



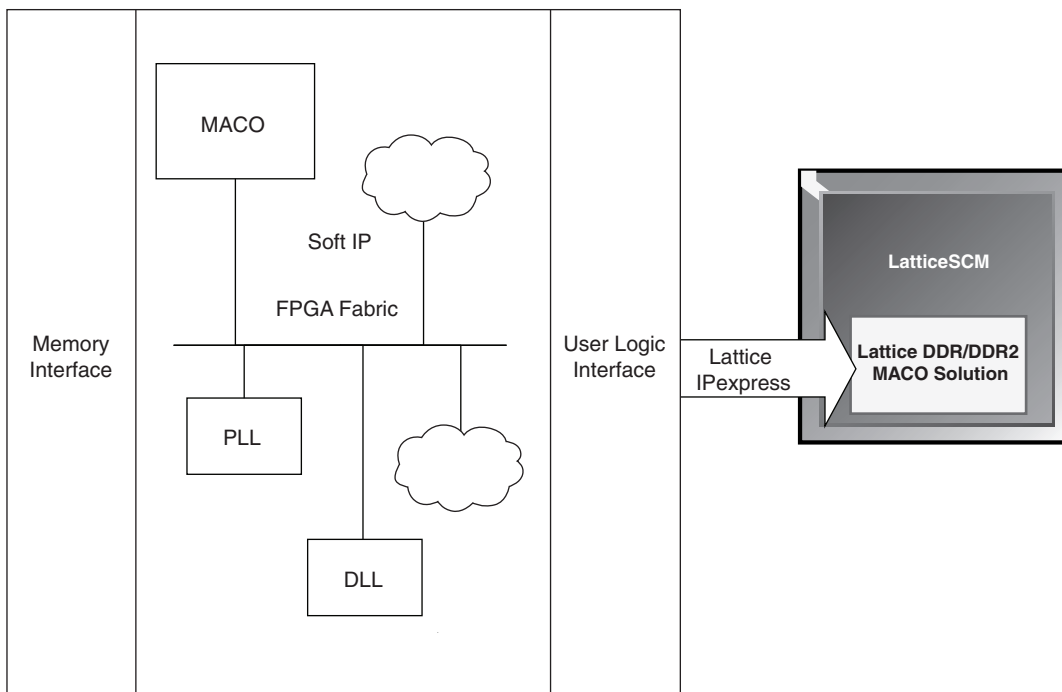
DDR/DDR2 SDRAM Controller MACO Cores

User's Guide

Introduction

Lattice's DDR/DDR2 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 on system architecture design. These blocks eliminate the need to “re-invent the wheel,” by providing industry-standard DDR and DDR2 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 support to generate a number of user-customizable cores with 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 help generate new configurations of this IP core. Specific 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 DDR/DDR2 Synchronous Dynamic Random Access Memory (SDRAM) Controller is a general-purpose memory controller that interfaces with industry standard DDR/DDR2 SDRAM devices and modules. The Lattice Semiconductor DDR SDRAM Controller is a parameterized core that provides the flexibility for modifying data widths, burst transfer rates, and CAS latency settings in a design. It provides a simple command interface for application logic. The controller can be configured to function as a DDR only or DDR2 memory controller.

The memory controller comprises an FPGA logic block and an ASIC block. The FPGA logic is sometimes referred to as the “soft IP” because it is programmed into the FPGA along with the user application. The embedded ASIC block is called the MACO “hard IP”, because as an ASIC, it is an unmodifiable part of the device. Two (one on SC-15) DDR MACO sites are available on the device.

Lattice technical note TN1099, [LatticeSC DDR/DDR2 SDRAM Memory Interface User's Guide](#) covers topics such as modes of operation, I/O buffer and termination issues, system clocking and timing.

In addition to supporting all the features of regular DDR memory, DDR2 memory also supports:

- The posted CAS functionality to maximize data throughput when successive read/write commands with auto pre-charge are presented to the memory.
- An on-die termination resistor. The on/off state of this resistor is controlled by a signal driven by the controller.

This user's guide explains the functionality of the Lattice DDR Controller IP core.

Features

- Interfaces to industry standard DDR and DDR2 SDRAM
- Programmable burst length of 4 or 8
- Posted CAS functionality
- ODT signal generation
- Programmable CAS latency of 3 or higher
- Intelligent bank management to minimize ACTIVE commands
- Synchronous implementation
- Command pipeline to maximize throughput
- Supports SDRAM data path widths of 8, 16, 32, 40, 64 and 72 bits. Data width of 72 is supported in flip-chip or wire bond packages only with single-ended DQS. Maximum data width with differential mode DQS is 40 in wire bond packages.
- Varying address widths for different memory devices
- Programmable timing parameters
- Internal core frequency and DDR-2 DRAM frequency of 333MHz with two chip selects used
- Byte-level writing through data mask signals
- Supports both true and complementary DQS during write (for a maximum of 40 data bits). During read, the complementary pin is unused.
- Maximum of two chip selects (includes the capability for both chip selects to be de-selected to allow for other chip selects to be added via FPGA gates)
- Supports PCB trace lengths of up to eight inches.

Design Kit Deliverables

- Sample instantiation (template)
- Synthesis black box for MACO core
- Pre-compiled ModelSim® MACO core model
- Verilog core source code
- Evaluation design
 - 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 DDR/DDR2 MACO Kit for details on the Kit's contents
- MACO License File
- See the IPexpress 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

DDR/DDR2 SDRAM is similar in function to regular SDRAM, but doubles the bandwidth of the memory by transferring data twice per cycle, on both the rising and falling edges of the clock signal.

The memory controller core 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 need to deal with the DDR/DDR2 SDRAM command interface. The timing parameters for the memory can be set through the signals that are input to the core. This enables the user to switch between different memory devices and/or to modify the timing parameters to suit the application using the IPexpress utility.

While most of the functionality of the memory controller remains the same for both DDR and DDR2 mode, certain differences exist.

Table 1. Basic Differences Between DDR and DDR2

Feature	DDR Mode	DDR2 Mode
CAS Latency	1, 2, or 3 clocks	2, 3, 4 or 5 clocks
Write Latency	1 clock	Read Latency - 1
Burst Length	2, 4, 8 words	4, 8 words
DQS as differential signals	No	Yes
Redundant DQS for read data (RDQS, RDQS#)	No	Yes (only for 32x8 configuration)
Ability to interrupt 8-word burst (write or read)	No	Yes
No of banks per device	4	4 or 8
On Die Termination	NA	Supported
Posted CAS Additive Latency Mode	NA	Supported

Top Level Block Diagram

Figure 2. DDR_IP_TOP Module

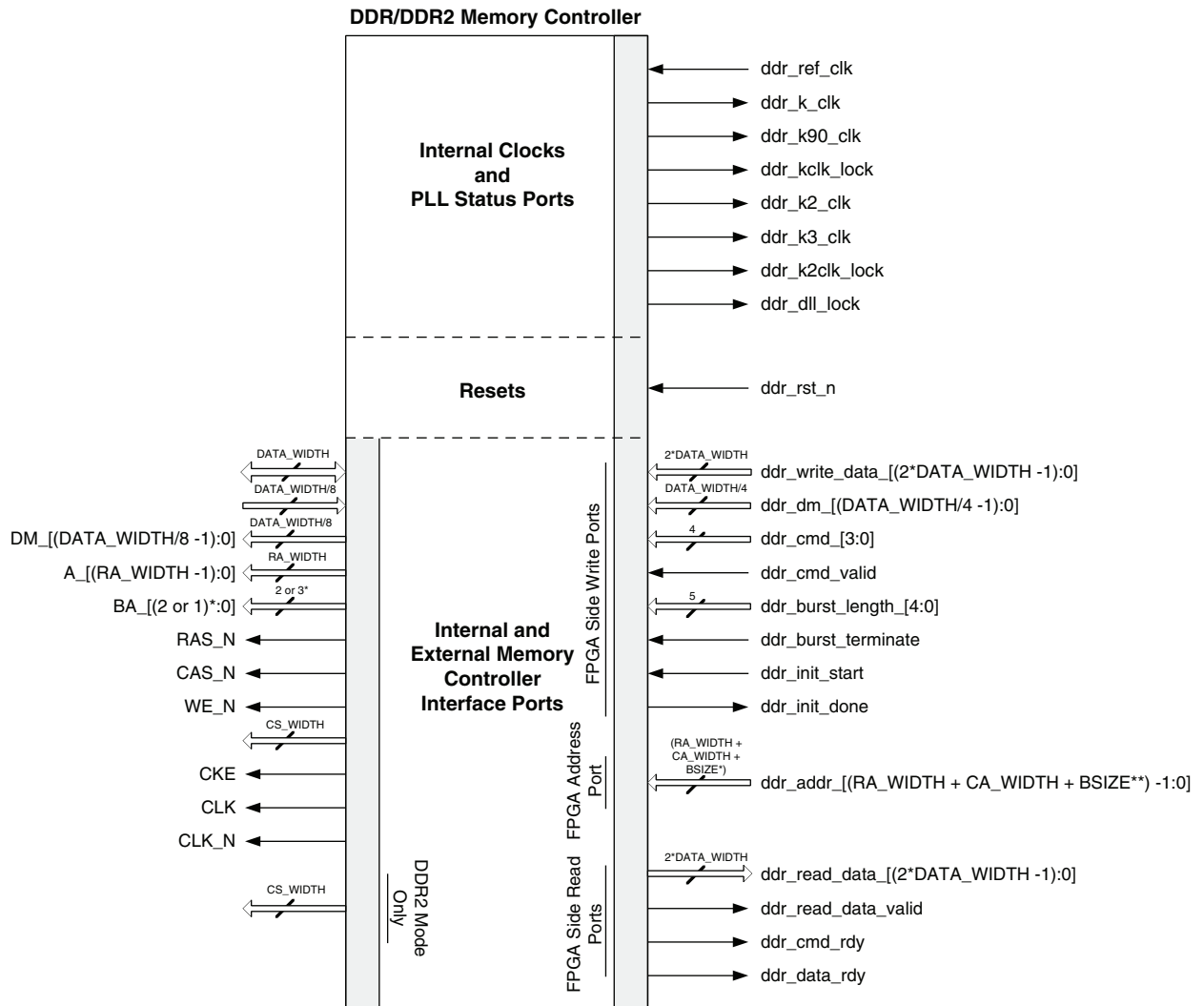
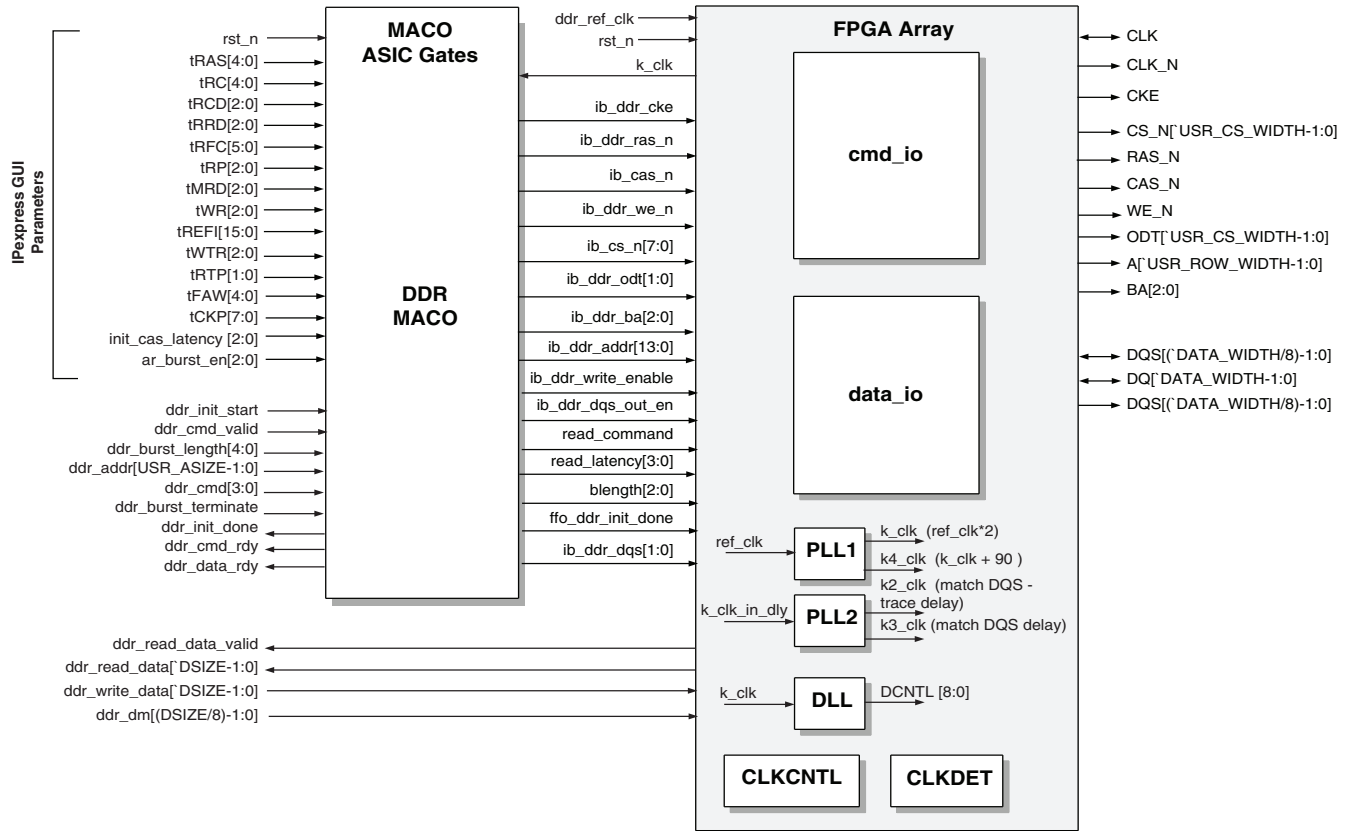


Figure 3. DDR_IP_TOP Detailed Diagram of DDR Controller

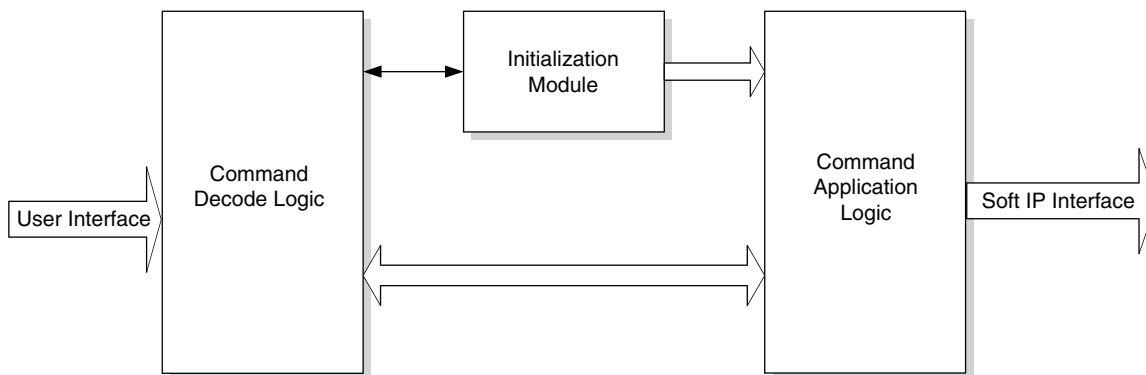


The DDR Controller core includes the following functional blocks:

1. DDR MACO hard-core of the LatticeSCM device.
2. `cmd_io` – instantiates I/Os for memory device command and address bus
3. `data_io` – instantiates I/Os for memory device data bus
4. PLLs and DLLs
5. Clock control and clock detection logic blocks

The MACO hard core includes three blocks as shown in Figure 4.

Figure 4. DDR MACO Block Diagram



Command Decode Logic

The commands presented by the user are decoded and placed into one of the two internal queues by this module. The controller asserts the signal `ddr_cmd_rdy`, whenever it is capable of accepting a new command from the user. To ensure that the available bandwidth is fully utilized at a burst length of 8, this module is capable of issuing a `ddr_cmd_rdy` signal once every 4-clock cycles. A command is accepted if the `ddr_cmd_valid` signal was asserted. A valid command is then decoded and the bank management logic compares the row and bank address of the current command with the list of open banks/rows to determine whether precharge and/or activate command should be applied.

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 register update is complete and the memory chip is reprogrammed with the new values.

This module also maintains a refresh counter and issues a request for a refresh command(s) to be generated. The controller allows up to eight auto-refresh commands to be issued to the memory chip. The user can select the exact number to be issued through the `ar_burst_en` signal.

The generic user interface integrates the core to standard bus interfaces. The user is required to only supply the Read, Write, Power down, Load Mode register, and Self Refresh commands through the interface. The controller can also accept the read/write with auto precharge command. The controller will apply the proper commands based on the address of the accessed location. Table shows the valid values for the `cmd[3:0]` bus.

Table 2. User Interface Commands

Command	Acronym	Command Decoding cmd[3:0]	Control Signals				SDRAM Address
			CS#	RAS#	CAS#	WE#	
Read	READ	0001	0	1	0	1	Column
Write	WRITE	0010	0	1	0	0	Column
Read with Auto Precharge	READA	0011	0	1	0	1	Column
Write with Auto Precharge	WRITEA	0100	0	1	0	0	Column
Power Down	PWRDN	0101					
Load Mode Register	LOAD_MR	0110	0	0	0	0	Opcode A15-A0
Self Refresh	SELF_REFRESH	0111	0	0	0	1	X
Read Interrupt	READ_INT	1001	0	1	0	1	Column
Read Interrupt with Auto Precharge	READ_INTA	1010	0	1	0	1	Column
Write Interrupt	WRITE_INT	1011	0	1	0	0	Column
Write Interrupt with Auto Precharge	WRITE_INTA	1100	0	1	0	0	Column

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.

Initialization State Machine

This module initializes the DDR SDRAM after power-up as indicated by the user. Initialization is done in a pre-defined manner as mentioned in the JEDEC specification. Since initialization must be performed at least 200 μ S after power-up, the user is required to initiate this process to meet the desired specification.

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

- Issue a NOP command
- * Activate internal DDR SDRAM clock signals by making ddr_cke signal HIGH
- Issue a PRECHARGE ALL command
- Enable the DLL by issuing a LOAD MODE REGISTER command to the extended mode register. Write default values to the register.
- Reset the DLL by issuing a LOAD MODE REGISTER command to the mode register
- Wait for 200 clock cycles for the DDR SDRAM DLL to lock
- Place the device in idle state by issuing a PRECHARGE ALL command
- Once in idle state, issue two AUTO REFRESH commands
- Issue a LOAD MODE REGISTER command to the mode register to program operating parameters with “reset DLL” deactivated. Writes CFG register value for BL (Burst Length), CL (CAS Latency) and sets the BT (Burst Type) to sequential mode.

The initialization sequence varies slightly in the DDR2 mode.

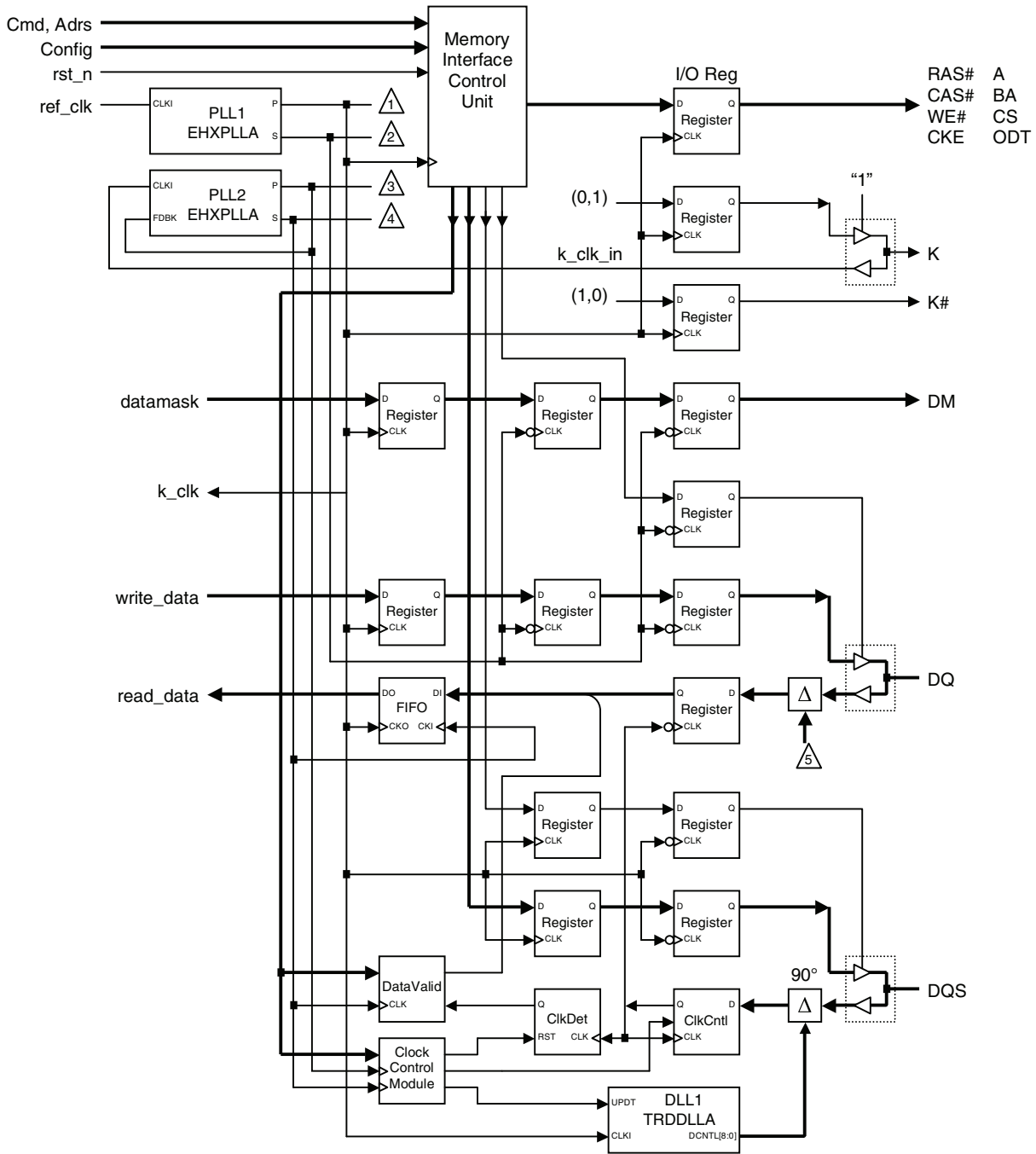
Command Application Logic

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.

Commands in the two pipelines are executed in parallel to maintain a high throughput. This module also meets the timing requirements set by the user through the configuration interface. To maximize data throughput at burst length of eight, this module is capable of accepting a new command every four clock cycles.

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.

Figure 5. Clocking Scheme



- Notes:
- 1 k_clk ($ref_clk \times 2$)
 - 2 $k4_clk$ ($k_clk + 90^\circ$)
 - 3 $k2_clk$ (match DQS – trace delay) + 90°
 - 4 $k3_clk$ (match DQS) + 90°
 - 5 Delay for DQS edge clock injection matching (Delay = 13 for SC25, other devices TBD)

The core also utilizes the FPGA fabric for I/O interfaces and clocking. This includes Data I/O and Command I/O as well as the PLLs, DLLs and the clock detect logic.

Data I/O

Data I/O interfaces with the user logic and I/O pads for transferring data between the two interfaces. The logic for this module is outside the MACO core. This module transfers write data from user to memory, read data from memory to user. During a write operation, the user data is transferred to the DRAM on the bi-directional DQ bus using k4_clk (k_clk+90 deg). Data is sent on both edges using ODDRXA pads. During a read operation, the data from the DRAM is captured using the DQS signal (shifted by 90 degrees) and is given to the user after synchronizing with system clk (k_clk).

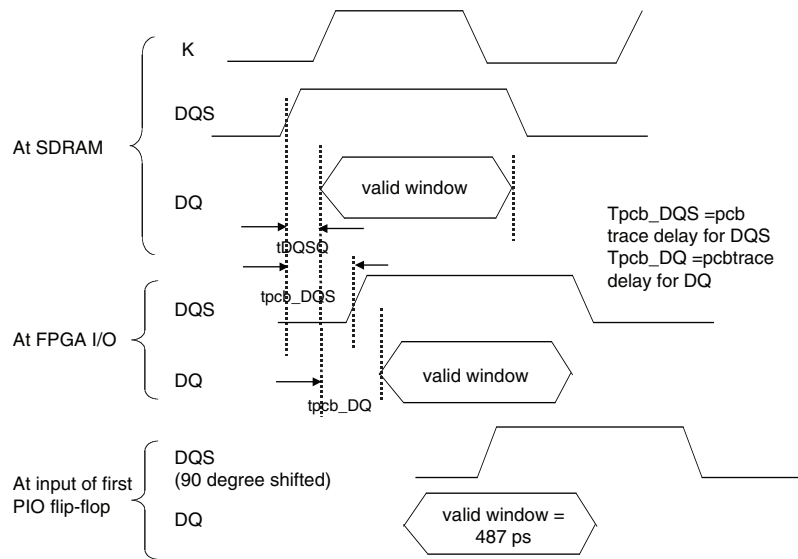
Command I/O

Command I/O interfaces with DDR MACO and I/O pads for transmitting DDR command to memory device. The DDR Commands from the MACO block are directly sent to the memory. This module is also part of the soft IP.

Clocking

Two PLLs and one DLL are used in the soft IP. PLL1 generates k_clk (core clock) and k4_clk which is a 90-degree phase shifted version of k_clk. K_clk is the main clock that is used for all the registers in the design. The same k_clk is sent to memory as K and K#. K4_clk is used for driving the write data from the user to the memory. The clocking scheme is shown in Figure 6.

Figure 6. Read Data Capture at SDRAM Controller



As shown in Figure 6, the read data DQ is captured on the rising or falling edge of the data strobe DQS (DQS# is not shown). Since DQ and DQS are edge-aligned coming from the SDRAM device, DQS needs to be delayed (ideally centered to DQ) to effectively capture the data. Methods such as using the cycle stealing delays or by pre-setting the INDEL to a given value can be used to delay the DQS with respect to the data, but using the DLL as shown in Figure 3 to control the INDEL to delay the DQS signals by 90 degrees gives the greatest timing margin over PVT and is independent of the interface speed. The INDEL can be set to a single value per device to match the edge clock injection delay variations over process, voltage and temperature, thus a fixed INDEL setting on the DQ inputs will be used to match the captured DQ data to the edge clock injection delay for DQS. The memory controller core uses the K clock. K and its complement K# are also sent to the SDRAM memory device. This core clock K is fed into DLL0TRDLL (which operates as the master DLL) to produce a T/4 digital control output called dcntlctrl0. This is a 9-bit bus that is used to control the INDEL delay cells within the PIOs used for DQS/DQS# read inputs and will provide a 90 degree time shift for the DQS/DQS# input signals. DLL0TRDLL can be adjusted to give additional margin on top of the 90 degree delay based on the customer's actual system.

As shown in Figure 4, the PCB routing delay of DQS is denoted by $tpcb_dqs$ and the PCB routing delay of DQ is denoted by $tpcb_dq$. DQS arrives at the FPGA I/O and gets routed to the clock pins of all the associated DQ pins. This results in extra on-chip DQS clock delay and a clock skew on the FPGA device which is estimated to be approximately 50 to 100 ps for the worst case edge clock under worst case conditions. The digital control $dctrl0$ delays DQS by $T/4$, which results in $DQS0d$ shown in Figure 4. $DQS0e$ with respect to DQS is $T_{cyc}/4 + \text{clock injection time } (T_{inj})$.

The available data window at the first capture flip flop = Data valid window at the SDRAM memory - ((setup+hold at FPGA + package skew + $tpcb_dqs$ + clock skew). Assume FPGA setup and hold is 100 ps.

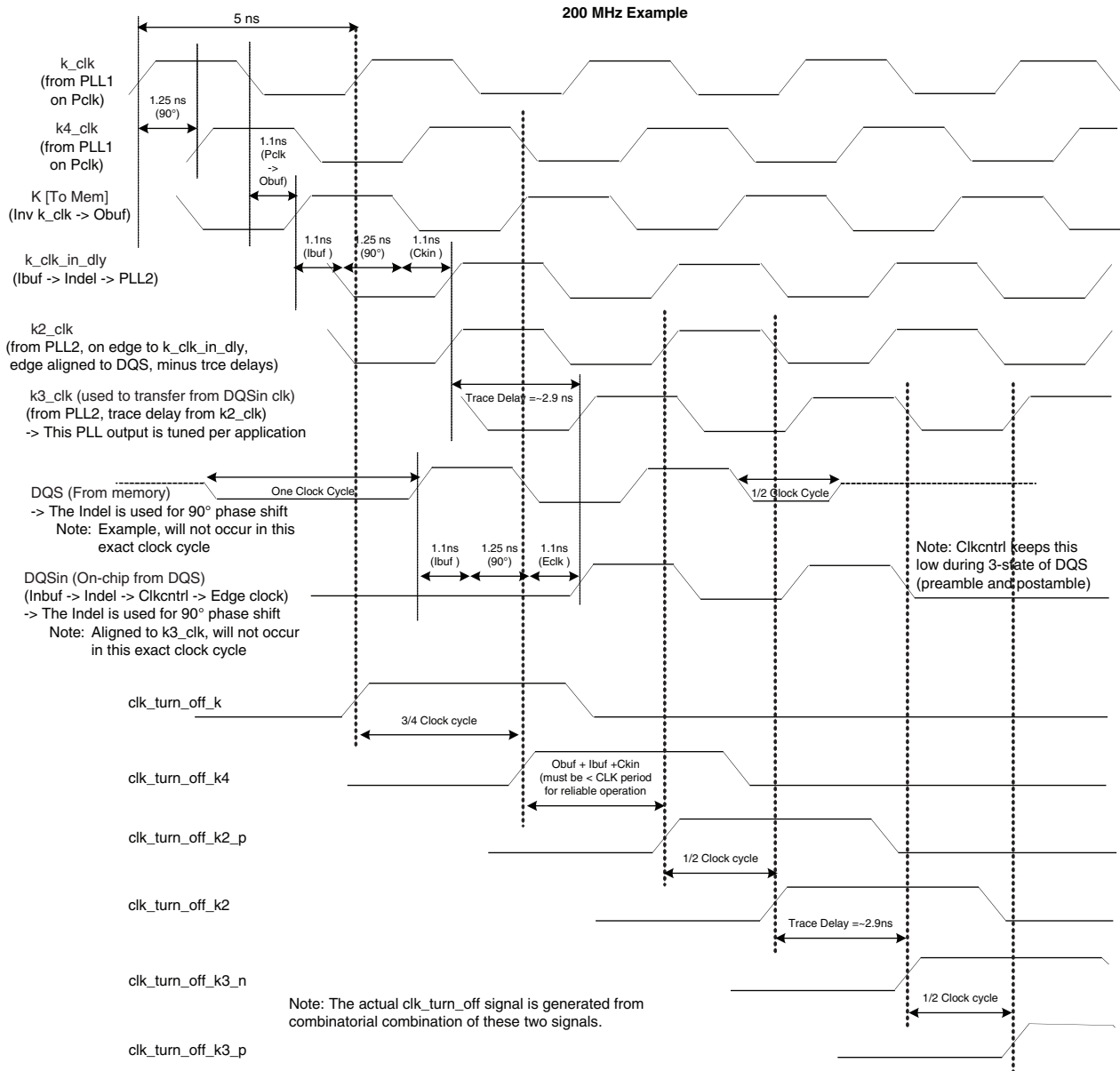
Example of the data valid window = 987 ps - ((100 + 100 + 50 + 100 + 50) = 487 ps.

Since DQS is a strobe and not a free-running clock, the read data captured with DQS should be recaptured using a free-running clock. As shown in Figure 3 and Figure 4, this is done using the K3 clock rather than the K clock. This is done because the DQS signal from the DDR2 memory is generated from the K clock signal sent from the FPGA device and then sent back to the FPGA device during a read. As shown in Figure 2, the K clock is looped back within the same I/O pad to the input clock routing in order to generate the K2 clock matched to $k_clk_in_dly$. Thus, this delay path has the same output buffer delay as K clock (including associated extrinsic loading delay) and matches the input buffer delay on the DQS/DQS# pins. It is delayed by $dctrl0$ control from TRDLL, which is the same control that is used to provide a 90 degree lag on the DQS pins. On the DDR2 device, the K clock input is used to generate the DQS strobe at $tDQSCK$ (+/-450 ps for the Micron device). Therefore the resulting clock signal $k2_clk$ has the same delay as the DQS signal coming back from the SDRAM except that the DQS strobe has extra delay associated with the K signal pcb trace delay ($tpcb_K$) and the DQS return pcb trace delay ($tpcb_DQS$) and the DQS also can be +/- this delay by $tDQSCK$ (+/-450 ps).

The DQS is then received at the FPGA to capture the read data. The output from the input buffer INDEL element at the pad for the K clock, referred to as $k_clk_in_dly$, is fed as the reference clock to PLL2 to generate $k2_clk$ and $k3_clk$. If the RAM device is close enough to the FPGA on the board and the SDRAM interface speed is slow enough, then the $k_clk_in_dly$ (possibly tuned further using INDEL) can be used to hand off from the DQS clock that will stop at the end of read instructions to an internal continuous clock. Generally however $k2_clk$ is phase matched to $k_clk_in_dly$ and $k3_clk$ is phase shifted from $k2_clk$ by a value equal to the pcb routing delay. Thus $k3_clk$ nominally matches the round trip delay of DQS. Generally k_clk is the clock used for other internal logic on the device.

The read data-timing diagram in Figure 5 shows the read data captured using DQS at the FPGA I/O, the relationship between k_clk , $k4_clk$, $k2_clk$ and $k3_clk$. It also shows an example for the number of K clock cycles of latency after which read data is available to the FPGA. The $data_valid$ read_data_start signal generated in the soft IP indicates the start of the read data burst. This is generated by sampling the first rising edge of DQS using the edge detect capability built into the FPGA PIOs. The naming conventions used in Figure 5 should be used only as a reference.

Figure 7. Read Data Timing



GOAL: Transfer Signals between k_clk and k3_clk, where k3_clk is created to be matched to the delay of DQS that is sent back from the DDR memory device when performing a read. The goal is for this circuit to work regardless of speed and trace length of K to the memory and DQS back from the memory. The only requirements are:

Note: Example shown is for transfer of clk_turn_off signal generated on core clock which is k_clk. This signal is transferred through k4_clk, k2_clk and finally to k3_clk in such a way that delays are not lumped between transfers. Various delays are as shown in the waveform. K1_clk, k2_clk, k3_clk and k4_clk are all shown as they appear at FFs after routing on primary clocks.

Three types of delays are possible :

1. Delays that depends on the clock cycle itself.
2. Trace delay of PMIK to the DDR memory, DLL delay at the DDR memory trace delay of DQS returning to the LatticeSCM device.
3. Output Buffer (with clk->out of ODDRXA with board load on PMIK) + Input Buffer (With Clkcntrl delay) + Edge clock insertion delay (ECLK).

The above scheme will work for all the clock frequencies as long as following conditions are MET.

1. Trace Delays + DLL delay < 1 Clock Cycle.
2. Output Buffer + Input Buffer + Edge Clock Insertion Delay < 1 Clock Cycle

Note: If all of these delays can be in one clock cycle, k2_clk can be removed and transfers from k4_clk -> k3_clk can be done., where k3_clk if used as the feedback to PLL2.

Input/Output Signals

Table 3 shows the signals connecting to the user interface.

Table 3. User Interface I/O Signals

Signal Name	Active State	Signal Direction (I/O)	Description
User Interface			
ddr_ref_clk	—	I	System Clock
ddr_rst_n	Low	I	System Reset
ddr_init_start	High	I	Asserted when an initialization routine is to be performed, and deasserted when ddr_init_done is asserted, indicating that the initialization routine is complete.
ddr_cmd_valid	High	I	Asserted when the contents of cmd and addr bus are valid
ddr_cmd[3:0]	—	I	Command for controller
ddr_addr[[^] USR_ASIZE-1:0]	—	I	Address for read/write. USR_ASIZE is a programmable parameter set based on size of memory, which is derived by the following formula: USR_ASIZE = USR_ROW_WIDTH + USR_BSIZE + USR_COL_WIDTH
ddr_burst_length [4:0]	—	I	Indicates the number of read/write commands to be issued to DRAM
ddr_burst_terminate	High	I	Asserted if the burst cycle is to be terminated.
ddr_write_data [[^] DSIZE-1:0]	—	I	Data input. DSIZE is set to DATA_WIDTH times 2
ddr_dm [([^] DSIZE/8) -1:0]	High	I	Data Mask for write data
ddr_cmd_rdy	High	O	Asserted to indicate that the controller is ready to accept a new command.
ddr_data_rdy	High	O	When asserted, the controller is ready to accept data on the write_data bus.
ddr_init_done	High	O	Asserted when the controller has completed the initialization routine.
ddr_read_data_valid	High	O	When asserted, the contents of the ddr_read_data bus are valid
ddr_read_data [[^] DSIZE-1:0]	—	O	Read Data Out
Configuration Interface Signals (set through ispLEVER/IPexpress GUI)			
trefij[15:0]	NA	I	Refresh Interval in clock cycles.

Table 4 shows the signals of the DDR SDRAM memory types.

Table 4. DDR/DDR2 External Interface I/O Signals

Signal Name	Active State	Signal Direction (I/O)	Description
DDR/DDR2 Memory Interface Primary Signals			
CLK	High	O	DDR/DDR2 SDRAM clock derived from the system clock
CLK_N	Low	O	Inverted DDR/DDR2 SDRAM clock derived from the system clock
CKE	High	O	Clock enable
CS_N [[^] USR_CS_WIDTH-1:0]	Low	O	Active low chip select which selects and deselects the DDR SDRAM
RAS_N	Low	O	Row Address Strobe

Table 4. DDR/DDR2 External Interface I/O Signals (Continued)

Signal Name	Active State	Signal Direction (I/O)	Description
CAS_N	Low	O	Column Address Strobe
WE_N	Low	O	Write Enable
ODT [^USR_CS_WIDTH-1:0]	Low	O	DDR2 only: Signals controlling the on-die termination registers on the memory chip.
BA [2:0]	NA	O	Bank address select. DDR2 Mode: [2:0] if INT_BANK is 8 [1:0] if INT_BANK is 4 DDR Mode: default value [1:0]
A [^USR_ROW_WIDTH-1:0]	NA	O	Row or column address lines depending whether the ddr_ras_n or ddr_cas_n is active.
DQ[^DATA_WIDTH-1:0]	NA	I/O	Bi-directional data bus.
DQS [^(DATA_WIDTH/8)-1:0]	NA	I/O	Bi-directional data strobe.
DM [^(DATA_WIDTH/8)-1:0]	NA	O	Data mask signals used to mask the byte lanes for byte level write control.

Parameter Descriptions

Several configuration and timing parameters must be set before the DDR SDRAM Controller Module can be interfaced to a memory device. To ensure maximum flexibility in using the IP core, these parameters are designed as inputs to the IP core that can be tied to desired values within the top level RTL file. These values are input via the IPexpress GUI utility capturing the parameters into the user's customized core. The user inputs physical and actual timing information to reflect their memory design into the GUI. This data is processed to format the pertinent parameters needed to compile their customized design.

Table 5. Programmable Parameters/User Interface I/O Signals

Signal Name	Active State	Signal Direction (I/O)	Description
Configuration Interface Signals (set through ispLEVER/IPexpress GUI)			
tRAS[4:0]	NA	I	ACTIVE to PRECHARGE command delay in clock cycles.
tRC[4:0]	NA	I	ACTIVE to ACTIVE/AUTO REFRESH delay in clock cycles.
tRCD[2:0]	NA	I	ACTIVE to READ/WRITE delay in clock cycles.
tRRD[2:0]	NA	I	ACTIVE bank a to ACTIVE bank b delay in clock cycles.
tRFC[5:0]	NA	I	AUTO REFRESH command period in clock cycles.
tRP[2:0]	NA	I	PRECHARGE command period in clock cycles.
tMRD[2:0]	NA	I	Loan Mode Register command period in clock cycles.
tWR[2:0]	NA	I	Write recovery time in clock cycles.
tREFI[15:0]	NA	I	Refresh Interval in clock cycles.
ext_reg_en	High	I	When Asserted, EMR is written into during initialization
tWTR[2:0]	NA	I	DDR2 only: Internal Write to Read command delay in clock cycles.
tRTP[1:0]	NA	I	DDR2 only: Internal READ to Precharge command delay.
tFAW[4:0]	NA	I	DDR2 only:
tCKP [7:0]	NA	I	DDR2 only: CKE assertion to Precharge command delay during initialization sequence.
Init_cas_latency [2:0]	NA	I	CAS latency during initialization sequence.
ar_burst_en[2:0]	NA	I	Number of Auto Refresh commands issued at a time.

Table 6. Mode Parameters I/O Signals

Parameter Name	Range	Description
Configuration Interface Signals (set through ispLEVER/IPexpress GUI)		
CONTROLLER_MODE	DDR, DDR2	DDR mode or DDR2 mode
DATA_WIDTH	8, 16, 24, 32, 40, 48, 56, 64, 72	Data bus width
USR_ROW_WIDTH	1 to 14	Row address width
USR_COL_WIDTH	8 to 13	Column address width
USR_CS_WIDTH ¹	1, 2, 4, 8	Number of chip selects
INT_BANK ²	4, 8	Number of banks
Operating Frequency	166, 200, 266	Frequency
BUFFER_TYPE	SSTL2-Class2, HSTL1	I/O buffers to be selected

1. For DDR, allowed values are 1, 2, 4 and 8. For DDR2, allowed values are 1 and 2.

2. For DDR2 only.

Table 7. Bank Size Dependency on CS_WIDTH and INT_BANK Parameters

CS_WIDTH	Parameters Derived	Value	Example
DDR Mode			
1	BSIZE	2	<code>`define BSIZE 2</code>
2	BSIZE	3	<code>`define BSIZE 3</code>
4	BSIZE	4	<code>`define BSIZE 4</code>
8	BSIZE	5	<code>`define BSIZE 5</code>
INT_BANK Set to 8 in DDR2 Mode			
1	BSIZE	3	<code>`define BSIZE 3</code>
2	BSIZE	4	<code>`define BSIZE 4</code>
INT_BANK Set to 4 in DDR2 Mode			
1	BSIZE	2	<code>`define BSIZE 2</code>
2	BSIZE	3	<code>`define BSIZE 3</code>

User Interface

After a power-on reset, the user requests the IP to initialize by asserting the `ddr_init_start` signal, and keeping it asserted until the `ddr_init_done` signal returns asserted, at which time `ddr_init_start` is deasserted and initialization is complete.

After initialization is complete, the user can issue a command by holding the `ddr_cmd` and `ddr_addr` buses valid for two consecutive rising edges of `k_clk`, the first being together with the assertion of `ddr_cmd_rdy` by the user and `ddr_cmd_valid` by the controller.

Along with a read or write command, the user also needs to place the `ddr_burst_length` and the `ddr_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 `ddr_read_data` bus when `ddr_read_data_valid` is active.

If the command issued was a write, the user has to provide the data to be written on the `ddr_write_data` bus when `ddr_data_rdy` is active.

The data mask signal `ddr_dm` is used to mask the data being written and should be provided along with the data.

In the case of BL=8 (burst length of 8), two cases of interrupt by a new burst access are allowed. A read can be interrupted by a read and a write can be interrupted by a write with 4 word burst boundary respectively. The minimum CAS to CAS delay is defined by tCCD and is a minimum of 2 clocks for read or write cycles. The following rules apply to burst interrupt:

1. The user command READ_INT will interrupt the immediately preceding READ command.
2. Interruption of a burst read or write cycle during BL=4 mode is not allowed.
3. A read burst with auto-precharge enabled (READA) cannot be interrupted (i.e. READ_INT cannot follow a READA command).
4. A read burst with auto-precharge enabled, can interrupt the current read burst (i.e. READ_INTA can follow a READ command).
5. When a current READ command is interrupted, the read data from the device memory is four words instead of eight.
6. All command timings will be referenced to the burst length mode set in the mode register and not the shortened burst.
7. The user command WRITE_INT will interrupt the immediately preceding WRITE command. This will cause only four words of data associated with the WRITE command to be written into memory.
8. WRITE_INT cannot interrupt a WRITEA command (autoprecharge enabled).
9. WRITE_INTA can interrupt a WRITE command.
10. When a WRITE_INTA or a READ_INTA is presented when a multiple burst write/read operation is in progress, the burst will be terminated.

User Address Mapping

For Single Chip Select

Example: 256Mb DDR2 device arranged as 16 Meg x 16 (16-bit data width)

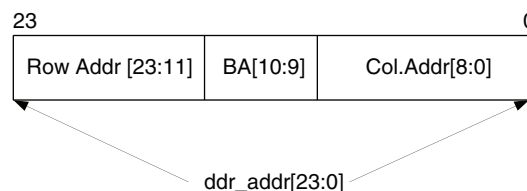
i.e., Four banks, each has 4Meg locations.

Row address	8K (A0-A12)
Column address	512 (A0-A8)
Bank Address	4 (BA0, BA1)

User Address = 13 + 9 + 2 = 24 bits

To address 64M locations, the required number of address bits is 24. Figure 8 shows how the user address is mapped to the memory address. If INT_BANK is set to 8 in DDR2 mode, the BA width becomes [2:0] (eight banks and 32Meg).

Figure 8. Mapping of User Address to Memory Address for Single Chip Select



For Two Chip Selects

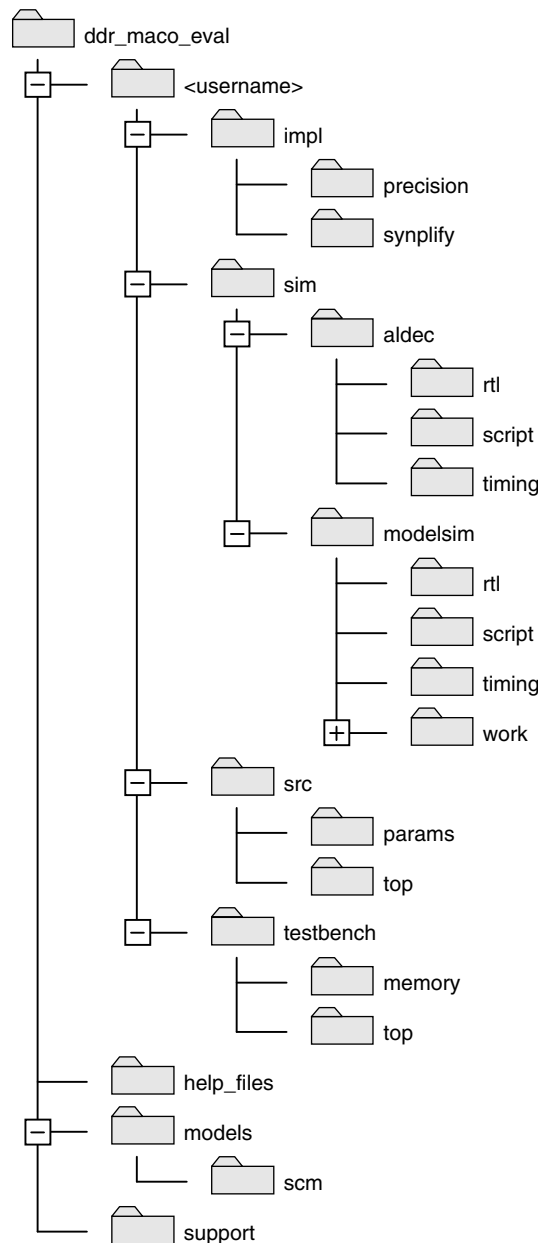
Two chip selects require one extra address line and the effective user address now is 1+24 bits. BA width becomes [2:0] and user address [11:9] is assigned to BA.

DDR/DDR2 MACO Memory Controller Design Kit Directory

The directory structure of the DDR/DDR2 MACO Memory Controller IP, as generated by the IPexpress GUI, is shown in Figure 9.

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.htm" file located in the ddr_maco_eval directory. This Readme file can be invoked in IPexpress by clicking on the "Help" button of the GUI, as shown in Figure 10. It can also be found in the ddr_maco_eval directory.

Figure 9. DDR/DDR2 MACO IP Design Kit Directory Structure



Parameter Descriptions

Figures 10 through 14 give examples of the four IPexpress GUI windows that allow the user to customize the generated IP to a particular application, and Tables 8 through 12 describe each parameter and its function.

Figure 10. GUI Dialog Box for DDR/DDR2 Memory Controller

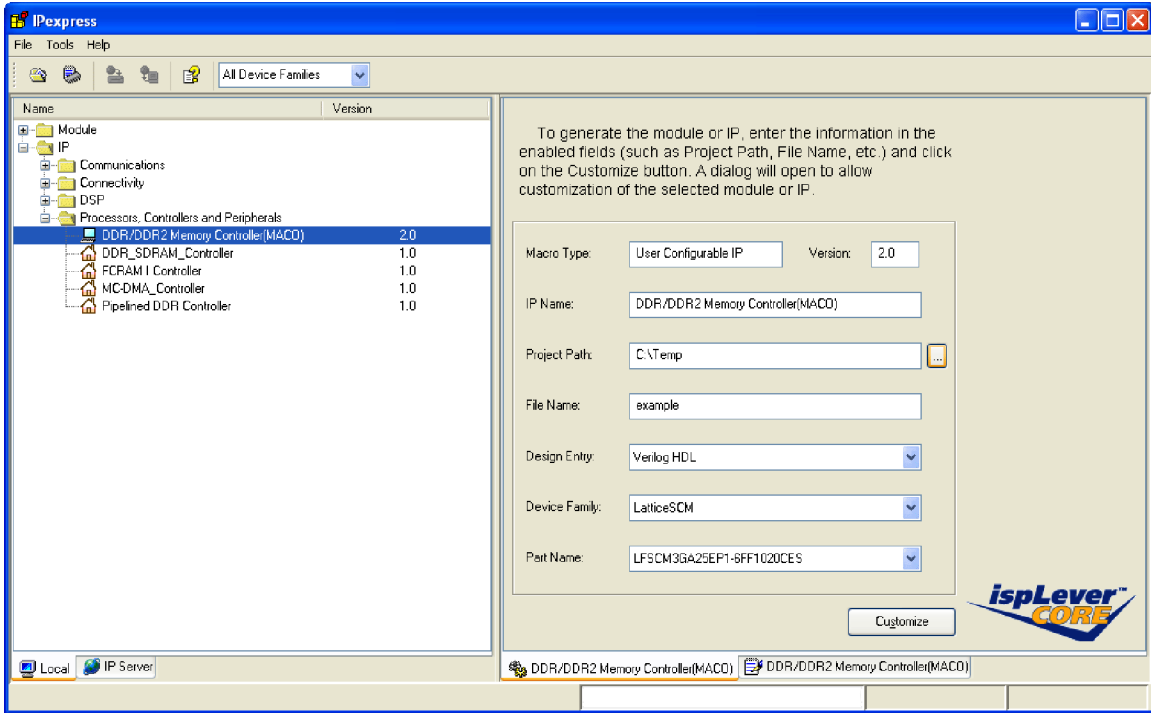


Table 8. GUI Dialog Box for DDR/DDR2 Memory Controller

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 mode is Verilog HDL
Device Family	The device family is LatticeSCM
Part Name	Select the desired LatticeSCM device size, speed grade and package

Figure 11. GUI Dialog Box for DDR/DDR2 Memory Controller Clocks

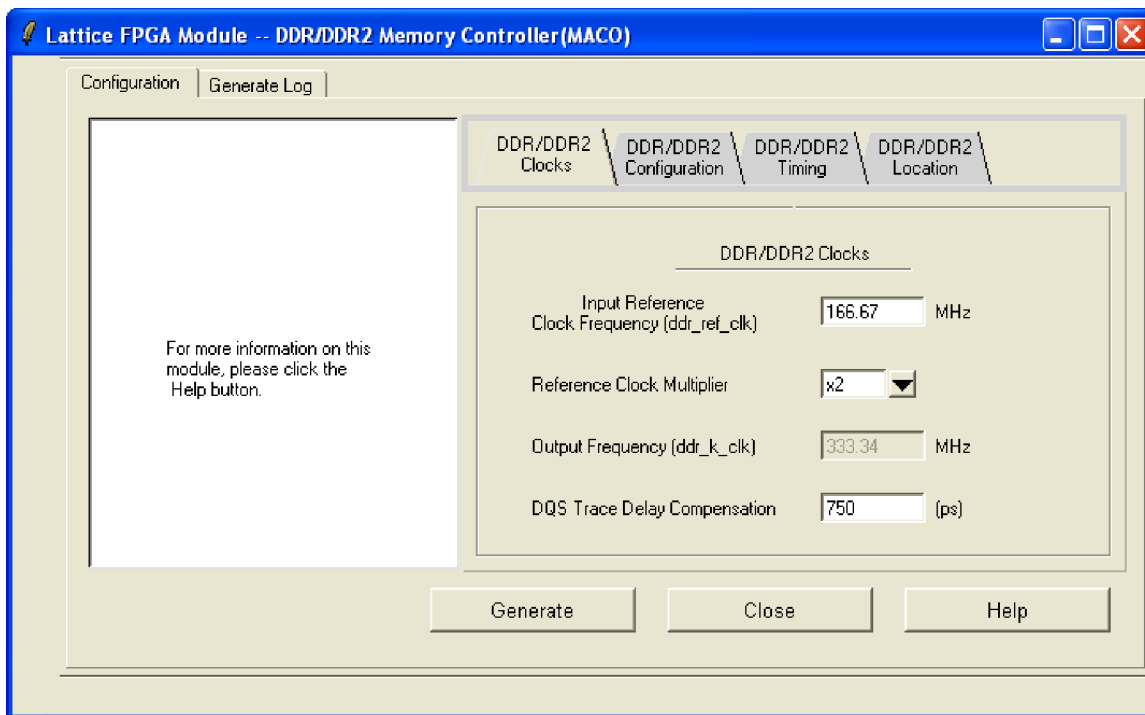


Table 9. GUI Dialog Box for DDR/DDR2 Memory Controller Clocks

Parameter	Description
Input Reference Clock Frequency	Specify the frequency of the input clock to the memory controller. Value range is 100 to 400MHz if the multiplier is set to 1, or 50 to 200MHz if the multiplier is set to 2, etc.
Reference Clock Multiplier	Set this value to the ratio of the desired Output Frequency and the selected Input Reference Clock Frequency. Choices are x1, x2, x4, x8. Default is x2.
Output frequency	The Output Frequency is the operating frequency of the DDR interface. It is calculated by IPexpress, and is set to (Input Reference Clock Frequency) * (Reference Clock Multiplier).
DQS Trace Delay Compensation	The DQS Trace Delay Compensation is set to the round-trip board trace delay (out-bound delay on K, plus inbound delay on DQS) in picoseconds. When the module is being generated for back-annotated simulation purposes, this value should be set to zero.

Figure 12. GUI Dialog Box for DDR/DDR2 Memory Controller Configuration

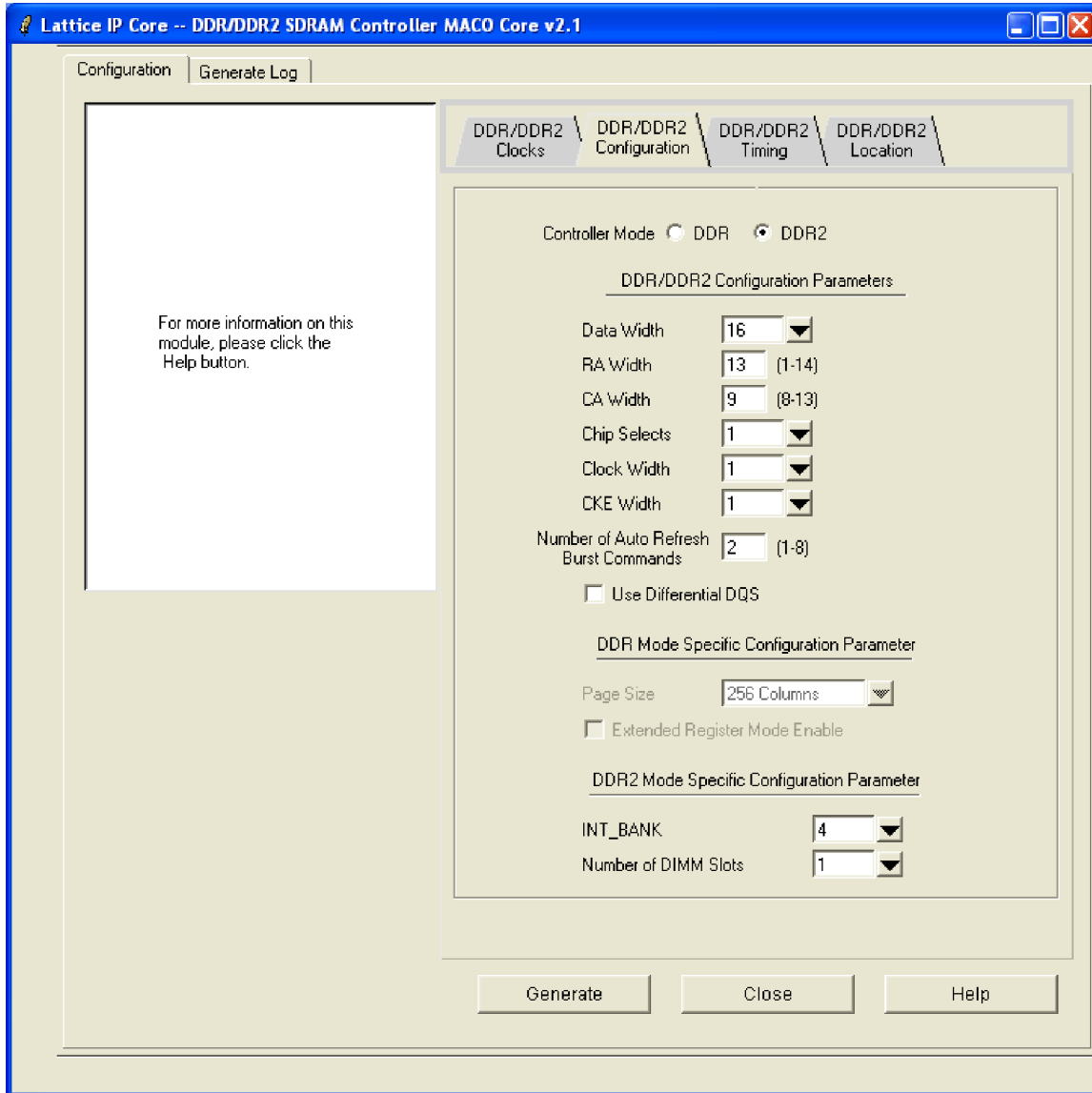


Table 10. GUI Dialog Box for DDR/DDR2 Memory Controller Configuration

Parameter	Description
Controller Mode	Select DDR or DDR2 Mode
Data Width	Width of DQ bus
RA Width	Row Address Width
CA Width	Column Address Width
Chip Selects	Number of chip selects required
Clock Width	Select the number of clocks to be driven out of the LatticeSC device. Valid choices are 1 or 2 and should be the same as CKE Width.
CKE Width	Select the number of clock enables to be driven out of the LatticeSC device. Valid choices are 1 or 2 and should be the same as Clock Width.
Number of Auto Refresh Burst Commands	Select number of refresh operations per auto refresh burst
Use Differential DQS	Check this box to enable differential DQS signals

Table 10. GUI Dialog Box for DDR/DDR2 Memory Controller Configuration (Continued)

Parameter	Description
DDR Mode Only Parameters	
Page Size	Select the desired page size
Extended Register Mode Enable	Enable/disable Extended Mode Register
DDR2 Mode Only Parameters	
INT_BANK	Set this to the internal bank structure of the target DDR device
Number of DIMM Slots	Set this to the number of DIMM slots that the target board supports

Figure 13. GUI Dialog Box for DDR/DDR2 Memory Controller Timing

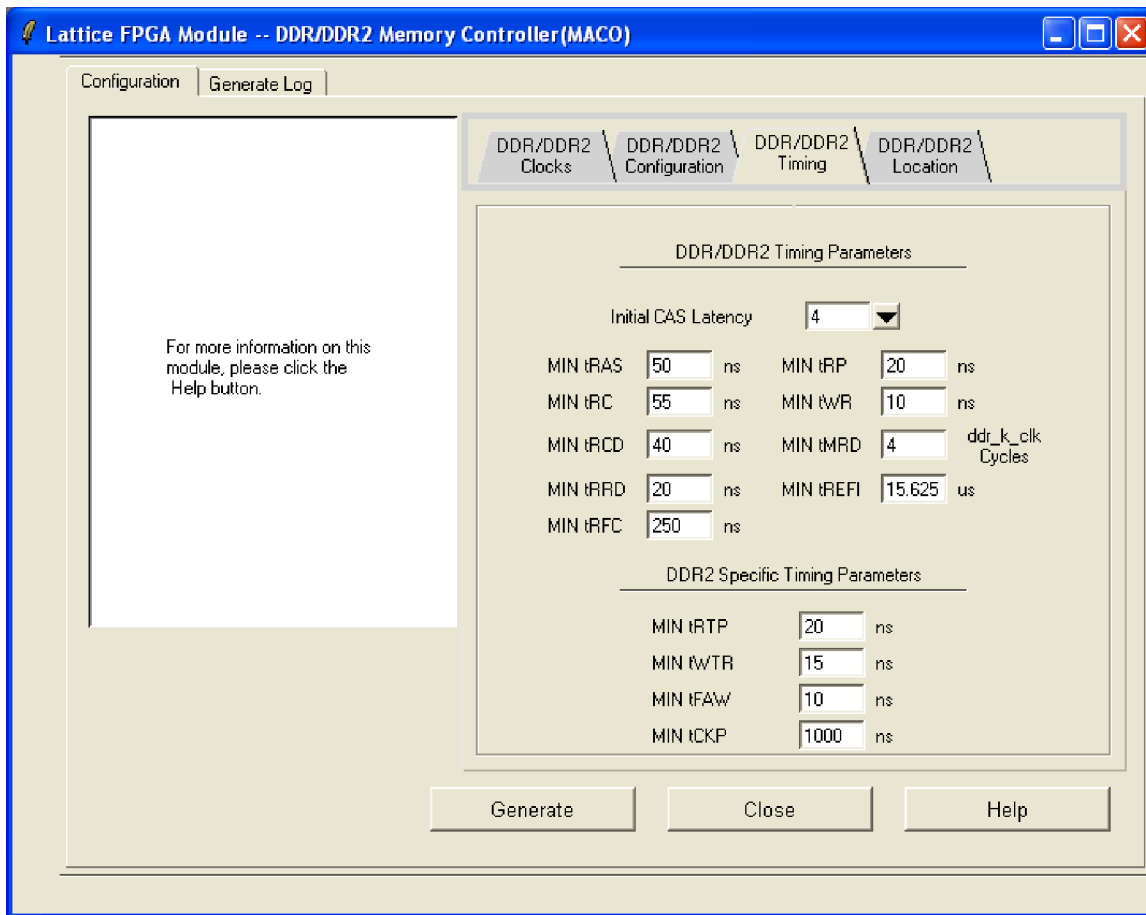


Table 11. GUI Dialog Box for DDR/DDR2 Memory Controller Timing

Parameter	Description
Initial CAS Latency	This is the CAS latency assigned during DDR device initialization
MIN t _{RAS}	ACTIVE to PRECHARGE command
MIN t _{RC}	ACTIVE to ACTIVE (same bank) command
MIN t _{RCD}	ACTIVE to READ or WRITE delay
MIN t _{RRD}	ACTIVE bank a to ACTIVE bank b command
MIN t _{RFC}	REFRESH to Active or Refresh to Refresh command interval
MIN t _{RP}	PRECHARGE command period
MIN t _{WR}	Write recovery time

Table 11. GUI Dialog Box for DDR/DDR2 Memory Controller Timing (Continued)

Parameter	Description
MIN t _{MRD}	LOAD MODE command cycle time
MIN t _{REFI}	Average periodic refresh interval
DDR2-Specific Parameters	
MIN t _{RTP}	Internal READ to precharge command delay
MIN t _{WTR}	Internal WRITE to READ command delay
MIN t _{FAW}	Four Bank Activate period
MIN t _{CKP}	CKE assertion to Precharge command delay during initialization sequence. Set this value to (output clock frequency) * 0.4, to produce a 400 ns delay.

Figure 14. GUI Dialog Box for DDR/DDR2 Memory Controller Location

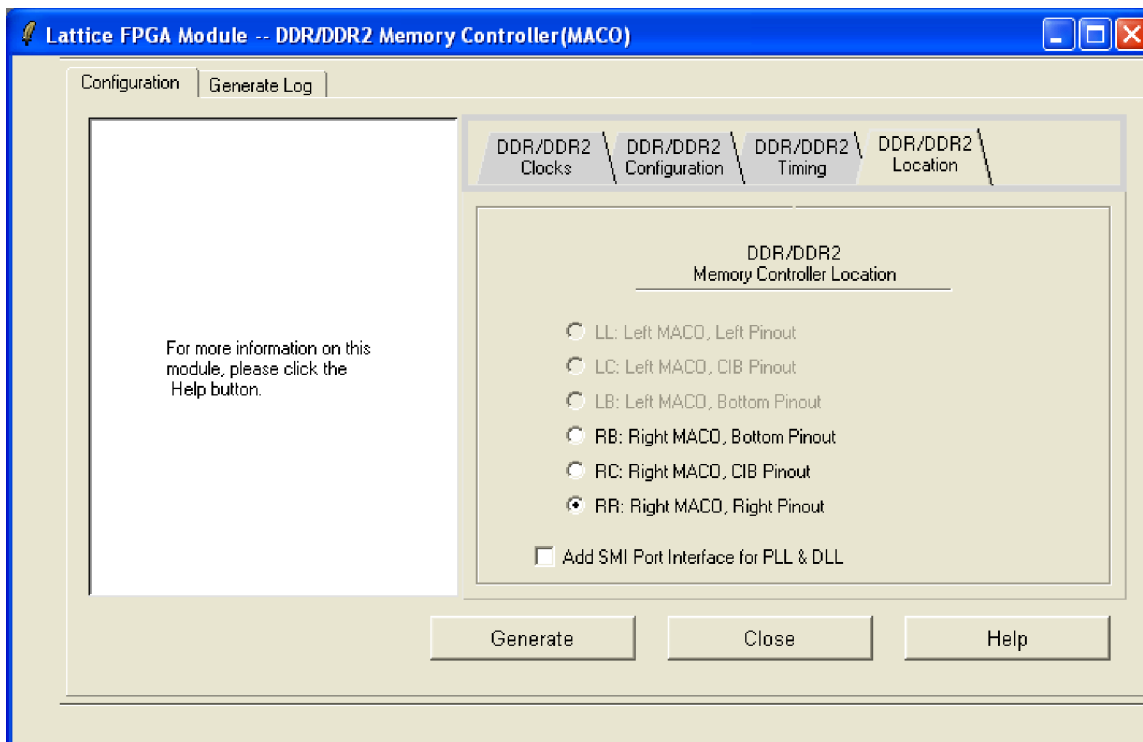


Table 12. GUI Dialog Box for DDR/DDR2 Memory Controller Location

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.
Add SMI Port Interface for PLL and DLL	Check this box to enable run-time access to PLL and DLL memory-mapped parameters

Design Guidelines to Optimize Performance

Master Clock

The master reference clock can be sourced from any clock source, either internal or external to the LatticeSCM device. It is fed to PLL PLL1. If the source is external, it should use the direct input pin for that PLL's CLKI input (refer to Table 13). Also, minimize clock jitter caused by coupling from noisy neighboring signals (refer to the accompanying discussion, "Selecting a Pin That Has Low Jitter Noise" below). Note that the PLL will filter some of the jitter that exists at the PLL's input.

Table 13. 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 14. 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 15. Preferred Pinout for Left Side Memory Controller

DDR/DDR2 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
ODT[0]	AF7	AK29	AN32	AW42	AA1	Y32	AF34	AG42
ODT[1]	AF6	AL29	AP32	AY42	Y1	W32	AE34	AH42
WE_N	AE5	AG28	AJ29	AL34	V4	W25	AA24	AG29
RAS_N	AJ1	AK32	AN33	AV42	V5	Y26	Y24	AF29
CAS_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
A[0]	AH4	AJ28	AN31	AW40	U5	W29	AA33	AD39
A[1]	AG5	AK28	AN30	AY40	U4	W30	Y33	AC39
A[2]	AF8	AJ31	AP31	AW39	T4	V30	Y31	AB42
A[3]	AG8	AH30	AP30	AW38	T5	V29	W31	AA42
A[4]	AH3	AM30	AM29	AV37	U1	V31	W33	AB38
A[5]	AJ3	AM29	AM28	AV36	T1	V32	V33	AA38
A[6]	AF9	AH29	AJ27	AM31	V3	U31	V34	Y41
A[7]	AE10	AH28	AJ26	AM32	U3	U32	U34	W41
A[8]	AK3	AJ27	AP29	BA40	T6	T27	V25	AA36
A[9]	AJ4	AK27	AP28	BB40	U2	T32	U33	Y40
A[10]	AE11	AL28	AN29	BA39	T2	T31	T33	W40
A[11]	AF10	AL27	AN28	BA38	R4	U24	Y27	AC32
A[12]	AH7	AM28	AL26	AW36	R1	R32	W30	Y39
A[13]	AH8	AM27	AL25	AW35	P1	R31	V30	W39
CKE	AE12	AG23	AG23	AM28	R3	T26	V28	AB35

Table 16. Preferred Pinout for Right Side Memory Controller

DDR/ DDR2 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
ODT[0]	AE25	AF27	AL3	AL6	AV4	AB25	W29	Y1	AF1	AG1
ODT[1]	AH28	AG26	AL4	AL7	AV3	AD30	V29	W1	AE1	AH1
WE_N	AD25	AH30	AK1	AM2	AU1	Y30	W26	W8	AA11	AG14
RAS_N	AE26	AG29	AH3	AH6	AL9	AA30	V26	Y7	Y11	AF14
CAS_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
A[0]	AK28	AF25	AJ5	AN3	AW3	T30	T27	W4	AA2	AD4
A[1]	AH21	AG25	AK5	AP3	AY3	W28	R27	W3	Y2	AC4
A[2]	AH23	AG24	AH4	AM6	BA2	U26	V27	V3	Y4	AB1
A[3]	AH22	AF24	AH5	AM7	AY2	U28	U27	V4	W4	AA1
A[4]	AG22	AH27	AM3	AP4	AV6	M30	R30	V2	W2	AB5
A[5]	AG21	AH26	AM4	AP5	AV7	R29	P30	V1	V2	AA5
A[6]	AF21	AE22	AF10	AK9	AN11	P29	U29	U2	V1	Y2
A[7]	AE21	AK29	AJ6	AN6	AY4	P27	T29	U1	U1	W2
A[8]	AE20	AK28	AK6	AN7	AY5	N29	T24	T6	V10	AA7
A[9]	AK25	AH25	AG8	AP6	BA4	N28	N30	T1	U2	Y3
A[10]	AH19	AH24	AG7	AP7	BA5	R25	M29	T2	T2	W3
A[11]	AK23	AE23	AL5	AN8	BB4	R28	U26	U9	Y8	AC11
A[12]	AJ21	AD23	AL6	AN9	BB5	N27	U28	R1	W5	Y4
A[13]	AG18	AH21	AC12	AF12	AT10	L30	T28	R2	V5	W4
CKE	AK21	AH23	AM5	AL9	AV8	J30	W30	AA1	AG2	AK3

Note that if there are multiple DDR2 Memory Controllers on the same LatticeSCM device that operate at the same rate, they can share a common PLL PLL1, in which case the two nets “k_clk” and “k4_clk” will also be common among them. This is accomplished by wrapping each memory controller in its own module that contains all logic except PLL1, connecting each module's internal nets “k_clk” and “k4_clk” to module inputs, instantiating one copy of PLL1 outside the DDR2 modules, and connecting that PLL's outputs to those inputs on each module.

DQS Strobe

The DQS strobe and its associated DQ and DM signals must all reside in banks served by a common edge clock. There is a single edge clock serving the two banks (2-3 or 6-7) on each of the two sides of the device, right and left. The two banks (4-5) on the bottom edge are each served by a separate edge clock.

The bidirectional DQS strobe to/from the DDR2 device can be implemented as either a single-ended or differential signal. Lattice recommends differential, and differential is required at clock rates above 200 MHz.

For differential DQS implementations, the differential pairs must be on A/B pairs (for example, PB17A/PB17B) that are semi-dedicated "PCLKT/C[7:2]_[7:0]" (for example, PCLKT5_2/PCLKC5_2). These pairs feature complementary outputs and differential inputs, and are able to directly drive edge clocks. If IPexpress™ is used to generate the DDR2 memory Controller, the pins assigned will conform to this requirement.

Differential DQS applications also require that the IOBUF preference for the DQS signals have the IO_TYPE changed from SSTL18_II to SSTL18D_II. If the design is generated by IPexpress, this is handled automatically.

K/K# Clocks

The K/K# clock pair (and K_copy/K_copy# pair, if used) must be placed on A/B pairs, since they form a complementary output pair.

In order to minimize skew and noise, the K/K# clock pair should be located on the pins that are the driving PLL's designated direct input pins (refer to Table 13). If this is not possible, then use an input driven by an edge clock. This may seem unusual, since these are outputs rather than inputs, but the driven signal is also fed back into PLL PLL2 to create the read data recapture clock, and it is this feedback that needs to be specially handled.

It is important to minimize clock jitter caused by coupling from noisy neighboring signals (refer to the accompanying discussion, "Selecting a Pin That Has Low Jitter Noise", below).

PLLs

PLL1 in Figure 5 generates a 90° phase shift between the address/data/control lines to the DDR device and the accompanying K/K# clock. PLL1 also performs optional clock frequency multiplication when necessary. No custom adjustment of PLL1 is needed.

PLL2 in Figure 5 performs a 90° phase shift on "k_clk_in", the internally reflected copy of the outbound clock. PLL2 also compensates for the total round-trip delay of the board traces to/from the memory device. The value of this delay is entered into IPexpress as the "DQS Trace Delay Compensation" when the DDR module is generated. In order to achieve optimum performance, especially at high clock rates, this delay value can be tuned for the specific implementation. This tuning need only be performed once, and should be performed using the final board layout. The simplest way to perform this tuning is by iteratively changing the DQS Trace Delay Compensation in IPexpress to determine the range of values that yield correct performance, and then using the "sweet spot" centered in that range. Alternatively, PLL2's behavior can be modified dynamically by writing the relevant parameters, PHASEADJ and CLKOS_VCODEL, via the System Bus. For details on modifying these parameters, refer to TN1098, [LatticeSC sysCLOCK and PLL/DLL User's Guide](#). Note that CLKOS_VCODEL is applied at PLL reset, so a reset must be applied after each change.

DLLs

The DLL is used to generate a 90° phase shift so that the receive DQ data eye is centered on the receiving DQS clock. It uses TRDDLLA (time reference delay) mode to achieve this result.

To achieve maximum performance, it may be necessary to adjust the DLL's ALU function +/- 1 or more taps in order to center the DQS in the DQ eye. The optimum setting should be determined experimentally, using the final board layout. The parameter to adjust is named DCNTL_ADJVAL, and can be set in the DLL's source code file "ddr_trdll.v". Do this by adding two lines similar to the following, which set DCNTL_ADJVAL to -2:

```
/* synthesis DCNTL_ADJVAL="-2" */
// exemplar attribute ddr_trdll_0_0 DCNTL_ADJVAL -2
```

Add each of the lines to the group of similar lines in the code. The value can also be modified dynamically by writing it via the LatticeSCM's SMI Bus, if the DLL has been assigned a unique SMI address (DCNTL_ADJVAL is byte

9). Also, if the device has the ORCAstra interface module implemented in the design, the value can be modified via the ORCAstra GUI.

Since this DLL drives the DCNTL bus, it should be located in the corner of the device adjacent to the DQS signal banks. Note that the clock to the DLL is driven internally, so there is no input pin to be located.

Board Layout and Trace Matching

All DQ, DQS and DM signals within a lane must have PCB board trace lengths matched to within 50 picoseconds, and across lanes, they must be matched to within 150 picoseconds.

All address, control signals, and their clocks (K, K#, K_copy and K_copy#) to the DDR2 device must be matched to within 50 picoseconds.

Lattice recommends simulation of simultaneous switching outputs (SSOs) for the device/package combination for performance targeted to over 200 MHz.

In order to ensure that potential conflicts are resolved and to provide maximum flexibility when assigning resources, Lattice recommends that the LatticeSCM device design be placed and routed in ispLever before commitment of the board design to manufacture.

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 DDR2's VREF pin, noise-free. This involves robust, wide-bandwidth decoupling, and isolation of quiet, noise-sensitive signals from noise sources.

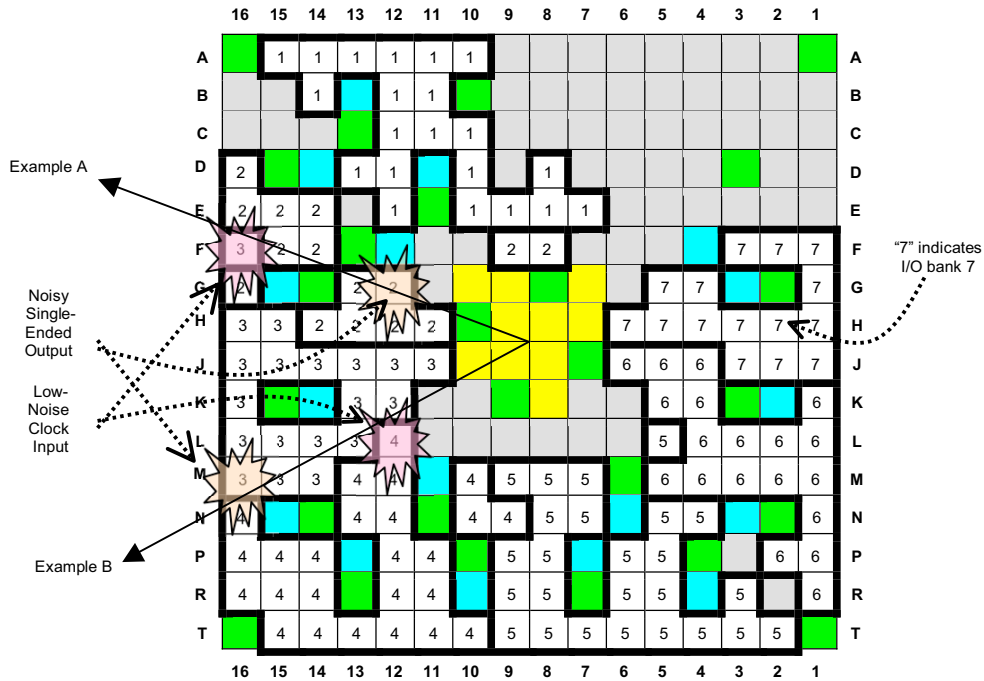
The physical distance between the LatticeSCM device and the DDR2 memory device needs to be minimized, since trace delays, skews and signal degradation will limit overall speed.

Selecting a Pin That Has Low Jitter Noise

When a signal, such as an input reference clock or the DDR2 clocks K/K# or DQS, 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 14 shows how bank groups form clusters around the package for a 256-pin fpBGA. The 256-pin fpBGA was used for simple illustration. The 256-pin package is too small to allow for complete dedicated pinout and thus performance is not guaranteed for this package. See Table 17 for performance data.

Figure 15. Selecting a Pin for Low Jitter Noise



- 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 14 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-die 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.

Timing Specifications

The timing diagrams below show the user interface for command and data. For memory interface timing diagrams, please refer to the data sheet of the memory device.

Figure 16. Write Timing

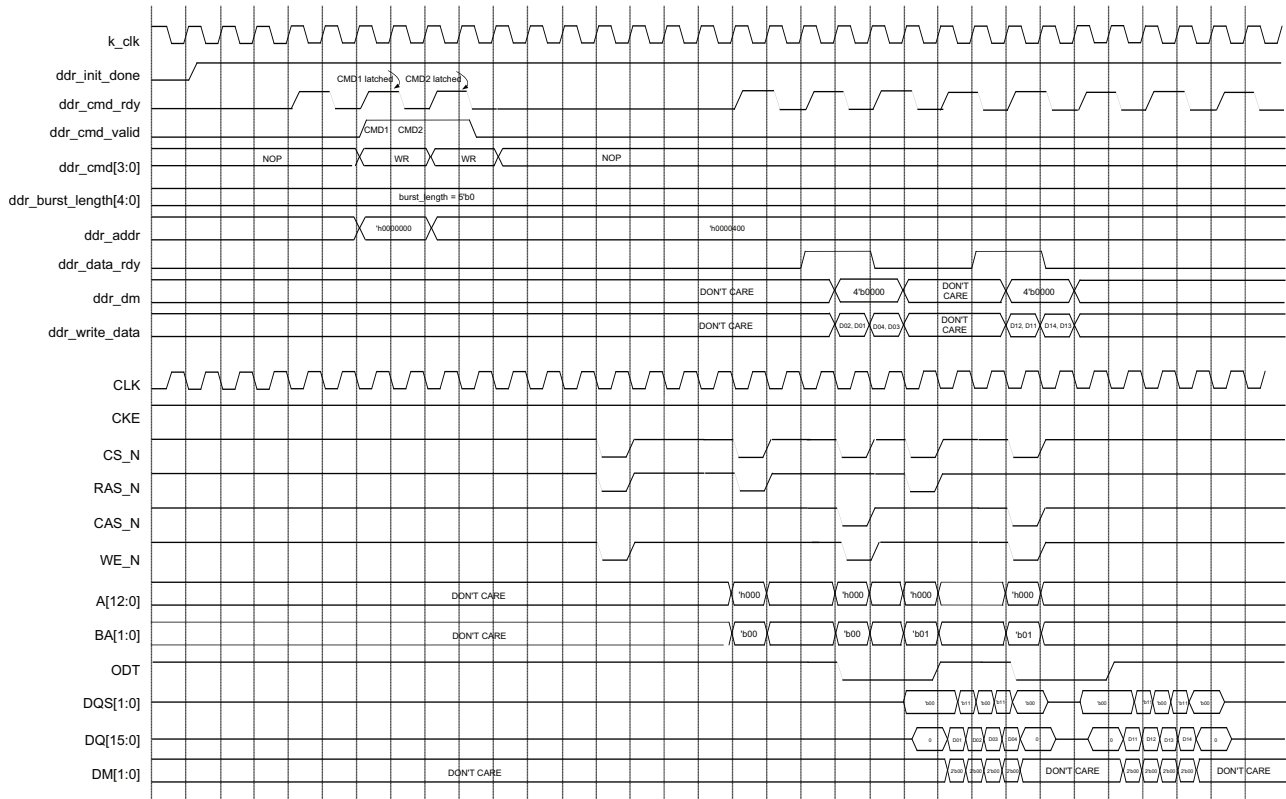
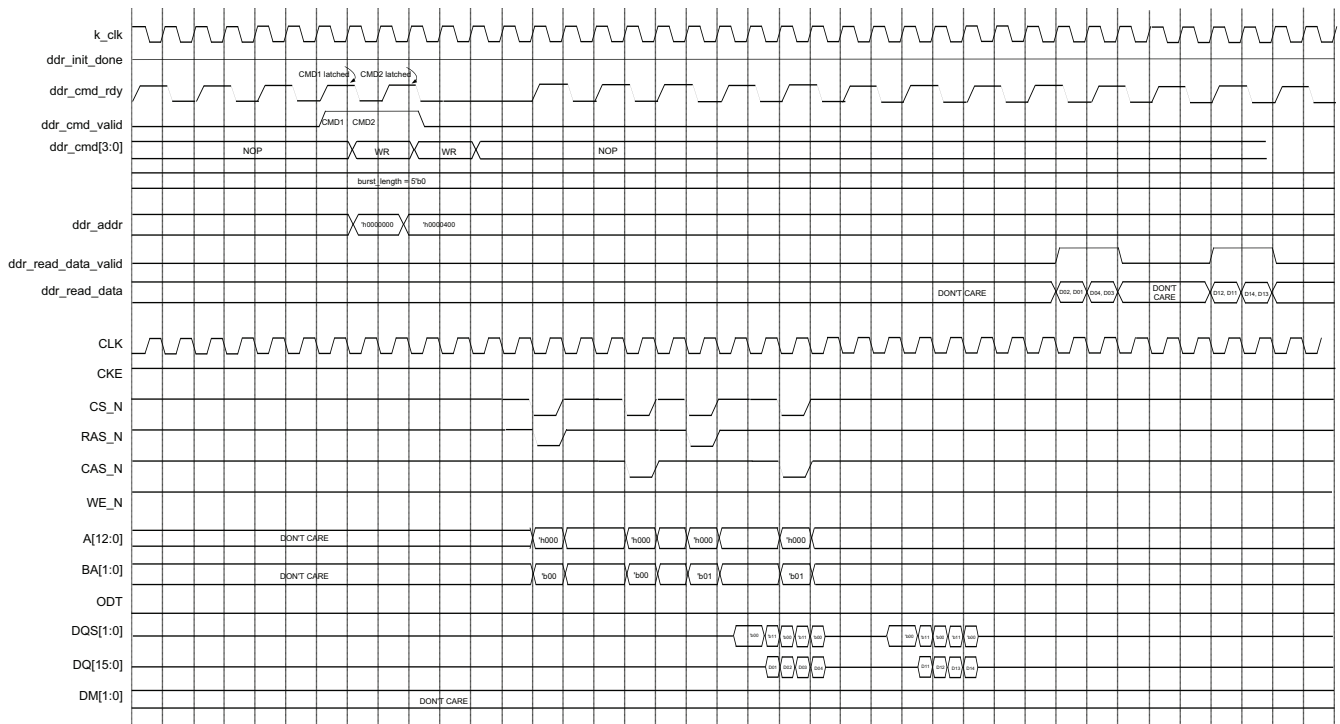


Figure 17. Read Timing

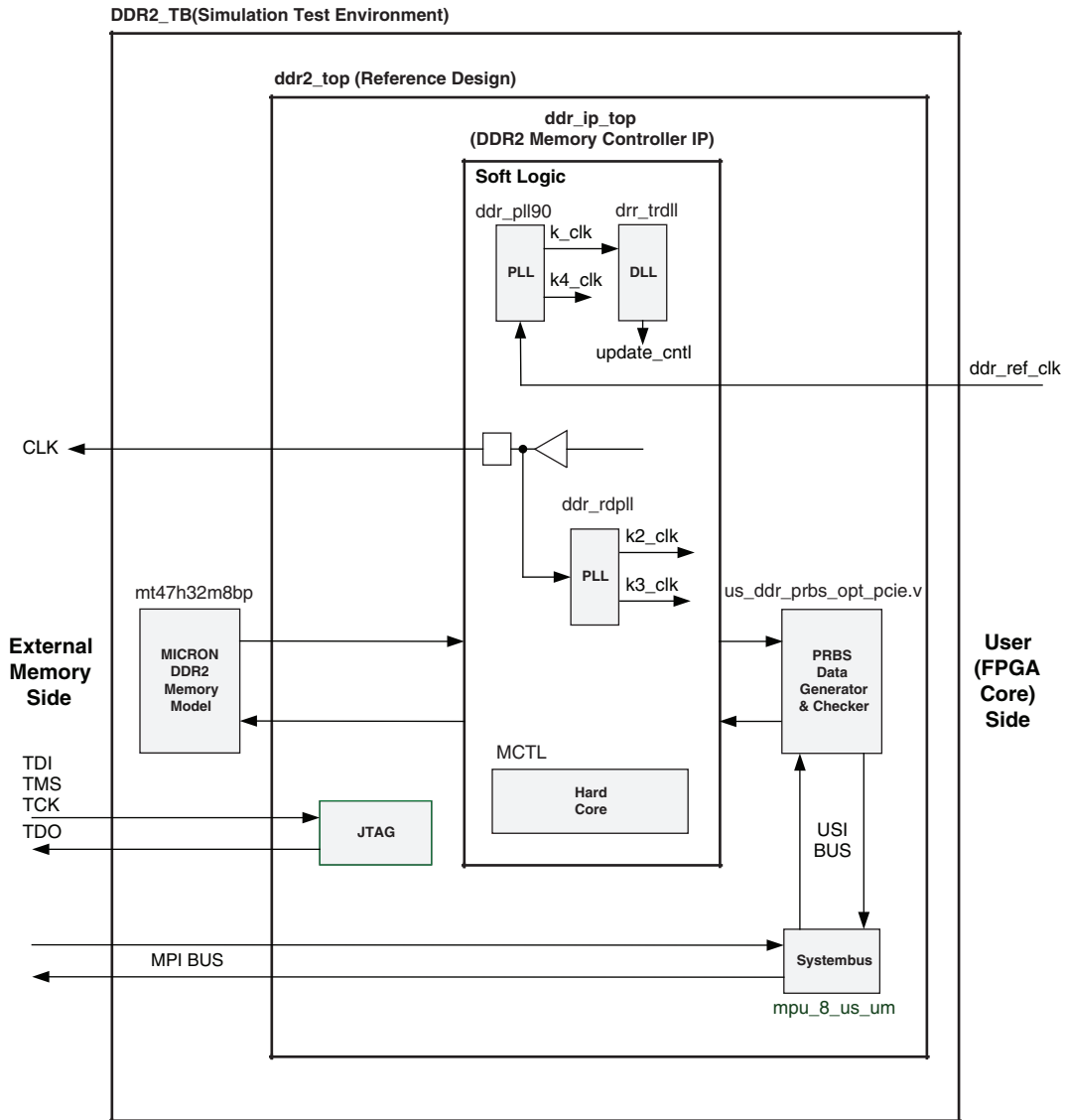


Reference Design/Test Bench

Lattice supplies a reference design along with the DDR or DDR2 controller core. While the core design is intended for use "as is", the reference design provides a framework for testing the core. In the absence of a real user application, the reference design provides synchronization between the external and internal clock domains and pseudo-random data generation.

Using the supplied reference design and test bench as a guide, users can easily customize the verification of the core by adding, removing and customizing tests.

Figure 18. DDR2 Reference Design



DDR/DDR2 SDRAM Memory Controller Performance

Table 17 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 17. DDR/DDR2 SDRAM Memory Controller Performance

Package	VCC = 1.0V ±5%			VCC = 1.2V ±5%			Units
	-5	-6	-7	-5	-6	-7	
Wirebond ¹ : 900	533	533	533	533	533	533	Mbps
Flip-Chip: 1020, 1152, 1704 ²	533	533	533	533	667	667	Mbps

1. For 72-bit configurations in SCM80 and SCM115 devices, a -7 speed grade will be needed to meet 667 Mbps.

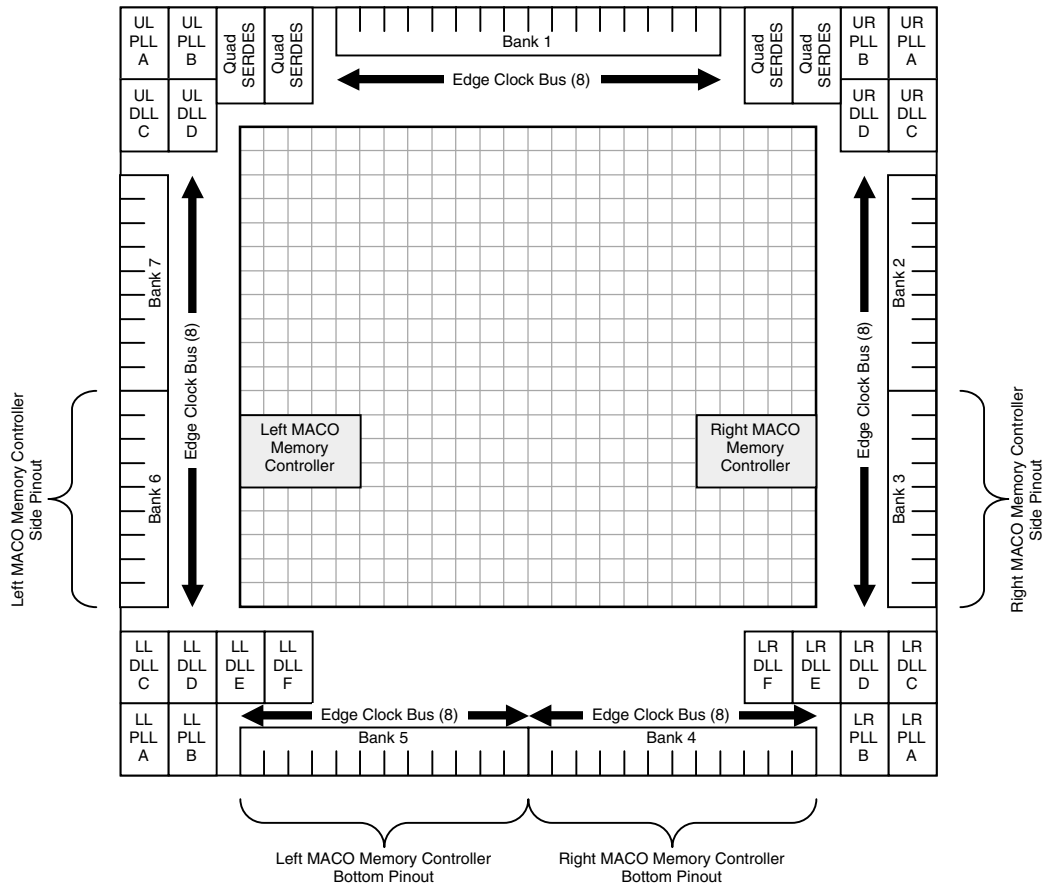
2. The 256-pin package is also wirebond. However, pins are too sparse to permit dedicated pinout of all critical signals and thus timing cannot be guaranteed.

DDR/DDR2 SDRAM Memory Controller On-Chip Resources

Figure 19 illustrates some of the resources on the LatticeSCM device that are available to the DDR/DDR2 SDRAM 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



Conclusion

Applications using DDR and DDR2 SDRAM are becoming popular in FPGA designs. LatticeSCM MACO devices offer a proven, flexible, high-performance interface to these SDRAM with consistent timing margins to meet your design needs. The ease of integration into the LatticeSCM gives the FPGA designer the freedom to choose different variations of SDRAM and reduces the risk of system complexity.

References

- TN1099, [LatticeSC DDR/DDR2 SDRAM Memory Interface User's Guide](#)
- TN1098, [LatticeSC sysCLOCK and PLL/DLL User's Guide](#)
- JEDEC Standard Publication JESD79C, DDR SDRAM Specification, JEDEC Solid State Technology Association
- JEDEC Standard Publications JESD79-2A, DDR2 SDRAM Specification, JEDEC Solid State Technology Association
- Micron Technical Note DDR333, Memory Design Guide for Two-DIMM Unbuffered Systems.

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
April 2006	01.0	Initial release.
June 2007	01.1	Updated Clocking Scheme diagram.
		Added Design Guidelines to Optimize Performance section.
		Added DDR/DDR2 SDRAM Memory Controller Performance section and DDR/DDR2 SDRAM Memory Controller On-Chip Resources section.
		Added DDR/DDR2 MACO Memory Controller Design Kit Directory section.
July 2007	01.2	Added PLLs section.
August 2007	01.3	Updated DDR/DDR2 SDRAM Memory Controller Performance table.
		Replaced references to "LatticeSC" with "LatticeSCM".
		Added LatticeSCM appendix.
January 2008	01.4	Updated User Interface text section.
		Updated Write Timing diagram.
		Updated GUI Dialog Box for DDR/DDR2 Memory Controller Configuration figure.
		Updated GUI Dialog Box for DDR/DDR2 Memory Controller Configuration table.
July 2008	01.5	Updated appendix for LatticeSCM FPGAs.
July 2008	01.6	Document title changed from "LatticeSCM DDR/DDR2 SDRAM Controller MACO Cores User's Guide" to "DDR/DDR2 SDRAM Controller MACO Cores User's Guide".
		Updated Performance and Utilization table footnote in the Appendix for LatticeSCM FPGAs.
July 2008	01.7	Added information regarding READA and WRITEA commands to the Command Decode Logic text section.
May 2010	01.8	Modified DDR/DDR2 SDRAM Memory Controller Performance table and Clocking Scheme figure. Changed references of ddr_ref_clk to k_clk.

Appendix for LatticeSCM FPGAs

Table 18. Performance and Resource Utilization¹

Configuration				Slices	LUTs	Registers	PIOs
Type	Data Width	RA / CA Widths	LatticeSCM Device Speed				
DDR2	16	13 / 9	Typ. (-6)	269	225	387	43
	32		Typ. (-6)	422	321	629	63
	64		Typ. (-6)	729	515	1113	103
	72		Max. (-7)	806	562	1234	113

1. Performance and utilization characteristics are generated using Lattice's ispLEVER[®] 7.1 software. When using this IP core with different software or in a different speed grade, performance may vary. Not all configurations will fit on smaller LatticeSCM devices. These results are from Synplify Pro v9.4L.

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.