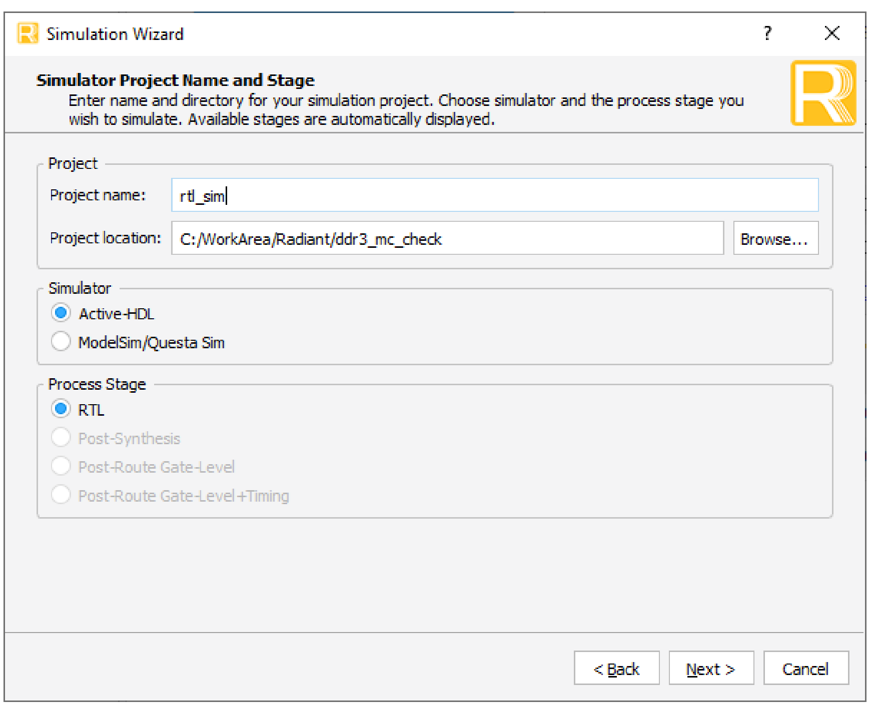

Figure 3.4. Simulation Wizard

3. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.5. Notice that the **Source Files** area only contain the generated IP (<*Component name*>.*v*) and the tb_top.v, which is added in Step 1. The tb_top.v includes all the necessary test files for simulation.

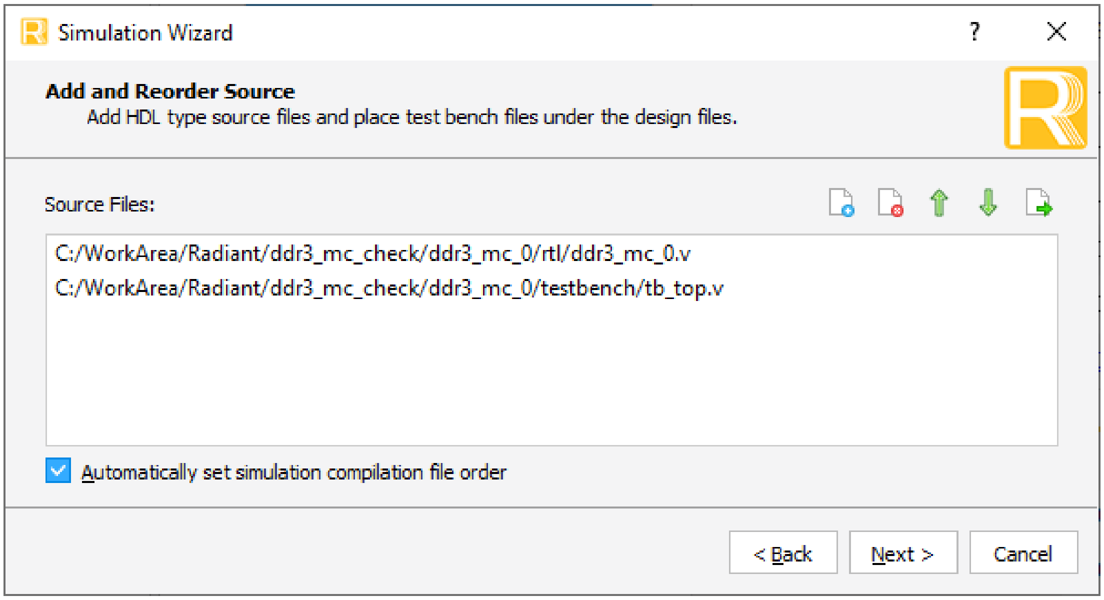

Figure 3.5. Adding and Reordering Source

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

   The results of the simulation in our example are provided in Figure 3.6.

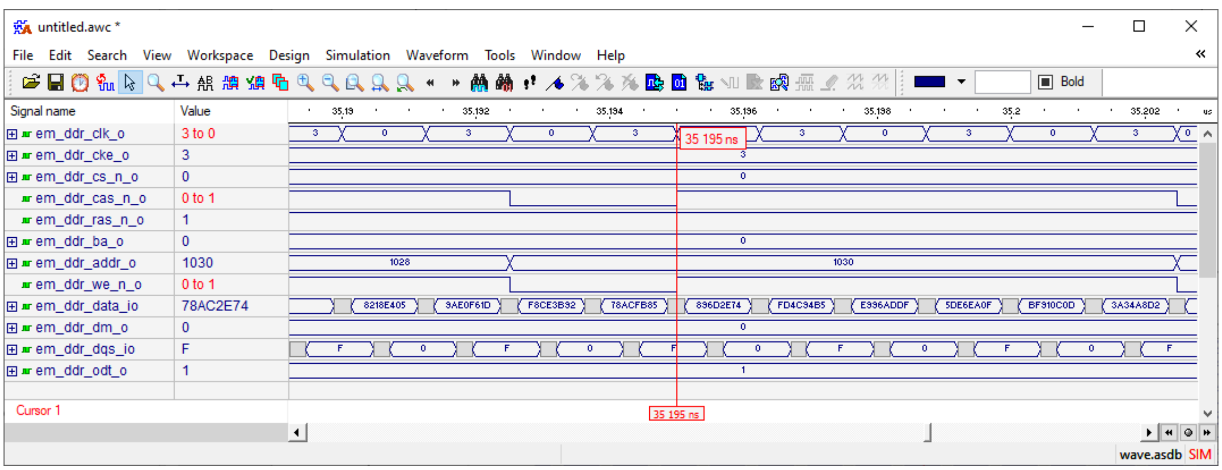

Figure 3.6. Simulation Waveform

Notes:

- It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant software suite.
- The following warning message is allowed during write leveling procedure, please ignore them.

KERNEL: tb_top.U0_ddr3_dimm.U3.main: at time 30171942.0 ps WARNING: tWLH violation on DQS bit0 positive edge. Indeterminate CK capture is possible.

...

KERNEL: tb_top.U0_ddr3_dimm.U3.dqs_pos_timing_check: at time 30366950.0 ps WARNING: tWLS violation on DQS bit 0 positive edge. Indeterminate CK capture is possible

KERNEL: [0042162188] MON INFO: Mode Register Set command issued -> MR1 = 0x0044

KERNEL: [0042162188] MON INFO: Write Leveling Ended.

The end of write leveling procedure is marked by the last message above. After this message, no more warning message is expected.

3.4. Hardware Evaluation

The DDR3 SDRAM Controller IP Core supports Lattice's IP hardware evaluation capability when used with CrossLink-NX devices. This makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.

4. Ordering Part Number

The Ordering Part Number (OPN) for this IP Core are the following:

- DDR3-P-CNX-U – DDR3 SDRAM Controller for CrossLink-NX - Single Design License
- DDR3-P-CNX-UT – DDR3 SDRAM Controller for CrossLink-NX - Site License
- DDR3-P-CTNX-U – DDR3 SDRAM Controller for Certus-NX - Single Design License
- DDR3-P-CTNX-UT – DDR3 SDRAM Controller for Certus-NX - Site License

Appendix A. Resource Utilization

Table A.1 show configuration and resource utilization for LIFCL-40-9BG400I using Synplify Pro of Lattice Radiant software 2.1.

Table A.1. Resource Utilization

Configuration	sclk_o Fmax (MHz)	Registers	LUTs	EBR	IDDR/ODDR/TDDR
Default	171.204	2843	2921	0	132
Memory Data Bus Size = 24, Others = Default	169.635	2469	2724	0	105
Memory Data Bus Size = 16, Others = Default	163.159	2102	2586	0	78
Memory Data Bus Size = 8, Others = Default	176.274	1713	2390	0	51
MemClock = 533, Others = Default	170.474	2861	2873	0	153

Notes:

1. The sclk_o Fmax is generated using the eval_top.v that is described in Table 3.1. This design only contains the DDR3 SDRAM Controller IP Core and a few linear-feedback shift registers. These values may be reduced when the IP Core is used with the user logic.
2. The *distributed RAM* utilization is accounted for in the total LUT4 utilization. The actual LUT4 utilization is distribution among *logic*, *distributed RAM*, and *ripple logic*.

Appendix B. Limitation

The read training logic does not yet support Gearing Ratio 8:1. This is being tracked internally as DNG-9914. Depending on the board-level routing delay, the read training logic may not compensate for the total read path delay. It is recommended to use Gearing Ratio 4:1 until this limitation is fixed.

References

- CrossLink-NX web page at www.latticesemi.com
- Certus-NX web page at www.latticesemi.com
- http://www.jedec.org

Technical Support Assistance

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

Revision History

Revision 1.4, October 2020

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

Revision 1.3, August 2020

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

Revision 1.2, June 2020

Section	Change Summary
Introduction	• Updated Table 1.1 to add Certus-NX as supported FPGA family, LFD2NX-40 as targeted device. Added new entry for Lattice Implementation and changed Synplify version to Pro for Lattice. • Updated Features section to include selectable gearing ratio: 4:1, 8:1.
Functional Description	• Updated note 3 in Table 2.1 for signal bit width change due to gearing ratio feature and added note 4 to indicate the change in signal names. • Updated Table 2.2 for gearing ratio feature and improved attribute dependency checks.
Core Generation, Simulation, and Validation	• Updated Figure 3.1, Figure 3.2 and Figure 3.3 for IP Core v1.0.2. • Updated Table 3.1 for the added eval folder. • Added Required Post-Synthesis Constraints section.
Ordering Part Number	Updated this section.
Appendix A. Resource Utilization	Updated this section.
References	Updated this section.

Revision 1.1, February 2020

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

Revision 1.0, December 2019

Section	Change Summary
All	Initial release