RLDRAM Controller MACO Core

User's Guide

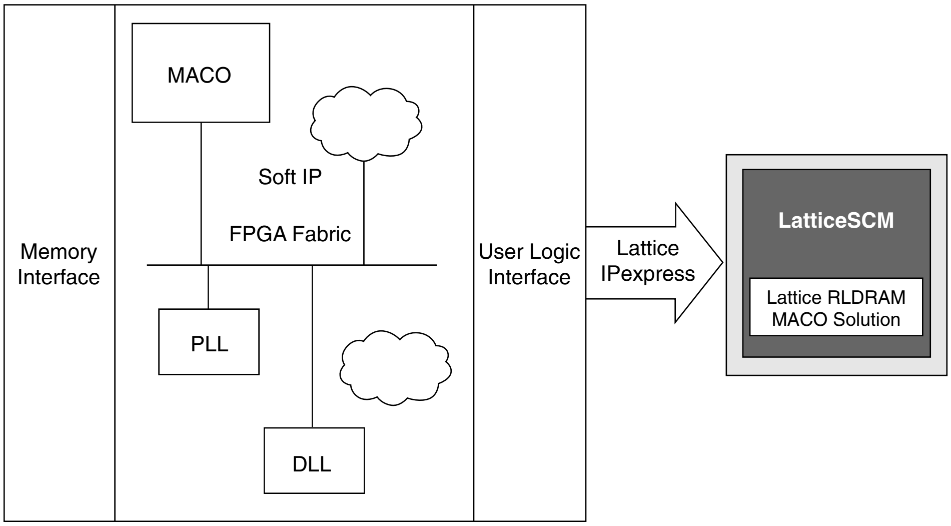
November 2009
ipug47_01.5

Introduction

Lattice's RLDRAM I/II Memory Controller MACO™ IP core assists the FPGA designer by providing pre-tested, reusable functions that can be easily plugged in, freeing the designer to focus instead on system architecture design.

These blocks eliminate the need to "re-invent the wheel" by providing industry-standard RLDRAM I/II memory controller modules. These proven cores are optimized utilizing the LatticeSCM device's MACO architecture, resulting in fast, small cores that utilize the latest architecture to its fullest.

Figure 1. Lattice MACO Conceptual Diagram



Complementing the Lattice ispLEVER® software is the ability to generate user-customizable cores using the IPexpress™ utility. This utility helps the designer to input design information into a parameterized design flow. Designers can use the IPexpress software tool to generate specific configurations of this IP core. Information on bus size, clocking, and memory device requirements are prompted by the GUI and compiled into the FPGA design database. The utility generates templates and HDL-specific files needed to synthesize the FPGA design.

IPexpress, the Lattice IP configuration utility, is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER online Help systems. For more information on the ispLEVER design tools, visit the Lattice web site at www.latticesemi.com/software.

Overview

The Reduced Latency Dynamic Random Access Memory (RLDRAM) Controller is a general-purpose memory controller that interfaces with industry standard RLDRAM. The controller can be configured to function as RLDRAM I, RLDRAM II CIO (Common I/O) or RLDRAM II SIO (Separate I/O). The RLDRAM is a high-speed memory device used mainly in high bandwidth, latency sensitive application like communications, data storage etc. Data is transferred on both edges of the clock, doubling the rate of data transfer. This IP provides an interface between the RLDRAM and the generic bus interface.

The RLDRAM controller is designed to support:

- RLDRAM I with target speeds up to 300 MHz DDR
- RLDRAM II CIO/SIO with target speeds up to 400 MHz DDR

The RLDRAM applications include, but are not limited to, transmitting and receiving buffers in telecommunication systems and data or instruction cache applications requiring large amounts of memory.

Features

- Interfaces to industry standard RLDRAM I, RLDRAM II CIO and RLDRAM II SIO devices
- Supports RLDRAM memory devices operating up to 400 MHz (800 Mbps/pin)
- Allows programmable burst lengths of
  - 2 or 4 in RLDRAM I mode
  - 2, 4 or 8 in RLDRAM II mode
- Includes a reconfigurable refresh counter
- Command & Write Data pipeline to maximize throughput
- Supports the following commands at the user interface
  - READ, WRITE, LOAD_MR
- Contains command pipeline to maximize throughput
- Supports:
  - RLDRAM data path widths of 16 and 32 bits in RLDRAM I mode
  - RLDRAM data path widths of 9,18, 36, 72 and 144 bits in RLDRAM II CIO mode
  - RLDRAM data path widths of 9, 18, 36 and 72 bits in RLDRAM II SIO mode
  - Data Mask signals
- Controller operates at half the memory clock frequency
- Supports up to 8 chip selects in RLDRAM II modes, one chip select in RLDRAM I mode
- Operates synchronously
- Generates differential output clocks (CK,CK#)
- Accepts differential input data clocks (DK,DK#)
- Supports RLDRAM II CIO MiniDIMM_72 and MiniDIMM_ 144, and RLDRAM II SIO MiniDIMM_72
- Includes DLL to align DQ and QK transitions with CK
- Supports user-selectable 1, 2, 4 and 8 DLLs for the 72- and 144-bit mini-DIMMs

Design Kit Deliverables

- Sample instantiation (template)
- Synthesis black box for MACO core
- Pre-compiled ModelSim® MACO core model
- Verilog core source code
- Verilog testbench
- Preference files

Getting Started

Requirements to implement a MACO core include:

- ispLEVER version 6.1 SP2 or later
- MACO Design Kit: see the ReadMe file supplied with the IPexpress RLDRAM MACO Kit for details on the Kit contents
- MACO license file
- See the Lattice ispLeverCORE IP Tutorial for more information on the ispLEVER design flow

For information on obtaining the above requirements, please contact your local Lattice Semiconductor sales representative.

Functional Description

The RLDRAM controller provides a generic command interface to the user's application. This interface reduces the effort to integrate the module with the user's design and minimizes the user's need to deal with the RLDRAM command interface. The configuration parameters for the memory can be changed which enables the user to switch between different memory devices and/or modify the timing parameters to suit the application using the ispLEVER GUI.

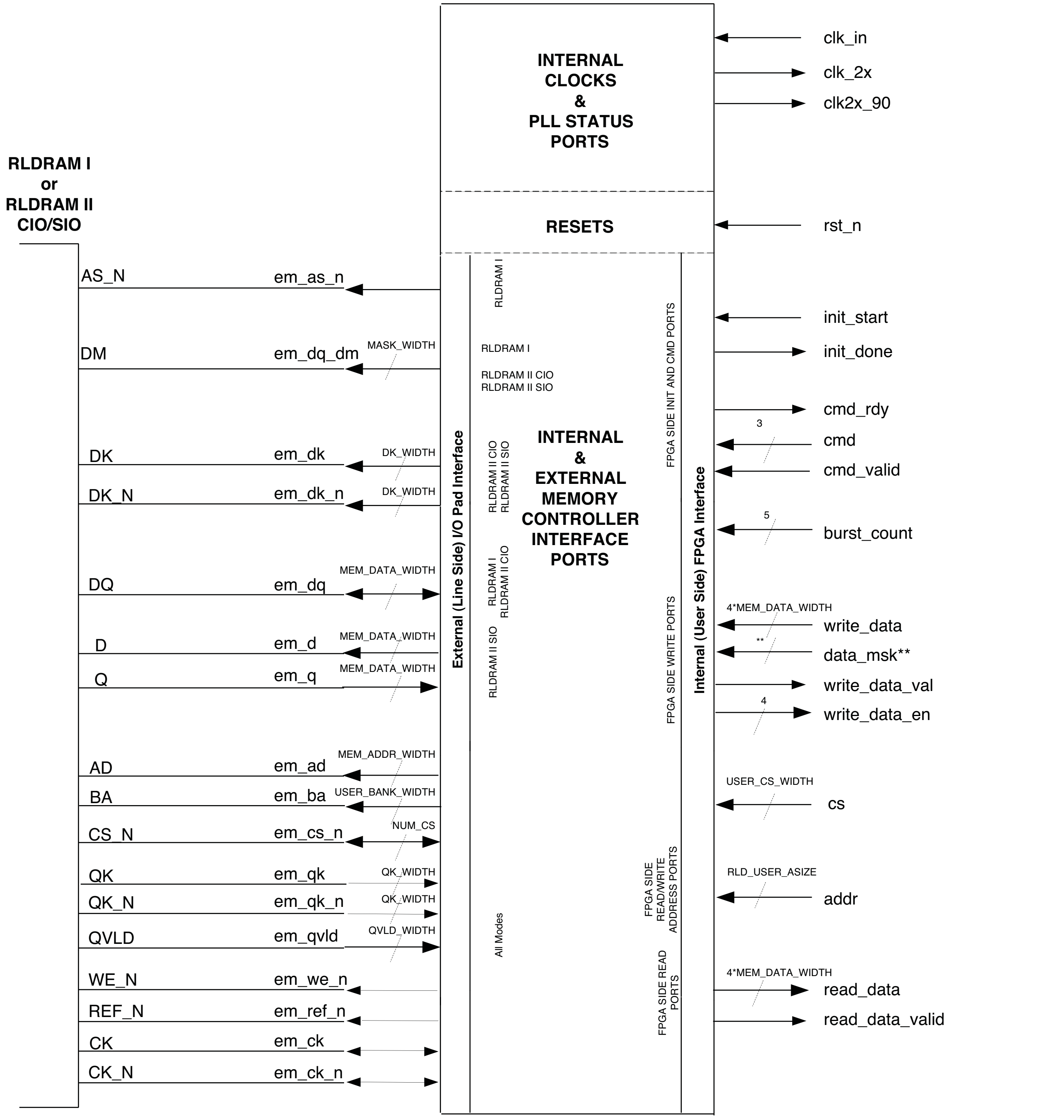
While most of the functionality of the memory controller remains the same for RLDRAM I, RLDRAM II CIO and RLDRAM II SIO modes, certain differences exist, as shown in Table 1.

Table 1. Differences Between RLDRAM Modes

Feature	RLDRAM I	RLDRAM II CIO	RLDRAM II SIO
Common I/O (CIO) or Separate I/O (SIO)	CIO	CIO	SIO
Multiplexed Address	No	Selectable	
Configurations Supported	1, 2, 3 and 4	1, 2 and 3	
Burst Length	2, 4	2, 4, 8	
On-Die Termination (ODT)	No	Supported	
Device Density	256 MB	288 MB	
Max. Clock Frequency	300 MHz	400 MHz	
Max Bus Bandwidth per Device	19.2 Gbps	28.8 Gbps	
RLDRAM I/O Impedance	50¾only	User-programmable 25¾ to 60¾	
Clock DLL in RLDRAM Device	No	Yes	

Top Level Block Diagram

Figure 2. RLDRAM_TOP Module



**data_msk = if RLDRAM I
data_msk[1:0]
if RLDRAM II
d_msk[(4* MASK_WIDTH - 1):0]

*cs[USER_CS_WIDTH-1:0]

Only if USER_NUM_CS_1
not true

Input/Output Signals

Table 2 lists the signals connecting to the User and RLDRAM interfaces from the embedded MACO block:

Table 2. I/O Signals

Pin Name List	Pin Direction	Conditions/Comments
Clock and Reset Pins		
rst_n	In	System reset. *Note: rst_n is synchronous, and requires a rising edge on clk_in order to be applied.*
clk_in	In	System clock
User Interface Pins		
init_start	In	Asserted when an initialization routine is to be performed.
init_done	Out	Asserted when the controller has completed the initialization routine.
cmd_rdy	Out	Asserted to indicate that the controller is ready to accept a new command.
cmd [2:0]	In	Command for the controller.
cmd_valid	In	Asserted when the contents of cmd, addr, burst_count are valid.
addr[RLD_USER_ASIZE -1 : 0]	In	Address for read/write. It includes bank address as well. For example: RLDRAM I - 576 Mbit - 18 bit data width 576/18 = 32M locations BL2 = For every address two locations Effective address locations = 32M/2 = 16M Addressing 16M locations requires 24 address bits. RLD_USER_ASIZE = 24
cs[USER_CS_WIDTH -1: 0]	In	Not present if RLDRAM I and USER_NUM_CS_1 is true
burst_count[4:0]	In	Indicates the number of read/write commands to be issued to the memory chip.
write_data[(4* MEM_DATA_WIDTH 1) :0]	In	Write data input. MEM_DATA_WIDTH is the width of the memory device connected.
write_data_en[3:0]	Out	Write data enable. When asserted, the controller is ready to accept data on the write_data bus.
write_data_valid	Out	Write data valid signals for the write_data bus. Gets asserted along with write_data_en signal.
data_msk[1:0]	In	RLDRAM I Only Data Mask for write data. Should be asserted along with the command. Valid only when controller is configured to RLDRAM I.
data_msk[4*`MASK_WIDTH-1:0]	In	RLDRAM II Only. data_msk[N] corresponds to write_data[((W*(((N%4)+1))+(4*(N\4)))-1) : (W*(((N%4)+0))+(4*(N\4)))] where: • N = data_msk bit number; takes on values of 0 to ((4*(MASK_WIDTH))-1). • W = data bus width of each RLDRAM device • % is modulo operation; \ is integer divide; + is add; * is multiply
read_data[(4*MEM_DATA_WIDTH -1) :0]	Out	Read Data from memory to user. MEM_DATA_WIDTH is the width of the memory device connected.
read_data_valid	Out	When asserted, the contents of the read_data bus are valid.

Table 2. I/O Signals (Continued)

Pin Name List	Pin Direction	Conditions/Comments
RLDRAM Memory Interface Pins		
RLDRAM CLK Pins		
em_ck	Inout	Address and commands are issued with respect to this clock.
em_ck_n	Inout	Differential em_ck
em_dk [DK_WIDTH-1: 0]	Out	RLDRAM II Only The data going into the RLDRAM on em_dq is aligned to this clock during memory write.
em_dk_n [DK_WIDTH-1: 0]	Out	RLDRAM II Only Differential Data Clock
em_qk [QK_WIDTH -1 : 0]	In	Data Strobe input signal from the memory.
em_qk_n [QK_WIDTH -1 : 0]	In	Differential Data Strobe
RLDRAM Data and Control Pins		
em_cs_n [NUM_CS -1 : 0]	Out	Active low chip select, which selects and deselects the RLDRAM devices. NUM_CS indicates the number of memory devices that controller is connected.
em_as_n	Out	RLDRAM I Only Address Strobe signal to the memory.
em_ad [ADDRESS_WIDTH -1: 0]	Out	Address Output signal to the memory.
em_ba [BANK ADDRESS - 1: 0]	Out	Bank Select signal to the memory.
em_d [MEM_DATA_WIDTH -1: 0]	Out	RLDRAM II SIO Only Write data bus.
em_dm [1: 0]	Out	RLDRAM I Only Data mask signals used to mask the data to be written.
em_dm [MASK_WIDTH-1:0]	Out	RLDRAM II Only
em_dq [MEM_DATA_WIDTH -1: 0]	Inout	RLDRAM I and RLDRAM II CIO Bi-directional data bus.
em_q [MEM_DATA_WIDTH -1: 0]	In	RLDRAM II SIO Only Read data bus.
em_qvld [QVLD_WIDTH -1: 0]	In	Data valid signal indicating valid data out from memory
em_we_n	Out	Write Enable signal to the memory.
em_ref_n	Out	Auto Refresh signal to the memory.

Table 3. Configuration Interface Signal[1]

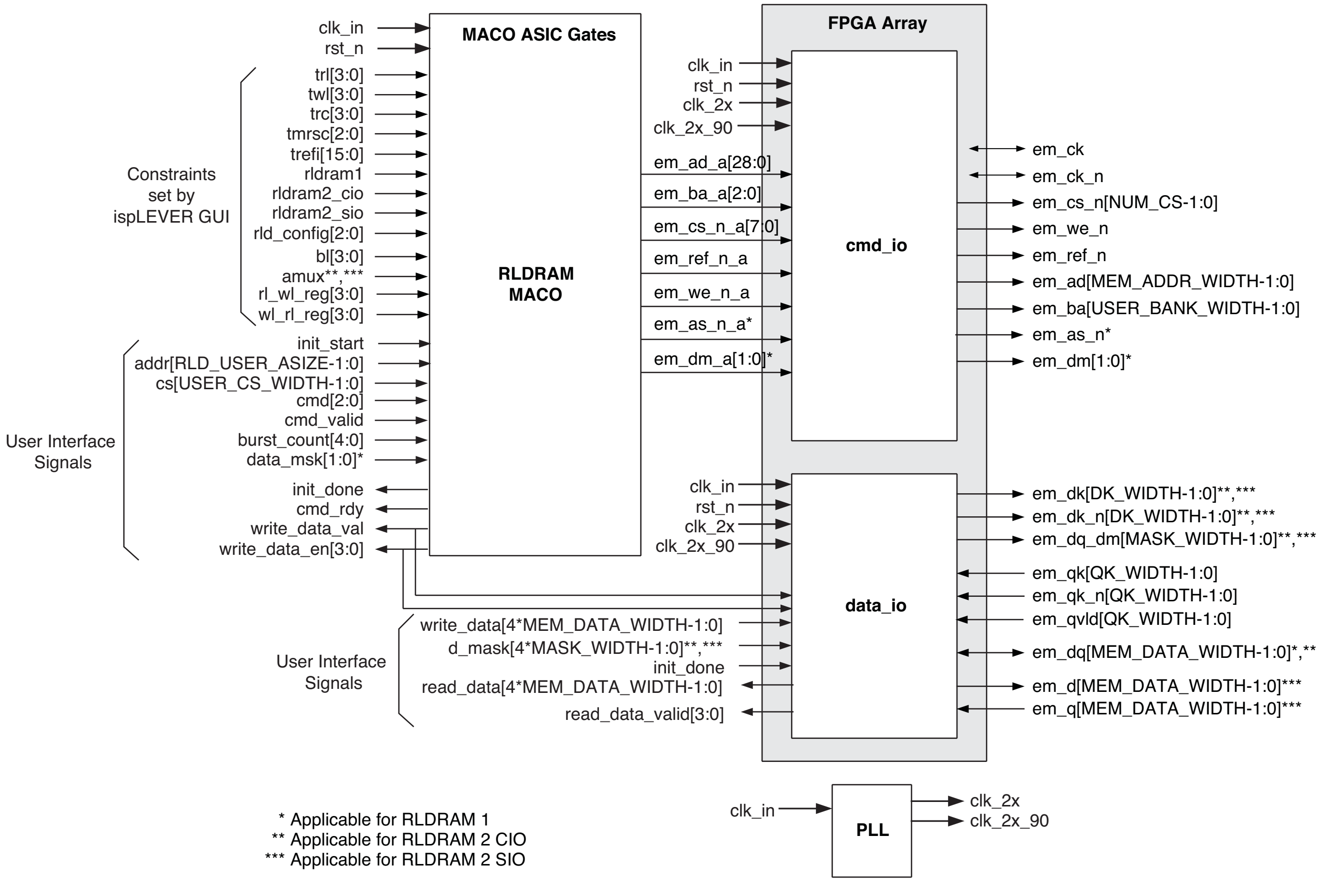
Signal Name	Active State	Signal Direction (I/O)	Signal Description
TRL	NA	I	Read latency in clk_in clocks.
TWL	NA	I	Write latency in clk_in clocks.
TRC	NA	I	Row cycle time in clk_in clocks.
TMRSC	NA	I	Mode register set cycle time to any command in clk_in clocks.
TREFI	NA	I	Refresh interval in clock cycles in clk_in clocks.
RLDRAM1	High	I	Signal indicating that the core is configured to RLDRAM I.
RLDRAM2_CIO	High	I	Signal indicating that the core is configured to RLDRAM II CIO.
RLDRAM2_SIO	High	I	Signal indicating that the core is configured to RLDRAM II SIO.
USER_CONFIG	NA	I	Configuration of RLDRAM device. This value is programmed during initialization.
BL<value>	NA	I	Burst length of RLDRAM device. This value is programmed during initialization.
AMUX	High	I	Address mux selection of RLDRAM II CIO/SIO device. This value is programmed during initialization.
MACO_LEFT	High	I	Uses the left MACO of the device.

Table 3. Configuration Interface Signal[1] (Continued)

Signal Name	Active State	Signal Direction (I/O)	Signal Description
MACO_RIGHT	High	I	Uses the right MACO of the device.
PINOUT_BOTTOM	High	I	Uses the bottom pinout of the device.
PINOUT_SIDE	High	I	Uses the side pinout of the device.
SPEED_<value>	N/A	I	Defines the speed of the clk_2x.
MEM_DATA_WIDTH_<value>	N/A	I	Memory data width specification.
QK_WIDTH_<value>	N/A	I	Specifies the QK width.
DK_WIDTH_<value>	N/A	I	Specifies the DK width.
USER_NUM_CS_<value>	N/A	I	Specifies the number of chip selects.

1. Automatically set up by ispLEVER GUI.

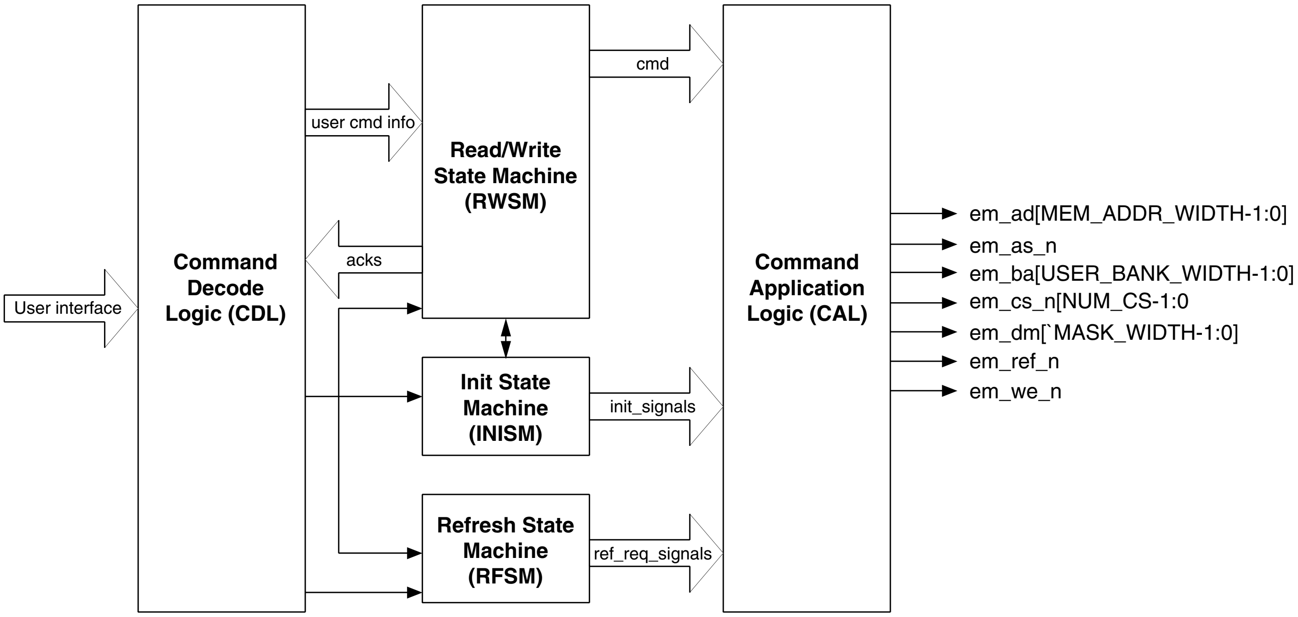
Figure 3. RLDRAM_TOP Detailed Diagram of the Controller



The RLDRAM Controller core (or IP core) is made up of the following blocks:

- RLDRAM MACO – implemented as a hard block in the LatticeSCM device
- cmd_io – instantiates I/Os for memory device command and address bus
- data_io – instantiates I/Os for memory device data bus
- PLLs and DLLs

Figure 4. RLDRAM MACO Block Diagram



Command Decode Logic

The commands presented by the user are decoded and placed into the internal queue by this module. The controller will assert cmd_rdy, whenever the controller is capable of accepting a new command from the user. The controller will register a new command, when cmd_rdy and cmd_valid signals are asserted. Command is then sent to the CDL block. If the user presents a command that is not listed above, it will be ignored. Command signals are comprised of addr, cs, cmd_valid, burst_count and cmd. Data_msk signal is part of the command when controller is configured to RLDRAM I. For RLDRAM II CIO/SIO data_msk should be applied along with the write data.

For write commands, the core asserts the write_data_val signal to indicate its willingness to accept the data that has to be written into the memory through the write_data bus. Along with the write_data_val signal write_data_en[3:0] will be asserted indicating valid data bytes on write_data bus. When burst length is 4/8, all bits of write_data_en[3:0] will be asserted along with the write_data_val. When burst length is 2, write_data_en[3:0] can be 4'b0011 or 4'b1111. Data is sampled on every clock edge that sampled the write_data_val signal asserted. It is possible for the write_data_val signal to be asserted continuously when successive write commands are presented. Note that the assertion times of write_data_val and cmd_rdy will differ based on different factors such as the rldram burst length (2, 4 or 8), write latency etc.

When the user presents a read command, the core presents the data read from the memory on the read_data bus. The contents of this bus are valid with respect to the read_data_valid[3:0] signal assertion. When burst length is 4/8, the read data is valid when all the read_data_valid[3:0] bits are asserted. When burst length is 2, read_data_valid[3:0] can be 4'b0011 or 4'b1111.

If the command received was for a mode register write, controller continues and completes execution of all commands in the queue ahead of the MODE register update command. New commands will be accepted once the mode register update is complete and the memory chip is reprogrammed with the new values.

This module also maintains a refresh counter and issues a requests for refresh commands to be generated. The controller will post eight auto-refresh commands to the memory chip to provide efficient refresh handling. Changing the value of trefi changes the interval between sets of eight auto-refresh commands being posted.

The controller supports a burst mode of command execution where the user provides a base address and a burst count. The read or write command is then executed as many times as specified by the burst_count signal. The controller supports a burst count of up to 31.

The generic user interface is designed to be simple enough for integrating the core to standard bus interfaces. The user is required to only supply the Read, Write and Load Mode register commands through the interface. The controller will apply the proper commands based on the address of the accessed location. Table 4 shows the valid values for the cmd bus. If the user presents a command that is not listed below, it will be ignored and considered as a NOP command.

The Mode Register will be configured according to the IspLEVER GUI settings, during initialization. Users should not change the burst length, configuration, and address mux bits after initialization.

Table 4. User Interface Commands

Command	cmd[2:0]	Description
READ	001	Start a read burst
WRITE	010	Start a write burst
Mode Register Set (MRS1)	011	Load the RLDRAM I mode register
MRS2	100	Load the RLDRAM II mode register

Note: The Read Write commands are issued by the controller based on the burst length selected, t_{WL}, t_{RL} and t_{RC} latencies. The Load Mode Register command causes an update to the Mode/Extended register.

Initialization State Machine

This block does initialization in a predefined manner, as specified in the RLDRAM specification. During the initialization process this logic uses the BL (Burst Length), CONFIG (configuration) values loaded during FPGA configuration when updating the mode register.

The initialization for the controller can only be performed after the supply voltages are stable (following the power sequencing steps described in the RLDRAM specifications) and a stable clock is running for at least 200 µs. At this time the user must assert init_start signal for at least one clock period of clk.

The following operations are done as a part of the initialization process:

1. Issues three Mode Register Set commands: two dummies plus one valid MRS. In case a RLDRAM II CIO/SIO device is configured to the address multiplex mode, an extra Mode Register Set command is issued. The contents of the mode register are derived from the settings of the ispLEVER GUI.

2. tMRSC after the valid MRS, issues eight AUTO REFRESH commands, one on each bank and separated by 2,048 cycles. The initial bank refresh order does not matter.

3. After t_{RC}, the device is ready for normal operation.

After the completion of the initialization process, the init_done signal will be asserted. Read commands are executed in the order received. Write commands are also executed in the order received. When there is a combination of reads and writes, the reads and writes will be interspersed together during execution but the order of the read and the order of the writes will be maintained.

Read/Write State Machine

This module gets its inputs from the command decode logic. The commands presented by the decode logic are applied to the memory in the order received.

Commands in the pipeline are executed continuously to maintain a high throughput. This module also meets the timing requirements set by the user through the ispLEVER GUI, which are t_{RC}, t_{WL}, t_{RL} and the burst length.

This logic also multiplexes the address lines on the 11 address pins when in the address multiplex mode. This mode is used to reduce the number of address pins connected to the memory.

Refresh State Machine

The function of the Refresh State Machine is to generate auto refresh commands at periodic intervals as required by the RLDRAM memory device. The Refresh State Machine generates a burst of eight auto refresh commands at an average periodic interval of 3.9 µs. The address lines on the memory interface are "don't cares" during a refresh command. The value on the bank address bits selects the bank to be refreshed. The refresh interval (t_{REFI}) is programmed during FPGA configuration. Value of t_{REFI} is (clk_2x frequency in MHz*3.9µs -50)/2. Clk_2x frequency is one of the settings that user configures through ispLEVER GUI.

As an example, if a user selects SPEED_400, the value of t_{REFI} = 400*3.9 - 50/2 = 755.

User Interface

After a power-on reset, the user requests IP initialization by pulsing the init_start signal. The IP signals initialization completion by setting init_done active high.

After initialization, the user can issue a command to the IP by placing a valid command on the cmd bus and making the cmd_valid active. The command will be accepted by the IP when the IP generates a cmd_rdy signal. Once the command is accepted the user can place next command on the cmd bus and wait until the cmd_rdy goes active.

Along with a Read or Write command, the user also needs to place the burst_count and the addr signals for that particular command. When using burst count, address will get incremented automatically by the controller and always lies within the same chip select. After reaching the last address within the same chip select, address will be wrapped to zero within the same chip select.

If the command issued was a Read, the read data will be available on the read_data bus when read_data_valid is active.

If the command issued was a Write, the user has to provide the data to be written on the write_data bus when write_data_en is active.

The data_msk signal is used to mask the data being written, in case of RLDRAMI this information has to be provided along with the command being applied on the cmd bus. In case of RLDRAM II the data_msk signal should be provided along with the data being written provided on write_data bus. See the Write Data Mask Timing Diagrams in the RLDRAM vendors' data sheets for more detail.

User Address Mapping

For Single External Chip Select
Consider RLDRAM I - 576 Mbit - 18 bit data width

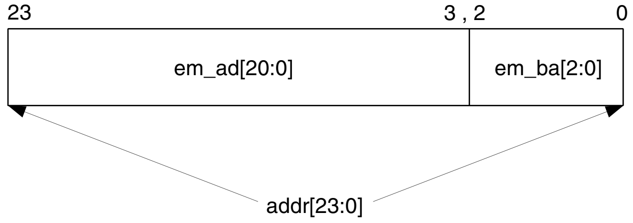
576/18 = 32M locations

BL2 = For every address two locations

Effective address locations = 32M/2 = 16M

To address 16M locations required number of address bits is 24. Figure 5 shows how the user address is mapped to the memory address.

Figure 5. Mapping of User Address to Memory Address for Single Chip Select



For Multiple External Chip Selects
For multiple chip selects (up to 7), the user can set the user-side cs[2:0] pin accordingly. These add to the overall user-side address width.

Requirements for Achieving Peak Bandwidth

For RLDRAM, both row and column addresses are latched at the same time, which eliminates the t_{RCD} timing parameter. The row cycle time t_{RC} is also reduced.

The peak bandwidth is calculated with the following assumptions:

1. 8-word burst with 36 bits of data in a word

2. Clock frequency of 400 MHz (2.5 ns) for RLDRAM II and 300 MHz for RLDRAM I

3. Read latency of eight clocks and write latency of nine clocks

Raw bandwidth with 36 bits of data at 400 MHz is 28.8 Gbps. Assuming 31 8-word bursts of writes followed by 31 8-word bursts of reads, the total clocks consumed = (31 * 4 clocks) + 9 + (31 * 4) + 8 + 1 clock for turnaround = 274 clocks. Total number of bits = 2* 31 bursts * 8 words/burst * 32 bits/word = 17856 15872 bits.

Peak bandwidth = (17856 * 10E+9)/ (274 * 2.5) = 26.07 Gbps

An additional 0.5% bandwidth reduction can be assumed due to periodic refreshes.

The bandwidth gains in RLDRAM II SIO would not be significant compared to CIO, since only the turnaround cycle is eliminated; however, the bandwidth of SIO is double that of CIO for a given bus width when the user issues single write commands followed by single read commands and so on, since this causes both the read and write buses to be fully utilized.

The following are recommendations for achieving peak performance:

- In cases where reads and writes occur with near equal frequency,
  - The SIO architecture is preferred;
  - Use burst length = 8;
  - For RLDRAM II CIO and RLDRAM I, use multiple read or write burst commands;
  - For RLDRAM II SIO, use single interleaved read and write burst commands.
- In cases where accesses are mostly reads or mostly writes,
  - The CIO architecture is preferred;
  - Use burst length = 8;
  - For RLDRAM II CIO, RLDRAM II SIO and RLDRAM I, use multiple read/write burst commands.

Table 5. Peak Bandwidth Summary

Memory Type	Clock Speed (MHz)	Peak Bandwidth (Gbps)	Address Width	Data Width	Total I/Os	I/O Type	Memory Density	Vendors
RLDRAM I	300 DDR	19.54	20	36	83	HSTL (1.5V or 1.8V Class I)	256 Mb	Micron
RLDRAM II CIO	400 DDR	26.07	23	36	86		256 Mb, 288 Mb, 576 Mb	
RLDRAM II SIO		26.16		18	81			

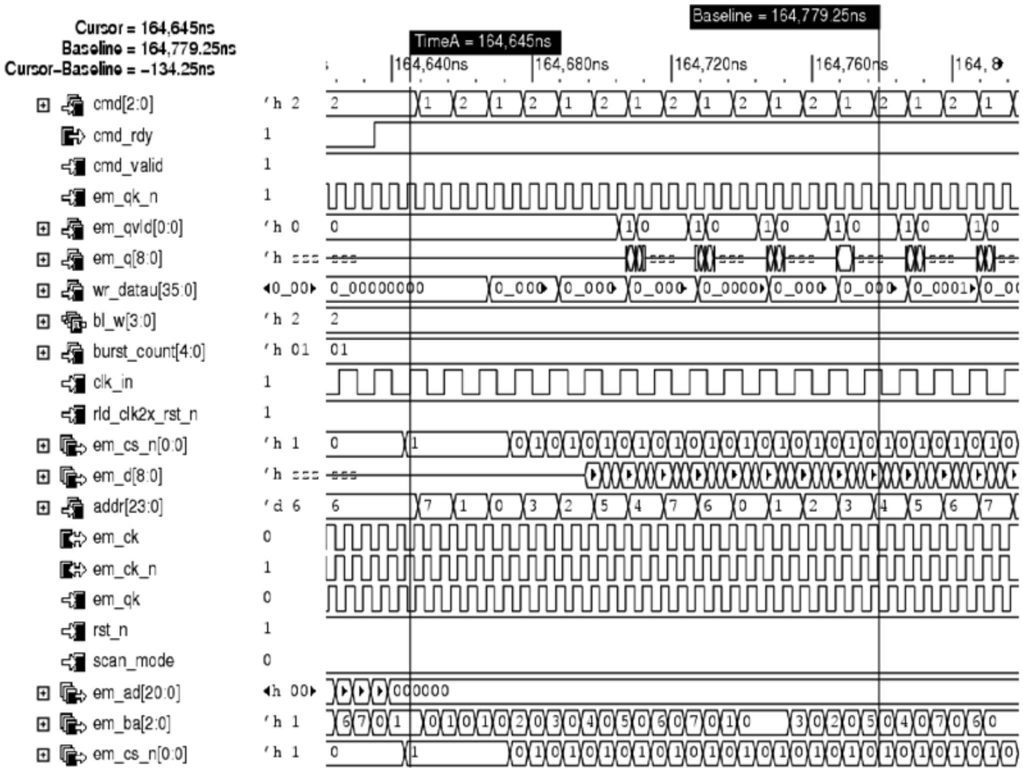
Bus Saturation

Figure 6 shows an example of bus saturation for burst length 2, data width 9, RLDRAM II SIO device. The signal cmd_rdy stays high, and the user does a back-to-back WRITE followed by READ commands to banks 0,1,2, ... 7 and back to bank 0.

Each back-to-back write or read must be done to a different bank.

In the case of burst length 8, the user cannot saturate the bus, since it takes two clocks for one command. Once the pipelines are filled, the user can give one command per two clocks. For burst length 4 as well burst length 2, saturation can be achieved as shown in Figure 6.

Figure 6. Bus Saturation Example



Command Application Logic

This command application logic module receives input from the configuration interface as well as the command decode logic. The commands presented by the decode logic are applied to the memory in the order received. This module takes does the address muxing, if required, for RLDRAM II CIO and RLDRAM II SIO devices.

The controller supports a burst mode of command execution where the user provides a base address and a burst count. The read or write command is then executed as many times as set at the burst_count[4:0] signal. The row address is fixed for every single burst while the column address is incremented. If the column address happens to reach the page boundary, it wraps around to the beginning of the same page. The controller supports a burst count of up to 31.

Data I/O Module

This module is part of the soft IP. Data I/O interfaces with the user logic and I/O pads for transferring data between the two interfaces. This module transfers write data from user to memory, and read data from memory to user. During a write operation, it converts user data, available with respect to clk to clk_2x dual data rate. During a read operation, dual data rate (clocked by clk2x) will be converted to single data rate (clocked by clk) and provided at the user interface. Data I/O will be synthesized to FPGA LUTs

Command I/O Module

This module is also part of soft IP. Command I/O interfaces with the RLDRAM MACO and I/O pads for transmitting the RLDRAM command to the memory device. I/O pads for command use clk_2x. Command I/O will be synthesized to the FPGA LUTs.

PLL

Input for the PLL is clk. PLL outputs two clocks, called clk_2x (2*clk) and clk_2x_90.

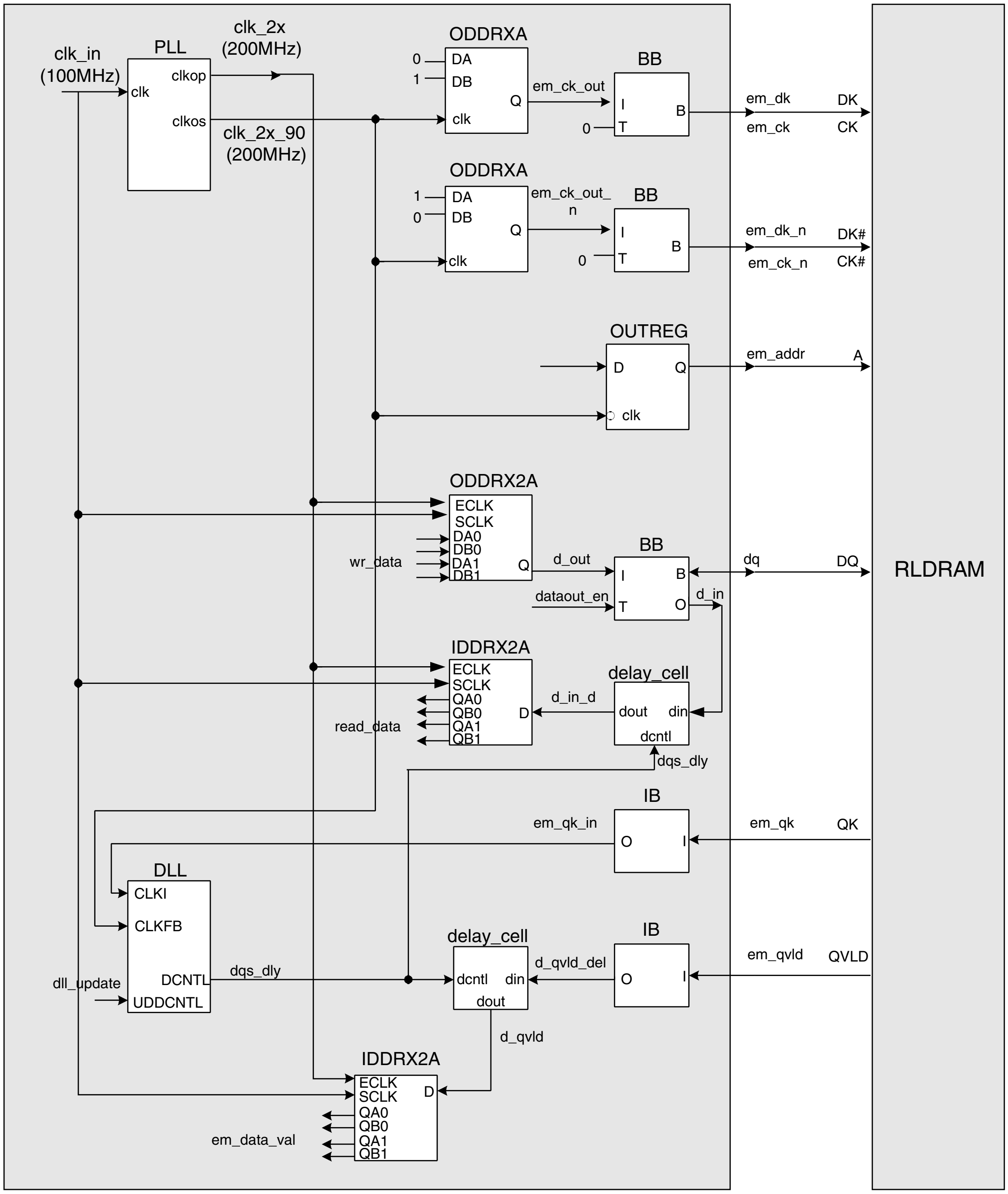
Clk_2x_90 is 90-degree phase shifted from clk_2x.

Clocking

Figure 10 illustrates the clocking scheme used for the MACO RLDRAM Memory Controller core. The key clocks include:

- RLDRAM I and RLDRAM II devices use CK and CK# signals to clock the command and address bus on a single edge (single data rate)
- RLDRAM I and RLDRAM II use a free-running QK,QK# clock pair for read data.
- RLDRAM I uses the system clock CK, CK# for timing the write data. RLDRAM II uses the DK, DK# clock pair for write data.
- The clock clk_in is used to drive much of the MACO IP core and the data bus pins. In order to transfer date on/off chip at a 2x rate, two clocks are generated by the PLL, clk_2x and clk_2x_90 (90 degree phase shifted from clk_2x). Clk_in and clk_2x are phase aligned at the PLL in order to facilitate transfers between all three clocks.

Figure 7. Clocking Scheme

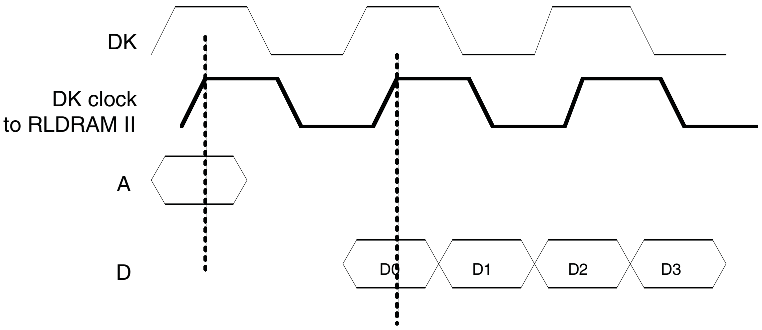


Write Data Timing

For RLDRAM I devices, command clock and write data clock are the same. For RLDRAM II devices (both SIO and CIO), command clock (em_ck) and write data clock (em_dk) are different. They could have a maximum phase difference of t_{CKDK} (500ps). As a result, write data clock and command clock should be same.

When interfacing to RLDRAM II devices, the write output clock DK should be phase shifted by 90 degrees to center it to the write data. When interfacing to RLDRAM I devices, the system clock CK should be centered to the write data DQ. This can be done by using a PLL to generate the 90 degree phase-shifted clock. This is shown in Figure 8.

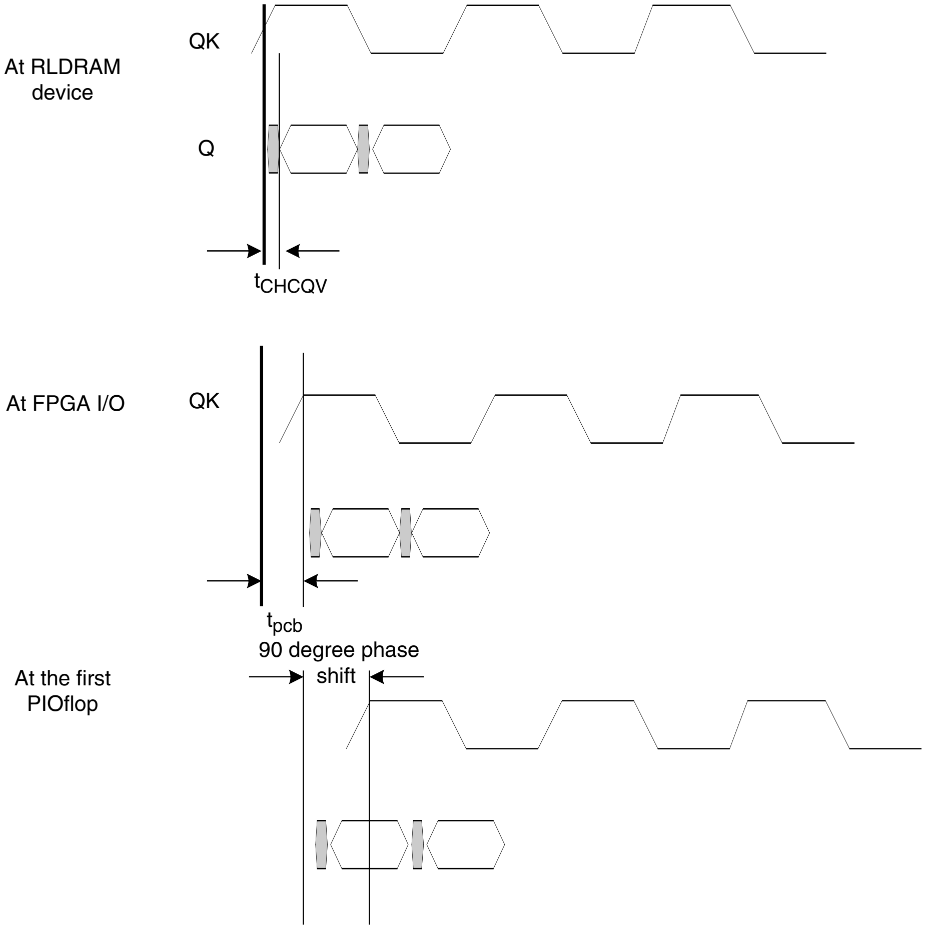
Figure 8. Write Data Timing



Read Data Timing

Since Q data and QK clock are edge-aligned coming from the RLDRAM devices, QK needs to be delayed (ideally centered to Q) to effectively capture the data. The controller uses the DLL to delay the QK signals by 90 degrees, and that gives the greatest timing margin over PVT. The QK pin is then sent to a clock divider to route the fast input clock to the IDDRX2 elements via an edge clock. The clock divider also provides the phase aligned 1/2 speed input clock routed on a primary clock.

Figure 9. Read Data Timing



RLDRAM Design Kit Directory

The directory structure of the RLDRAM IP, generated by the IPexpress GUI, is shown in Figure 10.

A more detailed description of the files generated, as well as information on installation, functional simulation, synthesis, design implementation and timing simulation, is given in the Readme file.

The Readme file can be invoked by IPexpress as shown in Figure 11. It can also be found in the rldram_controller_eval directory.

Figure 10. RLDRAM Design Kit Directory Structure

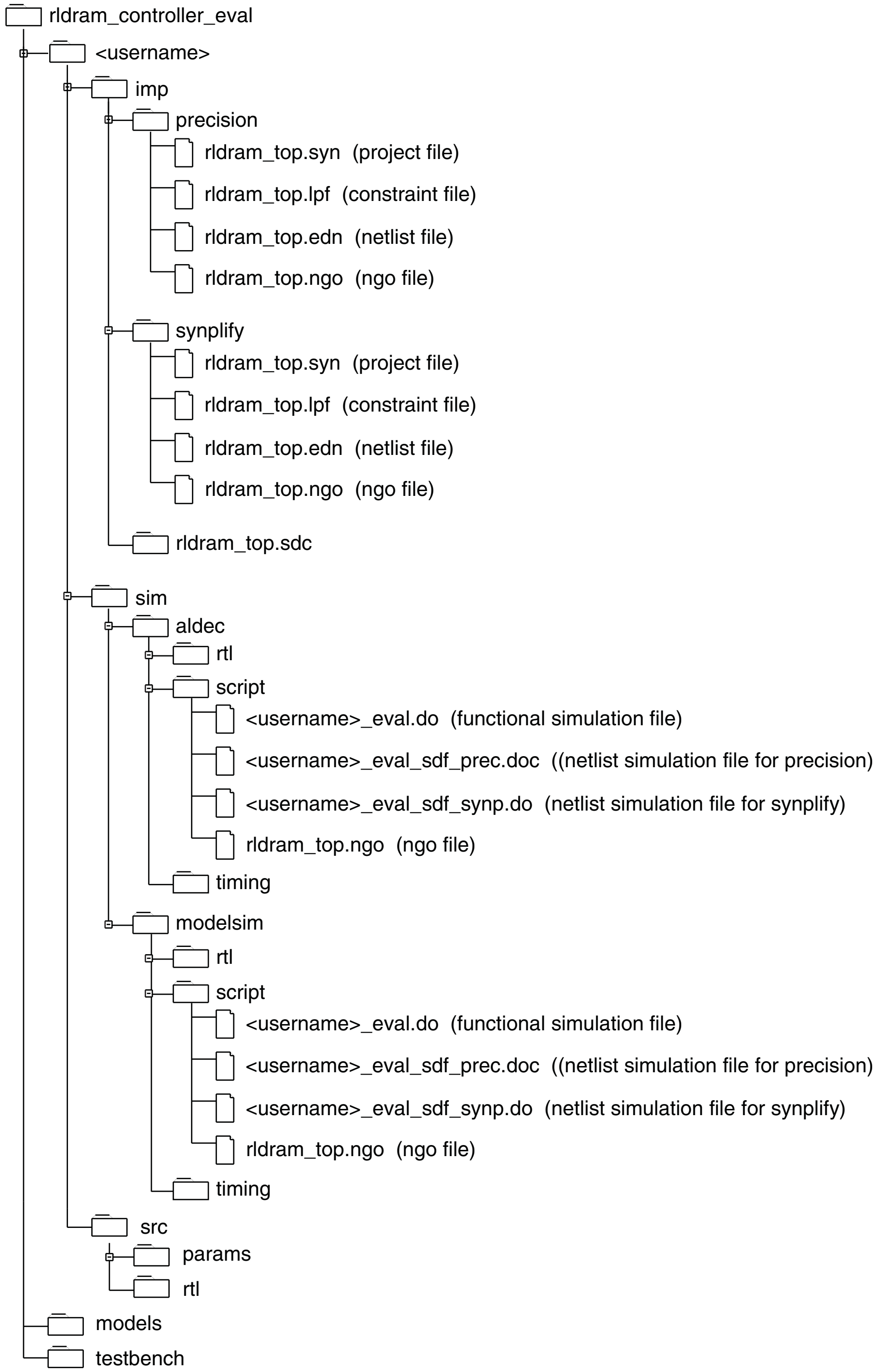
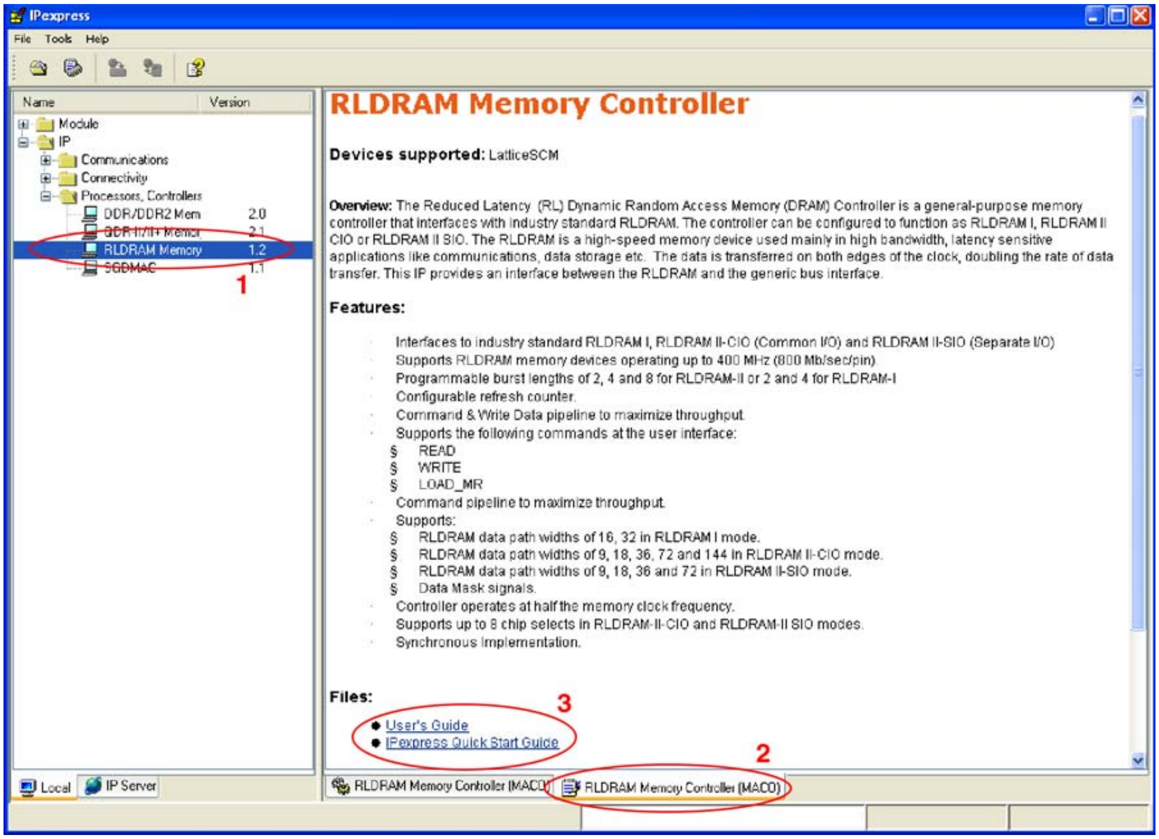

rldram_controller_eval
<username>
imp
precision
rldram_top.syn (project file)
rldram_top.lpf (constraint file)
rldram_top.edn (netlist file)
rldram_top.ngo (ngo file)
synplify
rldram_top.syn (project file)
rldram_top.lpf (constraint file)
rldram_top.edn (netlist file)
rldram_top.ngo (ngo file)
rldram_top.sdc
sim
aldec
rtl
script
<username>_eval.do (functional simulation file)
<username>_eval_sdf_prec.doc ((netlist simulation file for precision)
<username>_eval_sdf_synp.do (netlist simulation file for synplify)
rldram_top.ngo (ngo file)
timing
modelsim
rtl
script
<username>_eval.do (functional simulation file)
<username>_eval_sdf_prec.doc ((netlist simulation file for precision)
<username>_eval_sdf_synp.do (netlist simulation file for synplify)
rldram_top.ngo (ngo file)
timing
src
params
rtl
models
testbench

Figure 11. RLDRAM Memory Controller GUI Information and Readme File



RLDRAM Memory Controller

Devices supported: LatticeSCM

Overview: The Reduced Latency (RL) Dynamic Random Access Memory (DRAM) Controller is a general-purpose memory controller that interfaces with industry standard RLDRAM. The controller can be configured to function as RLDRAM I, RLDRAM II CIO or RLDRAM II SIO. The RLDRAM is a high-speed memory device used mainly in high bandwidth, latency sensitive applications like communications, data storage etc. The data is transferred on both edges of the clock, doubling the rate of data transfer. This IP provides an interface between the RLDRAM and the generic bus interface.

Features:

- Interfaces to industry standard RLDRAM I, RLDRAM II-CIO (Common I/O) and RLDRAM II-SIO (Separate I/O)
- Supports RLDRAM memory devices operating up to 400 MHz (800 Mb/sec/pin).
- Programmable burst lengths of 2, 4 and 8 for RLDRAM-II or 2 and 4 for RLDRAM-I
- Configurable refresh counter.
- Command & Write Data pipeline to maximize throughput.
- Supports the following commands at the user interface:
  - READ
  - WRITE
  - LOAD_MR
- Command pipeline to maximize throughput.
- Supports:
  - RLDRAM data path widths of 16, 32 in RLDRAM I mode.
  - RLDRAM data path widths of 9, 18, 36, 72 and 144 in RLDRAM II-CIO mode.
  - RLDRAM data path widths of 9, 18, 36 and 72 in RLDRAM II-SIO mode.
  - Data Mask signals.
- Controller operates at half the memory clock frequency.
- Supports up to 8 chip selects in RLDRAM-II-CIO and RLDRAM-II SIO modes.
- Synchronous Implementation.

Files:

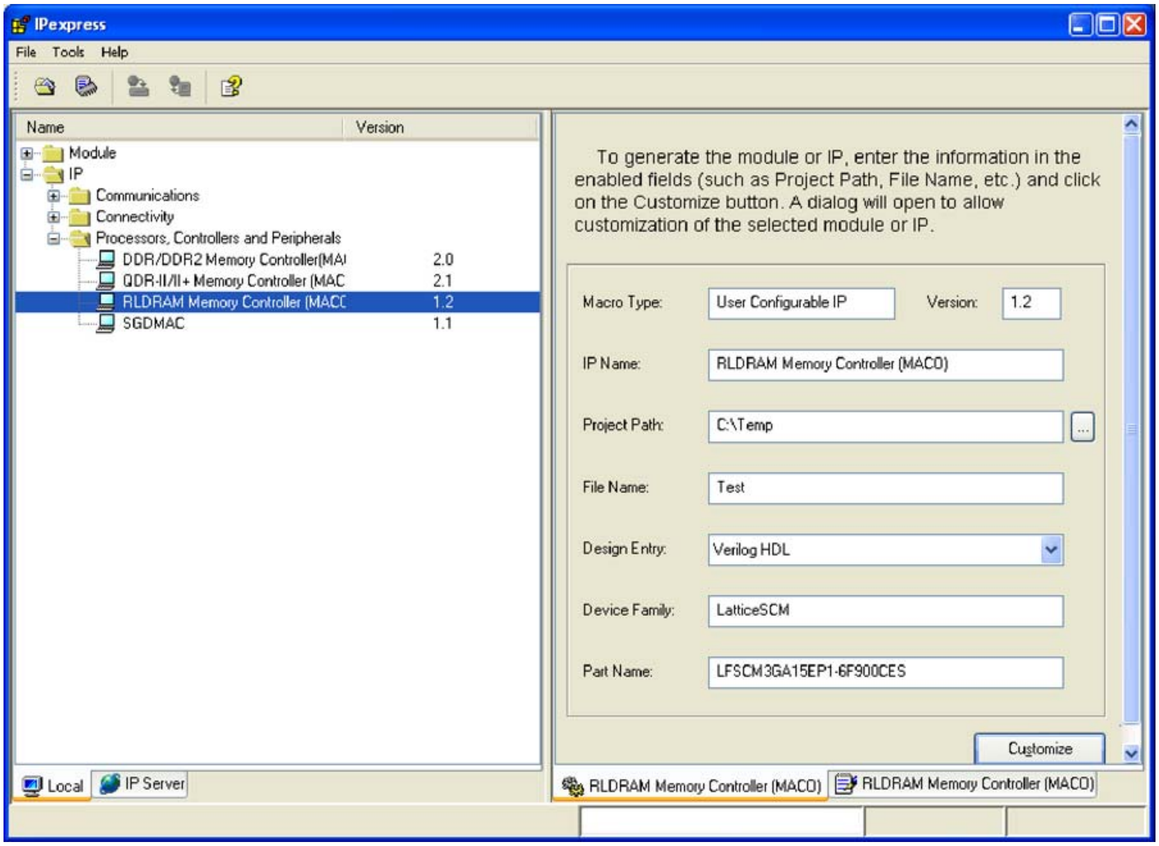
- User's Guide
- IPexpress Quick Start Guide

After selecting the RLDRAM Memory Controller IP in the IPexpress GUI, click on the About tab (right side RLDRAM Memory Controller tab) and then click on the hyperlinks to access the User's Guide (this document) and the IPexpress Quick Start Guide.

Parameter Descriptions

Figures 12 to 14 and their accompanying Tables 6 to 8 describe the parameters that can be set in the IPexpress GUI for the MACO RLDRAM Memory Controller.

Figure 12. GUI Dialog Box for RLDRAM Memory Controller


IPexpress
File
Tools
Help
Name
Version
Module
IP
Communications
Connectivity
Processors, Controllers and Peripherals
DDR/DDR2 Memory Controller(MA
2.0
QDR-II/II+ Memory Controller (MAC
2.1
RLDRAM Memory Controller (MACC
1.2
SGDMAC
1.1
Local
IP Server
Customize
RLDRAM Memory Controller (MACO)
RLDRAM Memory Controller (MACO)

To generate the module or IP, enter the information in the enabled fields (such as Project Path, File Name, etc.) and click on the Customize button. A dialog will open to allow customization of the selected module or IP.

Macro Type: User Configurable IP   Version: 1.2

IP Name: RLDRAM Memory Controller (MACO)

Project Path: C:\Temp

File Name: Test

Design Entry: Verilog HDL

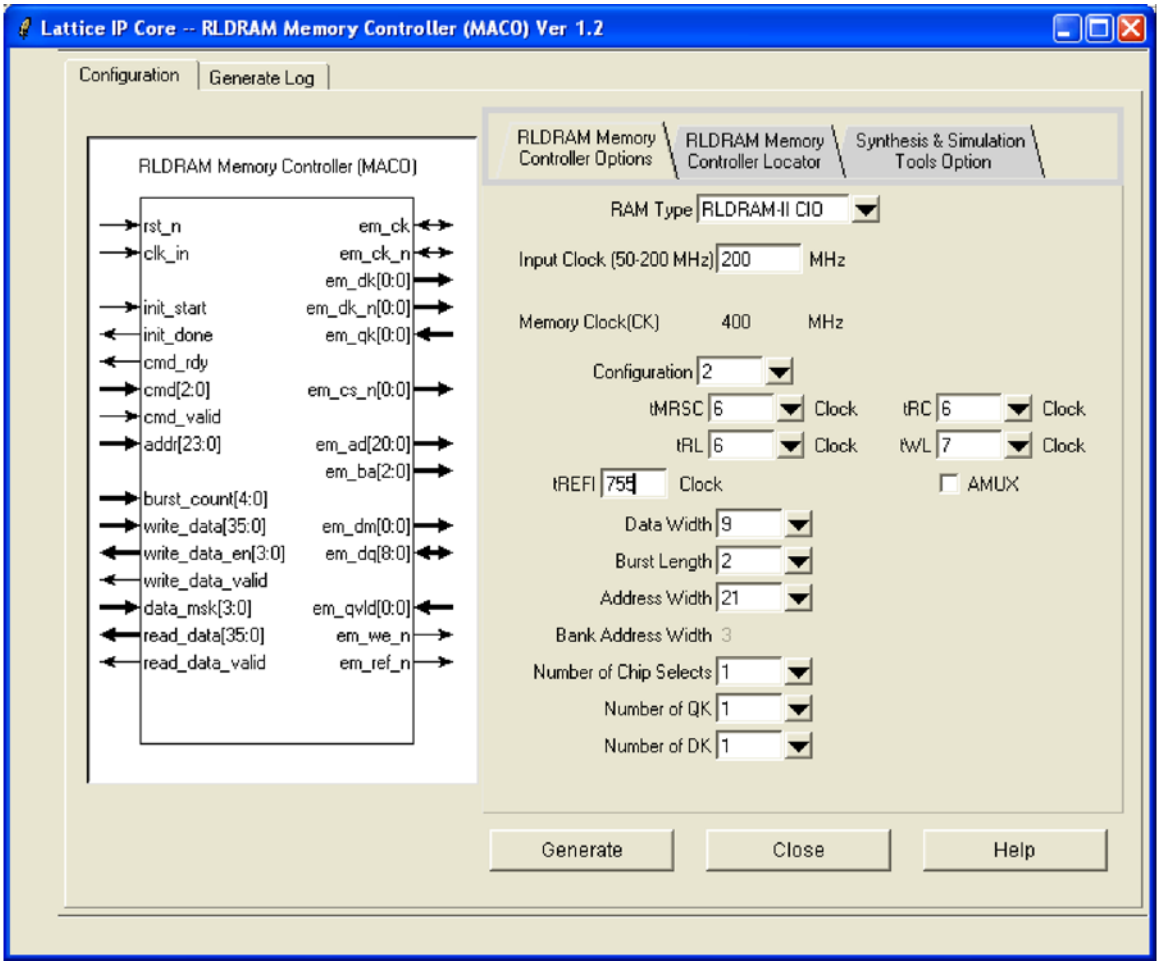
Device Family: LatticeSCM

Part Name: LFSCM3GA15EP1-6F900CES

Table 6. GUI Dialog Box for RLDRAM Memory Controller

Parameter	Description
Project Path	This is the directory in which the project will be generated
File Name	Enter the project name
Design Entry	The design entry is verilog HDL
Device Family	The device family is LatticeSCM
Part Name	Select the part as per required speed grade and package

Figure 13. GUI Dialog Box for RLDRAM Memory Controller Options



RAM Type RLDRAM-II CIO

Input Clock (50-200 MHz) 200 MHz

Memory Clock(CK) 400 MHz

Configuration 2

tMRSC 6 Clock   tRC 6 Clock

tRL 6 Clock   tWL 7 Clock

tREFI 755 Clock   □ AMUX

Data Width 9

Burst Length 2

Address Width 21

Bank Address Width 3

Number of Chip Selects 1

Number of QK 1

Number of DK 1

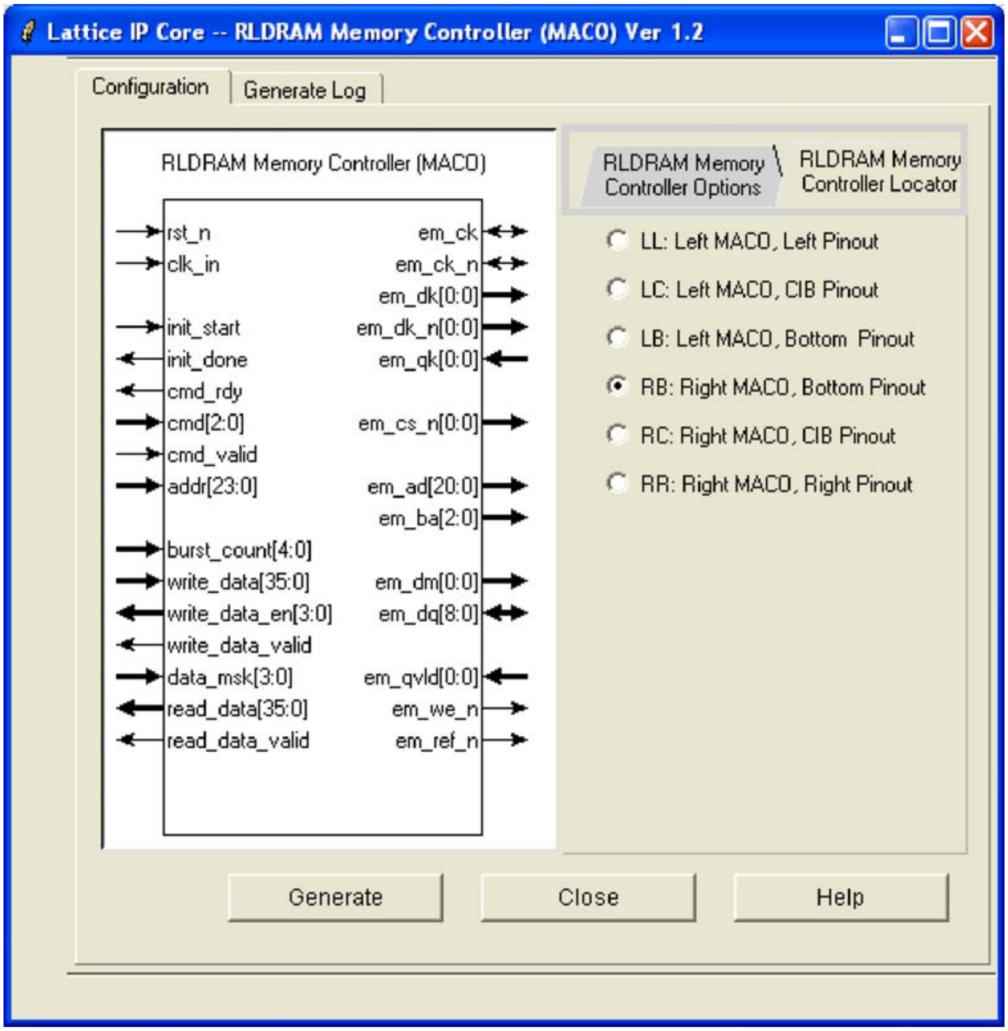
Table 7. GUI Dialog Box for RLDRAM Memory Controller Options

Parameter	Description
RAM Type	Select the type of RLDRAM controller. Choices are RLDRAM I, RLDRAM II CIO, or RLDRAM II SIO for RLDRAM II CIO/SIO or 50-150 for RLDRAM I.
Input Clock	Specify the frequency of the input clock to the memory controller. Value range is 50 to 200 MHz.
Memory Clock	The frequency of the memory side clock is always 2 * input clock frequency.
Configuration	Specify the RLDRAM Memory timing configuration. Choice of configurations for RLDRAM I: 1, 2, 3, 4 and for RLDRAM II CIO/SIO: 1, 2, 3.
t_{MRSC}	Mode Register Set Cycle time in terms of number of clock cycles. Available choices for RLDRAM I: 4, 5, 6, 7 and for RLDRAM II CIO/SIO: 6, 7.
t_{RC}	Row Cycle time in terms of number of clock cycles. Available choices for RLDRAM I: 5, 6, 7, 8 and for RLDRAM II CIO/SIO: 4, 5, 6, 7, 8.
t_{RL}	Read Latency time in terms of number of clock cycles. Available choices for RLDRAM I: 5, 6 and for RLDRAM II CIO/SIO: 4, 6, 8, 9.
t_{WL}	Write Latency time in terms of number of clock cycles. Available choices for RLDRAM I: 1, 2, 3 and for RLDRAM II CIO/SIO: 5, 6, 7, 8, 9, 10.
t_{REFI}	Memory Refresh Interval cycle time in terms of the number of clock cycles.
AMUX	Address MUX mode selection. Valid only for RLDRAM II CIO/SIO.
Data Width	Memory data width. Available choices for RLDRAM I: 16, 32, for RLDRAM II CIO: 9,18, 36, 72, 144 and for RLDRAM II SIO: 9, 18, 36, 72.

Table 7. GUI Dialog Box for RLDRAM Memory Controller Options (Continued)

Parameter	Description
Burst Length	Memory burst length. Available choices for RLDRAM I are 2 or 4. Available choices for RLDRAM II CIO/SIO are 2, 4, 8.
Address Width	Memory address width. Available choices 1-29.
Number of Chip Selects	Number of chip selects to the memory module. RLDRAM will have a value of 1. Available choices for RLDRAM II CIO/SIO are 1-8.
Number of QK	Number of input data clocks to the controller from memory module. RLDRAM I can only have a value of 1. Available choices for RLDRAM II CIO/SIO are 1, 2, 4, 8. The number of QKs selected will determine the number of DLLs that the controller will use.
Number of DK	Number of output data clocks from the controller to the memory module. Available choices 1-8.

Figure 14. GUI Dialog Box for RLDRAM Memory Controller Locator



○ LL: Left MACO, Left Pinout

○ LC: Left MACO, CIB Pinout

○ LB: Left MACO, Bottom Pinout

● RB: Right MACO, Bottom Pinout

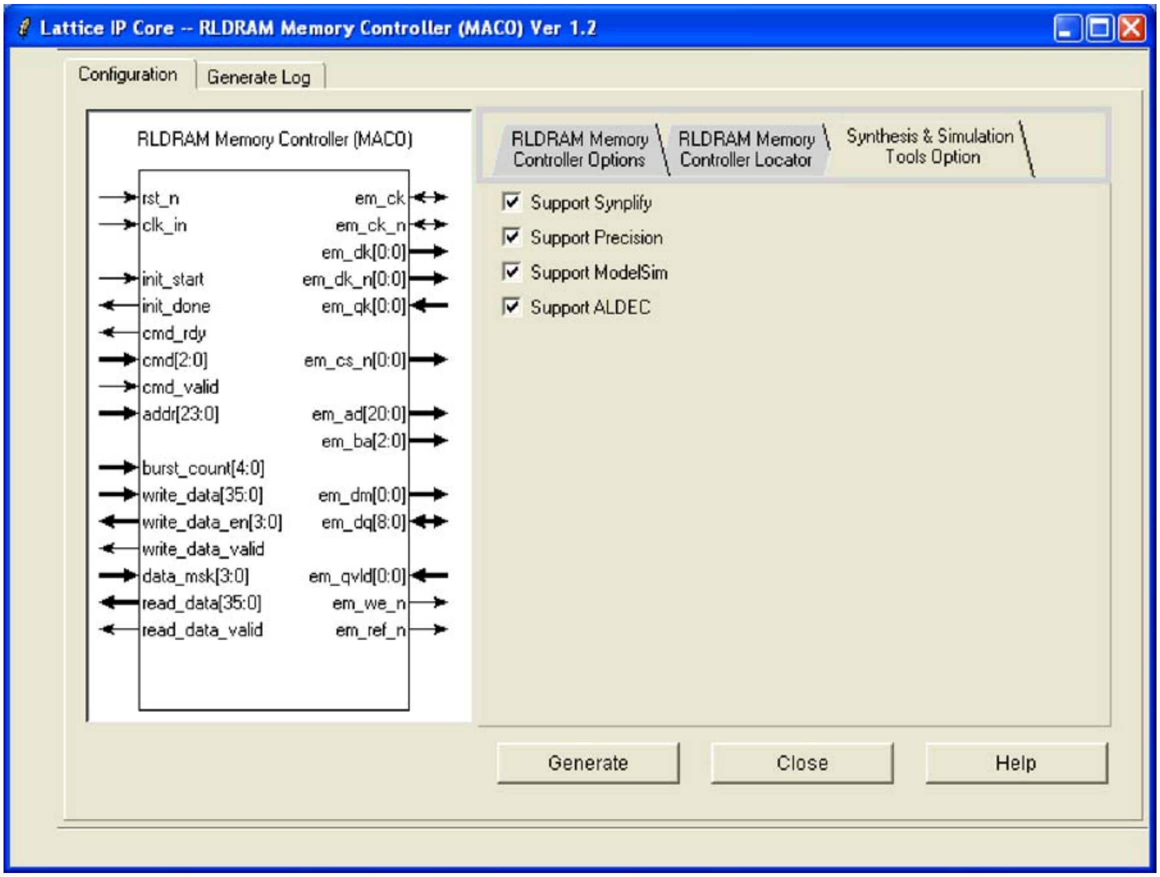
○ RC: Right MACO, CIB Pinout

○ RR: Right MACO, Right Pinout

Table 8. GUI Dialog Box for RLDRAM Memory Controller Locator

Parameter	Description
LL: Left MACO, Left Pinout	The left-side MACO used for the RLDRAM controller, and the pinout is on the left side.
LC: Left MACO, CIB Pinout	The left-side MACO used for the RLDRAM controller, and the pinout is CIB.
LB: Left MACO, Bottom Pinout	The left-side MACO used for the RLDRAM controller, and the pinout is on the bottom side.
RB: Right MACO, Bottom Pinout	The right-side MACO used for the RLDRAM controller, and the pinout is on the bottom side.
RC: Right MACO, CIB Pinout	The right-side MACO used for the RLDRAM controller, and the pinout is CIB.
RR: Right MACO, Right Pinout	The right-side MACO used for the RLDRAM controller, and the pinout is on the right side.

Figure 15.



Lattice IP Core -- RLDRAM Memory Controller (MACO) Ver 1.2

☑ Support Synplify

☑ Support Precision

☑ Support ModelSim

☑ Support ALDEC

Table 9.

Parameter	Description
Support Synplify	Creates rldram_controller_eval\<username>\impl\synplify which contains the eval project files when using Synplify.
Support Precision	Creates rldram_controller_eval\<username>\impl\precision which contains the eval project files when using Precision
Support ModelSim	Creates rldram_controller_eval\<username>\sim\modelsim which contains the eval project files to be used for a ModelSim simulation.
Support ALDEC	Creates rldram_controller_eval\<username>\sim\aldec which contains the eval project files to be used for an Aldec Active-HDL simulation.

Implementation Details

The following section discusses implementation details, such as pinout selection, clock, PLL and DLL considerations, as well as PCB layout guidelines for optimum performance.

PCB Layout and On-Chip Pinout Considerations

This section discusses some areas of the RLDRAM Memory Controller design that require particular attention, and offers recommendations that will lead to a more robust solution.

Master Clock and its PLL

- The Master Clock can originate from a variety of sources (input pin, another PLL, SERDES clock, FPGA logic, etc.). This clock drives a PLL via any primary clock net. The PLL can be any PLL that can directly drive all RLDRAM I/O pins via edge clocks. This requires that the PLL be placed in the corner of the device adjacent to the bank(s) with all RLDRAM I/O. If the Master Clock is driven by an input pin, refer to Table 10 for the input pin designated for the selected PLL.
- The PLL outputs, clk_2x and clk_2x_90, are used by the high-speed I/O logic for all data and control signals for the write interface to the memory, and to drive the DK, DK#, CK and CK# clocks off-chip (along with any derived copies for multiple lanes). These off-chip destinations are driven via edge clocks.
- For each RLDRAM device, only one of the QK clocks from the RLDRAM is used to clock both read lanes into the FPGA. This means that a DLL is used per pair of lanes into the FPGA. This may limit the pin density per bank due to the availability of a maximum of two DCNTRL buses per bank. It is possible to modify the clock plan to increase this pin density for RLDRAM SIO or slower-speed CIO operation. Contact your local Lattice sales representative for more information.
- If the Master Clock is sourced by an input pin (or pin pair), use the pin(s) designated for the chosen PLL for that purpose (refer to Table 10), and observe the recommendations below for minimizing jitter noise.

PCB Board Trace Matching

- All DK, DK# and D (DQ for CIO) pins must have PCB board trace lengths matched to within 50 psec within a lane, and 100 psec across lanes.
- Similarly, all QK, QK#, Q (DQ for CIO) and QVLD pins must have PCB board trace lengths matched to within 50 psec within a lane, and 100 psec across lanes.
- All CK, CK#, Address and Control pins must have PCB board trace lengths matched to within 50 psec.

Other Board-Level Considerations

- All dynamic signal traces must be 50 Ohm transmission lines.
- All power signals, including any VTT power, must be supplied by planes, not traces.
- Care must be taken to keep reference voltages, such as the RLDRAM's VREF pin, noise-free.This involves robust, wide-bandwidth decoupling, and isolation of quiet, noise-sensitive signals from noise sources.
- The physical distance between the LatticeSC device and the RLDRAM needs to be minimized, since trace delays, skews and signal degradation will limit overall speed, as previously discussed.

DLL

The DLL is used in CIMDLLA mode to determine the phase difference between QK and clk_2x_90, which is then sent via the DCNTRL bus to the INDEL of all Q pins. The Q pins are then latched by clk_2x, producing an effective 90° phase shift before capture.

To achieve maximum performance, it may be necessary to adjust the DLL's ALU function ±1 tap in order to center QK in the Q eye. The optimum setting should be determined experimentally, using the final board layout. This setting can then be used for all production units using that board layout.

Since this DLL drives the DCNTRL bus, it must be located in the corner of the device adjacent to the Q signal banks.

The clock to the DLL is driven by QK, so this signal must be located on the designated input pin that provides an optimized path into the DLL (refer to Table 11).

The preference file generated from the design kit provides an example implementation consisting of a single DLL. For configurations consisting of multiple DLLs, the user must locate the remaining DLLs in the preference file and must locate the corresponding QK pin as described above. The instance name of the DLL for the LOCATE preference can be obtained from the Map Report File (*.mrp).

Selecting a Pin That Has Low Jitter Noise

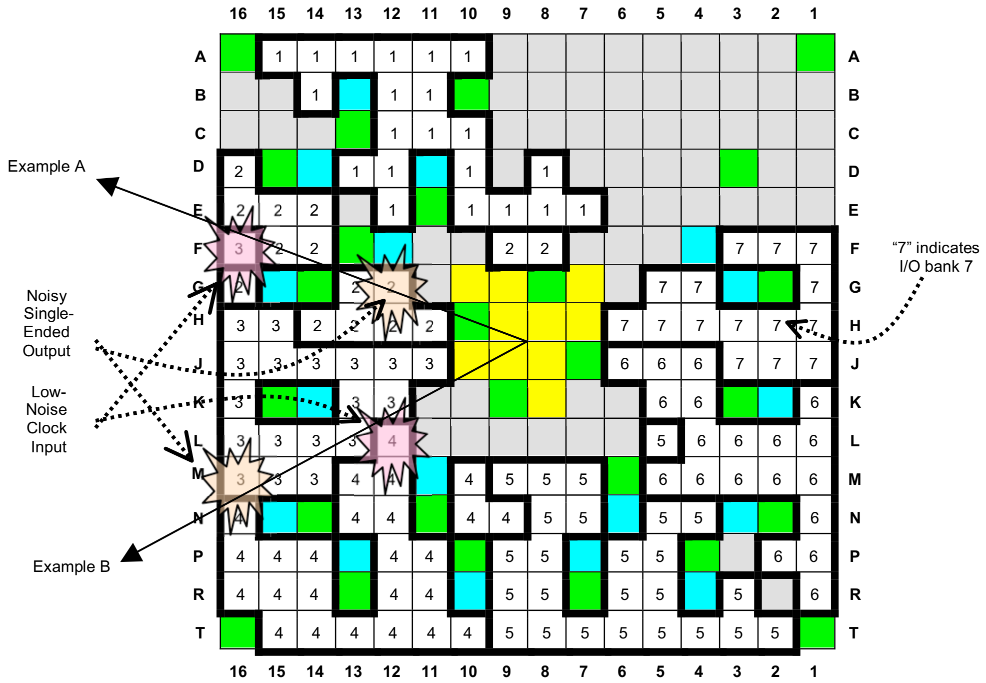
When a signal, such as an input clock or the RLDRAM clocks DK/DK#, CK/CK# and QK, needs to be especially quiet with low-jitter, some special design rules can help achieve this goal:

- It is highly preferable to place the pin in a bank that does not also contain single-ended output drivers. Figure 16 shows how bank groups form clusters around the package, in this case for a 256-pin fpBGA.
- If a quiet bank cannot be used, avoid creating inductively coupled paths linked to noisy signals on the package. These occur when the low-noise signal trace passes through an area on the package substrate from pin to pad that contains noisy signal pins or traces (in particular single-ended outputs, and especially when those single-ended outputs are unterminated).

  Figure 16 also illustrates this concept. Two examples are shown:

  - Example A shows a noisy output pin (G12, bank 2) that is near the package center, and a low-noise clock pin (F16, bank 3) that is situated radially outward from that pin. In this case, the pin-to-pad connection for the clock will route directly past the noisy output pin, resulting in coupled noise. This should be avoided.
  - Example B demonstrates the reverse situation, which is also to be avoided. In this case, a noisy output pin (M16, bank 3) is situated radially outward from a low-noise clock pin (L12, bank 4), so that the noisy output's pad-to-pin connection will pass over the clock pin.
  - In order to minimize this coupling, it is typically better to place noise-sensitive pins toward the center of the package. This reduces the trace length of this signal in the package, thus reducing coupling to this signal.
- Noise immunity may be further enhanced by providing extra "ground" pins around the sensitive signal, by driving adjacent outputs to a constant LO and tying them to signal ground on the PCB. This can enhance noise immunity in two ways: first, it provides extra signal current return paths, and second, it provides a buffer distance to nearby signal pins, thus reducing coupling to their signals. The buffers should be set to the maximum drive strength allowed at the bank's VCCIO voltage.

Figure 16. Selecting a Pin for Low Jitter Noise



Optimum Pinout Selection

In order to ensure that the demanding I/O timing requirements of RLDRAM devices will always be met, dedicated signal paths from the MACO core to the I/O pins have been designed into the LatticeSCM devices. If the designer chooses to use these optimized locations, I/O timing can be guaranteed, and will not change each time the device undergoes map/place/route. These designated pinout assignments are given in Tables 12 (for the left-side MACO) and 13 (for the right-side MACO). In addition, some flexibility has been provided by offering two sets of locations, one on the side edge and one on the bottom edge, so that conflicts with other pinout placement requirements can be resolved. Note that these special routings only apply to the signals that connect to the MACO block (address and control); other signals (data and their clocks) have more freedom of placement, restricted only by the need to place complete lanes in a single I/O bank adjacent to the PLLs and DLLs, as described previously.

In addition to the two pinout options described above, a third option is provided that interfaces the signals to the general FPGA routing fabric. This allows the signals to be routed to any pin, or even to FPGA logic, albeit at the penalty of additional and variable routing delay. This option should only be considered when the RLDRAM Memory Controller is being operated well below its maximum operating frequency.

General Considerations

- Lattice recommends simulation of Simultaneous Switching Outputs (SSOs) for the device/package combination for performance targeted to over 200 MHz.
- Lattice recommends that the LatticeSC device's design be placed and routed before commitment of the board design to manufacture.

Below is a example of the parameters file generated by IPexpress for an RLDRAM memory application. This file incorporates the user's design-specific information that is processed for the HDL generation.

```
`define SPEED_400
`define RLDRAM2_CIO
`define MEM_DATA_WIDTH_72
`define MEM_DATA_WIDTH 72
```

```
`define USER_CONFIG_2
`define TMRSC 6
`define TRC 6
`define TRL 6
`define TWL 7
`define TREFI 365
`define BL2
`define RLD_USER_ASIZE 23
`define BANK_WIDTH 3
`define USER_NUM_CS_1
`define QK_WIDTH 1
`define QK_WIDTH_1
`define DK_WIDTH 1
`define MACO_LEFT
`define PINOUT_BOTTOM
```

Setting Design Timing Constraints

In order to ensure that a design will meet a specific speed requirement, the requirement must be called out as a preference in the *.lpf file. The design kit gives an example of how this is done, and the values simply need to be adjusted to meet the specific design's requirements.

Note that the internal name of a clock net can change if the design is modified or if the synthesis engine version is changed. In this case, the net names given in the design example will not be correct. To find the new net name, run the synthesis flow through the map phase, and inspect the Map Report (*.mrp) file. It will list all the clock nets that the mapper detected. Find the new net name in question and put it in the preference file in place of the old name.

Preferred Pinouts

The tables below show connections from I/O to logic that have been designed-in to be fast and consistent, so that special signals such as clocks and timing-critical I/O can be guaranteed to always meet requirements.

Tables 10 and 11 give the designated pins for driving the PLLs and DLLs respectively. This information is extracted from the pinout tables in the LatticeSC Family Data Sheet.

Tables 12 and 13 show the designated optimum-performance pins for interfacing the RLDRAM Memory Controller to the RLDRAM device, for the left-side and right-side MACO respectively.

Table 10. PLL Direct Input Pins (True/Complement Pair)

	F900	FF1020	FC1152	FC1704
ULC PLL A	D3/D2	K25/J25	F30/G30	J37/J38
ULC PLL B	K4/J4	M23/N23	N25/P25	N33/P33
LLC PLL B	AC6/AC7	AC23/AD24	AG29/AG28	AN36/AP36
LLC PLL A	AH1/AJ1	AJ32/AK32	AM33/AN33	AU42/AV42
LRC PLL A	AJ30/AH30	AJ1/AK1	AN2/AM2	AV1/AU1
LRC PLL B	AD26/AC25	AC10/AD9	AG6/AG7	AN7/AP7
URC PLL B	K25/K26	M10/N10	N10/P10	N10/P10
URC PLL A	D28/E28	K8/J8	F5/G5	J6/J5

Table 11. DLL Direct Input Pins (True/Complement Pair)

	F900	FF1020	FC1152	FC1704
ULC DLL C	E3/E2	D32/D31	F31/G31	G40/H40
ULC DLL D	F3/G3	E32/E31	D33/E33	G41/H41
LLC DLL E	AB6/AC5	AE26/AE27	AJ30/AK30	AL37/AM37
LLC DLL F	AF2/AG2	AG32/AG31	AL32/AL31	AR39/AR40
LLC DLL C	AF4/AE5	AF27/AG28	AH29/AJ29	AL33/AL34
LLC DLL D	AG3/AH2	AK31/AL31	AM32/AM31	AU38/AV38
LRC DLL C	AJ29/AH29	AL2/AK2	AM3/AM4	AV2/AW2
LRC DLL D	AG28/AG29	AJ2/AH3	AJ6/AH6	AL10/AL9
LRC DLL F	AF29/AF28	AG1/AG2	AL3/AL4	AR4/AR3
LRC DLL E	AB26/AC26	AE7/AE6	AJ5/AK5	AL6/AM6
URC DLL D	G28/F28	E1/E2	D2/E2	G2/H2
URC DLL C	D29/D30	D1/D2	F4/G4	G3/H3

Table 12. Preferred Pinout for Left Side Memory Controller

RLDRAM Port	Bottom Edge Preferred Pinout				Left Edge Preferred Pinout			
	SC25 900	All 1020	All 1152	All 1704	SC25 900	All 1020	All 1152	All 1704
dm[0]	AF7	AK29	AN32	AW42	AA1	Y32	AF34	AG42
dm[1]	AF6	AL29	AP32	AY42	Y1	W32	AE34	AH42
we_n	AE5	AG28	AJ29	AL34	V4	W25	AA24	AG29
as_n	AJ1	AK32	AN33	AV42	V5	Y26	Y24	AF29
ref_n	AD6	AE25	AG27	AM34	W2	Y28	AC31	AG39
ba[0]	AJ2	AK30	AL29	AV41	V2	W28	AB31	AF39
ba[1]	AK2	AL30	AL28	AW41	V6	Y27	AA27	AH36
ba[2]	AD7	AD23	AH27	AK30	U6	W27	AA26	AG36
cs_n[0]	AH2	AL31	AM31	AV38	W1	Y31	AC32	AG40
cs_n[1]	AG3	AK31	AM32	AU38	V1	W31	AB32	AF40
cs_n[2]	AK9	AM22	AN20	BA26	AC1	AC31	AF31	AK38
cs_n[3]	AG14	AL20	AK20	AV24	Y6	AD30	AJ33	AM42
cs_n[4]	AK10	AJ19	AL20	BB24	AC3	AC32	AG34	AN42
cs_n[5]	AK11	AK19	AL19	BB25	AD3	AD32	AH34	AP42
cs_n[6]	AH15	AM21	AP21	AW24	AC4	AE30	AK33	AN41
cs_n[7]	AG15	AM20	AP20	AW23	AD4	AE29	AL33	AP41
ad[0]	AH4	AJ28	AN31	AW40	U5	W29	AA33	AD39
ad[1]	AG5	AK28	AN30	AY40	U4	W30	Y33	AC39
ad[2]	AF8	AJ31	AP31	AW39	T4	V30	Y31	AB42
ad[3]	AG8	AH30	AP30	AW38	T5	V29	W31	AA42
ad[4]	AH3	AM30	AM29	AV37	U1	V31	W33	AB38
ad[5]	AJ3	AM29	AM28	AV36	T1	V32	V33	AA38
ad[6]	AF9	AH29	AJ27	AM31	V3	U31	V34	Y41
ad[7]	AE10	AH28	AJ26	AM32	U3	U32	U34	W41
ad[8]	AK3	AJ27	AP29	BA40	T6	T27	V25	AA36
ad[9]	AJ4	AK27	AP28	BB40	U2	T32	U33	Y40
ad[10]	AE11	AL28	AN29	BA39	T2	T31	T33	W40

Table 12. Preferred Pinout for Left Side Memory Controller (Continued)

RLDRAM Port	Bottom Edge Preferred Pinout				Left Edge Preferred Pinout			
	SC25 900	All 1020	All 1152	All 1704	SC25 900	All 1020	All 1152	All 1704
ad[11]	AF10	AL27	AN28	BA38	R4	U24	Y27	AC32
ad[12]	AH7	AM28	AL26	AW36	R1	R32	W30	Y39
ad[13]	AH8	AM27	AL25	AW35	P1	R31	V30	W39
ad[14]	AE12	AG23	AG23	AM28	R3	T26	V28	AB35
ad[15]	AE13	AF22	AG22	AL28	R2	R29	T34	Y38
ad[16]	AK4	AG26	AN27	AV35	P2	R30	R34	W38
ad[17]	AK5	AG25	AN26	AV34	P3	P31	U30	V42
ad[18]	AJ5	AL26	AP27	AY36	N3	P32	T30	U42
ad[19]	AJ6	AM26	AP26	AY35	R6	T24	V29	W36
ad[20]	AJ7	AJ24	AN25	AV33	W3	AC27	AD29	AK34
ad[21]	AJ8	AK24	AN24	AV32	W5	Y24	AF30	AH32
ad[22]	AH11	AK22	AP24	AY26	Y5	AD29	AH33	AL42
ad[23]	AE14	AH22	AD20	AT24	AD2	AC28	AH32	AL38
ad[24]	AK7	AL22	AL22	AW26	AE2	AD28	AJ32	AM38
ad[25]	AF14	AH23	AH24	AU29	AF1	AE31	AL34	AN38
ad[26]	AF15	AH24	AH23	AU28	AG1	AE32	AM34	AP38
ad[27]	AJ12	AK21	AM22	BB26	AA2	AA32	AG33	AK40
ad[28]	AG13	AE19	AJ24	AR25	Y3	AB27	AC29	AJ34

Table 13. Preferred Pinout for Right Side RLDRAM Memory Controller

RLDRAM Port	Bottom Edge Preferred Pinout					Right Edge Preferred Pinout				
	SC15 900	SC25 900	All 1020	All 1152	All 1704	SC15 900	SC25 900	All 1020	All 1152	All 1704
dm[0]	AE25	AF27	AL3	AL6	AV4	AB25	W29	Y1	AF1	AG1
dm[1]	AH28	AG26	AL4	AL7	AV3	AD30	V29	W1	AE1	AH1
we_n	AD25	AH30	AK1	AM2	AU1	Y30	W26	W8	AA11	AG14
as_n	AE26	AG29	AH3	AH6	AL9	AA30	V26	Y7	Y11	AF14
ref_n	AG29	AE25	AD10	AH8	AK13	AA25	U30	Y5	AC4	AG4
ba[0]	AJ28	AD25	AE8	AG8	AM9	AE30	T30	W5	AB4	AF4
ba[1]	AE22	AE26	AE9	AG9	AM10	AB28	V25	Y6	AA8	AH7
ba[2]	AK29	AH29	AK2	AM4	AW2	AC28	U25	W6	AA9	AG7
cs_n[0]	AH30	AH28	AJ3	AN4	AY1	AF30	W28	Y2	AC3	AG3
cs_n[1]	AH29	AJ28	AK3	AN5	AW1	AG30	V28	W2	AB3	AF3
cs_n[2]	AE19	AH18	AM12	AL15	AW19	AC26	AB26	AE7	AJ5	AL6
cs_n[3]	AK24	AH17	AM13	AL16	AW20	AF28	AG30	AE1	AM1	AP5
cs_n[4]	AK22	AK19	AJ15	AM15	AY19	AC25	AC27	AF1	AJ4	AR2
cs_n[5]	AJ20	AK18	AK15	AM16	AY20	AB26	AC26	AE6	AK5	AM6
cs_n[6]	AF18	AH16	AM14	AK17	AV21	AF29	AC25	AD9	AG7	AP7
cs_n[7]	AK20	AE16	AD16	AE17	AP21	AB27	AF28	AG2	AL4	AR3
ad[0]	AK28	AF25	AJ5	AN3	AW3	T30	T27	W4	AA2	AD4
ad[1]	AH21	AG25	AK5	AP3	AY3	W28	R27	W3	Y2	AC4
ad[2]	AH23	AG24	AH4	AM6	BA2	U26	V27	V3	Y4	AB1
ad[3]	AH22	AF24	AH5	AM7	AY2	U28	U27	V4	W4	AA1

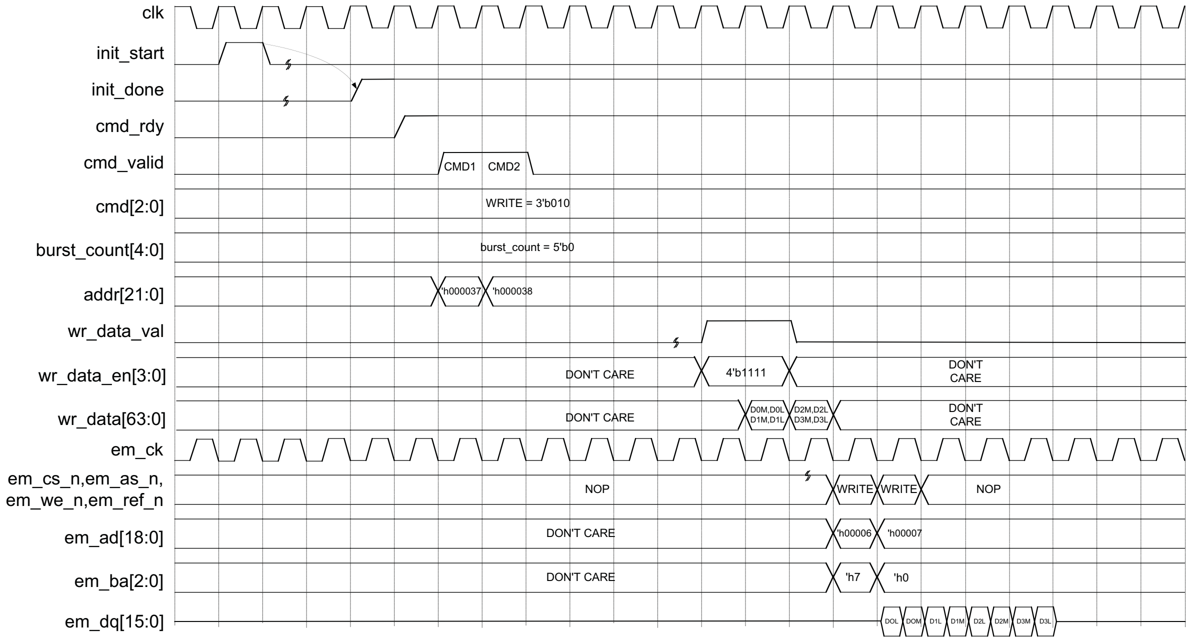
Table 13. Preferred Pinout for Right Side RLDRAM Memory Controller (Continued)

RLDRAM Port	Bottom Edge Preferred Pinout					Right Edge Preferred Pinout				
	SC15 900	SC25 900	All 1020	All 1152	All 1704	SC15 900	SC25 900	All 1020	All 1152	All 1704
ad[4]	AG22	AH27	AM3	AP4	AV6	M30	R30	V2	W2	AB5
ad[5]	AG21	AH26	AM4	AP5	AV7	R29	P30	V1	V2	AA5
ad[6]	AF21	AE22	AF10	AK9	AN11	P29	U29	U2	V1	Y2
ad[7]	AE21	AK29	AJ6	AN6	AY4	P27	T29	U1	U1	W2
ad[8]	AE20	AK28	AK6	AN7	AY5	N29	T24	T6	V10	AA7
ad[9]	AK25	AH25	AG8	AP6	BA4	N28	N30	T1	U2	Y3
ad[10]	AH19	AH24	AG7	AP7	BA5	R25	M29	T2	T2	W3
ad[11]	AK23	AE23	AL5	AN8	BB4	R28	U26	U9	Y8	AC11
ad[12]	AJ21	AD23	AL6	AN9	BB5	N27	U28	R1	W5	Y4
ad[13]	AG18	AH21	AC12	AF12	AT10	L30	T28	R2	V5	W4
ad[14]	AK21	AH23	AM5	AL9	AV8	J30	W30	AA1	AG2	AK3
ad[15]	AJ19	AH22	AM6	AL10	AV9	M26	Y27	AB6	AC6	AJ9
ad[16]	AJ18	AG22	AE12	AP8	AY7	G29	W27	AC6	AD6	AK9
ad[17]	AG17	AG21	AD12	AP9	AY8	F29	AA30	AC2	AF4	AK5
ad[18]	AH18	AF21	AJ8	AM9	AV10	H28	AA25	AD4	AH2	AL1
ad[19]	AH17	AE21	AK8	AM10	AV11	J28	AB25	AD3	AJ2	AM1
ad[20]	AK19	AE20	AH6	AD15	AU16	E30	AD30	AC5	AH3	AL5
ad[21]	AK18	AK25	AJ11	AN12	AW16	E29	AE30	AD5	AJ3	AM5
ad[22]	AG16	AH20	AH8	AH11	AP18	L25	AB28	AC1	AG1	AN1
ad[23]	AH16	AH19	AH9	AH12	AN18	F28	AC28	AD1	AH1	AP1
ad[24]	AE16	AK23	AL10	AL12	AY16	K26	AD29	AE3	AK2	AN2
ad[25]	AJ17	AK21	AL12	AN14	AV18	D30	AF30	AE2	AL1	AN5
ad[26]	AJ16	AJ19	AJ14	AP14	BB18	G26	AD28	AF2	AH4	AT2
ad[27]	AK17	AG18	AM10	AK14	BB16	H26	Y25	Y9	AF5	AH11
ad[28]	AK16	AF17	AH13	AG16	AP20	E28	AE29	AE4	AL2	AP2

Timing Specifications

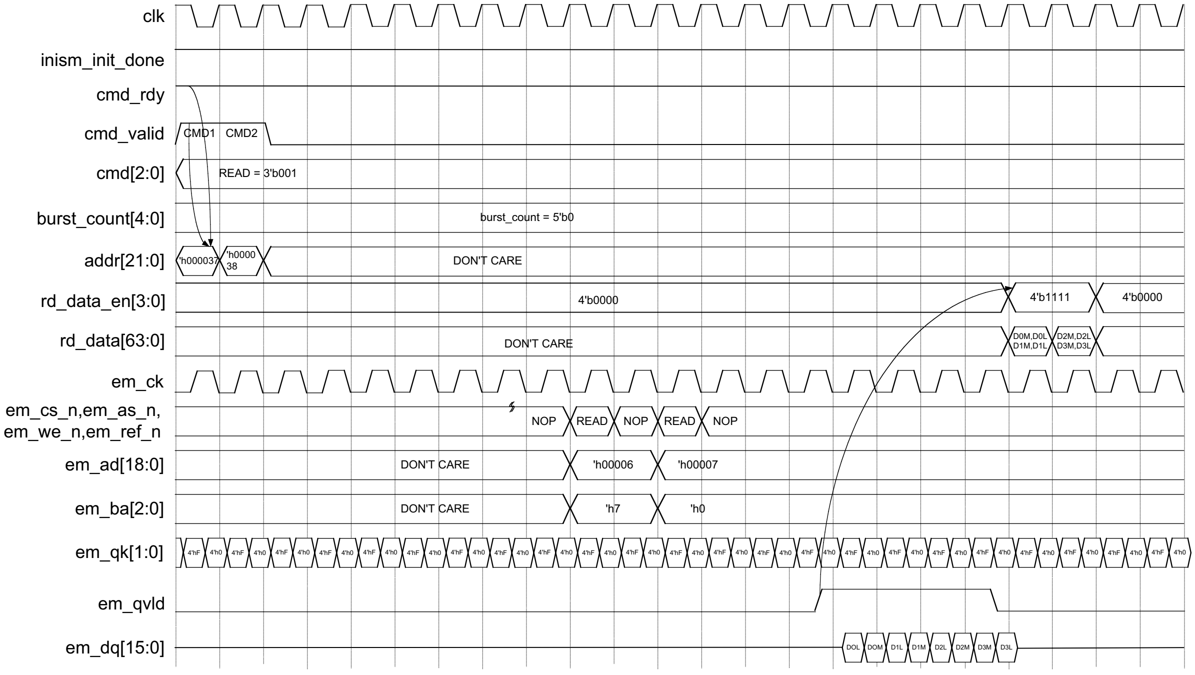
The timing diagrams below show the user interface for command and data. User interface for command and data remains constant for RLDRAM I, II CIO and II SIO. For memory interface timing diagrams, please refer to the data sheet of the memory device.

Figure 17. Timing Diagram Showing User Command and Data Interface for a Write Command



WRITE RLDRAM I, CONFIG 1, BL 4

Figure 18. Timing Diagram Showing User Command and Data Interface for a Read Command



READ RLDRAM I, CONFIG 1, BL 4

RLDRAM Memory Controller Performance

Table 5 lists the bandwidth performance per data bit for the various LatticeSCM packages, device supply voltages, and device speed grades. All timing is at a junction temperature of 105°C and below.

Table 14. RLDRAM Memory Controller Performance

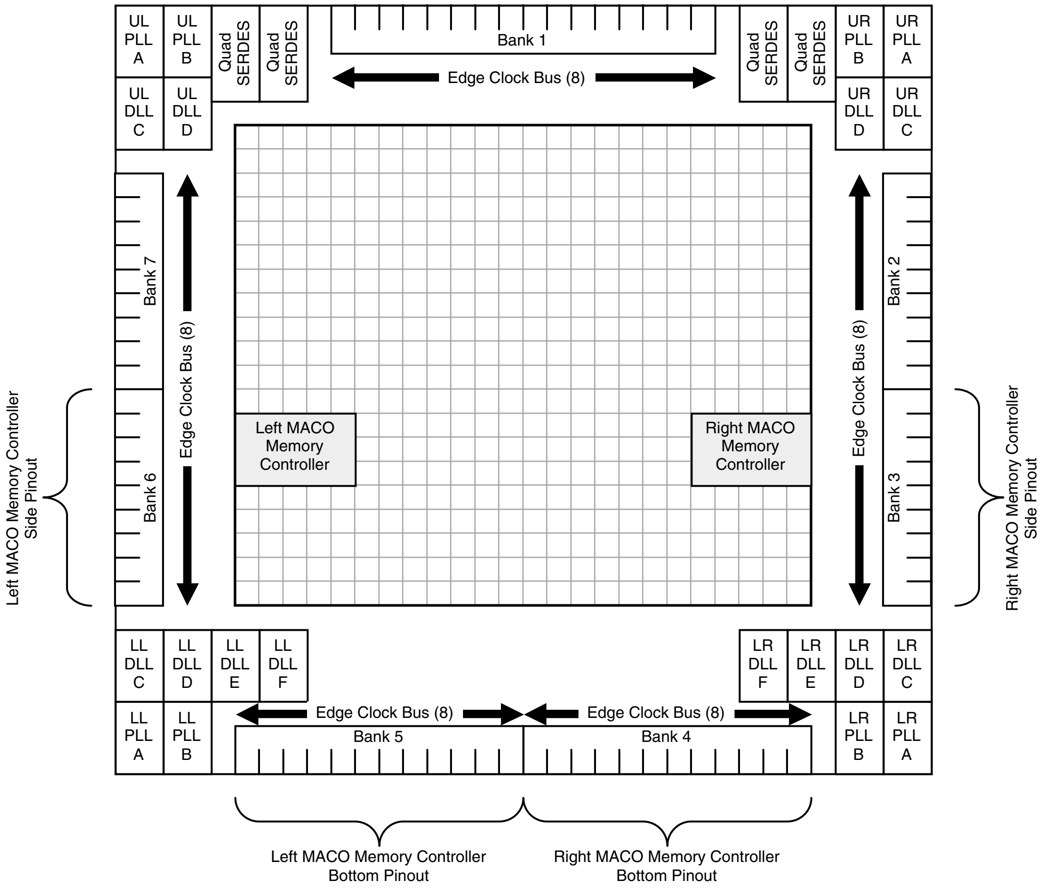
	VCC = 1.0V ±5%			VCC = 1.2V ±5%			
Package	-5	-6	-7	-5	-6	-7	Units
Wirebound: 256, 900	500	600	600	600	600	600	Mbps
Flip-Chip: 1020, 1152, 1704	500	600	800	600	700	800	Mbps

RLDRAM Memory Controller On-Chip Resources

Figure 19 illustrates some of the resources on the LatticeSCM device that are available to the RLDRAM Memory Controller, including:

- Seven banks of I/O pins;
- Dedicated routing to two sets of pins from each Memory Controller MACO block;
- Edge Clock buses containing eight clock lines per bus (shown), and two DCNTL buses per bank (not shown).
- PLLs for clock conditioning (up/down frequency shifting, duty cycle/phase adjusting, jitter filtering, etc.);
- DLLs for phase and delay adjustment.

Figure 19. MACO Memory Controller Resources



References

- RLDRAM Data sheets from Micron
- RLDRAM Evaluation Board Demonstration Design
- RLDRAM Characterization Report
- TN1033, High-Speed PCB Design Considerations

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
March 2007	01.0	Initial release.
July 2007	01.1	Added LatticeSCM appendix.
August 2007	01.2	Updated Features list.
		Replaced RLDRAM Memory Controller GUI Information and Readme File screen shot.
		Replaced GUI Dialog Box for RLDRAM Memory Controller screen shot.
		Updated Input Clock description in GUI Dialog Box for RLDRAM Memory Controller Clocks table.
November 2007	01.3	Replaced RLDRAM Memory Controller IPexpress GUI information and readme File screen shot.
August 2008	01.4	Changed document title from "LatticeSCM RLDRAM I/II Controller MACO Core User's Guide" to "RLDRAM Controller MACO Core User's Guide."
		Updated appendix.
November 2009	01.5	Command Decode Logic text section, updated third paragraph.
		Initialization State Machine text section, updated last paragraph.

Appendix for LatticeSCM FPGAs

Table 15. Performance and Resource Utilization[1]

Configuration							
Type	**Data Width**	**Address Width**	**Mem Speed (MHz)**	**Slices**	**LUTs**	**Registers**	**PIOs**
RLDRAM I	32	20	200	490	416	763	371
RLDRAM I	16	21	200	254	216	409	229
RLDRAM II CIO	72	21	400	882	902	1,513	750
RLDRAM II CIO	18	21	400	266	240	418	249
RLDRAM II SIO	72	21	400	882	906	1,513	822
RLDRAM II SIO	18	21	400	266	240	418	267

1. Performance and utilization characteristics are generated using Lattice's ispLEVER® v7.1 software and Synplify Pro v9.4. When using this IP core with different software, results may vary. Not all configurations will fit on smaller LatticeSCM devices.

Ordering Part Number

All MACO IP, including the Ethernet flexiMAC Core, is pre-engineered and hardwired into the MACO structured ASIC blocks of the LatticeSCM family of parts. Each LatticeSCM device contains a different collection of MACO IP. Larger FPGA devices will have more instances of MACO IP. Please refer to the Lattice web pages on LatticeSCM and MACO IP or see your local Lattice sales office for more information.

All MACO IP is licensed free of charge, however a license key is required. See your local Lattice sales office for the license key.