LatticeCORE

DDR & DDR2 SDRAM Controller for MachXO2 PLD Family IP Cores User Guide

(Piplelined Versions)

March 2015
IPUG93_1.2

Table of Contents

Chapter 1. Introduction 5
Quick Facts 5
Features 6
Chapter 2. Functional Description 7
Command Decode Logic 7
Configuration Interface 7
sysCLOCK PLL 8
Data Path Logic 8
Initialization State Machine 8
Command Application Logic 8
DDR I/O Modules 8
Signal Descriptions 8
Using the Local User Interface 9
Initialization and Auto-Refresh Control 10
Command and Address 11
Data Write 12
Data Read 13
Read/Write with Auto Precharge 13
Local-to-Memory Address Mapping 13
Mode Register Programming 14
Memory Interface 17
Chapter 3. Parameter Settings 18
Mode Tab 19
Type Tab 20
Select Memory 20
Clock 20
Memory Data Bus Size 20
Configuration 20
Data_rdy to Write Data Delay 20
Clock Width 20
CKE Width 21
Fixed Memory Timing 21
Setting Tab 21
Row Size 21
Column Size 21
Bank Size 22
Chip Select Width 22
User Slot Size 22
EMR Prog During Init 22
Auto Refresh Burst Count 22
External Auto Refresh Port 22
Mode Register Initial Setting 22
Timing Tab 22
Synthesis & Simulation Tools Option Tab 24
Info Tab 24
Chapter 4. IP Core Generation 25
Licensing the IP Core 25
Getting Started 25
IPexpress-Created Files and Top Level Directory Structure 27
Generated Files 27

DDR Memory Controller Core Structure 28
Top-level Wrapper 29
Encrypted Netlist 29
I/O Modules 29
Clock Generator 29
Parameter File 29
Core Header File 29
Preference Files 29
Evaluation Project Files 29
Simulation Files for Core Evaluation 30
Testbench Top 30
Obfuscated Core Simulation Model 30
Command Generator 30
Monitor 30
TB Configuration Parameter 31
Memory Model 31
Memory Model Parameter 31
Evaluation Script File 31
Hardware Evaluation 31
Enabling Hardware Evaluation in Diamond 31
Enabling Hardware Evaluation in ispLEVER 31
Updating/Regenerating the IP Core 31
Regenerating an IP Core in Diamond 31
Regenerating an IP Core in ispLEVER 32
Chapter 5. Application Support 33
Core Implementation 33
Understanding Preferences 33
Preference Localization 33
VREF Assignments 34
DLL Allocation 34
I/O Types for DDR 34
Skew Treatment 34
Dummy Logic Removal 35
Read Data Auto-Alignment Logic 35
PCB Routing Delay Compensation 35
DQS_PIO_READ Locate Constraints 36
Obtaining Location Values in Diamond Software 36
Obtaining Location Values in ispLEVER Software 37
Troubleshooting 38
Chapter 6. Core Verification 40
Chapter 7. Support Resources 41
Lattice Technical Support 41
Online Forums 41
Telephone Support Hotline 41
E-mail Support 41
Local Support 41
Internet 41
References 41
MachXO2 41
JEDEC Website 41
Micron Technology, Inc., Website 41
Revision History 42
Appendix A. Resource Utilization 43

MachXO2 Devices 43
Ordering Part Number 43

Introduction

The Double Data Rate (DDR) Synchronous Dynamic Random Access Memory (SDRAM) Controller is a general-purpose memory controller that interfaces with industry standard DDR/DDR2 memory devices/modules and provides a generic command interface to user applications. This core reduces the efforts required to integrate the DDR/DDR2 memory controller with the remainder of the application and minimizes the need to deal with the DDR/DDR2 memory interface. This core utilizes dedicated DDR input and output registers in the Lattice devices to meet the requirements for high-speed double data rate transfers. The timing parameters for a memory device or module can be set through the signals that are input to the core as a part of the configuration interface. This capability enables effortless switching among different memory devices by updating the timing parameters to suit the application without generating a new core configuration.

Throughout this user's guide, the term 'DDR' is used to represent the first-generation DDR memory. Since this document covers both the Lattice DDR and DDR2 memory controller IP cores, use of the term 'DDR' indicates both DDR and DDR2.

Quick Facts

Table 1-1 gives quick facts about the DDR IP core for MachXO2™ devices.

Table 1-1. DDR IP Core Quick Facts

		DDR IP Configuration		
		x16 1cs	**x16 1cs**	**x16 1cs**
Core Requirements	Device Family supported	MachXO2		
	Minimal Device needed	LCMXO2-2000HC-6FTG256CES		
Resource Utilization	Targeted Device	LCMXO2-2000HC-6FTG256CES	LCMXO2-4000HC-6FTG256CES	LCMXO2-7000HC-6FTG256CES
	Data Path Width	16		
	LUTs	1200		
	sysMEM EBRs	0		
	Registers	1150		
Design Tool Support	Lattice Implementation	Lattice Diamond™ 1.0 or ispLEVER® 8.1		
	Synthesis	Synopsys® Synplify™ Pro for Lattice D-2009.12L-1		
	Simulation	Aldec® Active-HDL™ 8.2 Lattice Edition		
		Mentor Graphics® ModelSim™ SE 6.3F		

Table 1-2 gives quick facts about the DDR2 IP core for MachXO2 devices.

Table 1-2. DDR2 IP Core Quick Facts

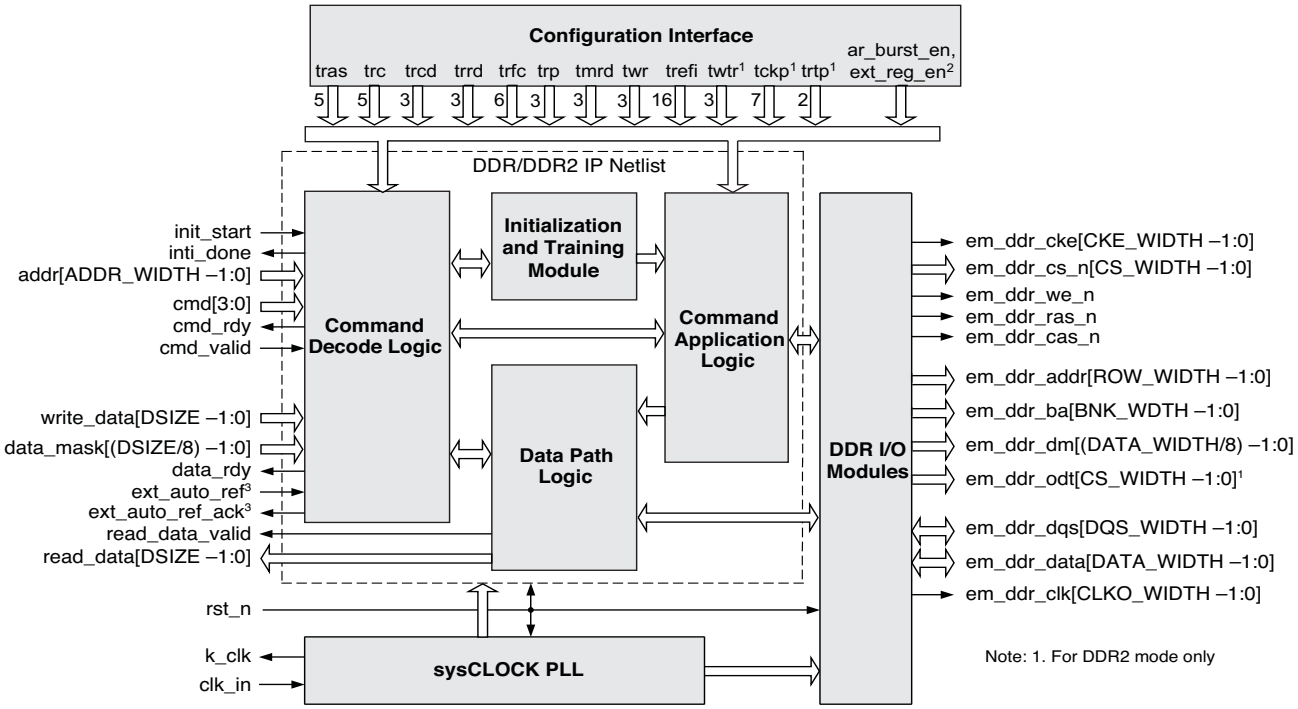
		DDR2 IP Configuration		
		x16 1cs	**x16 1cs**	**x16 1cs**
Core Requirements	Device Family supported	MachXO2		
	Minimal Device needed	LCMXO2-2000HC-6FTG256CES		
Resource Utilization	Targeted Device	LCMXO2-2000HC-6FTG256CES	LCMXO2-4000HC-6FTG256CES	LCMXO2-7000HC-6FTG256CES
	Data Path Width	16		
	LUTs	1325		
	sysMEM EBRs	0		
	Registers	1200		
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVE® 8.1		
	Synthesis	Synopsys® Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec® Active-HDL 8.2 Lattice Edition		
		Mentor Graphics ModelSim™ SE 6.3F		

Features

- Interfaces to industry standard DDR/DDR2 SDRAM devices and modules
- MachXO2 devices support DDR2 performance upto 266 Mbps. Although DDR2 SDRAM standard (JESD79-2F, www.jedec.org/standards-documents/docs/jesd-79-2-e) supports 400 Mbps and higher speeds, Micron Technology, Inc. (and possibly others) support operation below 400 Mbps.
- Programmable burst lengths of 2, 4 or 8 for DDR and 4 or 8 for DDR2
- Programmable CAS latency of 2 or 3 cycles for DDR and 3, 4, 5 or 6 cycles for DDR2
- Intelligent bank management to optimize performance by minimizing ACTIVE commands
- Supports all JEDEC standard DDR commands
- Two-stage command pipeline to improve throughput
- Supports unbuffered DIMM
- Supports all common memory configurations
  - — SDRAM data path width of 16 bits max.
  - — Variable address widths for different memory devices
  - — Up to 8 (DDR) or 4 (DDR2) chip selects for multiple SO/DIMM support
  - — Programmable memory timing parameters
  - — Byte-level writing through data mask signals

The DDR memory controller consists of two major parts, the encrypted netlist and I/O modules. The encrypted netlist comprises several internal blocks, as shown in Figure 2-1. The device architecture-dependent I/O modules are provided in RTL form. This section briefly describes the operation of each of these blocks.

Figure 2-1. DDR SDRAM Controller Block Diagram



Command Decode Logic

The Command Decode Logic (CDL) block accepts the user commands from the local interface. The accepted command is decoded to determine how the core will act to access the memory. When an accepted command is decoded as a write command, the CDL block asks the user logic to provide the write data. Once it receives the write data from the user logic, the CDL block delivers a write command to the Command Application Logic (CAL) block and the data is sent to the Data Path Logic (DPL) block. Similarly, when the accepted command is a read command, the CDL block sends a read command to the DPL block to generate a read command on the memory interface. The data read from memory is presented to the local user interface.

Intelligent bank management logic tracks the open/close status of every bank and stores the row address of every open bank. This information is used to reduce the number of PRECHARGE and ACTIVE commands issued to the memory. The controller also utilizes two pipelines to improve throughput. One command in the queue is decoded while another is presented at the memory interface.

Configuration Interface

The Configuration Interface (CI) block provides the DDR memory controller with the core reconfiguration capability for the memory timing parameters and other core configuration inputs. The configuration interface for the memory timing parameters can be enabled or disabled via a user parameter. When enabled, the DDR memory controller core can be reconfigured with an updated set of the memory timing parameters in the parameter file without generating a new IP core. When disabled, the reconfiguration logic is permanently removed from the core. It is generally expected that the IP core performance will be improved due to a lower utilization.

sysCLOCK PLL

The sysCLOCK™ PLL block generates the clocks used in all blocks in the memory controller core. If an external clock generator is to be used, it is possible to remove this block from the IP core structure.

Data Path Logic

The DPL block interfaces with the DDR I/O modules and is responsible for generation of the read data and read data valid signal in the read operation mode. This block implements the logic to ensure that the data read from the memory is transferred to the local user interface in a deterministic and coherent manner. The write data does not go through the DPL block; it is directly transferred to the Command Application Logic (CAL) block for the write operation mode. The implementation of the DPL block is also device dependent.

Initialization State Machine

The Initialization State Machine (ISM) block performs the DDR memory initialization sequence defined by JEDEC. Although the memory initialization must be done after the power-up, it is the user's responsibility to provide a user input to the block to start the memory initialization sequence. The ISM block provides an output that indicates the completion of the sequence to the local user interface.

Command Application Logic

The CAL block accepts the decoded commands from the Command Decode Logic on two separate queues. These commands are translated to the memory commands in a way that meets the timing requirements of the memory device. The CI block provides the memory timing parameters to the CAL block so that the timing requirements are satisfied during the command translations. Commands in the two stage queues are pipelined to maximize the throughput on the memory interface. The CDL and the CAL blocks work in parallel to fill and empty the queues respectively.

DDR I/O Modules

The DDR I/O modules are directly connected to the memory interface providing all required DDR ports for memory access. They convert the single data rate (SDR) data to DDR data for write operations and perform the DDR to SDR conversion for read operations. The I/O modules utilize the dedicated DDR I/O logic and are designed to reliably drive and capture the data on the memory interface.

Signal Descriptions

Table 2-1 describes the user interface and memory interface signals at the top level.

Table 2-1. DDR SDRAM Memory Controller Top-Level I/O List

Port Name	Active State	I/O	Description
Local User Interface			
clk_in	N/A	Input	Reference clock. It is connected to the PLL input.
rst_n	Low	Input	Asynchronous reset. It resets the entire core when asserted.
init_start	High	Input	Initialization start. It should be asserted at least 200 µs after the power-on reset to initiate the memory initialization.
cmd[3:0]	N/A	Input	User command input to the memory controller.
cmd_valid	High	Input	Command and address valid input. When asserted, the addr and cmd inputs are validated.
addr[ADDR_WIDTH –1:0]	N/A	Input	User address input to the memory controller.
write_data[DSIZE –1:0]	N/A	Input	Write data input from user logic to the memory controller.
data_mask[(DSIZE/8) –1:0]	High	Input	Data mask input for write_data. Each bit masks the corresponding byte on the write_data bus, in order
ext_auto_ref	High	Input	User auto-refresh control input. This port is enabled when EXT_AUTO_REF is defined.

Table 2-1. DDR SDRAM Memory Controller Top-Level I/O List (Continued)

Port Name	Active State	I/O	Description
k_clk	N/A	Output	System clock output. The user logic uses this as a system clock unless an external clock generator is used.
init_done	High	Output	Initialization done output. It is asserted for one clock cycle when the core completes the memory initialization routine.
ext_auto_ref_ack	High	Output	User auto-refresh control acknowledge output. This port is enabled when EXT_AUTO_REF is defined.
cmd_rdy	High	Output	Command ready output. When asserted, it indicates the core is ready to accept the next command and address.
data_rdy	High	Output	Data ready output. When asserted, it indicates the core is ready to receive the write data.
read_data[DSIZE –1:0]	N/A	Output	Read data output from the memory to the user logic.
read_data_valid	High	Output	Read data valid output. When asserted, it indicates the data on the read_data bus is valid.
DDR SDRAM Memory Interface			
em_ddr_clk[CLKO_WIDTH –1:0]	N/A	Output	DDR memory clock generated by the memory controller.
em_ddr_cke[CKE_WIDTH –1:0]	High	Output	DDR memory clock enable generated by the memory controller.
em_ddr_addr[ROW_WIDTH –1:0]	N/A	Output	DDR memory address. It has the multiplexed row and column address for the memory.
em_ddr_ba[BNK_WDTH –1:0]	N/A	Output	DDR memory bank address.
em_ddr_data[DATA_WIDTH –1:0]	N/A	In/Out	DDR memory bi-directional data bus.
em_ddr_dm[(DATA_WIDTH/8) –1:0]	High	Output	DDR memory write data mask. It is used to mask the byte lanes for byte level write control.
em_ddr_dqs[(DQS_WIDTH –1:0]	N/A	In/Out	DDR memory bi-directional data strobe. This strobe signal is associated with either 4 or 8 data pads.
em_ddr_cs_n[CS_WIDTH –1:0]	Low	Output	DDR memory chip select.
em_ddr_cas_n	Low	Output	DDR memory column address strobe.
em_ddr_ras_n	Low	Output	DDR memory row address strobe.
em_ddr_we_n	Low	Output	DDR memory write enable.
em_ddr_odt[CS_WIDTH –1:0]	High	Output	DDR memory on-die termination control. This output is present only in the DDR2 mode.

Using the Local User Interface

The local user interface of the DDR memory controller IP core consists of four independent functional groups:

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

Each functional group and its associated local interface signals are listed in Table 2-2.

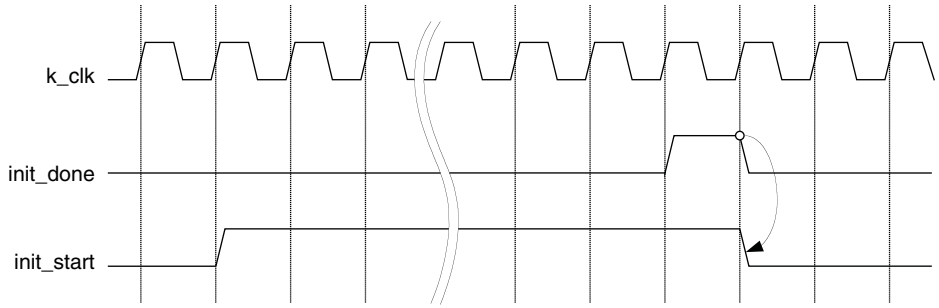
Table 2-2. Local User Interface Functional Groups

Functional Group	Signals
Initialization and Auto-Refresh Control	init_start, init_done, ext_auto_ref, ext_auto_ref_ack
Command and Address	addr, cmd, cmd_rdy, cmd_valid
Data Write	data_rdy, write_data, data_mask
Data Read	read_data, read_data_valid

Initialization and Auto-Refresh Control

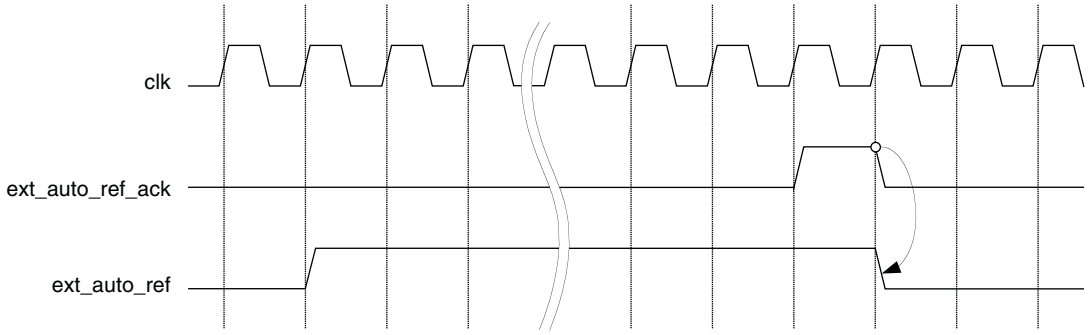
The DDR memory devices must be initialized before the memory controller can access them. The memory controller starts the memory initialization sequence when the init_start signal is asserted by the user interface. The user must wait at least 200 µs after the power-up cycle is completed and the system clock is stabilized, and then generate the initialization start input to the core. Once asserted, the init_start signal needs to be held high until the initialization process is completed. The init_done signal is asserted high for one clock cycle when the core has completed the initialization and training sequence and is now ready to access the memory. The init_start signal must be deasserted as soon as init_done is asserted. The memory initialization is required only once after the system reset. Note that the core will operate with the default memory configuration initialized in this process if the user does not program the MR and/or EMR registers. Figure 2-2 shows the timing diagram of the initialization control signals.

Figure 2-2. Timing of Memory Initialization Control



The memory controller core provides the user auto-refresh control feature. This feature can be enabled by the External Auto Refresh Port option. It is a useful function for applications that need to have a complete control on the DDR interface in order to avoid unwanted intervention caused by the memory refresh operations. Once enabled, ext_auto_ref is asserted by a user to force the core to generate a set of Refresh commands in a burst. The number of Refresh commands in a burst is defined by the Auto Refresh Burst Count option. The ext_auto_ref_ack signal is asserted high for one clock cycle to indicate that the core has generated the Refresh commands. The ext_auto_ref signal can be deasserted once the acknowledge signal is detected as shown in Figure 2-3.

Figure 2-3. Timing of External Auto Refresh Control



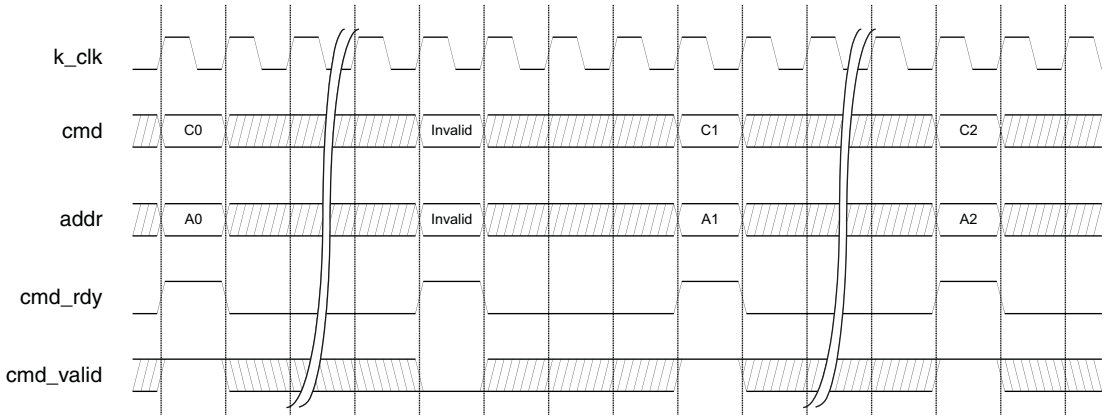
Command and Address

Once the memory initialization is done, the core waits for user commands that will 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 procedure described below:

1. The memory controller core tells the user logic that it is ready to receive a command by asserting the cmd_rdy signal for one clock cycle.
2. If the core finds the cmd_valid signal asserted by the user logic while it is asserting cmd_rdy, it takes the cmd input as a valid user command. The core also accepts the addr input as a valid start address or mode register programming data depending on the command type. If cmd_valid is not asserted, the cmd and addr inputs become invalid and the core ignores them.
3. The cmd, addr and cmd_valid inputs become “don’t care” while cmd_rdy is deasserted.
4. The cmd_rdy signal is asserted again to take the next command.

The timing of the command and address group is shown in Figure 2-4. The core will prevent cmd_rdy from being asserted when two queues in CDL are both occupied, if any queue in CDL is empty, the cmd_rdy will be asserted.

Figure 2-4. Timing of Command and Address



Each command on the cmd bus must be a valid command. Lattice defines the valid memory commands as shown in Table 2-3. All other values are reserved and considered invalid.

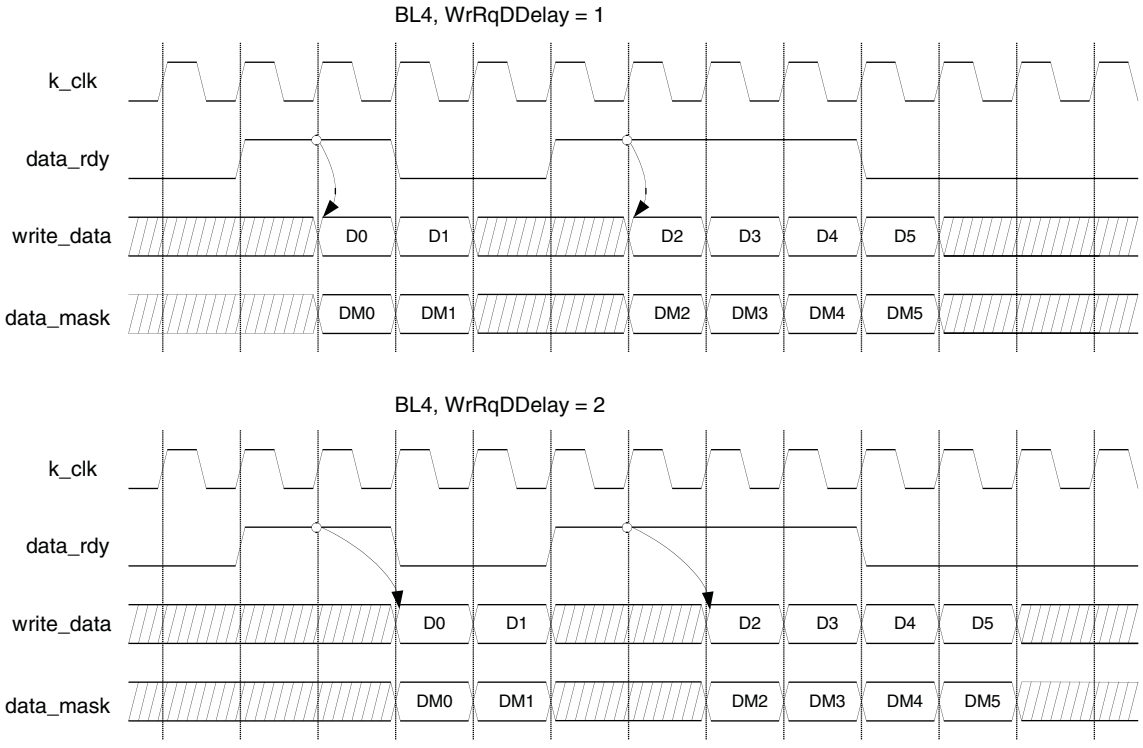
Table 2-3. Defined User Commands

Command	Mnemonic	cmd[3:0]
Read	READ	0001
Write	WRITE	0010
Read with Auto Precharge	READA	0011
Write with Auto Precharge	WRITEA	0100
Powerdown	PDOWN	0101
Load Mode Register	LOAD_MR	0110
Self Refresh	SELF_REF	0111

Data Write

After the WRITE command is accepted, the memory controller core asserts the data_rdy signal when it is ready to receive the write data from the user logic to be written into the memory. Since the duration from the time a write command is accepted to the time the data_rdy signal is asserted is not fixed, the user logic needs to monitor the data_rdy signal to detect when it is asserted. Once data_rdy is asserted, the core expects valid data on the write_data bus one or two clock cycles after the data_rdy signal is asserted. The write data delay is programmable by the user parameter, WrRqDDelay, providing flexible back-end application support. For example, setting WrRqD-Delay = 2 ensures that the core takes the write data out 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 the BL4 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.

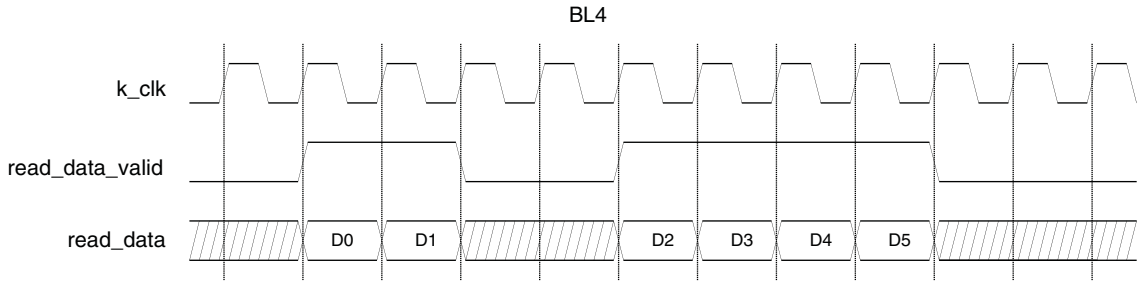
Figure 2-5. One-Clock vs. Two-Clock Write Data Delay



Data Read

When the READ command is accepted, the memory controller core accesses the memory to read the addressed data and brings it 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.

Figure 2-6. Read Data Timing on Local User Interface



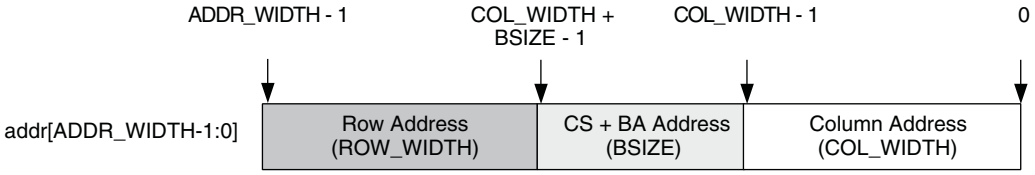
Read/Write with Auto Precharge

The DDR2 IP core automatically closes (precharges) and opens rows according to the user memory address accesses. Therefore, the READA and WRITEA commands are not used for most applications. The commands are provided to comply to the JEDEC DDR2 specification.

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. On 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 the memory device data sheet. Figure 2-7 shows the local-to-memory address mapping of the Lattice DDR memory controller cores.

Figure 2-7. Local-to-Memory Address Mapping for Memory Access

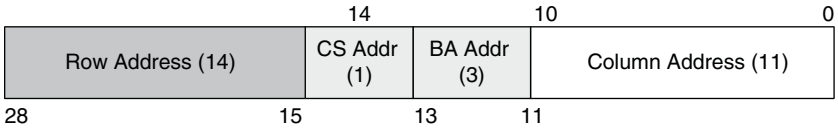


ADDR_WIDTH is calculated by the sum of COL_WIDTH, ROW_WIDTH and BSIZE. BSIZE is determined by the sum of the bank address size and chip select address size. For 4- or 8-Bank DDR2 devices, the bank address size is 2 or 3, respectively. When the number of chip select is 1, 2 or 4, the chip select address size becomes 0, 1, or 2, respectively. An example of the address mapping is shown in Table 2-4 and Figure 2-8.

Table 2-4. An Example of Address Mapping

User Selection Name	User Value	Parameter Name	Parameter Value	Actual Line Size	Local Address Map
Row Size	14	ROW_WIDTH	14	14	addr[28:15]
Column Size	11	COL_WIDTH	11	11	addr[10:0]
Bank Size	8	BNK_WDTH	3	3	addr[13:11]
Chip Select Width	2	CS_WIDTH	2	1	addr[14]
Total Local Address Line Size		ADDR_WIDTH	29	29	addr[28:0]

Figure 2-8. Mapped Address for the Example

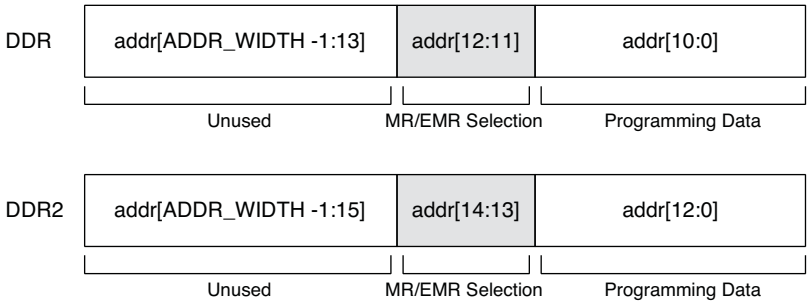


Mode Register Programming

The DDR SDRAM memory devices are programmed using the mode register (MR) and extended mode registers (EMR). The bank address bus (em_ddr_ba) is used for choosing one of the MR or EMR registers, while the programming data is delivered through the address bus (em_ddr_addr). The memory data bus cannot be used for the MR/EMR programming.

The Lattice DDR memory controller core uses the local address bus, addr, to program these registers. It uses different address mapping from the address mapping for memory accesses. The core accepts a user command, LOAD_MR, to initiate the programming of MR/EMR registers. When LOAD_MR is applied on the cmd bus, the user logic must provide the information for a target 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. Figure 2-9 shows how the local address lines are allocated for the programming of memory registers.

Figure 2-9. Local-to-Memory Address Mapping for MR/EMR Programming



The register programming data is provided through the lower side of the addr bus starting from the bit 0 for LSB. The programming data requires eleven (DDR mode) or thirteen (DDR2 mode) bits of the local address lines. Two more bits are needed to choose a target register as listed in Table 2-5. All other upper address lines are unused during the command patch cycle for the LOAD_MR command.

Table 2-5. Mode Register Selection Using Bank Address

Mode Register	Local Address	
	DDR (addr[12:11])	DDR2 (addr[14:13])
MR	00	00
EMR	01	01
EMR2[1]	—	10
EMR3[1]	—	11

1. DDR2 mode only

Figure 2-10 shows the use of local address for typical DDR2 memory configurations. The DDR memory configuration is accomplished the same way, except that it accesses only two registers, MR and EMR. Starting from DDR/DDR2 version 6.7, some of the registers such as Burst Type, Burst Length, CAS Latency and others can be configured directly from the IPexpress™ GUI for custom initialization.

The initialization default values for all mode registers are listed in Table 2-6.

Table 2-6. Initialization Default Values for DDR/DDR2 Mode Registers

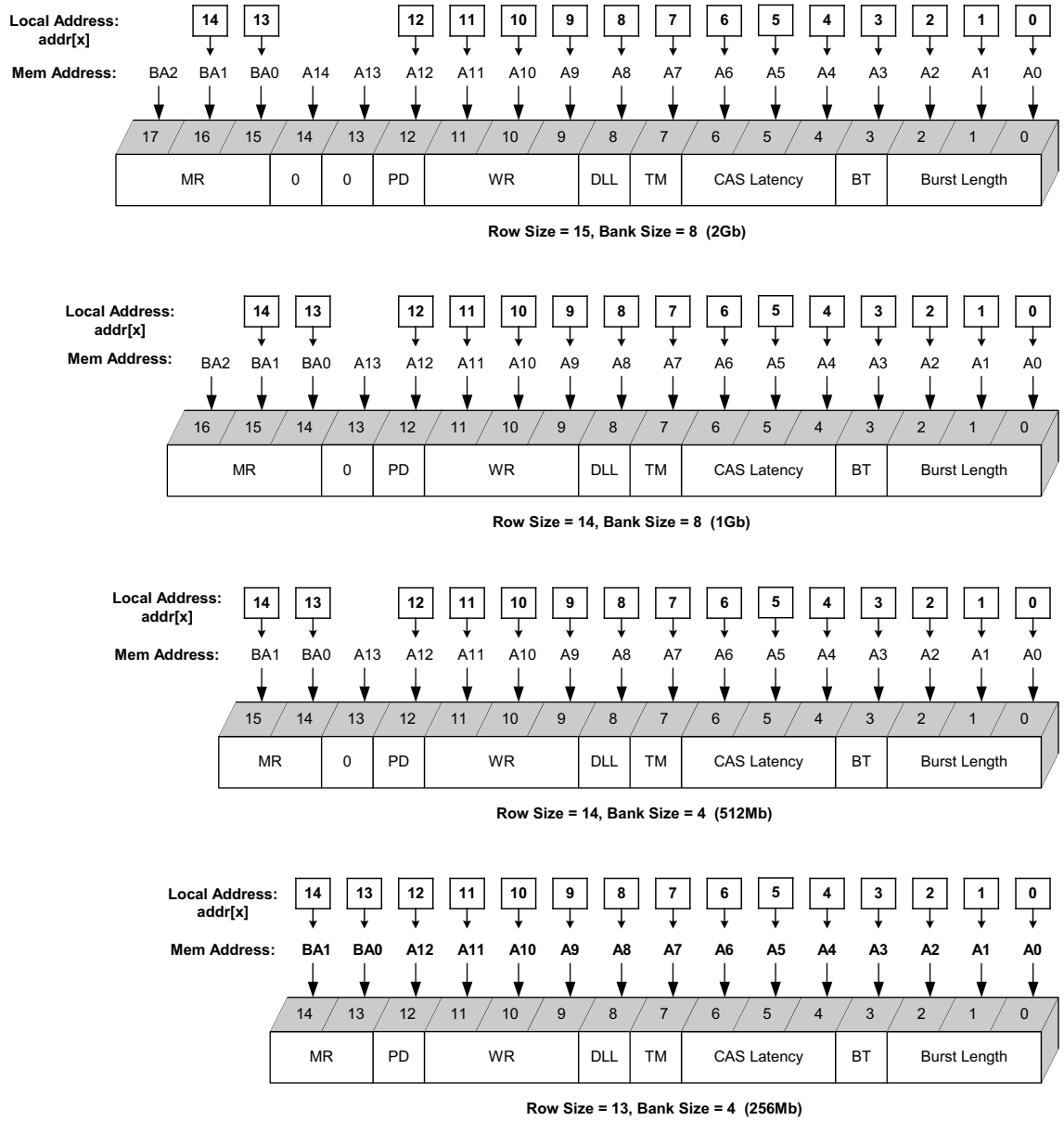
Type	Registers	Value	Description	Local address
DDR MR (BA[1:0] = 00)	Burst Length[1]	3'b010	BL = 4	addr[2:0]
	Burst Type[1]	1'b0	Sequential	addr[3]
	Cas Latency[1]	3'b010	CL = 2 Cycles	addr[6:4]
	Test Mode	1'b0	Normal	addr[7]
	DLL Reset	1'b1	DLL Reset = Yes	addr[8]
	All Others	0		addr[ROW_WIDTH-1:8]
DDR EMR (BA[1:0] = 01)	DLL	1'b0	DLL Enable	addr[0]
	Drive Strength	1'b0	Normal	addr[1]
	All Others	0		addr[ROW_WIDTH-1:2]
DDR2 MR (BA[1:0] = 00 or BA[2:0]=000)	Burst Length[1]	3'b010	BL = 4	addr[2:0]
	Burst Type[1]	1'b0	Sequential	addr[3]
	Cas Latency[1]	3'b100	CL = 4 Cycles	addr[6:4]
	Test Mode	1'b0	Normal	addr[7]
	DLL Reset	1'b1	DLL Reset = Yes	addr[8]
	WR Recovery[1]	3'b010	3 Cycles	addr[11:9]
	Power Down Exit[1]	1'b0	Fast	addr[12]
	All Others	0		addr[ROW_WIDTH-1:13]
DDR2 EMR (BA[1:0] = 01 or BA[2:0]=001)	DLL	1'b0	DLL Enable	addr[0]
	Drive Strength	1'b0	Normal	addr[1]
	RTT0[1]	1'b0	Disabled with RTT1=0	addr[2]
	Additive Latency[1]	3'b011	3 Cycles	addr[5:3]
	RTT1[1]	1'b0	Disabled with RTT0=0	addr[6]
	OCD	3'b000	OCD Not Applicable	addr[9:7]
	DQS Mode	1'b1	Differential Disabled	addr[10]
	RDQS	1'b0	Disable	addr[11]
	Outputs	1'b0	Enable	addr[12]
	All Others			addr[ROW_WIDTH-1:13]
DDR2 EMR2 (BA[1:0] = 10 or BA[2:0]=010)	All	0		addr[ROW_WIDTH-1:0]

Table 2-6. Initialization Default Values for DDR/DDR2 Mode Registers (Continued)

Type	Registers	Value	Description	Local address
DDR2 EMR3 (BA[1:0] = 11 or BA[2:0]=011)	All	0		addr[ROW_WIDTH-1:0]

1. This register can be initialized with a custom value through the IPexpress GUI.

Figure 2-10. Local Address Mapping for MR Programming (Typical DDR2 Memory Configurations)



Memory Interface

Table 2-7 lists the connections of the DDR interface between the Lattice DDR memory controller core and memory.

Table 2-7. DDR Interface Signal Connections to DDR Memory

Core Port Name	Memory Port Name[1]	Width	VREF[2]		I/O Type	
			DDR	DDR2	DDR	DDR2
em_ddr_clk[3]	CK, CK#	CLKO_WIDTH	—	—	SSTL25D_I	SSTL18D_I
em_ddr_cke	CKE	CKE_WIDTH	—	—	SSTL25_I	SSTL18_I
em_ddr_odt[4]	ODT	CS_WIDTH	—	—	N/A	SSTL18_I
em_ddr_cs_n	CS#	CS_WIDTH	—	—	SSTL25_I	SSTL18_I
em_ddr_ras_n	RAS#	1	—	—	SSTL25_I	SSTL18_I
em_ddr_cas_n	CAS#	1	—	—	SSTL25_I	SSTL18_I
em_ddr_we_n	WE#	1	—	—	SSTL25_I	SSTL18_I
em_ddr_addr	A	ROW_WIDTH	—	—	SSTL25_I	SSTL18_I
em_ddr_ba	BA	BNK_WDTH	—	—	SSTL25_I	SSTL18_I
em_ddr_data	DQ	DATA_WIDTH	1.25 V	0.9 V	SSTL25_I	SSTL18_I
em_ddr_dm	DM	DATA_WIDTH/8	—	—	SSTL25_I	SSTL18_I
em_ddr_dqs	DQS	DQS_WIDTH	1.25 V	0.9 V or none[5]	SSTL25_I	SSTL18_I or SSTL18D_I[6]

1. The listed DDR memory port names are from the Micron DDR memory data sheet.
2. In the banks with multiple VREFs, only VREF1 is used for DDR memory applications. VREF = VCCIO/2.
3. Lattice DDR memory controller core defines only the positive-end signal for the memory clock. The negative-end pad is allocated by the implementation software when a differential I/O type is assigned.
4. The ODT ports are available only in the DDR2 mode.
5. If DQS uses a differential pair, VREF is not required. However, VREF1 is still used for the DQS preamble detection.
6. In the DDR2 mode, either single-ended or differential type of DQS can be selected.

Parameter Settings

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

Table 3-1 provides the list of user configurable parameters for the DDR/DDR2 IP core. The parameter settings are specified using the DDR/DDR2 IP core Configuration GUI in IPexpress. The numerous IPexpress parameter options are partitioned across multiple GUI tabs as shown in this chapter.

Table 3-1. DDR SDRAM Memory Controller Parameters

Parameters	Range/Options	Default Value
Type		
Select Memory - DDR2	Micron DDR2 512 Mb -5E Custom	Custom
Select Memory - DDR	Micron DDR 512 Mb -5E Micron DDR 512 Mb -6E Micron DDR 512 Mb -75E Custom	Micron DDR 512Mb -75E
Clock	DDR - 133-200 DDR2 - 166 -333	133.333
Memory Data Bus size	16	16
Configuration	x8, x16	x8
Data_rdy to Write Data Delay	1, 2	1
Clock Width	1, 2	1
CKE Width	1, 2	1
Fixed Memory Timing	Disable, Enable	Disable
Use Differential DQS[1]	Disable, Enable	Enable
Setting		
Address		
Row Size	13 - 16	13
Column Size	9 - 11	10
Bank Size	4, 8[2]	4
Chip Select width	1, 2, 4	1
User Slot Size	1, 2	1
EMR Prog During Init[3]	Disable, Enable	Enable
Auto Refresh Control		
Auto Refresh Burst Count	2-8	8
External Auto Refresh Port	Disable, Enable	Disable
Mode Register Initial Setting - DDR2		
Burst Length	4, 8	4
CAS Latency	2 - 6	4
Additive Latency	0 - 4	3
Write Recovery	2 - 6	3
RTT_Nom (Ohm)	50 Ohm, 75 Ohm, 150 Ohm, Disable	Disable
Burst Type	Sequential, Interleaved	Sequential
DLL Control for PD	Fast End, Slow Exit	Fast Exit
Differential DQS	Enable, Disable	Enable

Table 3-1. DDR SDRAM Memory Controller Parameters (Continued)

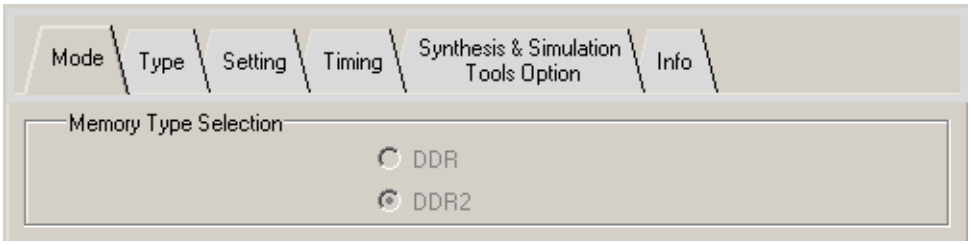
Parameters	Range/Options	Default Value
Mode Register Initial Setting - DDR		
Burst Length	2, 4, 8	4
CAS Latency	2, 3	2
Burst Type	Sequential, Interleaved	Sequential
Timing		
TRCD	1 - 7	3
TRAS	1 - 31	8
TRFC	1 - 63	DDR: 14 DDR2: 21
TMRD	1 - 7	2
TRP	1 - 7	3
TRRD	1 - 7	2
TRC	1 - 31	11
TREFI	1 - 65536	1563
TWTR	1 - 7	2
TRTP	1 - 4	2
Synthesis & Simulation Tools Option		
Support Synplify, Support ModelSim, Support ALDEC	Enable, Disable	Enable

1. DDR2 only.
2. DDR has 4 only.
3. The EXT_REG_EN parameter is effective only in the DDR mode.

Mode Tab

The Memory Type Selection field is not a user option but is selected by IPexpress when a DDR memory controller core is selected from the IPexpress IP core list. Figure 3-1 shows the contents of the Mode tab.

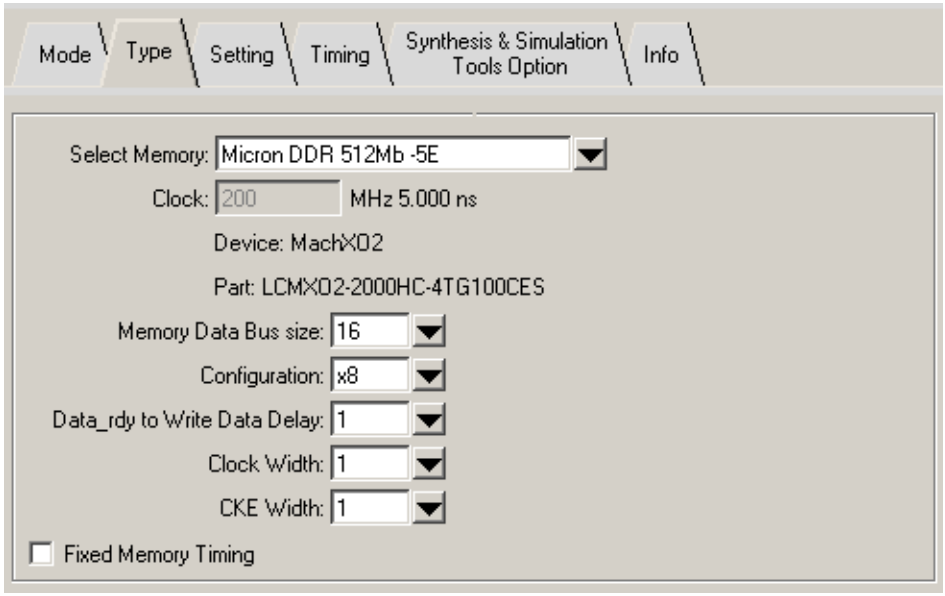
Figure 3-1. Mode Tab



Type Tab

The Type tab enables users to select various configuration options for the target memory device/ module and the core functional features. Figure 3-2 shows the contents of the Type tab.

Figure 3-2. Type Tab



Select Memory

A predefined DDR memory device is selected by default. The timing parameters for the selected memory are listed in the Timing tab. One or more different speed grade memory devices are also available for easy selection. If the desired memory device requires different timing parameters from the pre-defined ones, the Custom option should be selected.

Clock

When a predefined memory is selected, the Clock field displays the corresponding clock speed, and it cannot be modified. With the Custom option selected, a user target speed must be provided.

Memory Data Bus Size

This option means the memory data bus width to which the memory controller core is connected. If a memory module that has a wider data bus than required is to be used, only the required data width has to be selected.

Configuration

This option is used to select the device configuration of the DIMM. Device configurations x4, x8, and x16 are supported.

Data_rdy to Write Data Delay

This option is selected according to the user local back-end application's requirement. The user logic can send the write data to the core with either one-clock cycle or two-clock cycle delay.

Clock Width

This option sets the number of clocks with which the memory controller drives the memory. The IPexpress tool can generate either one or two memory clocks. Once a DDR memory controller core is generated, more memory clocks can be manually instantiated for those applications that need more than two memory clocks.

CKE Width

The number of memory clock enable signals is configured using this option. More clock enable signals can also be instantiated by the user.

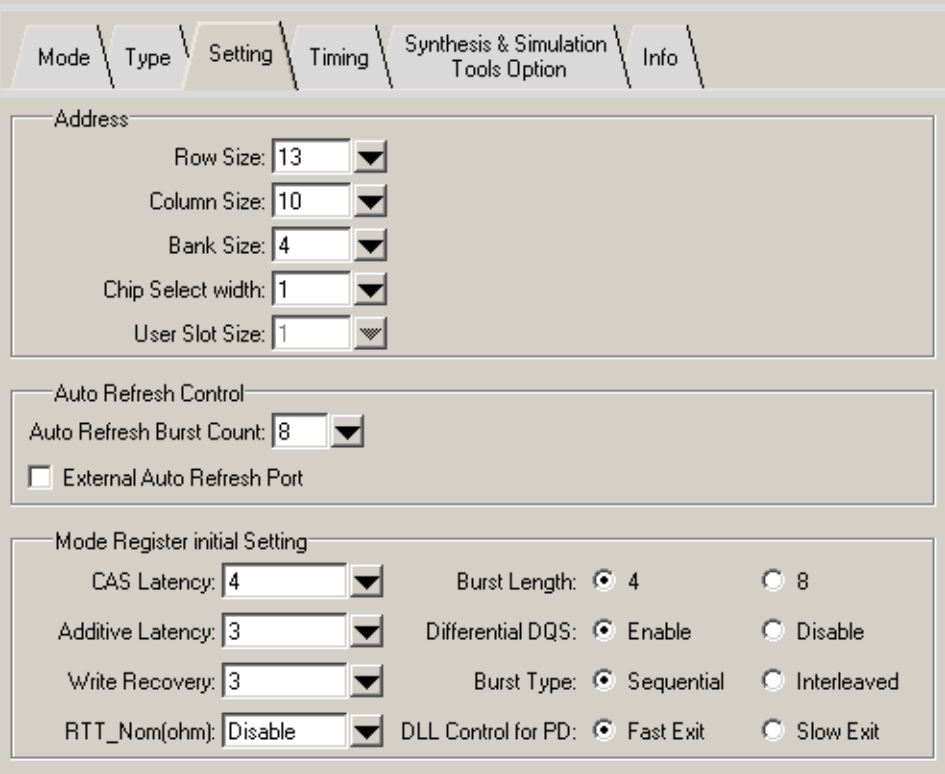
Fixed Memory Timing

This option disables the memory timing reconfiguration feature for a generated core. When disabled, the IP core only supports the timing parameter set applied at the time of the core generation. This option may provide somewhat improved performance with lower resource utilization by removing the reconfiguration logic from the core. This option should not be selected if it is necessary to support different memory timing parameters without regenerating the core. This option is unchecked by default.

Setting Tab

The target memory size and addressing scheme are determined in the Setting tab. The memory initialization and auto-refresh configurations are also covered in this tab. Figure 3-3 shows the contents of the Setting tab.

Figure 3-3. Setting Tab



Row Size

This option indicates the row address size for the memory ranging from 13 to 16, which is found in the memory data sheet.

Column Size

This option indicates the column address size for the memory ranging from 9 to 11, which is found in the memory data sheet.

Bank Size

This option indicates the bank address size for the memory. Either 4 or 8 is selected depending on the size and type of the memory to be used with the core.

Chip Select Width

The Lattice memory controller cores follow the JEDEC specifications for DDR/DDR2 SDRAM memories supporting two different sets of chip select signaling. The DDR mode provides up to eight chip selects supporting up to four memory modules. The DDR2 mode provides up to four chip selects supporting up to two DDR2 memory modules. Note that this option does not indicate the number of memory modules but the number of actual chip select signals.

User Slot Size

This option indicates the number of physical DIMM or SODIMM slots. The information of user slot number is used for the memory controller core to properly drive the ODT (On-Die Termination) signals to the DDR2 memory modules. Since DDR memory does not have ODT, this option is available only to the DDR2 mode with a choice of one or two slots.

EMR Prog During Init

Once disabled, the core does not program the EMR during the initialization; it can then be programmed after the initialization process is done. This option is required only in the DDR mode because the core must program the EMR during the initialization in the DDR2 mode. The default value for this option is Program. Keep the default for the DDR2 mode core configurations.

Auto Refresh Burst Count

This option indicates the number of Refresh commands that the memory controller core generates at once. DDR memories have at least an 8-deep Refresh command queue following the JEDEC specification and Lattice DDR memory controller cores support up to eight Refresh commands in one burst. It is recommended that the maximum number be used if the DDR interface throughput is a major concern of the system. If it is set to 8, for example, the core will send a set of eight consecutive Refresh commands to the memory at once when it reaches the time period of the eight refresh intervals (t_{REFI} x 8). Bursting refresh cycles increases the DDR bus throughput because it helps keep core intervention to a minimum.

External Auto Refresh Port

This option provides users with the capability of controlling the memory refresh command generation. If this option is disabled, the core takes control of the Refresh command generation according to the memory timing parameter, TREFI. Once enabled, the core adds the external auto refresh control ports to the local user interface with which users can take full control of the Refresh command generation.

Mode Register Initial Setting

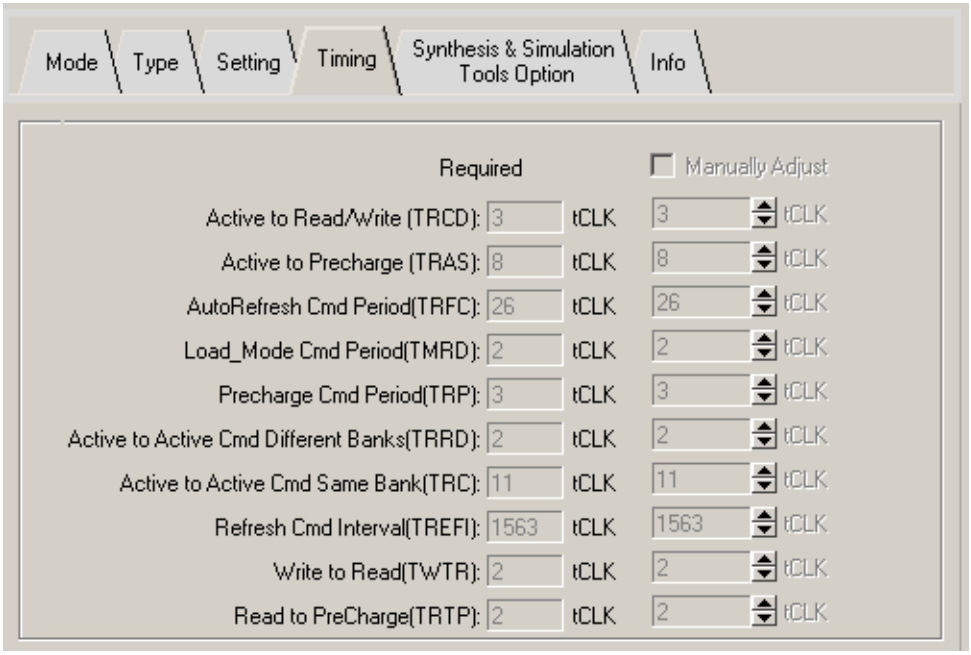
This option allows the user to program the mode registers during the core initialization process. Not all mode register bits are initialized from this option. Only the mode register configuration bits that are used for normal DDR operations are programmed using this setting. See Table 3-1 for the list of the covered mode register settings. The user does not need to program the mode registers after the core initialization is finished if the mode register is properly configured as desired.

Timing Tab

Lattice DDR memory controller core allows users to customize the memory timing parameters. This can be done when the Custom memory type is selected in the Type tab of the IPexpress GUI. The Manually Adjust box in the Timing tab must also be checked to adjust the parameters. Two timing parameters, TWTR and TRTP, are for the DDR2 mode only while all others are common to both modes. The numbers in the parameter boxes are decimal values indicating the number of clock cycles (t_{CLK}). Since the timing numbers available in the memory vendors' data sheets usually are actual time based, conversions from time numbers to clock numbers should be properly made. The conversion is easily made by dividing the time number by the clock period. When a timing parameter is

found to be a minimum value in the data sheet, the calculated number, if not a whole integer, should be the next whole integer to be safe. If it is a maximum value, then only the whole part is taken, and the decimal part is discarded. Figure 3-4 shows the contents of the Timing tab.

Figure 3-4. Timing Tab



The memory timing parameters are listed in Table 3-2.

Note: There is a timing parameter that is not shown in the Timing tab. The TCKP parameter is not a memory timing parameter but a memory controller core parameter used only in the DDR2 mode. It provides the wait cycles during the DDR2 memory initialization. The DDR2 specification requires a minimum of 400 ns wait before the PRECHARGE ALL command is executed. This parameter is found in the core parameter file with the default number `d107, which ensures 400 ns of wait up to 266 MHz speed. Although the wait time can be increased or decreased by adjusting the TCKP parameter, it may not be necessary to modify this parameter in most applications.

Table 3-2. Memory Timing Parameters for DDR Memory Controller

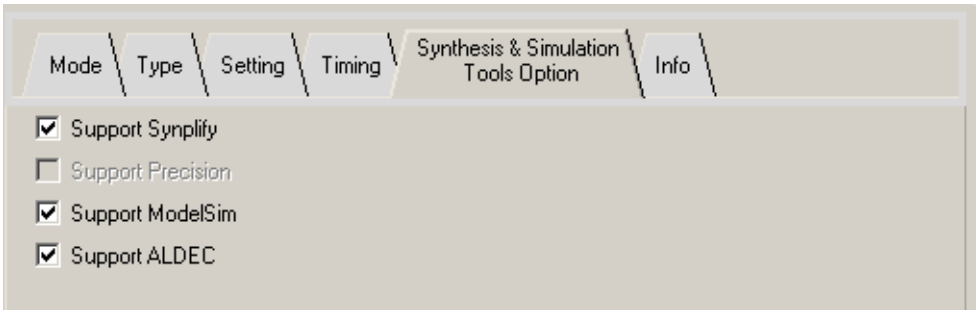
Signal Name	Description
tras[4:0]	ACTIVE to PRECHARGE command delay in clock cycles
trc[4:0]	ACTIVE to ACTIVE/AUTO REFRESH delay in clock cycles
trcd[2:0]	ACTIVE to READ/WRITE delay in clock cycles
trrd[2:0]	ACTIVE bank A to ACTIVE bank B delay in clock cycles
trfc[5:0]	REFRESH command period in clock cycles
trp[2:0]	PRECHARGE command period in clock cycles
tmrd[2:0]	LOAD MODE REGISTER command period in clock cycles
trefi[15:0]	Refresh Interval in clock cycles
trtp[1:0]	READ to PRECHARGE delay, DDR2 mode only
twtr[2:0]	WRITE to READ delay, DDR2 mode only
tckp[6:0][1]	Wait before PRECHARGE ALL during initialization, DDR2 mode only

1. Not available in the IPexpress GUI.

Synthesis & Simulation Tools Option Tab

The Lattice DDR memory controller cores support multiple synthesis and simulation tool flows. This tab allows users to deselect the unwanted flow supports. Figure 3-5 shows the contents of the Synthesis & Simulation Tools Option tab.

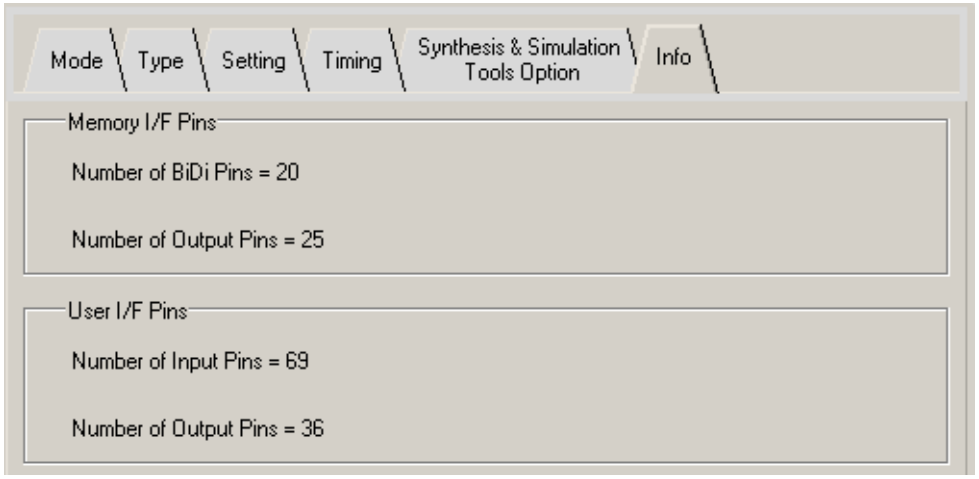
Figure 3-5. Synthesis & Simulation Tools Option Tab



Info Tab

The number of pins required on the DDR bus and the local user interface are reported in the Info tab. Figure 3-6 shows the contents of the Info tab.

Figure 3-6. Info Tab



Memory I/F Pins

The numbers displayed indicate the total required number of DDR bus I/O pads.

User I/F Pins

The numbers displayed indicate the total required number of local user interface signals. Although these signals usually do not use I/O pads in user applications, this information can indicate whether or not the evaluation project will insert the dummy logic. Note that all local user interface signals also use I/O pads in the core evaluation project.

IP Core Generation

This chapter provides information on licensing the DDR/DDR2 IP core, generating the core using the Diamond or ispLEVER software IPexpress tool, running functional simulation, and including the core in a top-level design. The Lattice DDR/DDR2 IP core can be used in MachXO2 PLDs.

Licensing the IP Core

An IP license is required to enable full, unrestricted use of the DDR/DDR2 IP core in a complete, top-level design. An IP license that specifies the IP core (DDR/DDR2) and device family (MachXO2) is required to enable full use of the DDR/DDR2 IP core in MachXO2 devices Instructions on how to obtain licenses for Lattice IP cores are given at:

http://www.latticesemi.com/products/intellectualproperty/aboutip/isplevercoreonlinepurchas.cfm

Users may download and generate the DDR/DDR2 IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The DDR/DDR2 IP core also 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 time (approximately four hours) without requiring an IP license. See the Hardware Evaluation section for further details. However, a license is required to enable timing simulation, to open the design in the Diamond or ispLEVER EPIC tool, and to generate bitstreams that do not include the hardware evaluation timeout limitation.

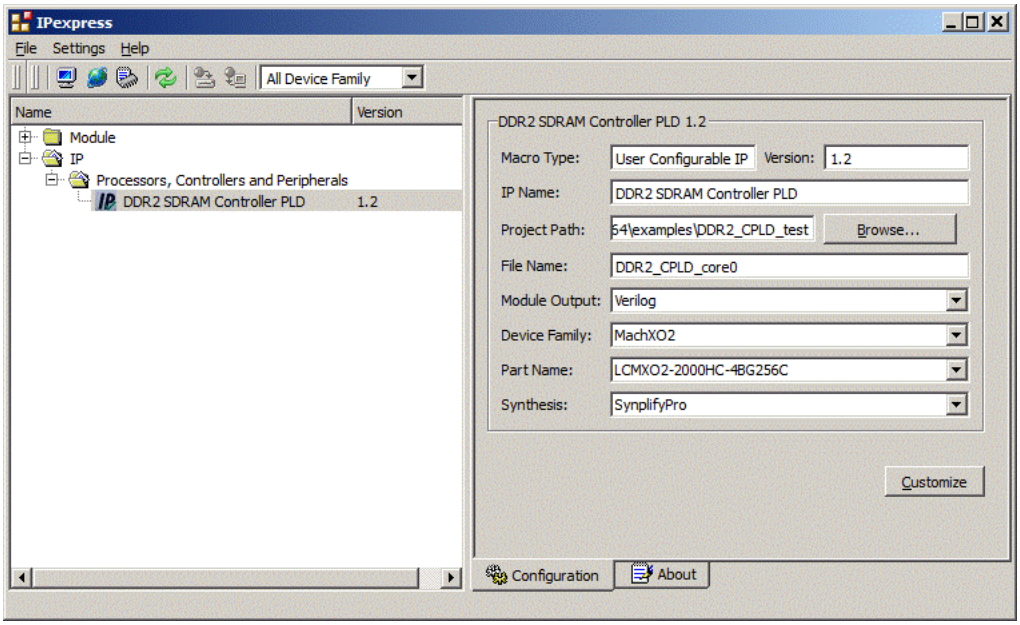
Getting Started

The DDR/DDR2 IP core is available for download from the Lattice IP Server using the IPexpress tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP core has been installed, the IP core will be available in the IPexpress GUI dialog box shown in Figure 4-1.

The IPexpress tool GUI dialog box for the DDR/DDR2 SDRAM IP core is shown in Figure 4-1. To generate a specific IP core configuration the user specifies:

- **Project Path** – Path to the directory where the generated IP files will be loaded.
- **File Name** – "username" designation given to the generated IP core and corresponding folders and files.
- **(Diamond) Module Output** – Verilog or VHDL.
- **(ispLEVER) Design Entry Type** – Verilog HDL or VHDL
- **Device Family** – Device family to which IP is to be targeted (e.g. LatticeSCM, Lattice ECP2M, LatticeECP3, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

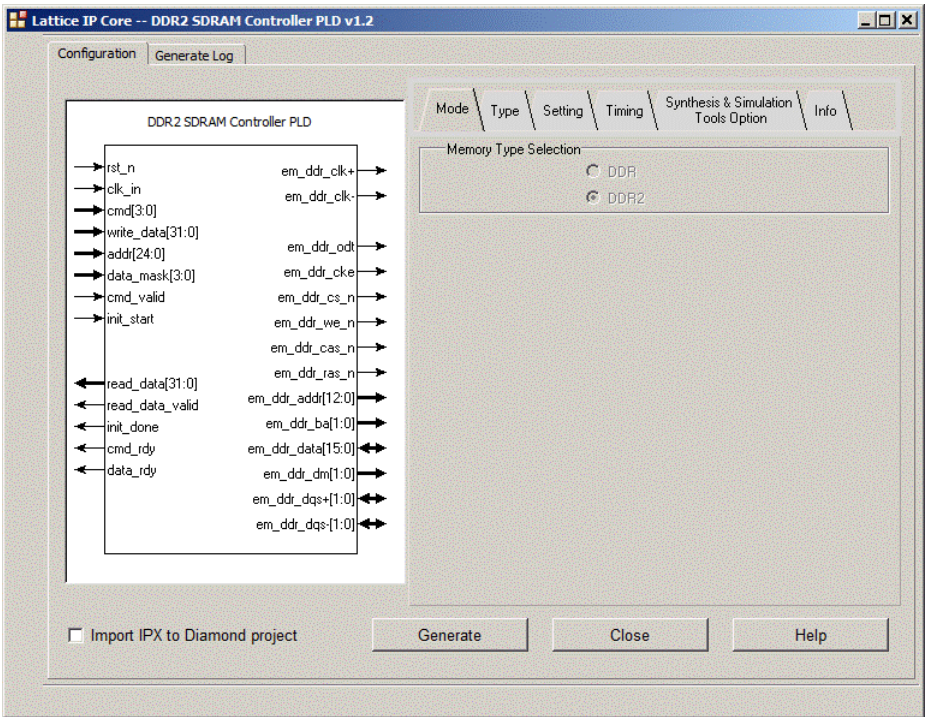
Figure 4-1. IPexpress Tool Dialog Box (Diamond Version)



Note that if the IPexpress tool is called from within an existing project, Project Path, Design Entry, Device Family and Part Name default to the specified project parameters. Refer to the IPexpress tool online help for further information.

To create a custom configuration, the user clicks the **Customize** button in the IPexpress tool dialog box to display the DDR/DDR2 SDRAM IP core Configuration GUI, as shown in Figure 4-2. From this dialog box, the user can select the IP parameter options specific to their application. Refer to Parameter Settings for more information on the DDR/DDR2 parameter settings.

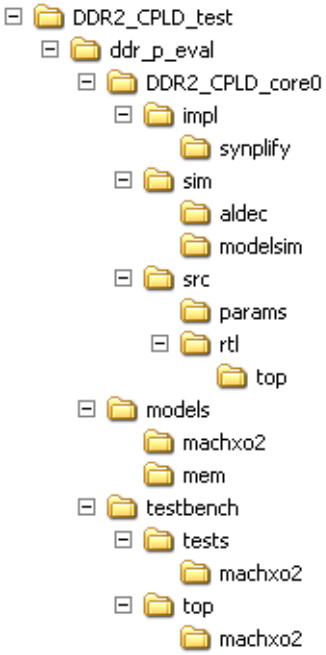
Figure 4-2. DDR/DDR2 SDRAM IP Configuration GUI (Diamond Version)



IPexpress-Created Files and Top Level Directory Structure

When the user clicks 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 4-3.

Figure 4-3. MachXO2 DDR/DDR2 Core Directory Structure



Generated Files

This section describes the structure of the DDR/DDR2 memory controller core that is generated by IPexpress as per user configuration. It also explains how the generated files are used in the structure. Understanding the core structure is an important step of a system design using the core. The summary of the files of the core for simulation are listed in Table 4-1.

Table 4-1. Files for Simulation and Implementation

File	Location	Modules	S[1]	P[2]
Top-level wrapper	.\ddr_p_eval\[core_name]\src\rtl\top\[device]\	ddr_sdram_mem_top ddr_sdram_mem_top_wrapper		X
Top-level wrapper	.\ddr_p_eval\[core_name]\sim	ddr_sdram_mem_top ddr_sdram_mem_top_wrapper	X	
Encrypted netlist	.\	[core_name].ngo		X
Core header[3]	.\	[core_name]_bb.v		X
I/O modules	.\ddr_p_eval\models\[device]\	ddr1_mem_io_top/ddr2_mem_io_top and its sub modules	X	X
Clock generator	.\ddr_p_eval\models\[device]\	clk_pll	X	X
Parameter file	.\ddr_p_eval\[core_name]\src\params\	ddr_sdram_mem_params	X	X
Preference files[4]	.\ddr_p_eval\[core_name]\impl\[synthesis]\	[core_name]_eval.lpf post_route_trace.prf		X
Evaluation project (GUI)[4]	.\ddr_p_eval\[core_name]\impl\[synthesis]\	[core_name]_eval.syn		X
Testbench top	.\ddr_p_eval\testbench\top\[device]\	test_mem_ctrl	X	
Obfuscated core simulation model	.\	[core_name]_beh	X	

Table 4-1. Files for Simulation and Implementation (Continued)

File	Location	Modules	S[1]	P[2]
Stimulus generator	.\ddr_p_eval\testbench\tests\[device]\	cmd_gen, test_case	X	
Monitor	.\ddr_p_eval\testbench\top\[device]\	monitor, odt_watchdog (DDR2 only)	X	
TB configuration parameter	.\ddr_p_eval\testbench\tests\[device]\	tb_config_params	X	
Memory model	.\ddr_p_eval\models\mem\	ddr1 or ddr2, (plus width configuration modules)	X	
Memory model parameter	.\ddr_p_eval\models\mem\	ddr1_parameters.vh or ddr2_parameters.vh	X	
Evaluation script[4]	.\ddr_p_eval\[core_name]\sim\modelsim\ .\ddr_p_eval\[core_name]\sim\aldec\	[core_name]_eval.do	X	
Simulation script[4]	.\ddr_p_eval\[core_name]\sim\modelsim\ .\ddr_p_eval\[core_name]\sim\aldec\	[core_name]_eval_timing_synplify.do	X	

1. S = Simulation.
2. P = Synthesis/Place and Route.
3. Not required for the VHDL flow.
4. Files are generated according to the Synthesis & Simulation Tools Option tab selection. See the Synthesis & Simulation Tools Option Tab section.

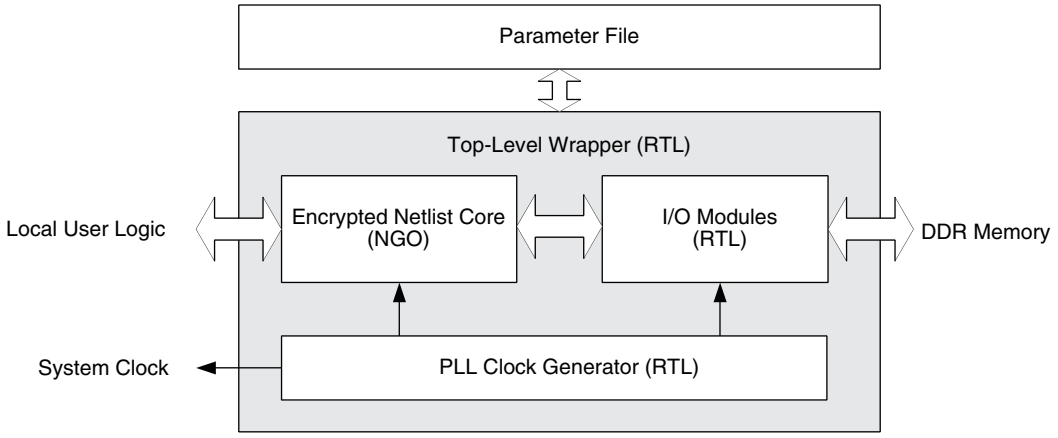
DDR Memory Controller Core Structure

The DDR memory controller core consists of the following five major functional blocks:

- Top-level wrapper (RTL)
- Encrypted memory controller block (NGO)
- I/O module block (RTL)
- Clock generator (RTL)
- Parameter file

All of these blocks are required to implement the core on the target device. Figure 4-4 shows the interconnection among those blocks.

Figure 4-4. Structure of DDR Memory Controller Core



Top-level Wrapper

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

Encrypted Netlist

The encrypted netlist contains the memory controller function that interfaces with the local user logic and the I/O modules that communicate with the DDR memory. The encrypted netlist must be located in the implementation project directory. IPexpress may generate another netlist for a PMI function when the core is generated. If this is the case, the PMI netlist must also be present along with the core netlist. The name of the PMI netlist is determined by IPexpress with the "pmi_xx..xx.ngo" form.

I/O Modules

The I/O module block provides device dependant DDR I/O functions. This block consist of one I/O module top file and several sub-modules that handle the DDR data (DQ), data mask (DM) and data strobe (DQS) signals. Note that the I/O modules are integrated into an NGO block when the core is generated for the VHDL flow. The simulation will continue to use the Verilog RTL modules to model the I/O Modules block behavior.

Clock Generator

The DDR memory controller core is designed to provide the system clock from the inside of the core. The clock output (k_clk) from the clock generator is used to drive the whole core logic as well as the external user logic. If a system that uses the DDR memory controller core is required to have a clock generator that is external to the core, the incorporated clock generator block can be removed from the core. The 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 following the system's need.

Parameter File

The IPexpress tool generates the parameter file based on the selected user options. The parameter file parameterizes the top-level wrapper and I/O modules. Note that the encrypted netlist (.ngo) file is created using the generated parameter file but is not a parameterized module. Therefore, the parameter definitions must not be altered. Otherwise, there will be connection problems between the netlist and other parameterized RTL modules.

Core Header File

The encrypted netlist is regarded as a black box during synthesis in the Verilog design environment. The header file that represents the netlist module must be included to bind the netlist to the wrapper in the Verilog flow. This file has a suffix "_bb" following the core name and is required only for the synthesis process.

Preference Files

The generated core contains preference files for the Synplify synthesis flow. The set contains two preference files. The implementation preference file ([core_name]_eval.lpf) contains a complete set of timing and physical preferences to force the implementation software to get a better performance margin. The trace preference file (post_route_trace.prf) is used to validate the timing results after the implementation is completed.

Refer to the Core Implementation section for more information about understanding preferences, preference localization, VREF assignments, DLL allocation, I/O types for DDR, skew treatment, data valid generation, dummy logic removal, read data auto-alignment logic, PCB routing delay compensation, and DQS_PIO_READ locate constraints.

Evaluation Project Files

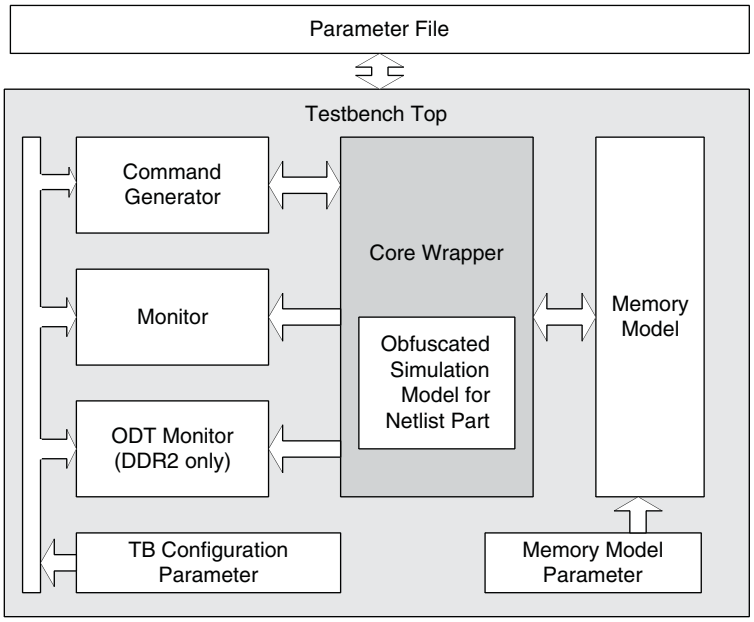
Several project files for implementation of the IP core are included for instant evaluation of the implementation result. A project file for Project Navigator is provided for the GUI-based flow, while a synthesis command script and a Place and Route (PAR) command script are included for the evaluation with the command-line flow. All required

files for synthesis and PAR processes are imported into the project files. These project files can be used as a starting point of a user application design.

Simulation Files for Core Evaluation

Once a DDR memory controller core is generated, it contains a complete set of testbench files that can be used to simulate some core activities for evaluation. This simulation structure for the DDR memory controller core is shown in Figure 4-5. This structure can be reused by system designers to accelerate their system validation. When a DDR memory controller core is simulated in VHDL, the core wrapper is provided in VHDL while other parts of the simulation structure are still in Verilog. Therefore, a simulation tool that has the mixed language capability such as the full version of ModelSim or an Aldec HDL simulator is required.

Figure 4-5. Simulation Structure for DDR 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 Core Simulation Model

The simulation model for the netlist part of the core is provided in the form of obfuscated RTL. This core model represents the functionality of the encrypted netlist and must be included in the simulation that contains the memory controller core.

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 blocks monitor both the local user interface and DDR interface activities and generate 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.

TB Configuration Parameter

The TB 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 DDR memory controller core contains a bus functional memory simulation model provided by one of the most popular memory vendors. If a different memory model is required, it can be done 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 bus functional 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

The functional and timing simulation macro script files are included for instant evaluation of the core. All required files for simulation are included in the macro script. These simulation scripts can be used as a starting point of a user simulation project. The generated scripts are based on the selection in the Synthesis & Simulation Tool Option tab (see the Synthesis & Simulation Tools Option Tab section).

Hardware Evaluation

The DDR/DDR2 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.

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.

Enabling Hardware Evaluation in ispLEVER

In the Processes for Current Source pane, right-click the **Build Database** process and choose **Properties** from the dropdown menu. The hardware evaluation capability may be enabled/disabled in the Properties dialog box. It is enabled by default.

Updating/Regenerating the IP Core

By regenerating an IP core with the IPexpress tool, you can modify any of its settings including: device type, design entry method, and any of the options specific to the IP core. Regenerating can be done to modify an existing IP core or to create a new but similar one.

Regenerating an IP Core in Diamond

To regenerate an IP core in Diamond:

1. In IPexpress, click the **Regenerate** button.
2. In the Regenerate view of IPexpress, choose the IPX source file of the module or IP you wish to regenerate.
3. IPexpress shows the current settings for the module or IP in the Source box. Make your new settings in the T**arget** box.
4. If you want to generate a new set of files in a new location, set the new location in the **IPX Target File** box. The base of the file name will be the base of all the new file names. The IPX Target File must end with an .ipx extension.

5. Click **Regenerate.** The module's dialog box opens showing the current option settings.
6. In the dialog box, choose the desired options. To get information about the options, click **Help**. Also, check the About tab in IPexpress for links to technical notes and user guides. 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.
7. To import the module into your project, if it's not already there, select **Import IPX to Diamond Project** (not available in stand-alone mode).
8. Click **Generate**.
9. Check the Generate Log tab to check for warnings and error messages.
10. Click **Close**.

The IPexpress package file (.ipx) supported by Diamond holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing the .ipx file to the associated Diamond project. To change the option settings of a module or IP that is already in a design project, double-click the module's .ipx file in the File List view. This opens IPexpress and the module's dialog box showing the current option settings. Then go to step 6 above.

Regenerating an IP Core in ispLEVER

To regenerate an IP core in ispLEVER:

1. In the IPexpress tool, choose **Tools > Regenerate IP/Module**.
2. In the Select a Parameter File dialog box, choose the Lattice Parameter Configuration (.lpc) file of the IP core you wish to regenerate, and click **Open**.
3. The Select Target Core Version, Design Entry, and Device dialog box shows the current settings for the IP core in the Source Value box. Make your new settings in the Target Value box.
4. If you want to generate a new set of files in a new location, set the location in the LPC Target File box. The base of the .lpc file name will be the base of all the new file names. The LPC Target File must end with an .lpc extension.
5. Click **Next**. The IP core's dialog box opens showing the current option settings.
6. In the dialog box, choose desired options. To get information about the options, click **Help**. Also, check the About tab in the IPexpress tool for links to technical notes and user guides. The IP core might come with additional information. As the options change, the schematic diagram of the IP core changes to show the I/O and the device resources the IP core will need.
7. Click **Generate**.
8. Click the **Generate Log** tab to check for warnings and error messages.

Application Support

This chapter provides application support information for the DDR/DDR2 IP core.

Core Implementation

This section describes the major factors that are important for a successful DDR memory controller implementation.

Understanding Preferences

The following preferences are found in the provided logical preference files (.lpf):

- **FREQUENCY**
  The DDR memory controller core is normally 10% over-constrained for obtaining optimal f_{MAX} results. The post-route trace preference file contains the preferences that have the real performance targets, and it should be used to validate the timing results.
- **MAXDELAY NET**
  The MAXDELAY NET preference ensures that the net for the READ input to the DQSBUF block has a minimal net delay and falls within the data valid clearing window. Since it is highly over-constrained, the post-route trace preference file should be used to validate the timing results.
- **MULTICYCLE / BLOCK PATH**
  These preferences are used to avoid an overruled performance report from the static timing results. They are not considered critical in terms of the core operability but still important for a correct static timing report.
- **IOBUF**
  The IOBUF preference assigns the required I/O types to the DDR I/O pads. See the I/O Types for DDR section for details.
- **LOCATE**
  Only the em_ddr_dqs pads are located in the provided preference file for evaluation purpose. It is a general practice that they be relocated to the desired locations when the core is instantiated in a user application. Note that not all I/O pads can be associated with a DQS (em_ddr_dqs) pad in a bank. Since there is a strict DQ-to-DQS association rule in each Lattice device, it is strongly recommended the DQ-to-DQS associations of the selected pinouts be validated using the implementation software before the PCB routing task is started. The DQ-to-DQS pad associations for a target device can be found in the data sheet or handbook of the target device. If there are LOCATE PGROUP preferences in the .lpf file, they must also be relocated to the closest locations to the corresponding DQS pads after the DQS pads are relocated by a user. The new locations can be found in the Design Planner's Floorplan View or EPIC device editor.

  Refer to the DQS_PIO_READ Locate Constraints section for the procedure to locate DQS_PIO_READ pgroups for DDR2 IP.

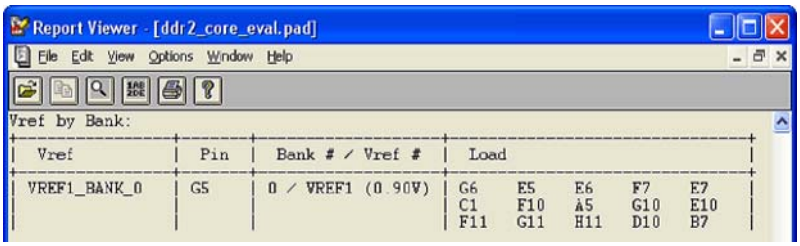
Preference Localization

Due to the nature of high-speed DDR operations, some of the internal nets must be constrained in order to achieve the functional and performance goal. However, the hierarchy structure and name of an internal net is subject to change when there are changes in the design or when a different version of a synthesis tool is used. It is the user's responsibility to track these changes and update them in the preference file. Since the FREQUENCY, MAXDELAY NET and LOCATE PGROUP preferences affect the functionality and performance of the core, it is good to pay close attention to tracking them after each run of the synthesis process. The updated net and path names can be found in the map report file (.mrp).

VREF Assignments

An SSTL I/O type pad requires a reference voltage input when it is operating as a receiving end. In the DDR design in a Lattice device, data and data strobe signals are bi-directional, and each of the banks that contain these bi-directional DDR interface signals must have a connection to the external reference voltage resource. Otherwise, the proper input level will not be detected. This can be done by connecting the VREF1 pad of the bank to the external reference voltage source. The VREF1 pad and its associated DDR input or bi-directional pads are listed in the pad report file (.pad), as shown in the example in Figure 5-1.

Figure 5-1. Example of VREF1 Connection Report (DDR2)



DLL Allocation

Lattice devices have dedicated DDR register structures in the input and output for read and write operations. The DQS delay block is required in order to correctly capture data at the input register. Since the data strobe signal (DQS) from the DDR memory is not free-running, a calibrated DLL function is required to precisely delay the incoming DQS. The DQSDLL block is used to generate the delay value on the dqs_del signal. The DQSBUFx block uses dqs_del to generate the 90-degree shifted DQS signal for read operations. Accuracy of this delay is crucial to maximize the capturing window for the read data. The calibration bus, UDDCNTL, controls the update and hold functions of the DLL to compensate for temperature, voltage and process variations. The DQSDLL block is updated when the core is not in the read mode. Note that UDDCNTL is an active-low update enable signal.

I/O Types for DDR

In the DDR mode, the SSTL25_I I/O type is used for all the DDR interface signals except the memory clock pads, em_ddr_clk, which need a differential I/O type, SSTL25D_I. When a DDR2 memory controller core is generated, both em_ddr_clk and em_ddr_dqs take the SSTL18D_I I/O type by default. Since the DQS mode is programmable in DDR2 memories, the I/O type for em_ddr_dqs can be easily replaced with the single-ended type, SSTL18_I, in the preference file. All other DDR2 interface pads use SSTL18_I.

Note that the local interface signals in the generated core are also assigned with the same I/O type as the DDR interface signals by default. Since the local interface signals are normally embedded inside the device once a system-level design is completed, the IOBUF preferences for them should be removed to avoid unnecessary preference warnings. If any of the local interface signals need to take the I/O pad including the clock and reset inputs, a proper I/O type for the signal must be selected to comply with the system requirement.

Skew Treatment

Lattice DDR memory controller is designed to use dedicated DDR I/O registers in order to minimize the skew among the DDR data and data strobe signals. Excessive skew between any two DDR data signals can be a major contributor to performance degradation. The skew of the DDR control signals among them and with respect to the DDR data is also crucial for high-speed implementation. The best way to minimize the skew of the DDR address and control signals is to implement all of them into the PIO registers instead of taking the registers inside the device fabric.

The IPexpress tool inserts the synthesis directives to push out those signals into the PIOs in the top-level wrapper when the core is generated. The implementation results can be found in the Design Summary section of the map report, which shows whether those signals are inside the PIO or not. The required I/O resource for each DDR interface signal is listed in Table 5-1. The table includes both the PIO and the dedicated DDR register resources. If the

PIO register number in the map report matches the total number of PIO registers calculated from the table, all of them are properly implemented into the PIO registers. The utilized IDDR/ODDR and PIO resources are reported separately in the Design Summary section.

Table 5-1. Required I/O Registers for Each DDR Interface Signals

DDR Pad	ODDR	IDDR	PIO Register	Total Required
em_ddr_dq	2	1	-	DATA_WIDTH x 3
em_ddr_dm	1	-	-	DATA_WIDTH / 8
em_ddr_dqs	2 (4)	1 (2)	-	DQS_WIDTH x 3
em_ddr_clk	1 (2)	-	-	CLK_WIDTH
em_ddr_odt	1	-	-	CS_WIDTH
em_ddr_addr	-	-	1	ROW_WIDTH
em_ddr_ba	-	-	1	BNK_WDTH
em_ddr_ras	-	-	1	1
em_ddr_cas	-	-	1	1
em_ddr_we	-	-	1	1
em_ddr_cs	-	-	1	CS_WIDTH
em_ddr_cke	-	-	1	CKE_WIDTH

Note: The numbers in parenthesis indicates that a differential pair is used. These are not included in the reported total.

Dummy Logic Removal

When a DDR IP core is generated, IPexpress assigns all the signals from both the DDR and local user interfaces to the I/O pads. The number of user interface signals is normally more than four times than that of the DDR interface. It makes the core impossible to be evaluated if the selected device does not have enough I/O pad resource. To facilitate core evaluation with smaller package devices, IPexpress inserts dummy logic to decrease the I/O pad counts by reducing the local read_data and write_data bus sizes by half. With the dummy logic, a core can be successfully 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 core with dummy logic, therefore, must be used only for evaluation purposes. The dummy logic parameter, DUMMY_LOGIC, is inserted in the top-level wrapper file if the generated core needs the dummy logic.

After a backend user logic design is attached to the core, most of the user interface signals are embedded. It is important to know that the DUMMY_LOGIC parameter should be removed from the top-level wrapper file before synthesizing the project. Another option is to use the simulation top-level wrapper file for synthesis.

Read Data Auto-Alignment Logic

Lattice devices have a dedicated DDR support circuitry that allows reliable capture of the read data from each DQS group with respect to the internal core clock. Because of possible PCB trace length differences among all DQS groups, the captured data from a DQS group may not be aligned with the ones from other DQS groups by one clock cycle. The data alignment logic in the DDR memory controller automatically aligns the read data received from the multiple DQS groups and presents it on the user interface.

PCB Routing Delay Compensation

After a read burst operation is completed, the read data valid generation logic must be initialized before the current read burst operation is started. The memory controller uses the incoming DQS signal (dqsi) from the memory for this operation. The valid timing window of this initialization is strictly defined to be within the preamble period ($t_{PREAMBLE}$). The DQS signal from the memory arrives after the round-trip delay that includes the following delay factors:

- Memory clock output delay from device

- Memory clock travel delay from device to memory through the routed PCB lines
- Memory internal delay from clock to DQS
- DQS travel delay back to device
- Device setup delay to the data valid generation logic

The IP will go through a training procedure before initialization is completed.

The training procedure will adjust each DQS lane's delay tap to ensure the read pulse is precisely positioned with the incoming DQS signal. Each tap lasts T/4.

DQS_PIO_READ Locate Constraints

The DDR/DDR2 IP has a few critical macro-like blocks called as DQS_PIO_READ pgroups that require specific placement locations. This placement is done by adding a “LOCATE” constraint in the .lpf file for each pgroup.

The user must manually update these constraints in the .lpf file by adding location values obtained as follows. In DDR2 mode, all locations of DQS pins are user selectable and the corresponding DQS_PIO_READ macros are automatically implemented. The following steps are for DDR1 mode, or for when the user changes DDR2 DQS pin locations after core generation.

Obtaining Location Values in Diamond Software

***Note**: Refer to Diamond online help for more information about using the Diamond software.*

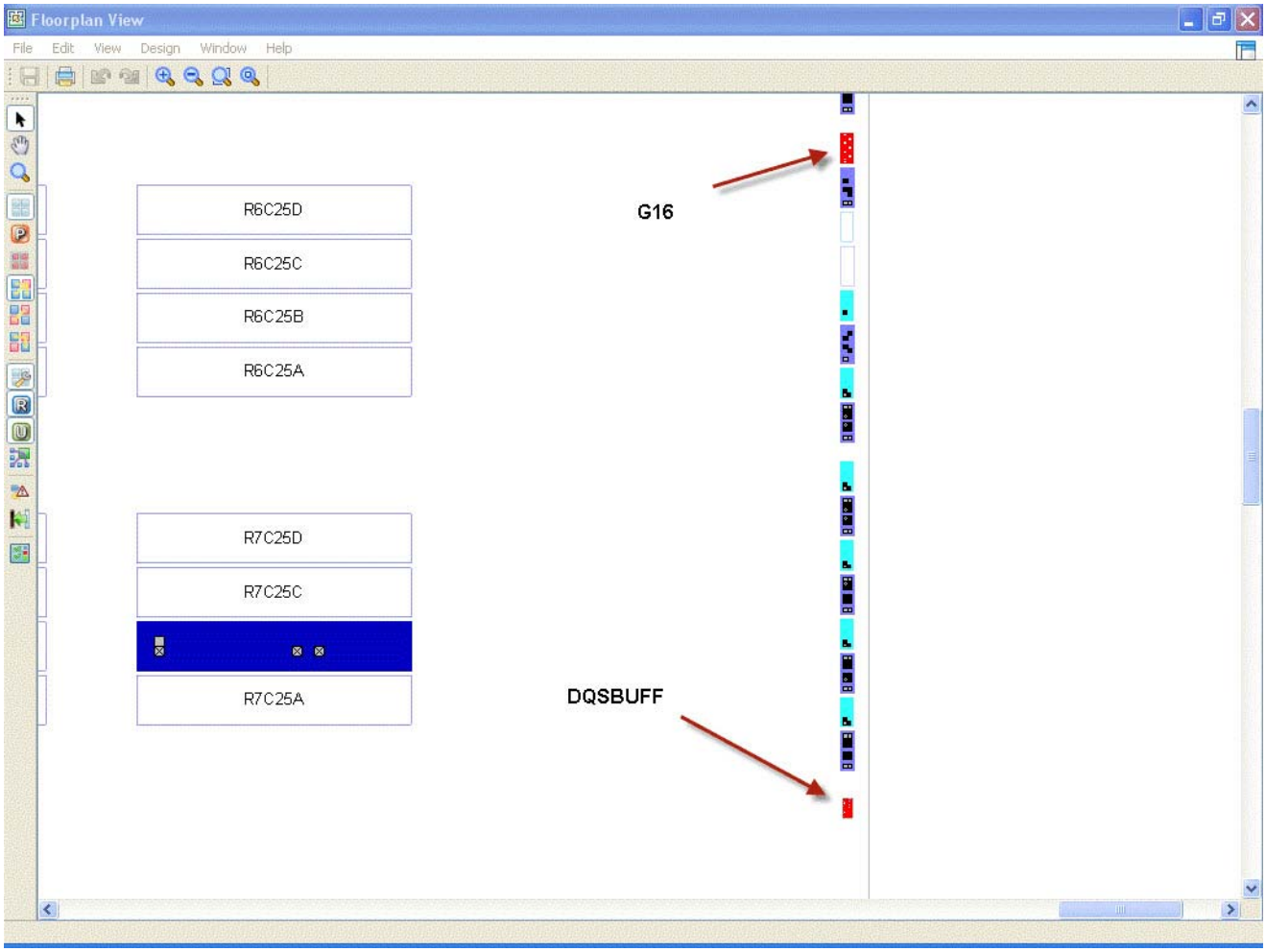
1. With the Eval project open in Diamond, run the **Place & Route** process without locating the DQS_PIO_READ pgroup in the .lpf file.
2. Enable the **Floorplan View**.
3. In the Floorplan View, find the location of the DQS pin (G16 as shown in the example in Figure 5-2).
   a. Find the nearby DQSBUF block.
   b. In the .lpf file, locate the DQS0_PIO_READ pgroup in the row and column of the closest SLICE (R7C25D as shown in the example in Figure 5-2) from this DQSBUF block.
4. Repeat Step 3 for each DQS pin of the design.
5. Once the .lpf file is updated for all DQS pins, re-run the **Place & Route** process using the updated .lpf file.

Note: If there is a MAXDELAY violation on any constraint listed below, locate the corresponding DQS_PIO_READ pgroup in the adjacent SLICE and re-run PAR.

MAXDELAY TO NET "/dqs_pio_read*"*

Figure 5-2 is an example Diamond Floorplan View showing typical locations of the DQS pin (N9), the corresponding DQSBUF block, and the closest SLICE (R7C25D) in a MachXO2 device.

Figure 5-2. Diamond Floorplan View



Obtaining Location Values in ispLEVER Software

***Note**: Refer to ispLEVER online help for more information about using the ispLEVER software.*

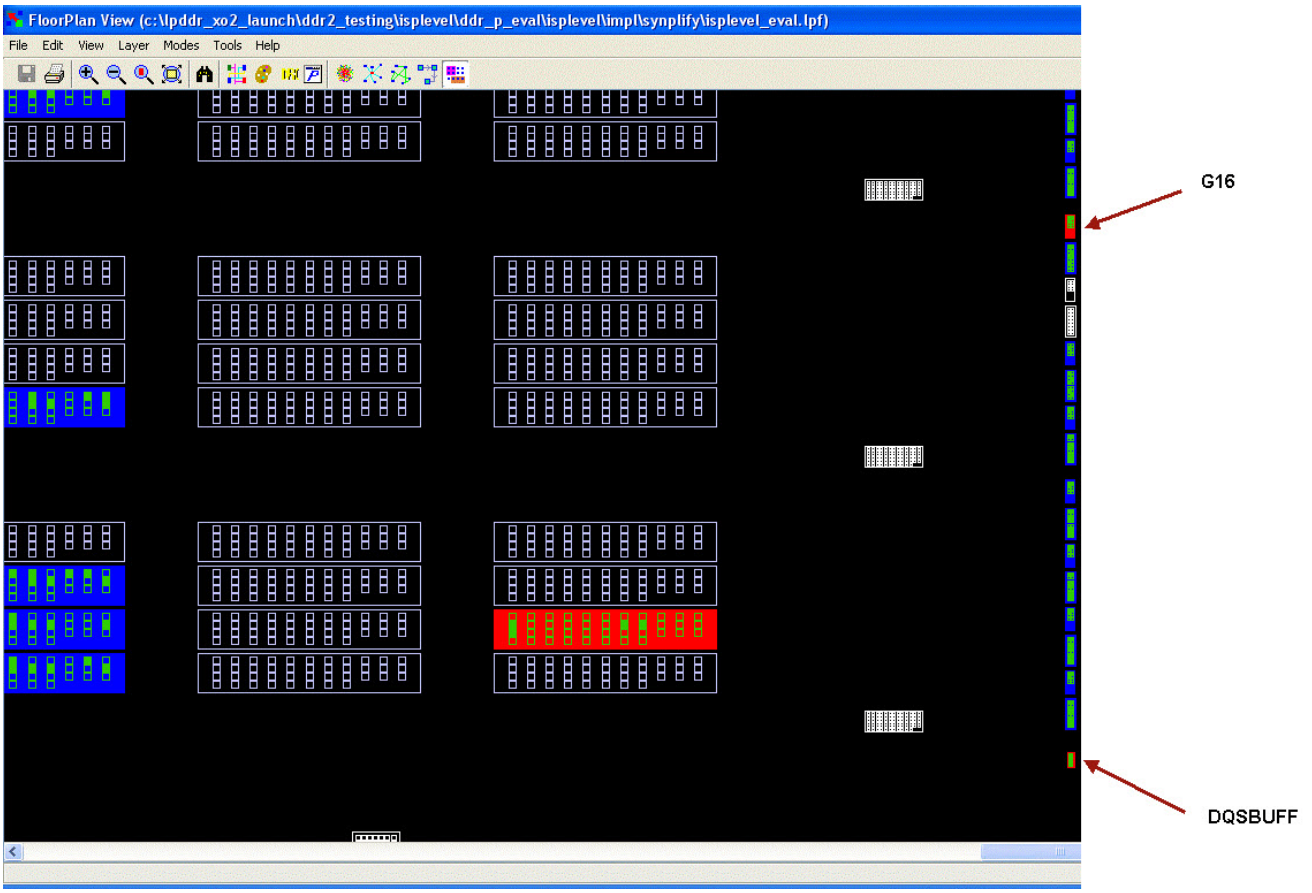
1. With the Eval project open in the ispLEVER Project Navigator, run the **Place & Route** process without locating the DQS_PIO_READ pgroup in the .lpf file.
2. Open the **Design planner [Pre-Map]** for this design and enable **Floorplan View**.
3. In the Floorplan View, find the location of the dqs0 pin (G16 as shown in the example in Figure 5-3.).
   a. Find the nearby DQSBUF block.
   b. In the .lpf file, locate the DQS0_PIO_READ pgroup in the row and column of the closest SLICE (for example R7C25D) from this DQSBUF.
4. Repeat Step 3 for each DQS pin of the design.
5. Once the .lpf file is updated for all DQS pins, re-run the **Place & Route** process using the updated .lpf file.

Note: If there is a MAXDELAY violation on the constraint listed below, locate the corresponding DQS_PIO_READ pgroup in the adjacent SLICE and re-run PAR.

MAXDELAY TO NET "/dqs_pio_read*"*

*Figure 5-3 is an example ispLEVER Floorplan View showing typical locations of the DQS pin (G16), DQSBUF block and the closest SLICE (*R7C25D*).*

Figure 5-3. ispLEVER Floorplan View



Troubleshooting

When a Lattice DDR memory controller-based system does not work as expected, there could be numerous reasons for the failure. Table 5-2 summarizes some approaches for troubleshooting during DDR system implementation.

Table 5-2. Troubleshooting of DDR Memory Controller Implementation

Symptom	Possible Reason	Troubleshooting
No read data received	Incoming DQS failure	Monitor the DQS signal from the memory. If no DQS is detected during the read operation, it may be a memory failure. Replace the memory.
Corrupted Read data	Incorrect read data valid timing	Check whether the READ input (to DQSBUF) has been properly constrained and implemented (MAXDELAY NET preference).
Data corruption in a specific frequency range	Data valid timing alignment failure	Although rare, it is possible for this to happen if the memory module is incompatible with the implemented core. Try different memory modules.
Read data is shifted in simulation while the hardware system is working	Clock delta delay	If a design has one or more clocks assigned from the original clock source, the design may have the clock delta delay issue when both the original and assigned clocks are used in the design. Bring the internal clock generator block out to the top-level of the system and distribute it to the rest of the sub-design blocks.

Table 5-2. Troubleshooting of DDR Memory Controller Implementation (Continued)

Symptom	Possible Reason	Troubleshooting
Unexpectedly low performance	Incorrect read data valid timing	See the description for “Corrupt read data”.
	Un-terminated memory control signals	If a system uses only a part of a DDR memory module and the unused memory control signals (such as chip select and clock enable) and the module remains unterminated, they may make the memory module sensitive to noise. Unused control signals must be terminated to their inactive state.
	Excessive skew on memory control signals	Excessive skew between any DDR interface signals can degrade performance. All address and control signals on the DDR interface must be implemented in the PIO registers to minimize the skew.

Core Verification

The functionality of the Lattice DDR and DDR2 IP cores have been verified via simulation with Micron Technology, Inc., DDR and DDR2 simulation models, including simulation environments that verify proper DDR and DDR2 functionality using Lattice's proprietary verification environment.

Chapter 7:

Support Resources

This chapter contains information about Lattice Technical Support, additional references, and document revision history.

Lattice Technical Support

There are a number of ways to receive technical support.

E-mail Support

techsupport@latticesemi.com

Local Support

Contact your nearest Lattice sales office.

Internet

www.latticesemi.com

References

MachXO2

- DS1035, *MachXO2 Family Data Sheet*

JEDEC Website

The JEDEC website contains specifications and documents referred to in this user's guide. The JEDEC URL is:

http://www.jedec.org

JESD79-2F, *DDR2 SDRAM Standard* (http://www.jedec.org/standards-documents/docs/jesd-79-2e)

Micron Technology, Inc., Website

http://www.micron.com

Revision History

Date	Document Version	IP Core Version	Change Summary
March 2015	1.2	1.2	Updated the Command Decode Logic section. Removed statement on CDL block providing command burst function.
			Updated the Signal Descriptions section. Revised cmd_valid description in Table 2-1, DDR SDRAM Memory Controller Top-Level I/O List.
			Updated Timing Tab section. Changed Figure 3-4, Timing Tab to match with v1.2 IP.
			Updated Synthesis & Simulation Tools Option Tab section. Changed Figure 3-5, Synthesis & Simulation Tools Option Tab to match with v1.2 IP.
			Updated Info Tab section. Changed Figure 3-6, Info Tab to match with v1.2 IP.
			Updated Getting Started section. Changed the following figures to match with v1.2 IP. — Figure 4-1, IPexpress Tool Dialog Box (Diamond Version). — Figure 4-2, DDR/DDR2 SDRAM IP Configuration GUI (Diamond Version).
			Updated Lattice Technical Support section.
February 2012	01.1	1.0	Updated document with new corporate logo.
November 2010	01.0	1.0	Initial release.

Appendix A:

Resource Utilization

This appendix gives resource utilization information for Lattice devices using the DDR/DDR2 IP core. The IP configurations shown in this chapter were generated using the IPexpress software tool. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the Diamond and ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and Diamond and ispLEVER help systems. For more information on the Diamond or ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

MachXO2 Devices

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

IP Core	Parameter Settings[2]	SLICEs	LUTs	Registers	I/O	f_{MAX} (MHz)
DDR	Table 3-1 parameter defaults	640	1193	1150	151	134 MHz (266 DDR)
DDR2	Table 3-1 parameter defaults	701	1311	1188	152	141 MHz (266 DDR2)

1. Performance and utilization characteristics are generated using LCMXO2-2000HC-6FTG256C in Diamond 1.1 software. Performance may vary when using this IP core in a different density, speed or grade within the MachXO2 family.
2. SDRAM data path width of 16 bits

Ordering Part Number

The Ordering Part Number (OPN) for the Pipelined DDR SDRAM Controller IP on MachXO2 devices is: DDRCT-WB-M2-U.

The Ordering Part Number (OPN) for the Pipelined DDR2 SDRAM Controller IP on MachXO2 devices is: DDR2CT-WB-M2-U.

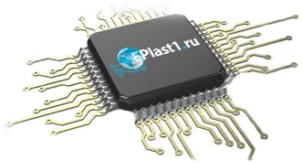
Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный)
Факс: 8 (812) 320-02-42
Электронная почта: org@eplast1.ru
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.