LPDDR2 SDRAM Controller Lite IP Core

User Guide

FPGA-IPUG-02046 Version 1.0

July 2018

Contents

Figures

Tables

1. Introduction

The Lattice Low Power Double Data Rate (LPDDR2) Synchronous Dynamic Random Access Memory (SDRAM) Controller Lite is a general-purpose memory controller that interfaces with industry standard LPDDR2 memory devices compliant with JESD209-2B and LPDDR2 SDRAM Standard, and provides a generic command interface to user applications. LPDDR2 SDRAM is the next-generation Low Power SDRAM memory technology which offers a higher data rate, higher density, greater bandwidth, and power efficiency. This core reduces the effort required to integrate the LPDDR2 memory controller with the user application design and minimizes the need to directly deal with the LPDDR2 memory interface.

1.1. Quick Facts

Table 1.1 provides quick facts about the LPDDR2 SDRAM Controller Lite IP core.

Table 1.1. LPDDR2 SDRAM Controller Lite IP Core Quick Facts*

		LPDDR2 SDRAM Controller Lite IP Configuration	
		x16	**X32**
Core Requirements	FPGA Families Supported	ECP5™	
	Minimal Device Needed	LFE5U-25F-6MG285C/ LAE5UM-25F-6MG285E	LFE5U-25F-6MG285C/ LAE5UM-25F-6MG285E1
Resource Utilization	Targeted Device	LFE5UM-85F-8BG756CES	
	Data Path Width	16	32
	LUTs	2241	2462
	sysMEM EBRs	114322	
	Registers	1599	1937
Design Tool Support	Lattice Implementation	Lattice Diamond® 3.10	
	Synthesis	Synopsys® Synplify Pro® for Lattice I-2014.03L-SP1	
		Lattice Synthesis Engine (LSE)	
	Simulation	Aldec® Active-HDL® 9.3 Lattice Edition II Mixed Language	
		Mentor Graphics® ModelSim® SE PLUS 6.5 or later	

***Note**: LFE5U-25F-6MG381C/LAE5UM-25F-6MG381E for the core evaluation project inside a generated core

1.2. Features

The LPDDR2 SDRAM Controller Lite IP core supports:

- All ECP5 devices
- Interface with industry standard LPDDR2 SDRAM components and modules compliant with JESD209-2B, LPDDR2 SDRAM standard
- Interface with LPDDR2 SDRAM at speeds up to 400 MHz / 800 Mbps in -8 speed grade devices
- Memory data path widths of 16 and 32 bits
- x16 and x32 device configurations
- Single rank of an LPDDR2 device (one chip select)
- Burst lengths of eight (fixed)
- Programmable read and write latency set
- Automatic LPDDR2 SDRAM initialization and refresh
- Automatic read training for each DQS
- Deep Power Down mode and Power Down mode
- Partial Array Self Refresh (PASR) operations on Bank and Segment
- Automatic programmable interval refresh or user-initiated refresh
- Burst or distributed auto refresh

Note: This *Lite* version IP only supports features and commands that LPDDR2 have in common with LPDDR3. This means burst lengths of 8 is supported, but burst lengths of 4 and 16 are not supported. Also, flash memory commands (LPDDR2-N) are not supported.

2. Functional Description

This chapter provides a functional description of the LPDDR2 SDRAM Controller Lite IP core.

2.1. Overview

This section briefly describes the operation of LPDDR2 modules. The LPDDR2 memory controller consists of two major parts: the Memory Controller (MC) module and the I/O (PHY) Module. In addition to these, a separate Clock Synchronization Module (CSM) is also provided.

Figure 2.1 provides a high-level block diagram illustrating the main functional blocks and the technology used to implement the LPDDR2 SDRAM Controller Lite IP core functions.

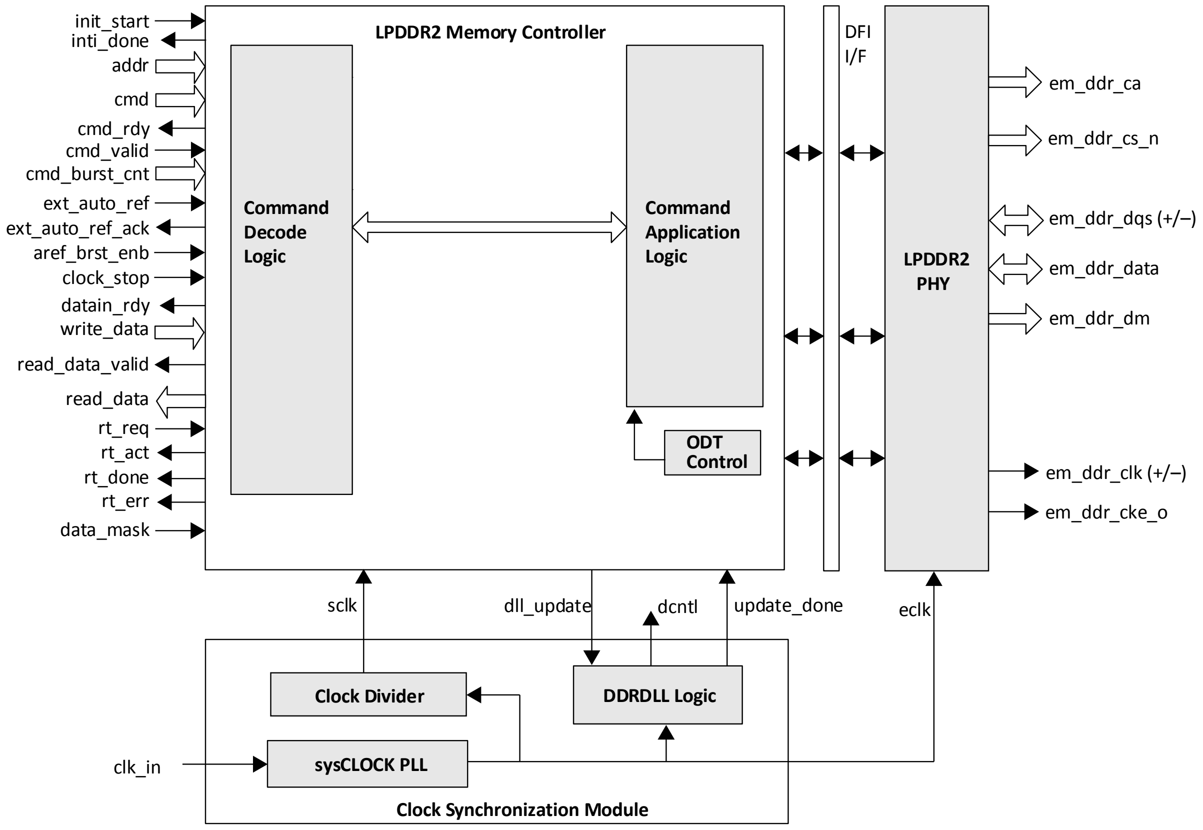


Figure 2.1. LPDDR2 SDRAM Controller Lite Block Diagram

2.2. LPDDR2 MC Module

The LPDDR2 MC module has the following functional sub-modules:

- Command Decode Logic (CDL) block
- Command Application Logic (CAL) block

2.2.1. Command Decode Logic

The Command Decode Logic (CDL) block accepts user commands from the local interface and decodes them to generate a sequence of internal memory commands 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 of depth 2 to improve throughput. With this capability, the next command in the queue is decoded while the current command is presented at the memory interface.

2.2.2. Command Application Logic

The Command Application Logic (CAL) block accepts the decoded internal command sequence from the Command Decode Logic, and translates each sequence into appropriate 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.

2.3. LPDDR2 PHY Module

The LPDDR2 PHY modules provide the PHY interface to the memory device. This block mostly consists of ECP5 device DDR I/O primitives supporting compliance with LPDDR2 electrical and timing requirements. In addition, this module consists of the logic for memory initialization, read training, write/read data path control, and address/command timing control.

The LPDDR2 PHY module implements a set of soft logic in the FPGA fabric for initialization, read training, and read/write paths. This module also implements a set of hard logic, called DDR2 I/O modules, for 1:2 clock gearing and LPDDR2 memory interface.

The LPDDR2 I/O modules are ECP5 device primitives that directly connect to the LPDDR2 memory. These primitives implement all of the interface signals required for memory access. They convert the single data rate (SDR) data from the user to double rate LPDDR2 data for the write operation and perform the DDR to SDR conversion in the read mode.

2.3.1. Initialization

The Initialization block performs the LPDDR2 memory initialization sequence as defined by the LPDDR2 JEDEC protocol. After power on or a normal reset of the LPDDR2 controller, the memory must be initialized before sending any command to the controller. It is the user's responsibility to assert the init_start input to the LPDDR2 controller to initiate the memory initialization sequence. The completion of initialization is indicated by the init_done output provided by this block.

2.3.2. Read Training

For every read operation, the LPDDR2 I/O primitives of the ECP5 device must be initialized at the appropriate time to identify the incoming DQS preamble. Upon a proper detection of the preamble, the primitive DQSBUFM extracts a clean signal out of the incoming DQS signal from the memory and generates the DATAVALID output signal that indicates the correct timing window of the valid read data.

The memory controller generates a positioning signal, READ[1:0], to the primitive DQSBUFM that is used for the above-mentioned operation. In addition to the READ[1:0] input, another fine control input signal READCLKSEL[2:0] and an output signal, BURSTDET, of the DQSBUFM block are provided to the controller to accomplish the READ signal positioning.

Due to the DQS round trip delay that includes PCB routing and I/O pad delays, proper internal positioning of the READ signal with respect to the incoming preamble is crucial for successful read operations. The ECP5 DQSBUFM block supports a dynamic READ signal positioning function called read training that enables the memory controller to position the internal READ signal within an appropriate timing window by progressively shifting the READ signal and monitoring the positioning result. During the read training, the memory controller generates the READ[1:0] pulse as a coarse positioning control.

READCLKSEL[2:0] is also generated to provide a finer position control (1/4 T per step). The BURSTDET output of DQSBUFM is used to monitor the result of the current position. The READ[1:0] signal is needed to be set high at least 5.5T before the read preamble starts. The internal READ is progressively shifted by READ[1:0] and READCLKSEL[2:0] once the read training routine starts. When the internal READ signal is properly positioned, the BURSTDET signal will be asserted high to indicate that the preamble has been detected correctly. This will guarantee that the generated DATAVALID signal is indicating the correct read valid time window. The READ[1:0] and READSELCLK[2:0] signals are generated in the system clock (SCLK) domain and required to stay asserted for the total burst length of the read operation.

The memory controller determines the proper position alignment when there is not a single failure on BURSTDET assertions during the multiple trials. If there is any failure, the memory controller shifts the READCLKSEL[2:0] and READ[1:0] signals accordingly to position the internal READ to a safer position and tries again until it detects no BURSTDET failure during the entire read training process. The memory controller stores the delay value of the successful position of the READCLKSEL[2:0] and READ[1:0] 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.

2.3.3. PHY Control Block

The PHY Control Block includes the data path control and PHY timing control functions. The data path control function interfaces with the LPDDR2 I/O modules and is responsible for generating the read data and read data valid signals during the read operations. This block also implements all the logic that are needed to ensure that the LPDDR2 command/controls and data write/read to and from the memory are properly transferred to the local user interface in a deterministic and coherent manner.

2.3.4. Data Path Logic

The Data Path Logic (DPL) block interfaces with the LPDDR2 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.

2.4. Clock Synchronization Module (CSM)

Along with the LPDDR2 SDRAM Controller Lite IP core, a separate module, called the Clock Synchronization Module (CSM), is also provided. The CSM generates all the clock signals, such as system clock (SCLK) and edge clock (ECLK) for the IP core.

The CSM logic ensures that all DDR components in the ECP5 device are synchronized after a system reset. CSM also provides the logic for the DLL update operation. 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 LPDDR2 memory support, the MC module operates with a 200 MHz system clock (SCLK), the I/O logic 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.

2.5. Signal Descriptions

Table 2.1 describes the user interface and memory interface signals at the top level.

Table 2.1. LPDDR2 SDRAM Memory Controller Top-Level I/O List

Port	Active State	I/O	Description
General I/O			
clk_in	N/A	Input	Reference clock to the PLL of the CSM block.
Clock Synchronization Logic (CSM) Interface			
sclk	N/A	Input	System clock used by controller's core module. User may use this clock for LPDDR2 controller interface logic.
eclk	N/A	Input	Edge clock used by controller's PHY module. Usually twice the Frequency of sclk.
clocking_good	High	Input	Signal from CSM module indicating stable clock condition.
update_done	High	Input	DLL update done input from CSM to the core. Upon receiving an update_done assertion, the core de-asserts dll_update in the next clock cycle.
dll_update	High	Output	DLL update request output from the core to the CSM block. Once asserted, dll_update needs to stay asserted until an update_done assertion is sampled. The CA bus stays in NOP during the DLL update.
Local User Interface			
rst_n	Low	Input	Asynchronous reset to the entire IP core.
init_start	High	Input	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. Refer to the Initialization Control section for more details.
cmd[3:0]	N/A	Input	User command input to the memory controller. Refer to the User Commands section for available commands.
cmd_valid	High	Input	Command and address valid input. When asserted, the addr, cmd and cmd_burst_cnt inputs are considered valid. Refer to the Command and Address section for more details.
addr[ADDR_WIDTH-1:0]	N/A	Input	User read or write address input to the memory controller. Refer the Local-to-Memory Address Mapping section for more details.
cmd_burst_cnt[4:0]	N/A	Input	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.
write_data[DSIZE-1:0]	N/A	Input	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[(DSIZE/8)[1:0]	High	Input	Data mask input for write data. Each bit masks a corresponding byte of local write data.
ext_auto_ref	High	Input	Refresh request from user – This signal is available only when the External Auto Refresh Port is selected in the interface.
aref_brst_enb	High	Input	Enable or disable Auto Refresh Burst mode. 1 = Burst mode 0 = Distributed mode

Port	Active State	I/O	Description
init_done	High	Output	Initialization done output – Asserted for one clock period after the core completes memory initialization. When sampled high, the input signal init_start must be immediately de-asserted at the same edge of the sampling clock. Refer to the Initialization Control section for more details.
cmd_rdy	High	Output	Command ready output – When asserted, indicates that the core is ready to accept the next command and the corresponding address. This cmd_rdy signal is active for one clock period.
datain_rdy	High	Output	Data ready output – When asserted, indicates the core is ready to receive the write data.
read_data[DSIZE-1:0]	N/A	Output	Read data output from memory controller to the user logic.
read_data_valid	High	Output	Read data valid output – When asserted, indicates the data on the read_data bus is valid.
ext_auto_ref_ack	High	Output	Completion of memory refresh in response to ext_auto_ref signal assertion. This pin is available only when the External Auto Refresh Port is selected in the interface.
clock_stop	High	Input	Signal from the user to stop the LPDDR2 memory clock. The memory clock stays Low while this signal is asserted High. Only NOP is allowed on the LPDDR2 bus during clock stop.
rt_req	High	Input	Read pulse training request from the user.
rt_act	High	Output	Signal to indicate that the read pulse training is active.
rt_done	High	Output	Signal to indicate that the read pulse training has been completed.
rt_err	High	Output	Signal to indicate a failure during the read pulse training.
LPDDR2 SDRAM Memory Interface			
em_ddr_clk[CLKO_WIDTH-1:0]	N/A	Output	Up to 400 MHz differential pair memory clock generated by the controller.
em_ ddr_cke[CKE_WIDTH-1:0]	High	Output	Memory clock enable generated by the controller.
em_ddr_ca[9:0]	N/A	Output	Memory Command and Address (CA) bus - multiplexed row address, column address and command to the memory.
em_ ddr_data[DATA_WIDTH-1:0]	N/A	In/Out	Memory bi-directional data bus.
em_ ddr_dm[(DATA_WIDTH/8)-1:0]	High	Output	LPDDR2 memory write data mask – to mask the byte lanes for byte-level write.
em_ ddr_dqs[DQS_WIDTH-1:0]	N/A	In/Out	Memory bi-directional, differential pair data strobe.
em_ ddr_cs_n[CS_WIDTH-1:0]	Low	Output	Memory chip select.
Optional I/Os			
ce	1	I	Clock Enable. While this signal is de-asserted, the core will ignore all other synchronous inputs and maintain its current state
sr	1	I	Synchronous Reset. When asserted for at least one clock cycle, all the registers in the IP core are initialized to reset state.

***Notes**:

- Width for signed type and symmetric interpolation is Coefficients width +1.
- Width for unsigned and symmetric interpolation is Coefficients width +2.
- Width for all other cases is Coefficients width.

2.6. Using the Local User Interface

The local user interface of the LPDDR2 SDRAM Controller Lite IP core consists of five independent functional groups:

- Initialization Control
- Command and Address
- Data Write
- Data Read

Each functional group and its associated local interface signals are listed in Table 2.2.

Table 2.2. Local User Interface Functional Groups

Functional Group	Signals
Initialization Control	init_start, init_done, rt_req, rt_act, rt_done, rt_err,
Command and Address	addr, cmd, cmd_rdy, cmd_valid, cmd_burst_cnt
Data Write	datain_rdy, write_data, data_mask
Data Read	read_data, read_data_valid
Auto Refresh	aref_brst_enb, ext_auto_ref, ext_auto_ref_ack

2.6.1. Initialization Control

LPDDR2 memory devices must be initialized before they can be accessed by the memory controller. The memory controller starts the memory initialization sequence when the init_start signal is asserted by the user interface. Once asserted, the init_start signal needs to be held high until the initialization process is completed. The output signal init_done 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 signal must be de-asserted as soon as init_done is sampled high at the rising edge of sclk. If the init_start is left high at the next rising edge of sclk 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 memory controller ensures a minimum gap of 200 µs between em_ddr_cke assertion and reset command to the memory. Figure 2.2 shows the timing diagram of the initialization control signals.

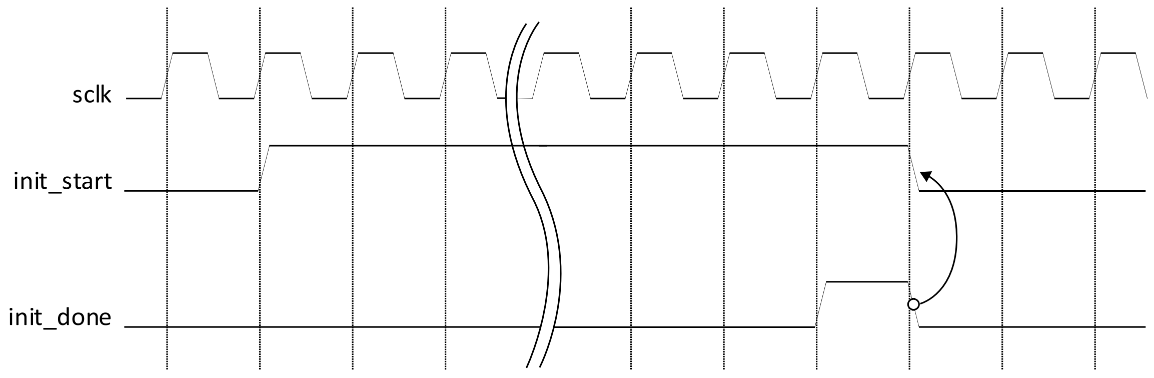


Figure 2.2. Timing of Memory Initialization Control

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 LPDDR2 memory does not use a DLL function, the clock to DQS driving time can vary significantly with the process, voltage, and temperature (PVT) variations. Due to this reason, periodic retraining of the read pulse position may be necessary for guaranteeing stable read transactions over the PVT variations during the normal operation. The LPDDR2 IP core provides you with a function that can perform read retraining for PVT calibration during the normal operation using the following signals:

- rt_req: read retraining request
- rt_act: read training active
- rt_done: read retraining done

You need to assert the rt_req signal during the normal operation but only when the LPDDR2 bus is in an idle state. No activity other than NOP command is allowed when the core enters to the read retraining mode. Once asserted, rt_req needs to keep holding its assertion state until the rt_done signal is sampled High. The rt_act signal indicates that the IP core is in the read training mode, and the LPDDR2 bus should remain idle with only NOP commands until the retraining process is completed. The rt_done signal is asserted only for one clock cycle, and the rt_req signal must be deasserted immediately after the sampling of the rt_done assertion to avoid unnecessary reentering the training. The rt_err signal is asserted if the read training is not successful. Since a failure during the read training usually causes the IP core unable to transfer the read data from the LPDDR2 SDRAM device to the FPGA fabric, rt_err should be the first signal that you need to check if the LPDDR2 interface does not work properly.

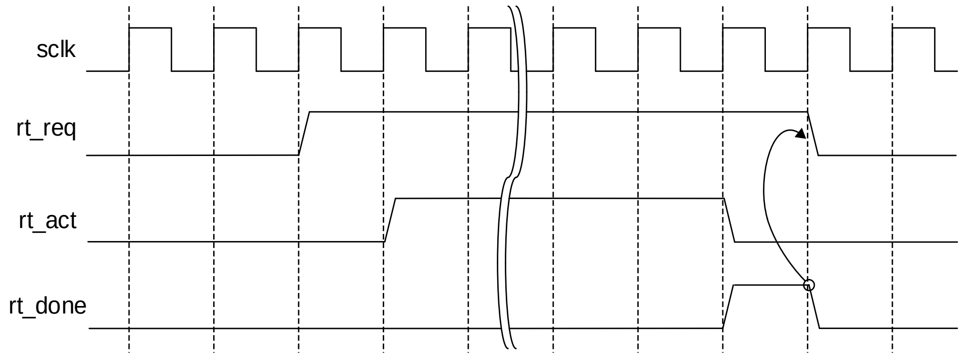


Figure 2.3. Timing Diagram for Read Retraining

2.6.2. Command and Address

After memory initialization is completed, the core waits for user commands in order to set up and/or access the memory. The user logic needs to provide the command and address to the core along with the control signals. The commands and addresses are delivered to the core using the following procedure.

The memory controller core informs the user logic that it is ready to receive a command by asserting the cmd_rdy signal for one cycle. If the core finds the cmd_valid signal asserted by the user logic while its cmd_rdy is asserted, it takes the cmd input as a valid user command. Usually, cmd_valid is de-asserted at the rising edge of the clock that samples cmd_rdy high. The core also accepts the addr input as a valid start address or mode register programming data depending on the command type. Along with addr input the core also accepts the signal cmd_burst_cnt. If cmd_valid is not asserted, the cmd and addr inputs become invalid and the core ignores them. The cmd, addr, cmd_burst_cnt and cmd_valid inputs become "don't care" while cmd_rdy is de-asserted. The cmd_rdy 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 once every two-clock cycles 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 core prevents cmd_rdy from being asserted until the command burst is completed. While the core is operating in the command burst mode, it can keep maximum throughput by internally replicating the command. The memory controller repeats the given READ or WRITE command up to 32 times. The cmd_burst_cnt[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 that the core accesses becomes the beginning of the same page. This results in overwriting the contents of the location or reading unexpected data. Therefore, you must 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 is 0, the controller will perform 32 commands (reads or writes). The cmd_burst_cnt input is sampled the same way as cmd signal. The timing of the Command and Address group is shown in Figure 2.4. The timing for burst count in Figure 2.4 shows only the sampling time of the bus. When cmd_burst_cnt is sampled with a value greater than 00001 and the command queue becomes full, the cmd_rdy signal will not be asserted and the memory address is automatically increased by the core until the current command burst cycle is completed.

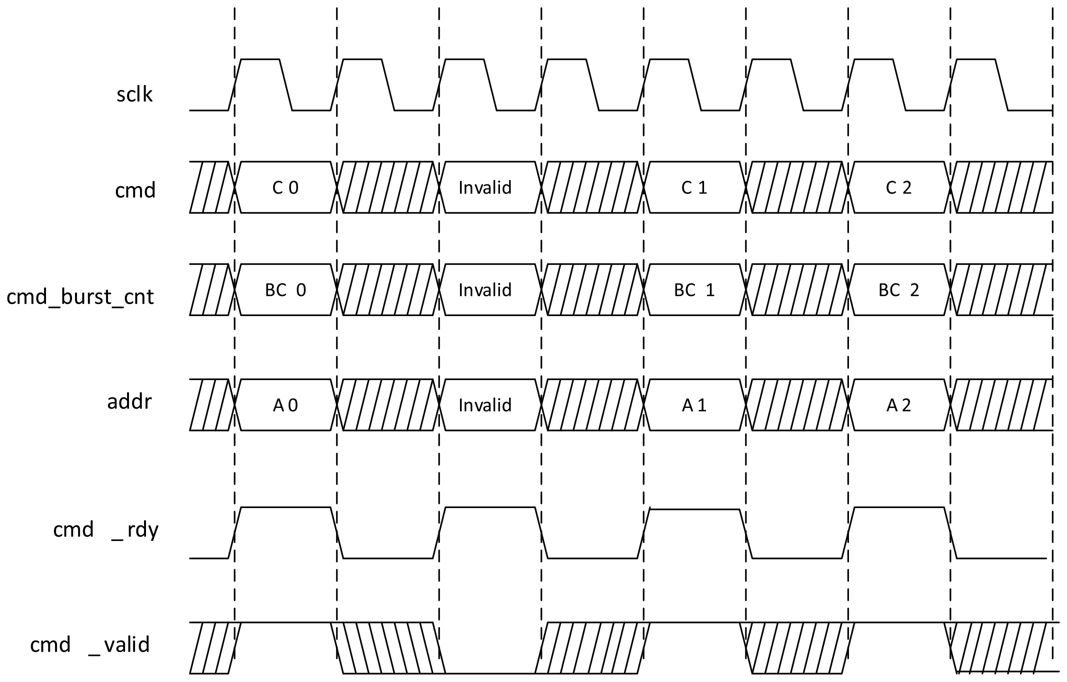


Figure 2.4. Timing of Command and Address

2.6.3. User Commands

The user initiates a request to the memory controller by loading a specific command code in cmd input along with other information like 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 2.3. All other values are reserved and considered invalid.

Table 2.3. Local User Interface Functional Groups*

Command	Mnemonic	cmd[3:0]
Read	READ	4'h1
Write	WRITE	4'h2
Read with Auto Precharge	READA	4'h3
Write with Auto Precharge	WRITEA	4'h4
Power down Entry	PD_ENT	4'h5
Mode Register Write	MRW	4'h6
Mode Register Read	MRR	4'h7
Self Refresh Entry	SEL_REF_ENT	4'h8
Self Refresh Exit	SEL_REF_EXIT	4'h9
Deep Powerdown Entry	DPD_ENT	4'ha
Powerdown/Deep Powerdown Exit	DPD_EXIT	4'hb

***Notes**:

- The controller accepts only the cmd codes listed above as legal commands. Any other cmd code is discarded as invalid command.
- The controller discards Self Refresh Entry or Deep 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 Deep Power Down Exit command if the memory is already not in Self Refresh mode or Power Down mode respectively.
- The controller needs at least 2 sclk gap between DPD_ENT command and DPD_EXIT command.

2.6.4. WRITE

The user initiates a memory write operation by asserting cmd_valid along with the WRITE or WRITEA command and the address. After the WRITE command is accepted, the memory controller core asserts the datain_rdy 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 signal is asserted is not fixed, the user logic needs to monitor the datain_rdy signal. Once datain_rdy is asserted, the core expects valid data on the write_data bus one or two clock cycles after the datain_rdy signal is asserted. The write data delay is programmable by the user, by setting desired value for Data_rdy to Write data delay in the interface, 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 2.5 shows two examples of the local user interface data write timing. Both cases are in BL8 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. The memory controller then 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. The memory controller then 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.

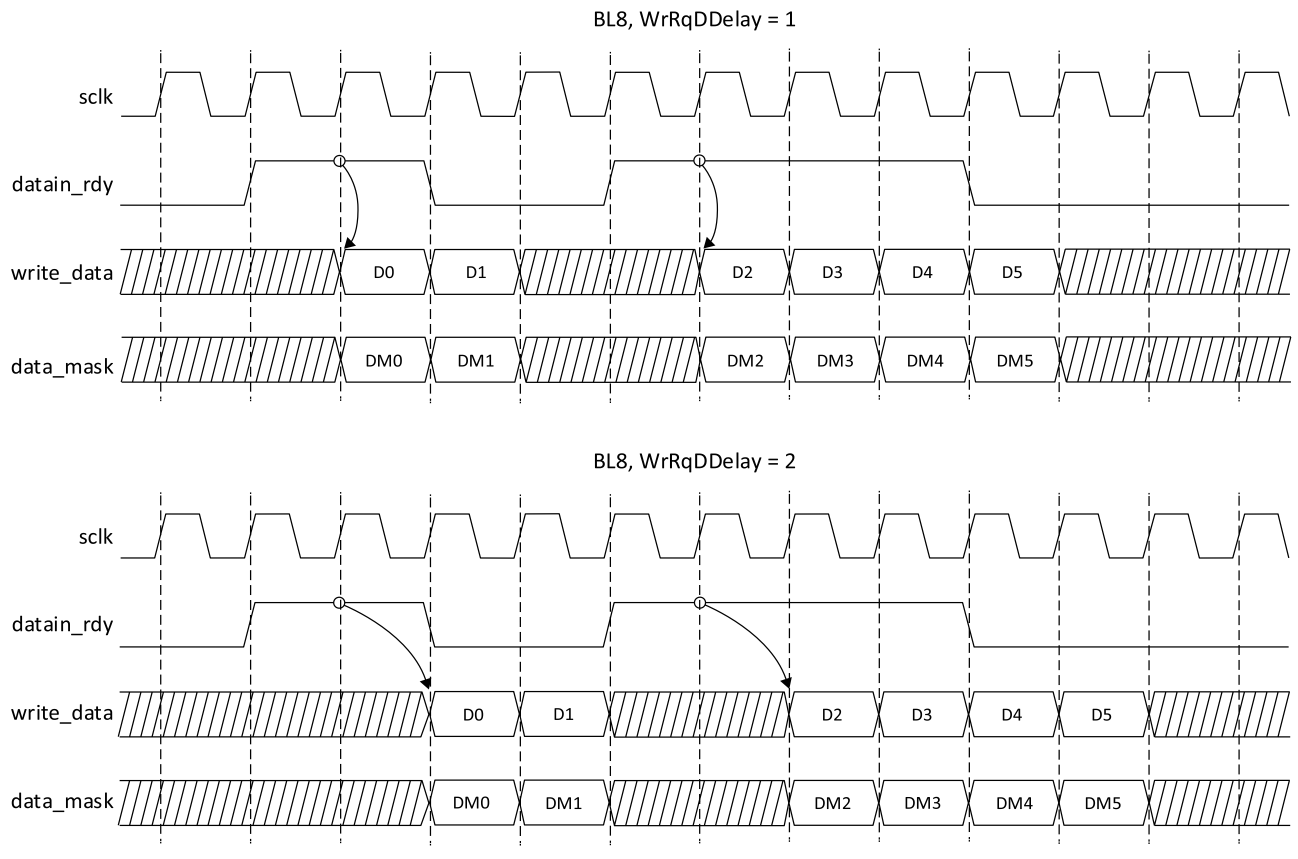


Figure 2.5. Timing One-Clock vs. Two-Clock Write Data Delay

Note: WrRqDDelay is Data_rdy to Write data delay.

2.6.5. WRITEA

WRITEA is treated in the same way as WRITE command except for the difference that the core issues a Write with Auto Pre-charge command to the memory instead of just a Write command. This causes the memory to automatically close the current row after completing the write operation.

2.6.6. 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 signal to tell the user logic that the valid read data is on the read_data bus. The read data timing on the local user interface is shown in Figure 2.6.

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. Refer to the description mentioned in Write operation for more details.

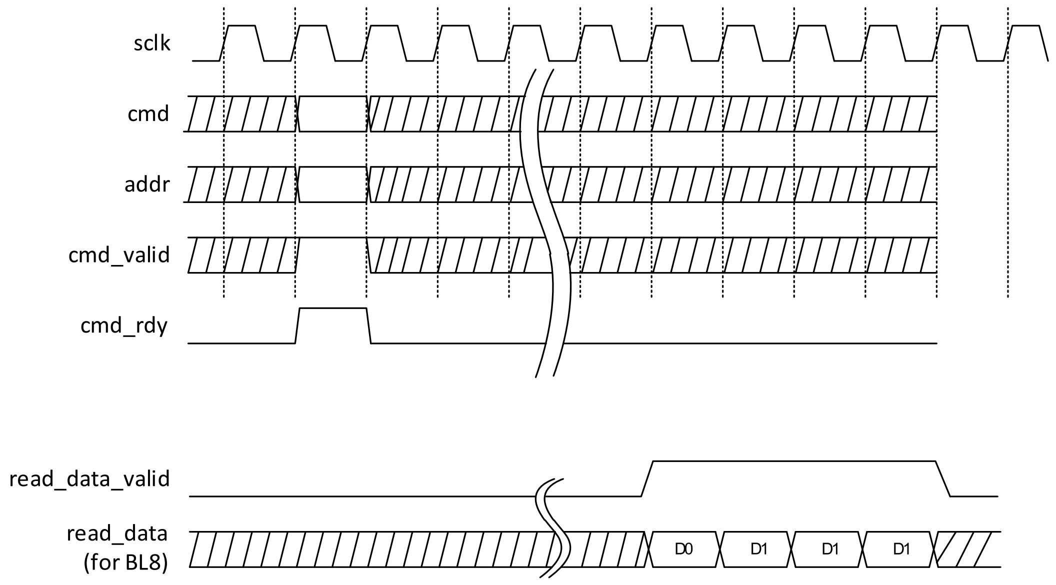


Figure 2.6. User-Side Read Operation

2.6.7. READA

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

2.6.8. REFRESH Support

Since LPDDR2 memories have at least an 8192-deep Auto Refresh command queue as per JEDEC specification, Lattice's LPDDR2 memory controller core can support up to 8192 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) times the Auto refresh burst count selected in the interface.

It is recommended that the optimum number of burst Refresh commands be used if the LPDDR2 interface throughput is a major concern of the system. If the refresh burst count is set to n, for example, the core will send a set of n consecutive Auto Refresh commands to the memory at once when it reaches the time period of the n refresh intervals (tREFI x n). Sending out n number of refresh commands in a burst takes about n x 4 x tRFCab ns in All bank refresh mode. During this time, no other command is sent to the memory. When a refresh burst is used, the controller issues a Pre-charge command only for the first Refresh command and the subsequent Refresh commands of the burst are issued without the associated Pre-charge commands. This is to improve the LPDDR2 throughput.

Alternatively, you can enable the External Auto Refresh Port which will add 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.

2.7. 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. Users 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. Users can obtain this information from a given datasheet. Figure 2.7 shows the local-to-memory address mapping of the Lattice LPDDR2 memory controller cores.

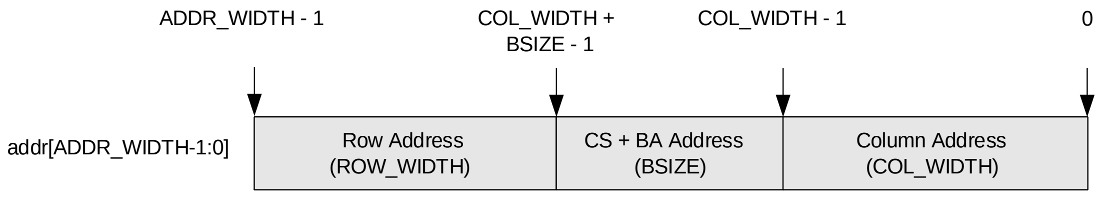


Figure 2.7. Local-to-Memory Address Mapping for Memory Access

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. Since the number of chip select is 1, the chip select address size becomes 0. An example of a typical address mapping is shown in Table 2.4 and Figure 2.8.

Table 2.4. Address Mapping Example

User Selection Name	User Value	Parameter Name	Parameter Value	Actual Line Size	Local Address Map
Column Size	11	COL_WIDTH	11	11	addr[10:0]
Bank Size	8	BANK_WIDTH	3	3	addr[13:11]
Chip Select Width	1	CS_WIDTH	1	0	—
Row Size	14	ROW_WIDTH	14	14	addr[27:14]
Total Local Address Line Size		ADDR_WIDTH	29	28	addr[27:0]

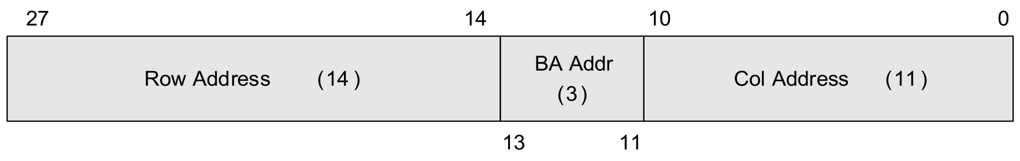


Figure 2.8. Mapped Address for the Example

2.8. Mode Register Access

The LPDDR2 SDRAM memory devices are programmed using a set of up to 256 Mode Registers. Mode Register address and the Opcode to the Mode Register (for MRW only) are sent to the memory device through the em_ddr_ca bus along with the MRW/MRR command. The memory data bus cannot be used for the Mode Register programming.

The Lattice LPDDR2 memory controller core uses the local address bus, addr, to program these registers. The core accepts a user command, MRW, to initiate the programming of mode registers. When MRW is applied on the cmd bus, the user logic must provide the information for the targeted mode register and the programming data on the addr bus. When the target mode register is programmed, the memory controller core is also configured to support the new memory setting. Table 2.5 shows how the local address lines are allocated for the programming of memory registers.

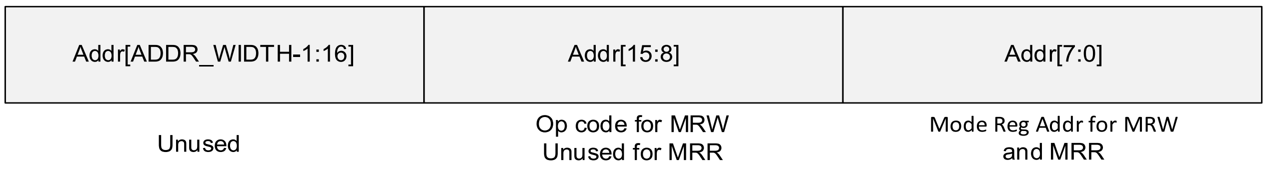


Figure 2.9. User-to-Memory Address Mapping for MR Programming

The register address (8 bits) is provided through the lower side of the addr bus starting from the bit 0 for LSB. The programming data requires 8 bits of the local address lines.

Table 2.5. Mode Register Selection Using Bank Address Bits

Mode Register	(addr[7:0])
MR0	8'h0
MR1	8'h1
—	—
MR255	8'ff

The initialization process uses the Mode register initial values selected through the interface. If these registers are not further programmed by the user logic, using MRW user command, they will remain in the configurations programmed during the initialization process. Table 2.6 shows the list of available parameters and their initial default values from the interface if they are not changed by the user.

Table 2.6. Initialization Default Values for Mode Register Setting

Type	Register	Value	Description	Local Address	GUI Setting
MR1	Burst Length	3'b011	BL=8	addr[10:8]	Yes
	Write Recovery	3'b100	6	addr[15:13]	Yes
MR2	Read & Write Latency	4'b0100	RL=6, WL=3	addr[11:8]	Yes
MR3	Drive strength (DS)	4'b0010	40 Ω PD/PU	addr[11:8]	Yes

The IP core supports mode register read operation using MRR user command. Since LPDDR2 supports only BL8 mode, any mode register read operation will generate a read_data_valid signal of 2 sclk duration at the user side. Content of the mode register is placed in the read_data bus at the first clock time of the read_data_valid signal. The value on read_data[7:0] at the first clock time of the active read_data_valid signal provides the content of the mode register accessed by MRR command.

3. Parameter Settings

The Clarity Designer tool is used to create IP and architectural modules in the Diamond software. Refer to the IP Core Generation and Evaluation section for a description on how to generate the IP.

Table 3.1 provides the list of user configurable parameters for the LPDDR2 SDRAM Controller Lite IP core. The parameter settings are specified using the LPDDR2 SDRAM Controller Lite IP core Configuration interface in Clarity Designer. The numerous LPDDR2 SDRAM Controller Lite parameter options are partitioned across multiple interface tabs as shown in this section.

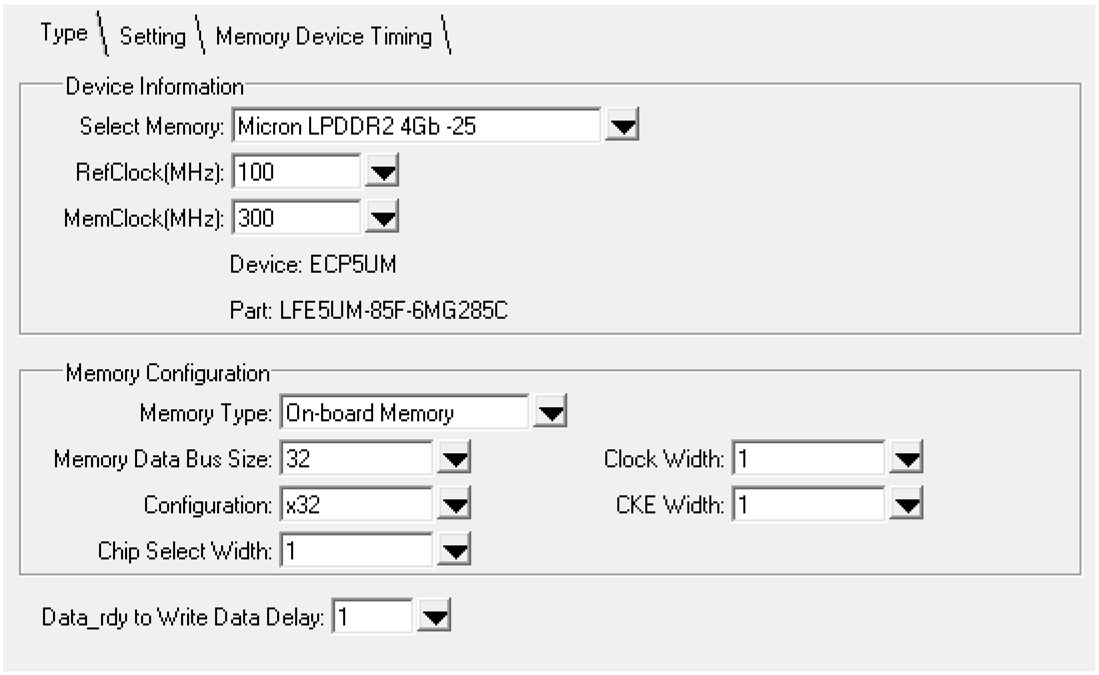
Table 3.1. IP Core Parameters

Parameters	Range/Options	Default
Type		
Device Information		
Select Memory	Micron LPDDR2 4Gb 25 / Micron LPDDR2 8Gb 25 / Custom	Micron LPDDR2 4Gb 25
Clock	400/350/300/250/200/150/100 MHz (for -8 device) 350/300/250/200/150/100 MHz (for -7 device) 300/250/200/150/100 MHz (for -6 device)	400 (for -8 device) 350 (for -7 device) 300 (for -6 device)
Reference Clock	25/50/100 MHz (for Clock=400 MHz) 25/50/87.5/100 MHz (for Clock=350 MHz) 25/50/75/100 MHz (for Clock=300 MHz) 25/50/62.5/75/100 MHz Clock=250 MHz) 25/50/75/100 MHz(for Clock =200 MHz) 25/37.5/50/75/100 MHz(for Clock=150 MHz) 25/50/75/100 MHz (for Clock=100 MHz)	100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz
Memory Configuration		
Memory Type	On-board Memory	On-board Memory
Memory Data Bus Size	16 / 32	32
Configuration	x16 (for Data Bus size 16) x32 (for Data Bus size 32)	x32
Chip Select Width	1	1
Clock Width	1	1
CKE Width	1	1
Data_rdy to Write Data Delay	1 / 2 / 3	1
Setting		
Address		
Bank Addr Width	2 - 3	3
Row size	12 - 15	14
Column size	8 - 12	10
Auto Refresh Control		
Refresh Type	Burst / Distributed	Burst
Refresh Bank	All bank / Per bank	All bank
Auto Refresh Control Burst Count	8 / 32 / 512 / 1024 / 2048 / 4096 /8192	32
External Auto Refresh Port	Unselected / Selected	Unselected
Mode Register Initial Setting		
Burst Length	Fixed 8	Fixed 8
Write Recovery	6 / 8 / 9 / 10 / 11 / 12	6
Read Latency	6 / 8	6
Write Latency	3 (for Read Latency 6) 4 (for Read Latency 8)	3

Parameters	Range/Options	Default
DS (Ω)	34.3 PD/PU 40.0 PD/PU 48.0 PD/PU 60.0 PD/PU 80.0 PD/PU 120.0 PD/PU	40.0 PD/PU
Memory Device Timing		
Command and Address Timing		
Manually Adjust	Unselected/Selected	Selected
TRTP (tCLK)	4 - 65536	4
TWTR (tCLK)	4 - 65536	4
TMRW (tCLK)	10 - 65536	10
TMRR (tCLK)	4 - 65536	4
TRCD (tCLK)	8 - 65536	8
TRP_AB (tCLK)	9 - 65536	9
TRP_PB (tCLK)	8 - 65536	8
TRC (tCLK)	26 - 65536	26
TRAS (tCLK)	17 - 65536	17
TFAW (tCLK)	20 - 65536	20
TRRD (tCLK)	4 - 65536	4
Calibration Timing		
TZQINIT (tCLK)	512 - 65536	512
TZQCS (tCLK)	64 - 65536	64
Refresh, Reset, and Power Down Timing		
TRFC_AB (tCLK)	52 - 65536	52
TRFC_PB (tCLK)	24 - 65536	24
TREFI_AB (tCLK)	44 - 1560	1560
TREFI_PB (tCLK)	44 - 196	196
TCKE (tCLK)	3 - 65536	4
TCKESR (tCLK)	6 - 65536	6
TXP (tCLK)	3 - 65536	3
TXSR (tCLK)	56 - 65536	56

3.1. Type Tab

The Type tab allows you to select the LPDDR2 controller configuration for the target memory device and the core functional features. These parameters are considered as static parameters since the values for these parameters can only be set in the interface. The LPDDR2 controller must be regenerated to change the value of any of these parameters. Figure 3.1 shows the contents of the Type tab.



Device Information

Select Memory: Micron LPDDR2 4Gb -25
RefClock(MHz): 100
MemClock(MHz): 300
Device: ECP5UM
Part: LFE5UM-85F-6MG285C

Memory Configuration

Memory Type: On-board Memory
Memory Data Bus Size: 32
Configuration: x32
Chip Select Width: 1
Clock Width: 1
CKE Width: 1

Data_rdy to Write Data Delay: 1

Figure 3.1. LPDDR2 SDRAM Controller Lite IP Core Type Options in the Clarity Designer Tool

The Type tab provides the following options.

3.1.1. Select Memory

The Micron LPDDR2 4GB -25 is provided as the default LPDDR2 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 LPDDR2 8GB -25 and Custom. When a desired configuration needs one or more different options from the default setting, the Custom option must be selected.

3.1.2. Clock

This parameter specifies the frequencies of the LPDDR2 memory clock. The allowed range for MemClock is from 100 MHz to 400 MHz. The default value is linked to the speed grade of Lattice device selected. For example, the default memory clock for ECP5 -8 devices is 400 MHz. The corresponding value for ECP5 -7 devices is 350 MHz, and the corresponding value for ECP5 -6 devices is 300 MHz.

3.1.3. Reference Clock

This parameter is dependent on memory clock selected. The allowed range for RefClock is 25 MHz to 100 MHz. The default value is 100 MHz.

3.1.4. Memory Type

This option is used to select the LPDDR2 module type. Only On-board memory type is supported.

3.1.5. Memory Data Bus Size

This option allows you to select the data bus width of the LPDDR2 memory module to which the memory controller core is connected.

3.1.6. Configuration

This option is used to select the device configuration of the on-board memory. The memory controller supports device configurations x16 and x32.

3.1.7. Chip Select Width

When On-board Module is selected as Memory Type, this option allows the user to select the number of Chip selects required for the on-board memory. Only one chip select is supported.

3.1.8. Clock Width

This field shows the number of clocks with which the memory controller drives the memory. The controller provides one differential clock per Chip select, as default.

3.1.9. CKE Width

This field shows the number of Clock Enable (CKE) signals with which the memory controller drives the memory. The controller provides one CKE signal per Chip select, as default.

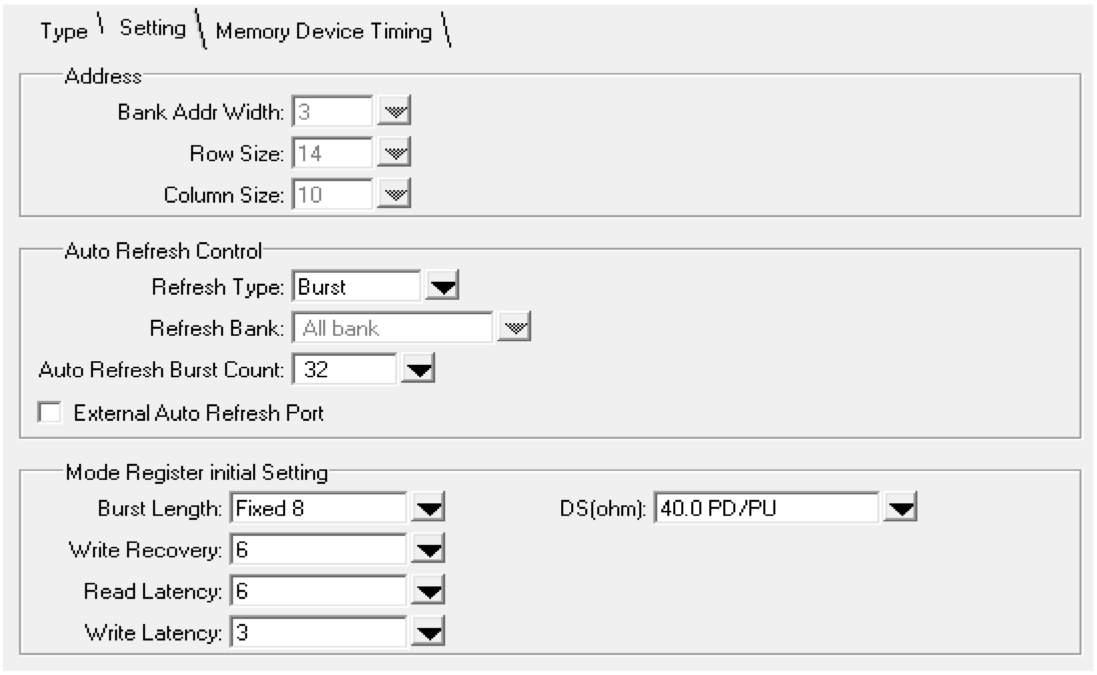
3.1.10. Data_rdy to Write Data Delay

This option allows you to select the number of clock cycles sending write data to the controller after the controller is ready to accept write data. User logic is allowed to send the write data to the controller after a one-clock cycle or two-clock cycle delay with respect to datain_rdy signal. Refer to the WRITE section for more information.

3.2. Setting Tab

The Setting tab enables you to select various configuration options for the target memory device/module. Parameters under the group, Mode Register Initial Setting, are dynamic parameters. Initialization values are set from the interface. These values are dynamically changeable using MRW user commands. (Refer to JESD209-2B, LPDDR2 SDRAM Standard, for allowed values).

Figure 3.2 shows the contents of the Setting tab.



Bank Addr Width: 3
Row Size: 14
Column Size: 10
Refresh Type: Burst
Refresh Bank: All bank
Auto Refresh Burst Count: 32
External Auto Refresh Port
Burst Length: Fixed 8
DS(ohm): 40.0 PD/PU
Write Recovery: 6
Read Latency: 6
Write Latency: 3

Figure 3.2. LPDDR2 SDRAM IP Core Setting Options in the Clarity Designer Tool

The Setting tab provides the following options.

3.2.1. Bank Address Width

This option indicates the default Bank Address Width used in selected memory configuration. If the option Custom is selected in Select memory field of Type Tab, you can choose a value other than the default value.

3.2.2. Row Size

This option indicates the default Row Address size used in the selected memory configuration. If the option Custom is selected in Select memory field of Type tab, you can choose a value other than the default value.

3.2.3. Column Size

This option indicates the default Column Address size used in the selected memory configuration. If the option Custom is selected in Select memory field of Type Tab, you can choose a value other than the default value.

3.2.4. Refresh Type

This option allows the user to select either Burst auto refresh or Distributed auto refresh. Refer to the REFRESH Support section for more details.

3.2.5. Refresh Bank

This option indicates that the core supports All Bank refresh. Per Bank refresh is not supported. Refer to the REFRESH Support section for more details about All Bank refresh.

3.2.6. Auto Refresh Burst Count

This option indicates the number of Auto Refresh commands that the memory controller core is set to send in a single burst. Refer to the REFRESH Support section for more details.

3.2.7. External Auto Refresh Port

This option, if selected, allows the user logic to generate a Refresh request to the controller. If this option is not selected, the controller automatically generates refresh commands to the memory at the interval defined by the Auto Refresh Burst count and memory refresh timing requirement. Refer to the REFRESH Support section for more details

3.2.8. Burst Length

This option sets the Burst length value in Mode Register 1 during initialization. In this release, only Burst Length 8 is supported.

3.2.9. READ Latency

This option sets the READ Latency value in Mode Register 1 during initialization. This value remains until you write a different value to the Mode Register.

3.2.10. Write Latency

This only displays the corresponding Write latency for the selected Read Latency. This value remains until you write a different Read Latency value to the Mode Register.

3.2.11. Write Recovery

This option sets the Write Recovery value in Mode Register 1 during initialization. It is set in terms of Memory clock. This value remains until you write a different value to the Mode Register.

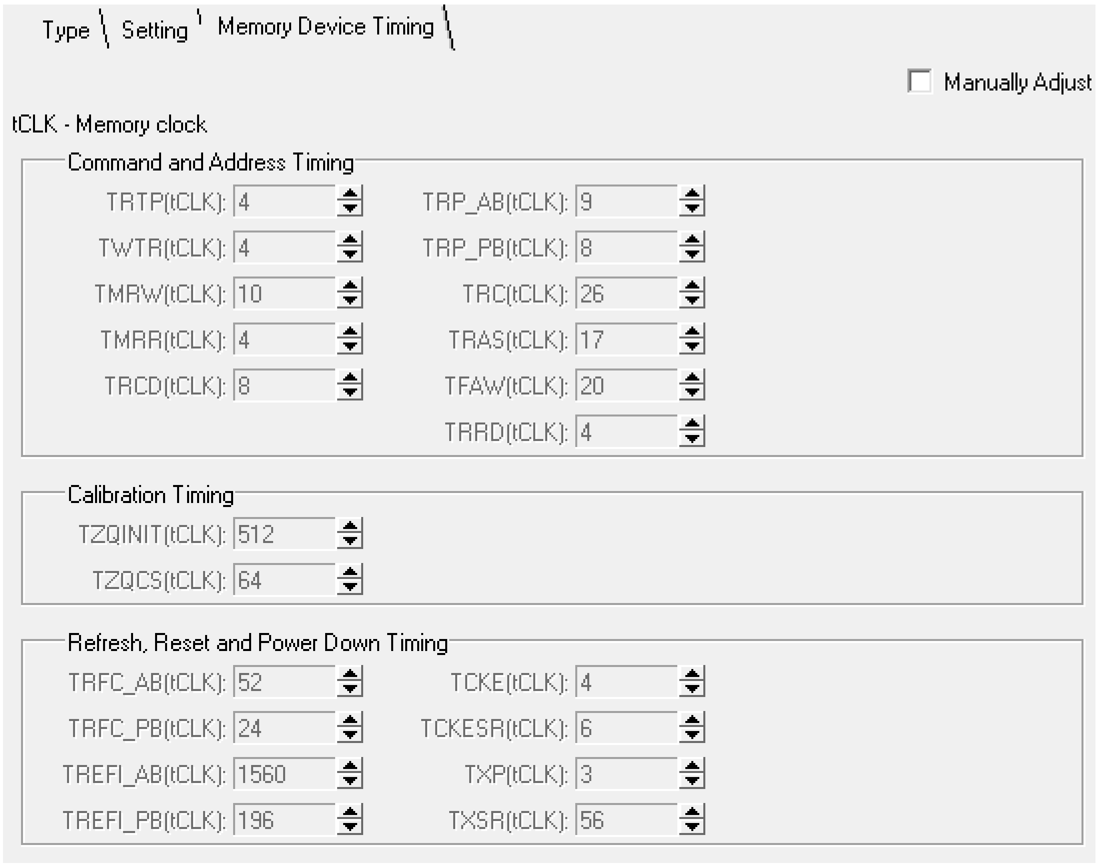
3.2.12. DS (Ω)

This option sets the Output Driver Strength Control value in Mode Register 3 during initialization. This value remains until you write a different value to the Mode Register. A list of Pull-down (PD) and Pull-up (PU) values are provided as defined by the LPDDR2's Mode Register 3.

3.3. Memory Device Timing Tab

Figure 3.3 shows the contents of the Memory Device Timing tab. The default memory timing parameters displayed in this tab are the default values of the Micron LPDDR2 4 Gb -25 memory device. You can adjust these parameters by selecting the Manual Adjust checkbox.

It is important that the values in this Memory Device Timing tab are adjusted to the timing parameters of the onboard memory device that you plan to use in their application. The LPDDR2 Controller also uses these timing parameters when generating memory commands.



☐ Manually Adjust

tCLK - Memory clock

Command and Address Timing

TRTP(tCLK):	4	TRP_AB(tCLK):	9
TWTR(tCLK):	4	TRP_PB(tCLK):	8
TMRW(tCLK):	10	TRC(tCLK):	26
TMRR(tCLK):	4	TRAS(tCLK):	17
TRCD(tCLK):	8	TFAW(tCLK):	20
		TRRD(tCLK):	4

Calibration Timing

TZQINIT(tCLK):	512
TZQCS(tCLK):	64

Refresh, Reset and Power Down Timing

TRFC_AB(tCLK):	52	TCKE(tCLK):	4
TRFC_PB(tCLK):	24	TCKESR(tCLK):	6
TREFI_AB(tCLK):	1560	TXP(tCLK):	3
TREFI_PB(tCLK):	196	TXSR(tCLK):	56

Figure 3.3. DDR2 SDRAM IP Core Memory Device Timing Options in the Clarity Designer Tool

The Memory Device Timing tab provides the following options.

3.3.1. Manually Adjust

Checking this box allows you to manually set (via increment/decrement) 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 will enable you to increment/decrement the memory timing parameters.

3.3.2. tCLK - Memory clock

This is a notation signifying that the memory timing parameters shown in this tab are specified in terms of tCLK LPDDR2 memory clock cycles.

3.3.3. Command and Address Timing

The Memory Device Timing parameters listed in this tab are standard parameters as defined in JESD209-2B, LPDDR2 SDRAM Standard. Refer to the memory device data sheet for detailed descriptions and allowed values of these parameters.

3.3.4. Calibration Timing

The Memory Device Timing parameters listed in this tab are standard parameters as defined in JESD209-2B, LPDDR2 SDRAM Standard. Refer to the memory device data sheet for detailed descriptions and allowed values of these parameters.

3.3.5. Refresh, Reset and Power Down Timing

The Memory Device Timing parameters listed in this tab are standard parameters as defined in JESD209-2B, LPDDR2 SDRAM Standard. Refer to the memory device data sheet for detailed descriptions and allowed values of these parameters.

4. IP Core Generation and Evaluation

This chapter provides information on how to generate the LPDDR2 SDRAM Controller Lite IP core using the Lattice Diamond design software Clarity Designer tool, and how to include the core in a top-level design.

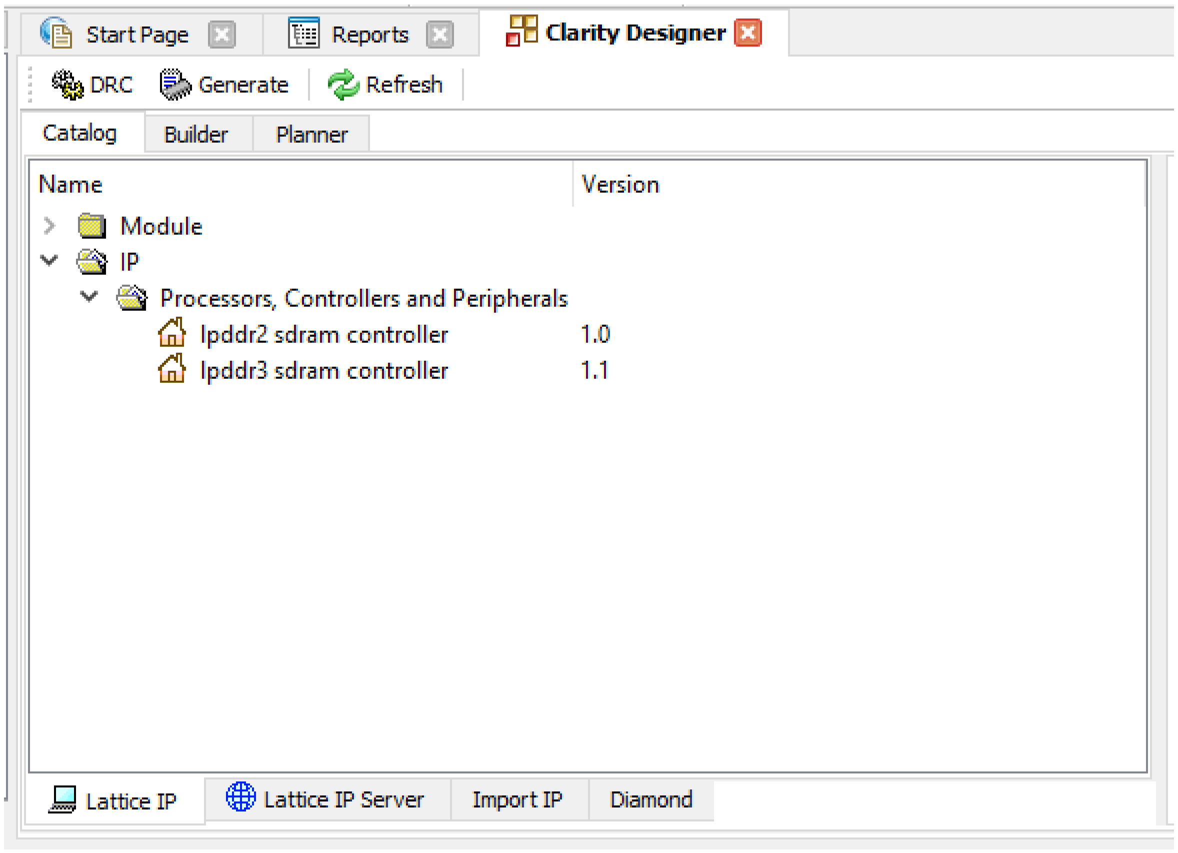
The LPDDR2 SDRAM Controller Lite IP core can be used in the ECP5 device family.

For example, information and known issues on this core can be found in the Lattice LPDDR2 IP ReadMe document. This file is available once the core is installed in Diamond. This document provides information on creating an evaluation version of the core for use in Diamond and simulation.

IP configuration and generation are done using Lattice's Clarity Designer tool. Please refer to the Clarity Designer User Manual for more information about this tool. This User Manual explains the procedure for IP or Module Generation, Building and DDR I/O planning. Detailed information about Clarity Designer interface is available in this user guide for better understanding of the tool.

4.1. Getting Started

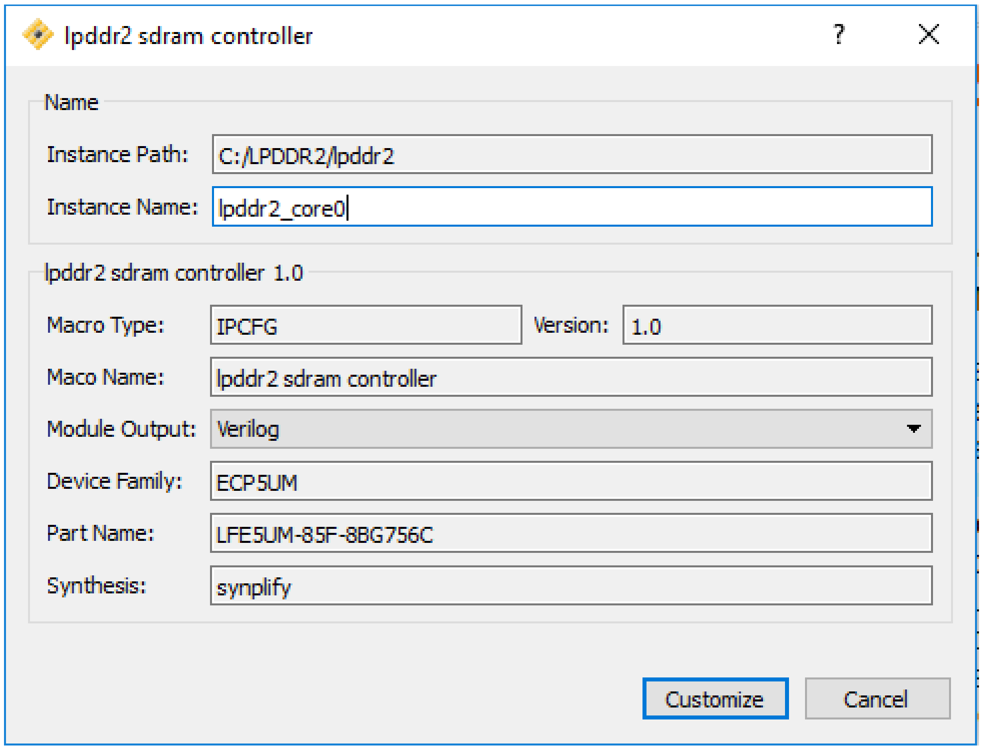
LPDDR2 SDRAM IP core is available for download from Lattice's IP Server using Clarity Designer tool. The IP core can be downloaded and installed from the Lattice IP Server tab. After installed, the LPDDR2 SDRAM Controller Lite IP core will be available in the Clarity Designer IP Catalog window as shown in Figure 4.1.



Name	Version
Module	
IP	
Processors, Controllers and Peripherals	
lpddr2 sdram controller	1.0
lpddr3 sdram controller	1.1

Figure 4.1. Clarity Designer IP Catalog Window

To configure the IP core, double click the LPDDR2 IP shown in the Catalog tab of the interface. This step opens the IP Project Dialog box as shown in Figure 4.2.



Instance Path: C:/LPDDR2/lpddr2
Instance Name: lpddr2_core0
Macro Type: IPCFG
Version: 1.0
Maco Name: lpddr2 sdram controller
Module Output: Verilog
Device Family: ECP5UM
Part Name: LFE5UM-85F-8BG756C
Synthesis: synplify

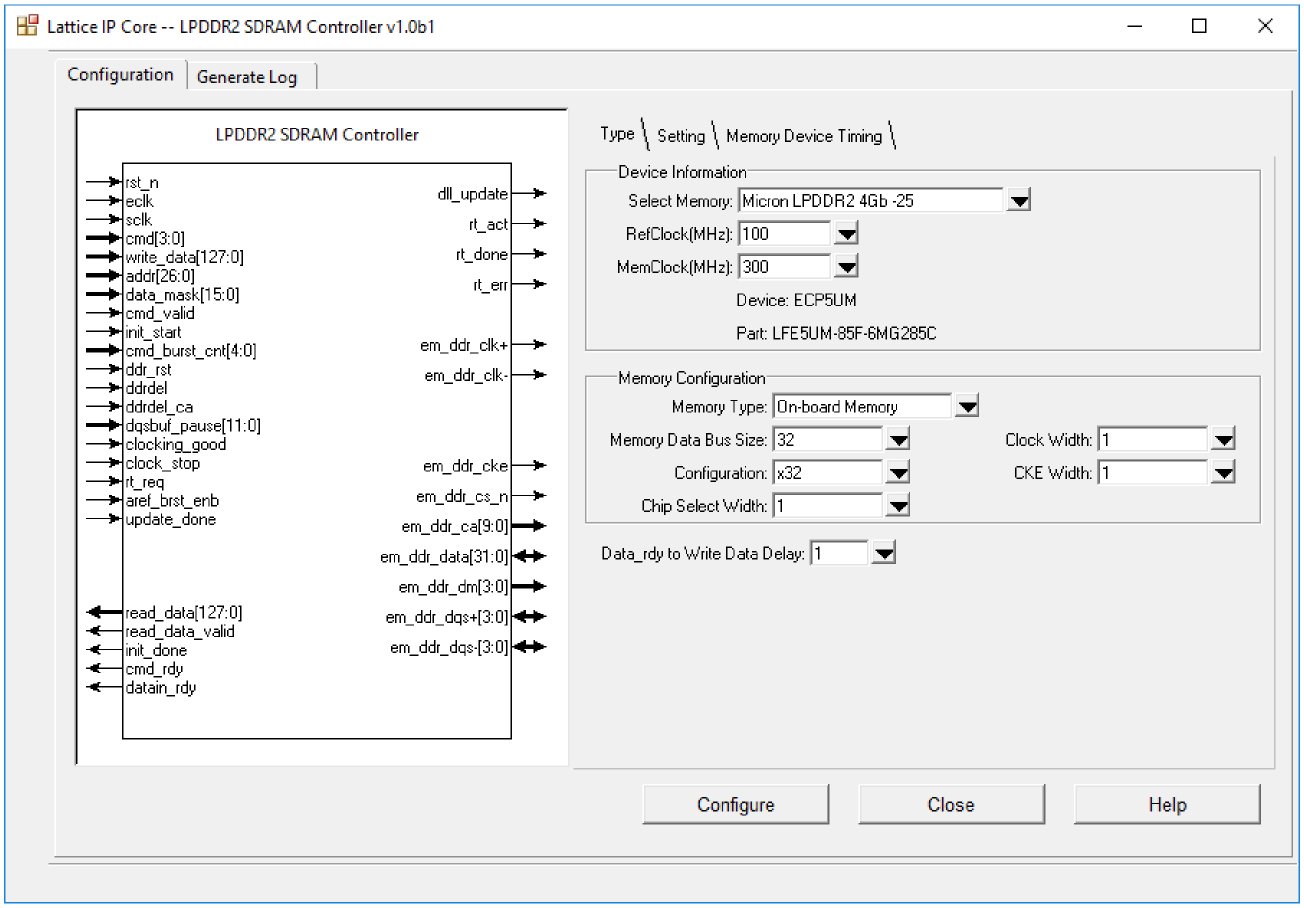
Figure 4.2. LPDDR2 IP Project Dialog Box

Enter the following information as described below:

- **Instance Path** – Path to the directory where the generated IP files will be loaded.
- **Instance Name** – Username designation given to the generated IP core and corresponding folders and files. (***Caution***: mobile_ddr2 and lpddr2_sdram_core are Lattice reserved names. You should not use any of these names as file name.)
- **Module Output** – Verilog
- **Device Family** – Shows the selected device family.
- **Part Name** – Shows the selected part within the selected device family.

To create a custom configuration:

1. Click the **Customize** button in the IP Project dialog box to display the LPDDR2 SDRAM IP core Configuration interface as shown in Figure 4.3.
2. Select the IP parameter options specific to their application. Refer to the Parameter Settings section for more information on the LPDDR2 parameter settings.



Device Information

Select Memory: Micron LPDDR2 4Gb -25

RefClock(MHz): 100

MemClock(MHz): 300

Device: ECP5UM

Part: LFE5UM-85F-6MG285C

Memory Configuration

Memory Type: On-board Memory

Memory Data Bus Size: 32

Clock Width: 1

Configuration: x32

CKE Width: 1

Chip Select Width: 1

Data_rdy to Write Data Delay: 1

Figure 4.3. Configuration Interface

4.2. Clarity Designer-Created Files and IP Top Level Directory Structure

When you click the Generate button in the IP Configuration dialog box, the IP core and supporting files are generated in the specified Project Path directory. The directory structure of the generated files is shown in Figure 4.4.

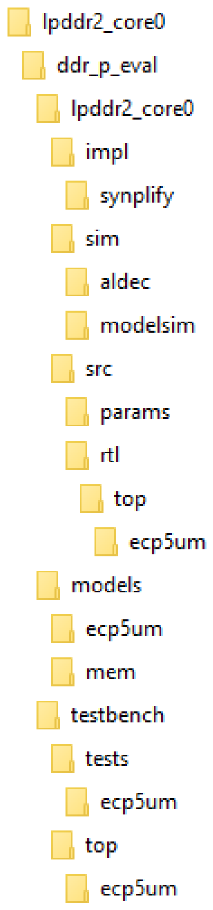


Figure 4.4. ECP5 LPDDR2 Core Directory Structure

Understanding the core structure is an important step of a system design using the core. A summary of the files of the core for simulation and synthesis are listed in Table 4.1.

Table 4.1 provides a list of key files and directories created by the Clarity Designer tool and how they are used. The Clarity Designer tool creates several files that are used throughout the design cycle. The names of most of the created files are customized to the user's module name specified in the Clarity Designer tool.

Table 4.1. File List

File	Simulation	Synthesis	Description
Source Files			
<username>.lpc			This file contains the Clarity Designer tool options used to recreate or modify the core in the Clarity Designer tool.
..\src\params\lpddr2_sdram_mem_params_<username>.v	Yes		This file provides user options of the IP for the simulation models
<username>_beh.v	Yes		This is the obfuscated core simulation model.
..\src\rtl\top\ecp5um\lpddr2_sdram_mem_top_wrapper_<username>.v		Yes	This is the top Level file for simulation and synthesis (.v file if

File	Simulation	Synthesis	Description
..\src\rtl\top\ecp5um\lpddr2_sdram_mem_top_wrapper_<username>.vhd			Verilog is selected or .vhd file if VHDL is selected). This file has black-box instantiations of the core and I/O modules and also source instantiation of clock synchronization module. Refer the section Dummy Logic in Core Evaluation for more details.
<username>.ngo		Yes	This file provides the synthesized IP core.
Model Files			
ddr_clks_<username>.v	Yes		These are source files of Clock synchronization logic. PLL and DQSDLL are included in this module and are used to generate system clock (SCLK) for the core and edge clock (ECLK) for I/O logic.
..\models\mem\lpddr2.sv	Yes		Memory simulation model and memory wrapper. (mem_data_width: 16/32).
..\models\mem\lpddr2_dimm_<mem_data_width>.v	Yes		
..\models\mem\mobile_ddr2_parameters.vh	Yes		
Evaluation Testbench Files			
..\testbench\top\ecp5um\test_mem_ctrl.v	Yes		This is the evaluation testbench top level file.
..\testbench\top\ecp5um\monitor.v	Yes		These are monitor files for the evaluation testbench.
..\tests\ecp5um\cmd_gen.v	Yes		This is the command generator for the evaluation testbench.
..\tests\ecp5um\tb_config_params.v	Yes		This file is the testbench configuration parameter.
..\tests\ecp5um\testcase.v	Yes		This file is the evaluation testbench.
Evaluation Script Files			
..\sim\aldec\<core_name>_eval.do	Yes		This file is the Aldec Active-HDL script for RTL simulation.
..\sim\aldec\<core_name>_gatesim_synplify.do	Yes		This file is the Aldec Active-HDL script for netlist simulation. This file is generated only if the selected device package has enough I/Os for all the user side and memory side signals.
..\sim\modelsim\<core_name>_eval.do	Yes		This file is the ModelSim script for RTL simulation.
..\sim\modelsim\<core_name>_gatesim_<synthesis>.do	Yes		This file is the ModelSim script for netlist simulation. This file is generated only if the selected device package has enough I/Os for all the user side and memory side signals. <synthesis>: synplify or lse
..\impl\<synthesis>\<username>_eval.ldf		Yes	This is the Diamond project file for evaluation.
..\impl\<synthesis>\<username>_eval.lpf		Yes	This is the par preference file for evaluation.

4.2.1. LPDDR2 Memory Controller IP File Structure

The LPDDR2 memory controller IP consists of the following four major functional blocks:

- Top-level wrapper (RTL)
- Obfuscated model for Memory Controller and PHY modules for simulation and corresponding encrypted netlist for synthesis
- Clock Synchronous Module (RTL for simulation and Verilog flow synthesis or netlist for VHDL flowsynthesis)

All of these blocks are required to implement the IP on the target FPGA device. Figure 4.5 depicts the interconnection among those blocks.

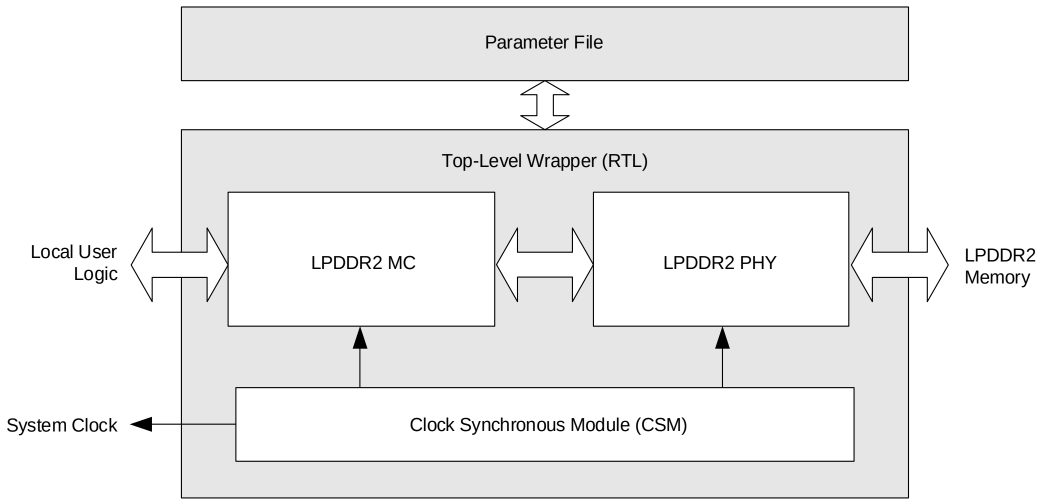


Figure 4.5. File Structure of LPDDR2 Memory Controller IP

Top-level Wrapper

The core, I/O modules, and the CSM block are instantiated in the top-level wrapper. When a system design is made with the Lattice LPDDR2 memory controller core, this wrapper must be instantiated. The wrapper is fully parameterized by the generated parameter file.

Clock Synchronization Module

The LPDDR2 memory controller has a clock synchronization module that generates the system clock (sclk) for the core and edge clock (eclk) for the I/O modules. The PLL implemented this block takes the reference clock input and generates the SCLK and ECLK outputs. In addition to clock generation, this block performs a synchronization process after every reset to lock a pre-defined phase relationship between these clocks. This clock synchronization block uses a DDRDLL to extract a PVT-compensated 90-degree delay count to the I/O block that appropriately shifts the DQS signal during write and read operations.

The clock output (sclk) from the clock generator that is used to drive the core logic is also made available to the external user logic. If a system that uses the LPDDR2 memory controller IP is required to have a clock generator that is external to the IP, the incorporated clock generator block can be shifted out from the IP. Connections between the top-level wrapper and the clock generator are fully RTL based, and therefore, it is possible to modify the structure and connection of the core for the clock distribution to meet system needs.

This module is provided as RTL source for all cases of simulation and for Verilog flow synthesis. For VHDL flow synthesis, this module is available as a netlist.

4.2.2. Simulation Files for IP Evaluation

Once an LPDDR2 memory controller IP is generated, it contains a complete set of testbench files to simulate a few example core activities for evaluation. The simulation environment for the LPDDR2 memory controller IP is shown in Figure 4.6. This structure can be reused by system designers to accelerate their system validation.

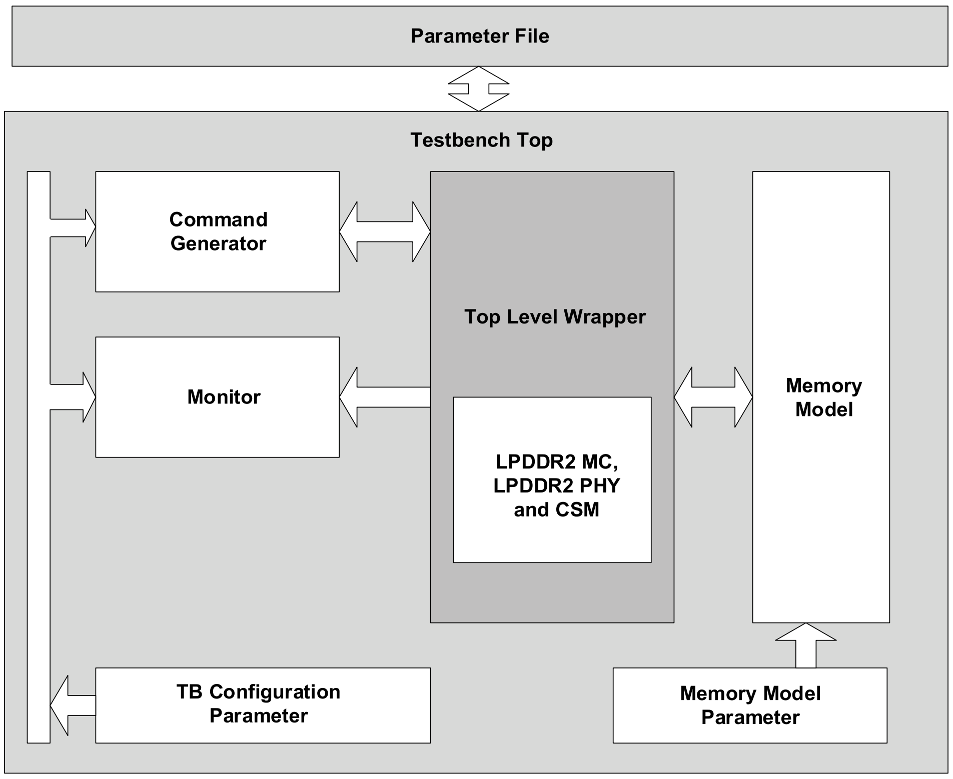


Figure 4.6. Simulation Structure for LPDDR2 Memory Controller Core Evaluation

Testbench Top

The testbench top includes the core under test, memory model, stimulus generator and monitor blocks. It is parameterized by the core parameter file.

Obfuscated Controller Simulation Model

The obfuscated simulation model for the controller includes the MC and PHY modules. This obfuscated simulation model must be included in the simulation.

Command Generator

The command generator generates stimuli for the core. The core initialization and command generation activities are predefined in the provided test case module. It is possible to customize the test case module to see the desired activities of the core.

Monitor

The monitor block monitors both the local user interface and LPDDR2 interface activities and generates a warning or an error if any violation is detected. It also validates the core data transfer activities by comparing the read data with the written data.

Testbench Configuration Parameter

The testbench configuration parameter provides the parameters for testbench files. These parameters are derived from the core parameter file and are not required to configure them separately. For those users who need a special memory configuration, however, modifying this parameter set might provide a support for the desired configuration.

Memory Model

The LPDDR2 memory controller testbench uses a memory simulation model provided by one of the most popular memory vendors. If a different memory model is required, it can be used by simply replacing the instantiation of the model from the memory configuration modules located in the same folder.

Memory Model Parameter

This memory parameter file comes with the memory simulation model. It contains the parameters that the memory simulation model needs. It is not necessary for users to change any of these parameters.

Evaluation Script File

A ModelSim and Aldec ACTIVE-HDL simulation macro script files are included for instant evaluation of the IP. All required files for simulation are included in the macro script. This simulation script can be used as a starting point of a user simulation project.

Note on Shortening Simulation Run Time

The memory controller implements many timers to comply with JEDEC specifications. Due to these timers, the functional simulation takes longer time at various stages of the simulation. In order to reduce the simulation run time, the controller has an option to lower the timer counts, particularly on those timers for waiting period. This option can be enabled by adding a define SIM in the simulation script. It is important to note that this reduced timer value is good only for the simulation and this define should not be included in the synthesis script.

4.3. Hardware Evaluation

The LPDDR2 SDRAM IP core supports Lattice's IP hardware evaluation capability, which 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.

4.3.1. Enabling Hardware Evaluation in Diamond:

Choose **Project > Active Strategy > Translate Design Settings**. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default.

4.4. Regenerating/Recreating the IP Core

By regenerating an IP core with the Clarity Designer tool, you can modify any of the options specific to an existing IP instance. By recreating an IP core with Clarity Designer tool, you can create (and modify if needed) a new IP instance with an existing LPC/IPX configuration file.

4.4.1. Regenerating an IP Core in Clarity Designer Tool

To regenerate an IP core in Clarity Designer:

1. In the Clarity Designer Builder tab, right-click on the existing IP instance and choose **Config**.
2. In the dialog box, choose the desired options.

   For more information about the options, click **Help**. You may also click the **About** tab in the Clarity Designer window for links to technical notes and user guides. The IP may come with additional information. As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module will need.
3. Click **Configure**.

4.4.2. Recreating an IP Core in Clarity Designer Tool

To recreate an IP core in Clarity Designer:

1. In the Clarity Designer Catalog tab, click the **Import IP** tab at the bottom.
2. In the Import IP tab, choose the existing IPX/LPC source file of the module or IP to regenerate.
3. Specify the instance name in **Target Instance**. Note that this instance name should not be the same as any of the existing IP instances in the current Clarity Design project.

4. Click **Import**. The module's dialog box opens showing the option settings.
5. In the dialog box, choose the desired options.

   For more information about the options, click **Help**. You may also click the **About** tab in the Clarity Designer window for links to technical notes and user guides. The IP may come with additional information. As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module will need.
6. Click **Configure**.

5. Application Support

This chapter provides supporting information on using the LPDDR2 SDRAM Controller Lite IP in complete designs.

5.1. Understanding Preferences

The generated preference file has many preferences that will fall mainly into one of these categories:

5.1.1. FREQUENCY Preferences

Each clock domain in the controller is defined by a Frequency preference.

5.1.2. MAXDELAY NET

The MAXDELAY NET preference ensures that the net has a minimal net delay and falls within the allowed limit. Since this preference is highly over-constrained, the post-route trace preference file should be used to validate the timing results.

5.1.3. MULTICYCLE / BLOCK PATH

The MULTICYCLE preference is applied to a path that is covered by the FREQUENCY constraint, but is allowed to be relaxed from its FREQUENCY constraint. The FREQUENCY constraint is relaxed in multiples of the clock period.

The BLOCK preference is applied to a path that is not relevant for the timing analysis.

5.1.4. IOBUF

The IOBUF preference assigns the required I/O types and attributes to the LPDDR2 I/O pads.

For more details on memory controller pinout guidelines, refer to ECP5 High-Speed I/O Interface (TN1265).

5.2. Dummy Logic in Core Evaluation

When an LPDDR2 IP core is generated, Clarity Designer assigns all the signals from both the LPDDR2 and local user interfaces to the I/O pads. The number of LPDDR2 IP's user interface signals for read and write data buses together is normally more than eight times than that of the LPDDR2 memory interface. It is impossible for the core to be generated if the selected device does not have enough I/O pad resources. To facilitate the core evaluation with smaller package devices, Clarity Designer inserts dummy logic to decrease the I/O pad counts by reducing the local read_data and write_data bus sizes. With the dummy logic, a core can be successfully generated and evaluated even with smaller pad counts. The PAR process can be completed without a resource violation so that one can evaluate the performance and utilization of the core. However, the synthesized netlist will not function correctly because of the inserted dummy logic. The top-level wrapper that includes such dummy logic must be used only for the implementation evaluation, and therefore, is located under a different directory from the one that includes the regular top-level wrapper.

5.3. Top-level Wrapper File Only for Evaluation Implementation

For evaluation implementation using the Verilog core, a separate top level wrapper file, lpddr2_sdram_mem_top_wrapper.v is provided in the directory ..\ddr_p_eval\<usr_name>\impl. This wrapper file has a reduced number of local side data busses for the reason mentioned in the previous paragraph. The eval par project file <usr_name>_eval.ldf in the directory ..\ddr_p_eval\<usr_name>\impl\<synthesis> points to this wrapper file for running evaluation implementation.

Note that this top-level wrapper file is not used for evaluation simulation.

5.4. Top-level Wrapper file for All Simulation Cases and Implementation in a User's Design

In real applications, since back end user logic design is attached to the core, most of the user side interface signals are embedded within the FPGA fabric and will not be connected to the pads of the FPGA fabric. There is a main top level wrapper file, lpddr2_sdram_mem_top_wrapper.v, in the directory

..\ddr_p_eval\<usr_name>\src\rtl\top\ecp5um. This wrapper is generated with a full local side data bus and is meant for simulation as well as for the final integration with user's logic for synthesis. The user's par project file should point to this top-level wrapper file while implementing the IP core in the user's application.

6. Core Validation

The functionality of the LPDDR2 SDRAM Controller Lite IP core has been verified via simulation using Lattice's in-house test bench environment and hardware validation.

Reference

ECP5 High-Speed I/O Interface (TN1265)

Technical Support Assistance

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

Appendix A. Resource Utilization

This appendix provides resource utilization information for Lattice FPGAs using the LPDDR2 SDRAM Controller Lite IP core. The IP configurations shown in this chapter were generated using the IPexpress software tool and Clarity Designer tool. IPexpress and Clarity Designer are the Lattice IP configuration utility, and are included as a standard feature of the Diamond design tool. Details regarding the usage of IPexpress and Clarity Designer can be found in the IPexpress, Clarity Designer and Diamond help systems. For more information on the Diamond design tool, visit the Lattice web site at: www.latticesemi.com/software.

ECP5 Devices

Table A.1. Performance and Resource Utilization[1]

Parameter	Slices	LUTs	Registers	I/O[2]	f_{MAX} (MHz)[3]
Data Bus Width: 16 (x16)	1599	2241	1639	34	400 MHz (800 Mbps)
Data Bus Width: 32 (x32)	1818	2462	1937	54	

***Notes:**

1. Performance and utilization data are generated targeting an LFE5UM-85F-8BG756CES device using Lattice Diamond 3.10 design software with an LFE5UM control pack. Performance may vary when using a different software version or targeting a different device density or speed grade within the ECP5 family.
2. Numbers shown in the I/O column represent the number of primary I/Os at the LPDDR2 memory interface. User interface (local side) I/Os are not included.
3. The LPDDR2 IP core can operate at 400 MHz (800 LPDDR2) in the fastest speed-grade (-8) when the data width is 32 bits or less and one chip select is used.

Ordering Part Number

The Ordering Part Number (OPN) for the LPDDR2 SDRAM Controller Lite IP on ECP5 devices is LPDDR2L-E5-U (for single-design license) or LPDDR2L-E5-UT (for multi-site license).

Revision History

Date	Document Version	IP Version	Change Summary
July 2018	1.0	1.0	Initial release.