QDRII+ SRAM Controller MACO Core

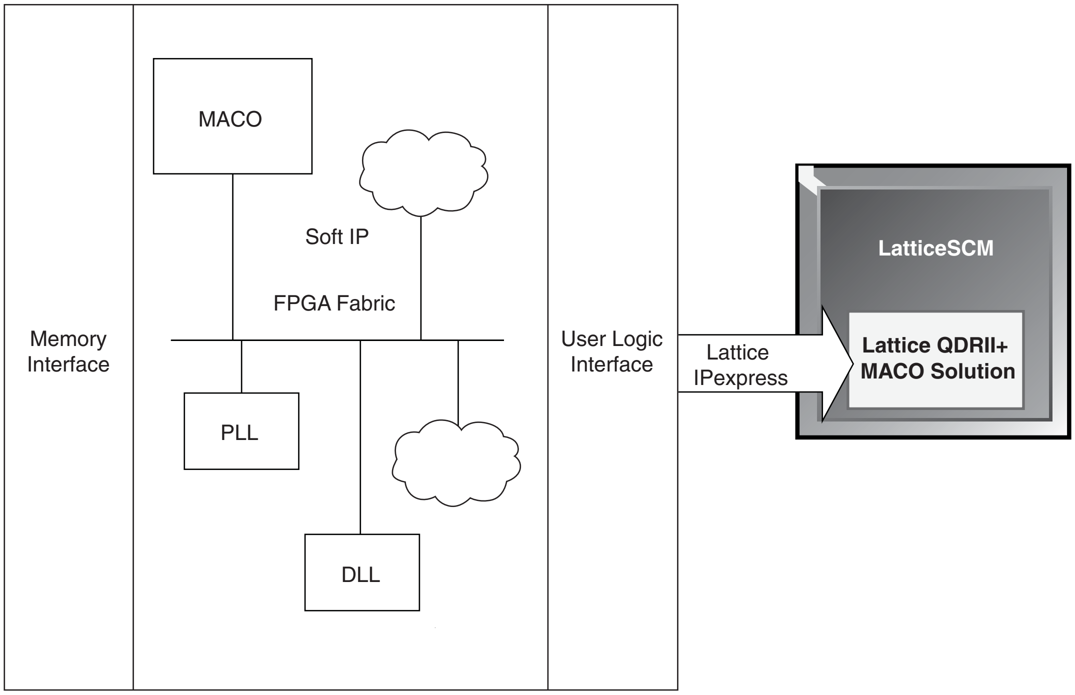
User's Guide

June 2008
ipug45_01.5

Introduction

Lattice's QDRII and QDRII+ (QDRII/II+) SRAM Controller MACO™ core assists the FPGA designer's efforts by providing pre-tested, reusable functions that can be easily plugged in, freeing designers to focus on their unique system architecture. These blocks eliminate the need to "re-invent the wheel," by providing industry-standard QDRII/II+ memory controller modules. These proven cores are optimized utilizing the LatticeSCM™ device's MACO architecture, resulting in fast, small cores that use the latest architecture to its fullest.

Figure 1. Lattice Semiconductor 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 assists 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 on-line Help systems. For more information on the ispLEVER design tools, visit the Lattice web site at www.latticesemi.com/software.

Overview

The second generation Quad-Data-Rate (QDRII/II+) Static Random Access Memory (SRAM) Controller is a general-purpose memory controller that interfaces with industry standard QDRII/II+ SRAM. The controller can be configured to function in two-word burst or four-word burst modes. It can also be configured to have an 18-bit bus or a 36-bit data bus. The data is transferred on both edges of the clock, doubling the rate of data transfer. Separate read and write data buses again double the data rate.

This user's guide explains the functionality of the Lattice's QDRII/II+ Controller core.

Features

- Interfaces to industry standard QDRII/II+ SRAM
- Supports QDRII SRAM memory devices operating up to 250MHz
- Supports QDRII+ SRAM memory devices operating up to 375MHz (highest speed grade)
- FPGA can be configured for 18-bit or 36-bit read and write memory data buses (on FPGA, 36-bit or 72-bit data buses)
- Shared address bus can be configured from 17 bits to 20 bits wide
- Programmable burst lengths of two or four
- Maximum read/write blocks of 31 consecutive locations

Core Deliverables

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

Getting Started

Requirements to implement a MACO core include:

- ispLEVER 7.0 or later
- MACO design kit
- MACO license file

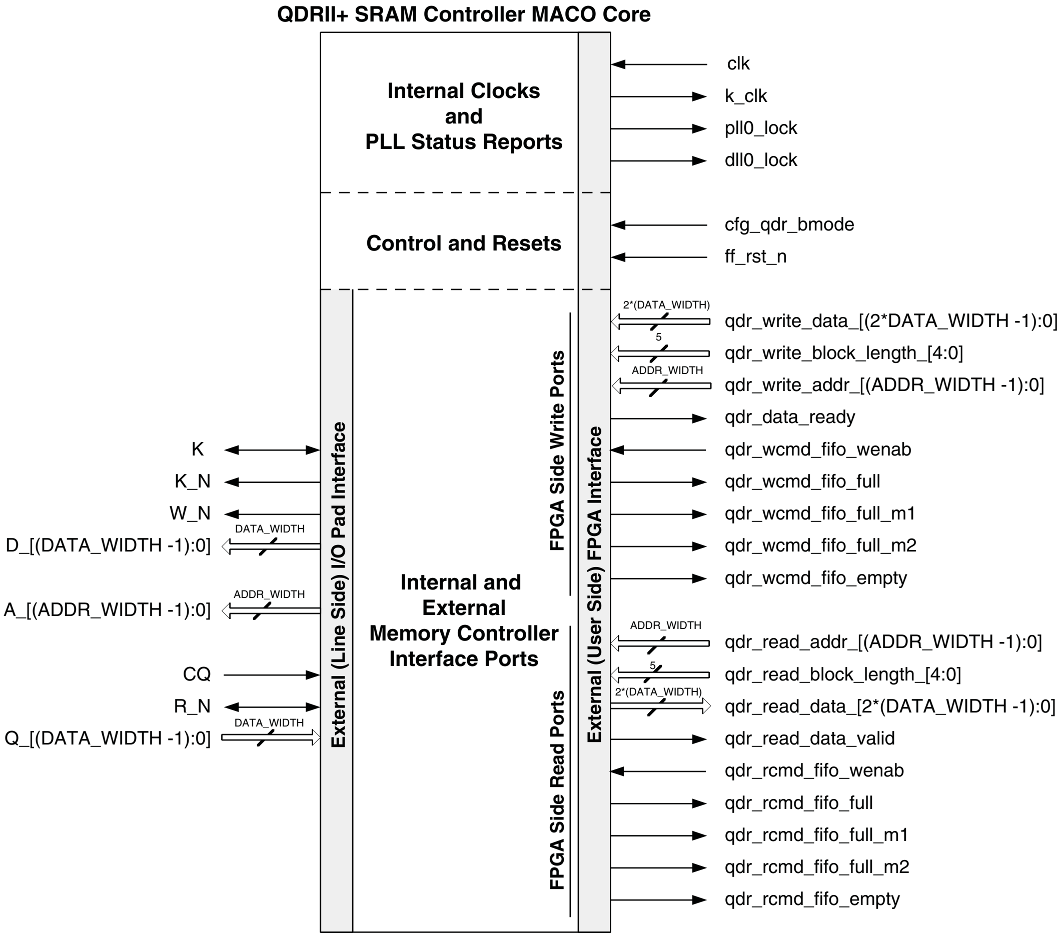
For information on obtaining the above requirements, please contact your local Lattice Semiconductor sales representative.

Functional Description

The QDRII/II+ 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 a permanent part of the device.

Together, these components are provided as intellectual property (IP) by Lattice Semiconductor in a single unit, called qdr_ip_top. This should be instantiated as a single component in the user's design. Figure 2 depicts the interface to the qdr_ip_top.

Figure 2. QDR Controller Core, qdr_ip_top



There are two major interfaces to the qdr_ip_top, the FPGA User Application Interface and the QDRII/II+ SRAM interface. The FPGA User Application Interface communicates with the on FPGA application logic designed by the user. The QDRII/II+ SRAM interface communicates directly with the FPGA pins connected to the external SRAM device. No additional user logic is required between the QDRII/II+ Controller core and the QDRII/II+ SRAM.

Differences Between QDRII and QDRII+

The LatticeSCM QDR Memory Controller supports both the QDRII and the QDRII+ protocols. The QDRII+ protocol has been introduced as a higher-speed enhancement to the earlier QDRII protocol. QDRII+ incorporates the signal QVLD, which accompanies the Q bus and indicates valid data on that bus. QVLD is edge-aligned with CQ/CQ#, and precedes the valid Q data by one-half clock cycle. The differences between QDRII and QDRII+ are summarized in Table 1.

Table 1. Differences Between QDRII and QDRII+

Feature	QDRII	QDRII+
Bit Rate per Data Bus Signal (Max)[1]	400 Mbps*	750 Mbps*
Total Bandwidth for 36-Bit Read/Write Buses	28.8 Gbps	54.0 Gbps
QVLD Support	No	Yes
C Clock Support	Yes	No
Burst Mode Size	2-Word 4-Word	4-Word
Read Latency	1.5 Clocks	2 Clocks 2.5 Clocks
I/O	1.5V HSTL 1.8V HSTL	1.5V HSTL

1. See the QDRII/II+ Memory Controller Performance table in this document for device-specific supported speeds.

Parameter Descriptions

Several configuration and timing parameters must be set before the QDRII/II+ SRAM 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.

The QDRII/II+ IP parameters include clocking preferences. The user can customize the width of the address and data buses and can choose between 4- or 2-word memory modes (4-word only for QDRII+). Sizing of the write and read command FIFO is also permitted.

Below is an example of the "qdr2_define.v" file generated by IPexpress for a QDRII/II+ memory application. This file incorporates the user's design-specific information that is processed for the HDL generation.

```
`define MCTL25
`define QDR_II_PLUS
`define QDR_DATA_18
`define QDR_ADDR_WIDTH       18
`define MAX_QDR_ADDR_WIDTH     20
`define QDR_4WB
`define QDRPLS_2P0L_4WB
`define WCMD_FIFO_ASIZE 2
`define RCMD_FIFO_ASIZE 2
`define PINOUT_BOTTOM
```

Internal PLL and DLL

A PLL is used to derive the internal clock, k_clk, from the reference clock, clk. The user can define the relationship between these two clocks via a setting in the GUI. The pll0_lock output can be used to determine when the PLL frequency is locked.

The SRAM clocks, K and K_N, operate at the same frequency as k_clk. For this reason, the user should set the PLL to produce a clock frequency that matches the desired memory operation frequency.

The QDRII/II+ SRAM standard requires the clocks to the memory to be phase-shifted 90 degrees with respect to data. Likewise, it is necessary to shift the echo clock, CQ, coming back from the SRAM, by 90 degrees. This keeps the clocks in the center or "eye" of the data, providing ample setup and hold times. This is accomplished by a DLL internal to qdr_ip_top.

I/O Signals

System Signals

Table 2 shows the system clock and signals. All signals are active-high unless otherwise noted. These signals are in the reference clock domain. All other signals are in the k_clk clock domain.

Table 2. System Clock and Signals

Signal Name	Signal Direction (I/O)	Signal Width	Signal Description
clk	I	1	Reference Clock. This clock is the input to the QDR_IP PLL.
ff_rst_n	I	1	System Reset (active-low). Synchronized to K clock.
cfg_qdr_bmode1	I	1	QDR Burst Mode Configuration 1 = 2 word burst (QDRII only) 0 = 4 word burst This signal is static, not clocked. It should be assigned to the value set in the GUI.

1. This signal is set by one of the parameters set by the ispLEVER GUI.

User Application Interface Signals

The QDRII/II+ Controller core provides a simple FIFO-based interface to receive read and write commands. Since the FIFOs reside in the soft IP, the user may change their depth via the GUI. By default, these FIFOs are four commands deep.

When the write enable signal for either FIFO goes high, the data present on the address and block length buses will be written on the rising edge of k_clk. The width of the address bus can also be varied by a setting in the GUI, to match the address bus of the memory in use. The maximum block length is fixed at 31.

Each FIFO provides empty and full signals. In addition, they provide full minus one and full minus two signals two provide advanced warning and avoid loss of commands.

Two buses and two handshake signals are provided to manage data traffic. When the QDRII/II+ Controller drives qdr_data_rdy active, it indicates that it has accepted the previous value on the write data bus and is ready for a new one. When it drives qdr_read_data_valid active, it indicates that the data on the read bus is valid. The read and write bus widths are configured via the GUI when the user selects either 18-bit or 36-bit memory data words.

Table defines the signals that communicate data and control between the user application and the QDRII/II+ Controller core. All signals are active-high unless otherwise noted. These signals are in the k_clk clock domain.

Table 3. FPGA Application Interface I/O Signals

Signal Name	Signal Direction (I/O)	Signal Width	Description
qdr_write_addr	I	17-20	Write address.
qdr_write_block_length	I	5	Write block length.
qdr_wcmd_fifo_wenab	I	1	Causes write command address and block length to be written.
qdr_read_addr	I	17-20	Read address.
qdr_read_block_length	I	5	Read block length.
qdr_rcmd_fifo_wenab	I	1	Causes read command address and block length to be written.
qdr_write_data	I	36/72	Write data bus.
k_clk	O	1	Internal clock, derived from system clock, clk.

Table 3. FPGA Application Interface I/O Signals (Continued)

Signal Name	Signal Direction (I/O)	Signal Width	Description
qdr_wcmd_fifo_empty	O	1	Write command FIFO is empty.
qdr_wcmd_fifo_full	O	1	Write command FIFO is full.
qdr_wcmd_fifo_full_m1	O	1	Write command FIFO is almost full (minus 1).
qdr_wcmd_fifo_full_m2	O	1	Write command FIFO is almost full (minus 2).
qdr_rcmd_fifo_empty	O	1	Read command FIFO is empty.
qdr_rcmd_fifo_full	O	1	Read command FIFO is full.
qdr_rcmd_fifo_full_m1	O	1	Read command FIFO is almost full (minus 1).
qdr_rcmd_fifo_full_m2	O	1	Read command FIFO is almost full (minus 2).
qdr_data_rdy	O	1	The controller has accepted the data on qdr_write_data.
qdr_read_data_valid	O	1	Accompanies valid data on bus qdr_read_data.
qdr_read_data	O	36/72	Read data bus.

QDR SRAM I/O Signals

This group of signals provides a standard interface to a QDRII/II+ SRAM device. The outputs consist of a clock and its inverse, a read strobe, write strobe, address bus and a write data bus. The width of the address bus and data bus are both configured via the GUI. The controller provides an internal DLL to shift the clocks by 90 degrees. This is done to provide adequate setup and hold time for the SRAM address and data input.

The inputs consist an echo clock and the read data bus. The width of the read data is also determined by a setting in the GUI. The echo clock is sent from the QDRII/II+ SRAM along with the read data. This clock is used to account for data-flight time across the board. Since both data and clock are in phase, the controller uses its internal DLL to shift this clock by 90 degrees, insuring adequate setup and hold time for the read data.

Table 4 shows the signals connecting the QDRII/II+ Controller to the QDR SRAM. All signals are active-high unless otherwise noted.

Table 4. QDR SRAM Interface I/O Signals

Signal Name	Signal Direction (I/O)	Signal Width	Signal Description
K	O	1	Input. K is the Memory Controller clock, delayed 90°.
K_N	O	1	KN is the inverse of K.
A	O	17-20	Address bus.
D	O	18/36	Write data bus.
W_N	O	1	Active-LOW write enable.
R_N	O	1	Active-LOW read enable.
CQ	I	1	CQ is the clock for the read data bus, "Q". Note: the CQ# signal from the QDRII/II+ SRAM is not used. Instead, both the rising and falling edges of CQ are used to clock incoming data.
Q	I	18/36	Read data bus.
QVLD	I	1	Valid signal for read data bus Q.

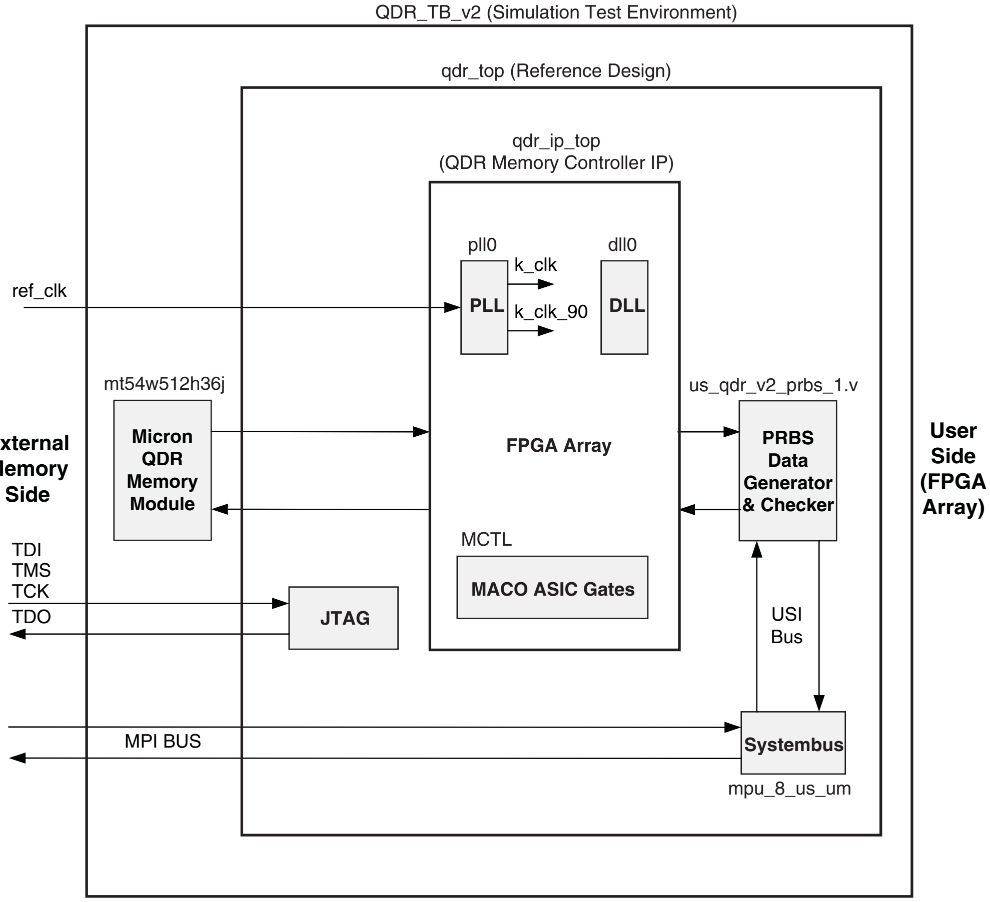
Reference Design/Test Bench

Lattice supplies a reference design along with the QDRII/II+ 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. The reference design is included in this package to demonstrate how a design using the QDRII/II+ Controller core can be implemented.

Reference Design Block Diagram

Figure 3. Block Diagram of QDRII/II+ Controller Reference Design



The QDRII/II+ Controller reference design consists of the following blocks:

1. Pseudo-random data generator
2. System bus
3. JTAG
4. QDR IP module
5. Micron memory module test bench

The external QDRII/II+ SRAM Interface I/O signals run directly between the QDRII/II+ IP core and the pads. There is no extra logic between them in the reference design. Their function is identical to that described in the previous section.

QDRII/II+ MACO Memory Controller Design Kit Directory

The directory structure of the QDRII/II+ MACO Memory Controller IP, as generated by the IPexpress GUI, is shown in Figure 4.

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. This Readme file can be invoked in IPexpress by clicking on the “Help” button of the GUI, as shown in Figure 5. It can also be found in the qdr_maco_eval directory.

Figure 4. QDRII/II+ MACO IP Design Kit Directory Structure

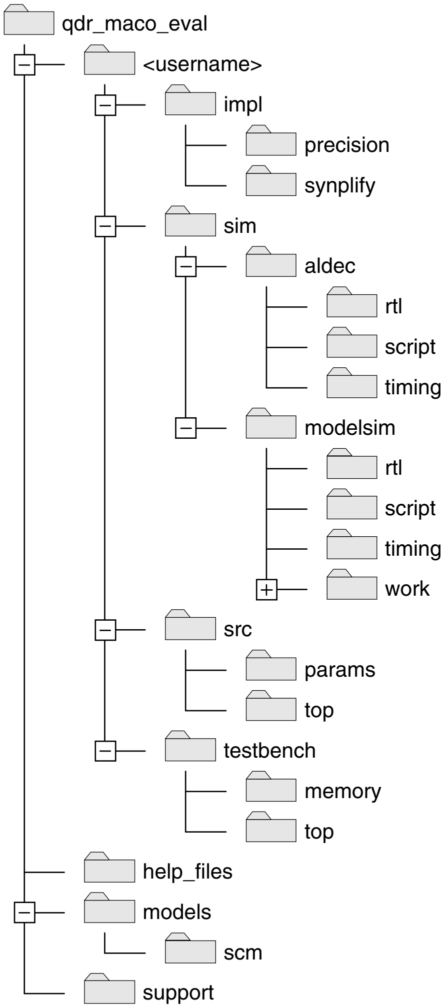
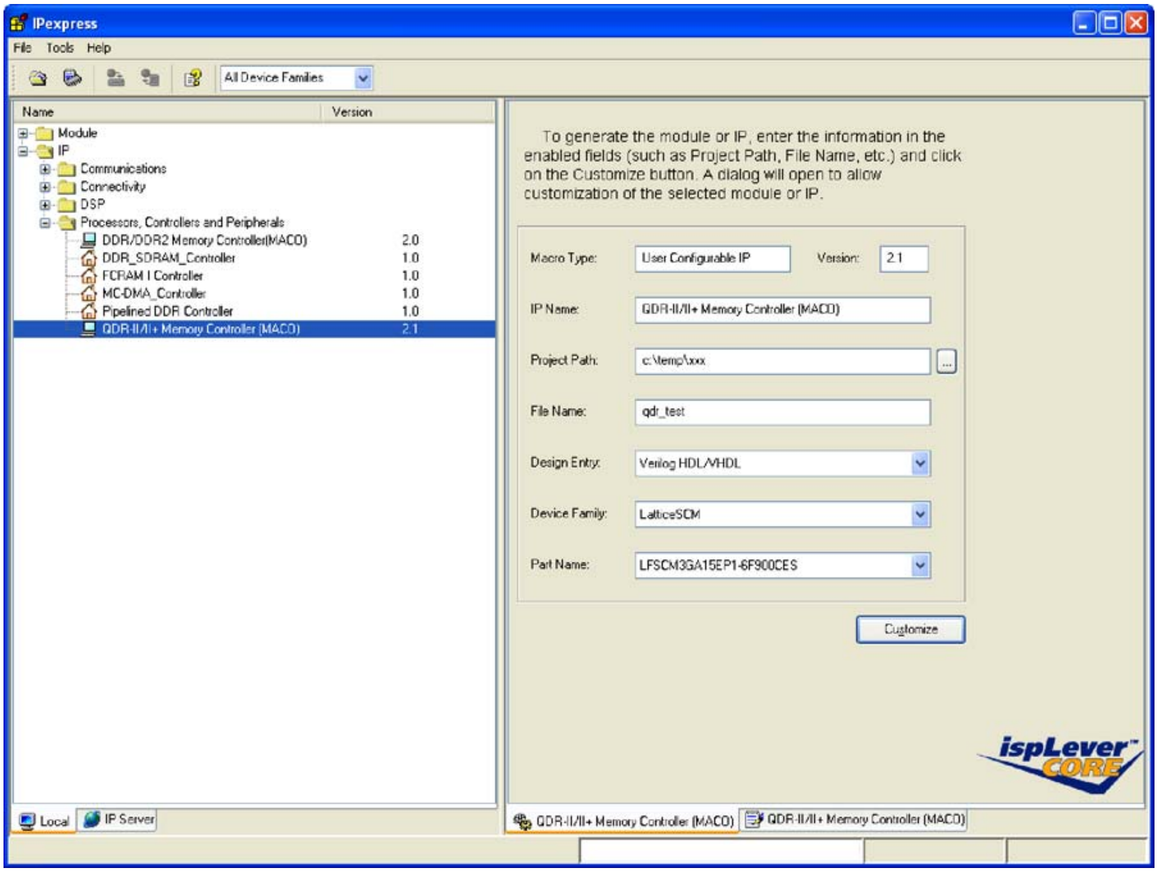


Figure 5. GUI Dialog Box for QDRII/II+ Memory Controller



Name	Version
Module	
IP	
Communications	
Connectivity	
DSP	
Processors, Controllers and Peripherals	
DDR/DDR2 Memory Controller(MACO)	2.0
DDR_SDRAM_Controller	1.0
FCRAM I Controller	1.0
MC-DMA_Controller	1.0
Pipelined DDR Controller	1.0
QDR-II/II+ Memory Controller (MACO)	2.1

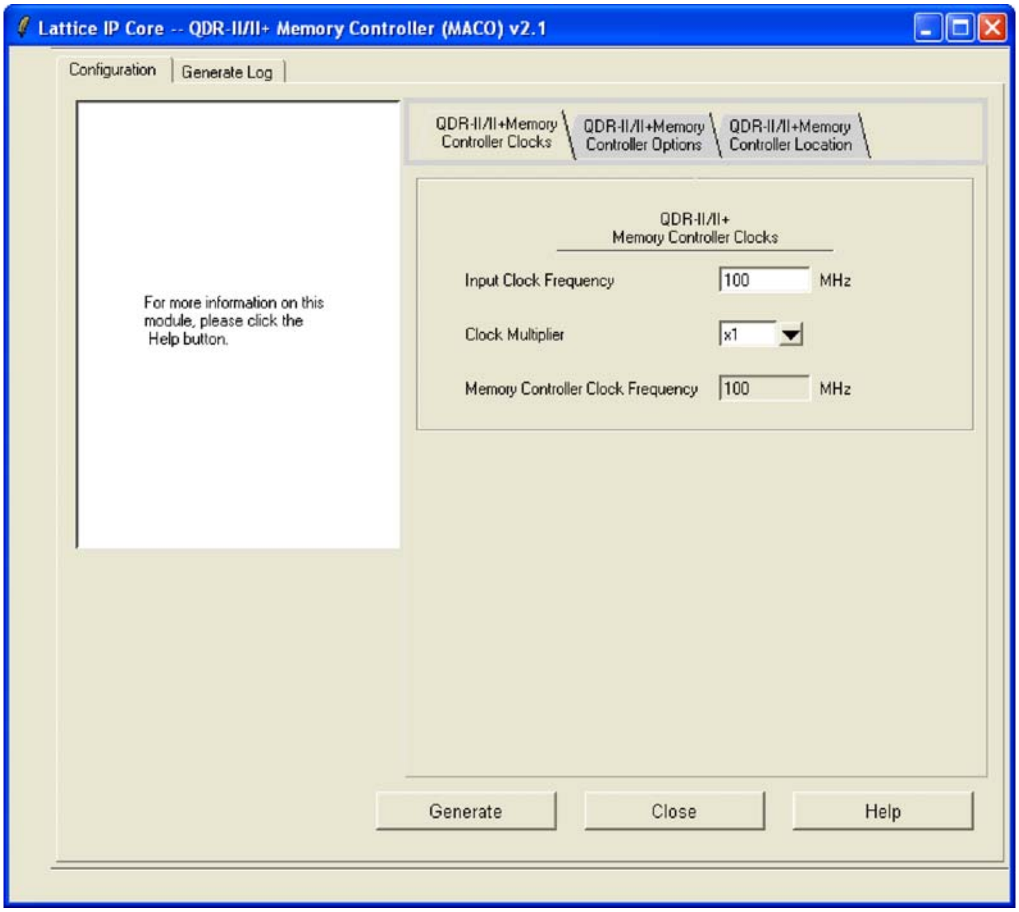
To generate the module or IP, enter the information in the enabled fields (such as Project Path, File Name, etc.) and click on the Customize button. A dialog will open to allow customization of the selected module or IP.

Macro Type: User Configurable IP Version: 2.1
IP Name: QDR-II/II+ Memory Controller (MACO)
Project Path: c:\temp\xxx
File Name: qdr_test
Design Entry: Verilog HDL/VHDL
Device Family: LatticeSCM
Part Name: LFSCM3GA15EP1-6F900CES

Table 5. GUI Dialog Box for QDRII/II+ 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 LatticeSC
Part Name	Select the desired LatticeSC device size, speed grade and package

Figure 6. GUI Dialog Box for QDRII/II+ Memory Controller Clocks



Lattice IP Core -- QDR-II/II+ Memory Controller (MACO) v2.1

For more information on this module, please click the Help button.

QDR-II/II+
Memory Controller Clocks

Input Clock Frequency 100 MHz

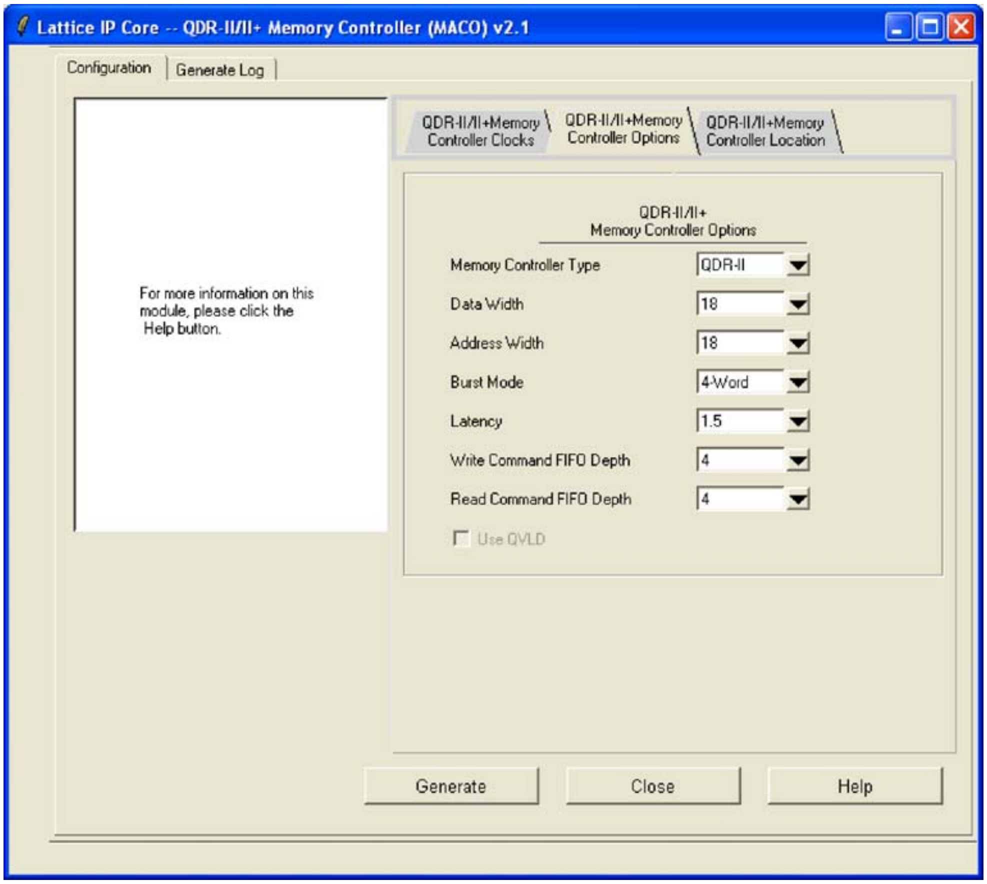
Clock Multiplier x1

Memory Controller Clock Frequency 100 MHz

Table 6. GUI Dialog Box for QDRII/II+ Memory Controller Clocks

Parameter	Description
Input Clock Frequency	Specify the frequency of the input clock to the memory controller
Clock Multiplier	Set this value to the ratio of the desired Memory Controller Clock Frequency and the selected Input Clock Frequency.
Memory Controller Clock Frequency	The memory Controller Clock Frequency is the operating frequency of the QDRII/II+ device. It is calculated by IPexpress, and is set to (Input Clock Frequency) * (Clock Multiplier). Result value is up to 375 MHz.

Figure 7. GUI Dialog Box for QDRII/II+ Memory Controller Options



For more information on this module, please click the Help button.

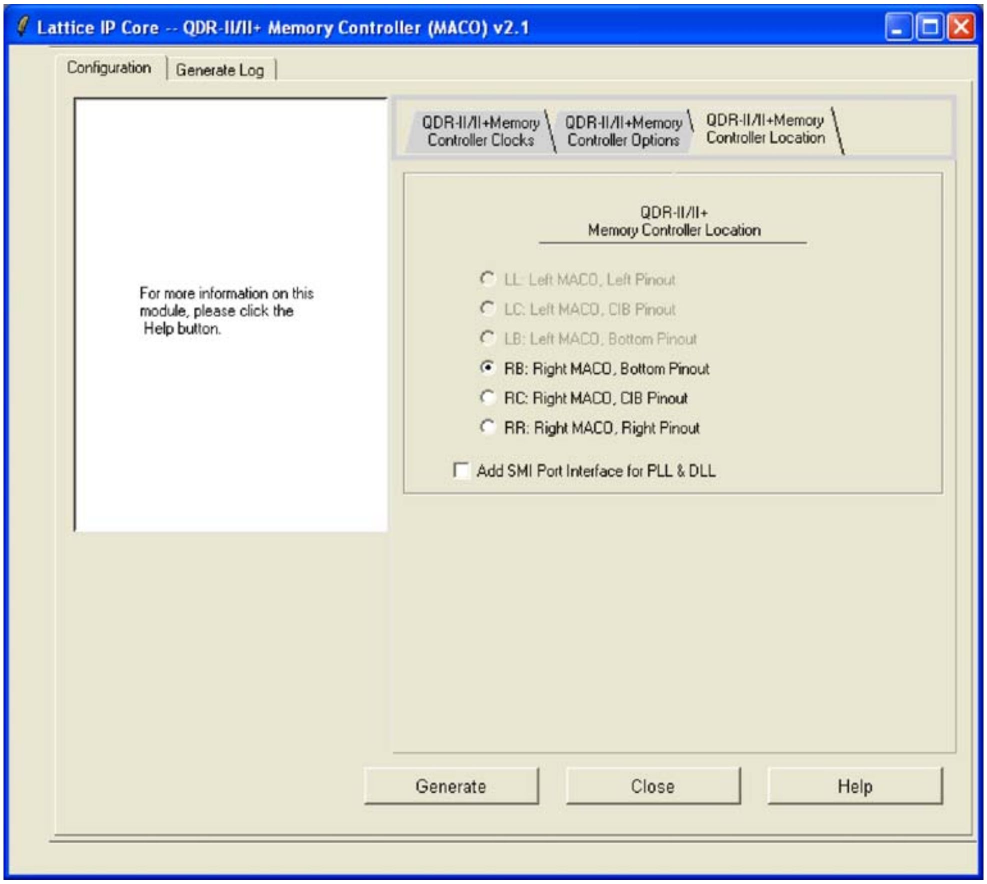
QDR-II/II+
Memory Controller Options

Memory Controller Type: QDR-II
Data Width: 18
Address Width: 18
Burst Mode: 4-Word
Latency: 1.5
Write Command FIFO Depth: 4
Read Command FIFO Depth: 4
☐ Use QVLD

Table 7. GUI Dialog Box for QDRII/II+ Memory Controller Options

Parameter	Description
Memory Controller Type	QDRII or QDRII+
Data Width	Data bus width: 18 or 36 bits
Address Width	Address bus width: 17-20 bits
Burst Mode	2-word or 4-word (depending on memory controller type)
Latency	1.5, 2 or 2.5 (depending on memory controller type)
Write Command FIFO Depth	4, 8, 16, 32 or 64
Read Command FIFO Depth	4, 8, 16, 32 or 64
Use QVLD	Use QVLD (QDRII+ only)

Figure 8. Figure X4. GUI Dialog Box for QDRII/II+ Memory Controller Location



For more information on this module, please click the Help button.

QDR-II/II+
Memory Controller Location

○ 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

☐ Add SMI Port Interface for PLL & DLL

Table 8. GUI Dialog Box for QDRII/II+ Memory Controller Location

Parameter	Description
LL: Left MACO, Left Pinout	The left-side MACO used for the QDRII/II+ controller, and the pinout is on the left side.
LC: Left MACO, CIB Pinout	The left-side MACO used for the QDRII/II+ controller, and the pinout is CIB.
LB: Left MACO, Bottom Pinout	The left-side MACO used for the QDRII/II+ controller, and the pinout is on the bottom side.
RB: Right MACO, Bottom Pinout	The right-side MACO used for the QDRII/II+ controller, and the pinout is on the bottom side.
RC: Right MACO, CIB Pinout	The right-side MACO used for the QDRII/II+ controller, and the pinout is CIB.
RR: Right MACO, Right Pinout	The right-side MACO used for the QDRII/II+ controller, and the pinout is on the left 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 LatticeSC device. If the source is external, it should use the direct input pin for that PLL's CLKI input (refer to Table 9). 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.

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 QDRII/II+ 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.
- 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 9), and observe the recommendations below for minimizing jitter noise.

Clocking Challenges and Solutions

Figure 9 illustrates the clocking network. Several unique features of the LatticeSC architecture are utilized in this design. A PLL [1] is used to perform frequency multiplication of the input clock "refclk", and at the same time to generate a second clock that is 90° lagging, so that the clocks "K" and "K#" to the QDRII/II+ SRAM can transition in the center of the data eye of bus "D".

Both "k_clk" and "k_clk shifted 90°" are typically routed on primary clock nets so that there is very little skew from the ideal 90° offset. The clocks "K" and "K#" are then generated using the same DDR output buffer elements as are used in the buffers for output data and control signals, so that once again very little skew is introduced. These two clocks are generated by simply sending a constant "10" pattern on outputs that are in every other respect identical to the data and control outputs.

A Valid Timing Chain [2] generates a data valid signal at the correct time to line up with the returning read data by duplicating the latency in the external QDRII/II+ SRAM and board routing. This is necessary because there is nothing returned from the QDRII/II+ SRAM with the read data to identify the valid data. Note that for 2-word bursts, the valid is asserted for one full clock (two half-clocks), and for 4-word bursts, it is asserted for two full clocks (four half-clocks). Note also that the number of registers in the timing chain varies to match the read latency (1.5, 2.0 or 2.5) of the QDRII/II+ SRAM. The Valid Timing Chain straddles two clock domains having the same frequency but different phases, and performs the clock domain transition between them. The phase difference represents all the cumulative delays in the external path: board trace delays (in both directions), and delay from K/K# to CQ/CQ#. The clocking scheme described here can accommodate and compensate for approximately 1/2 clock cycle of variation in this delay.

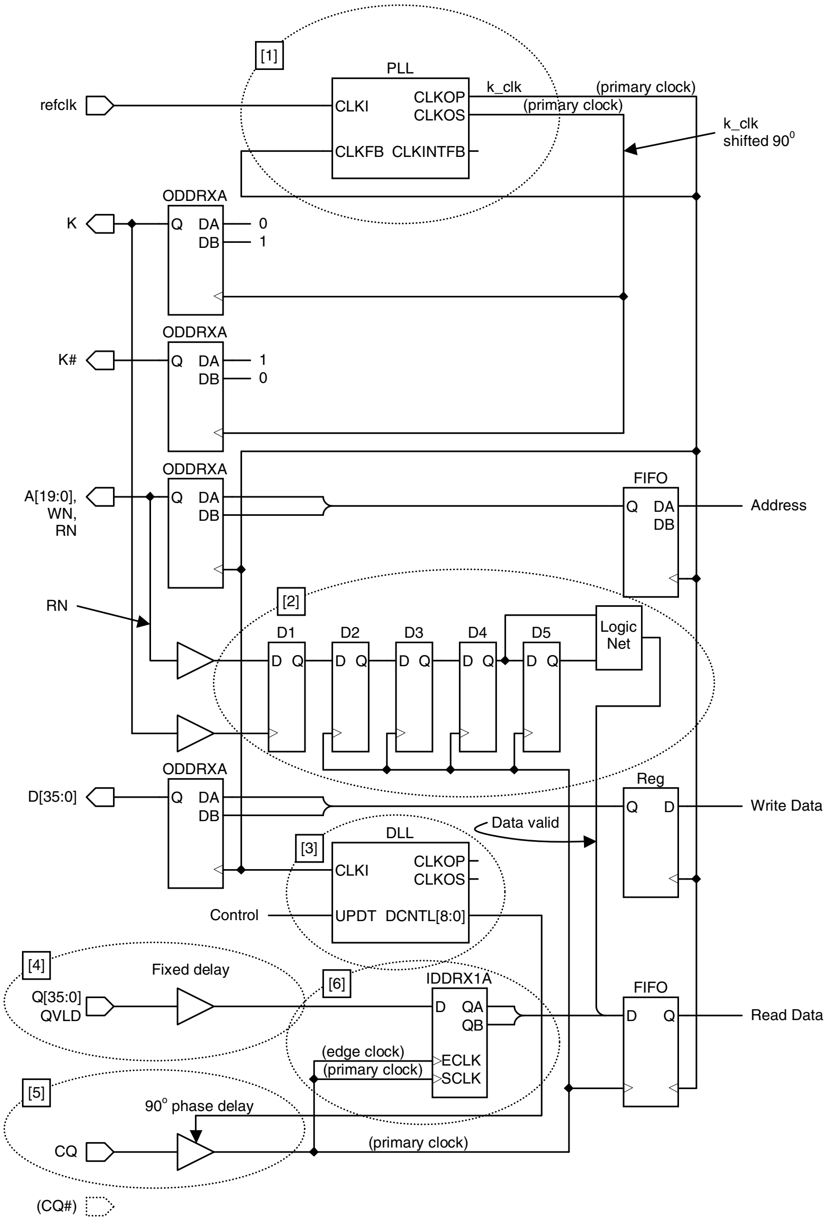
The input registers for the read data bus "Q" and signal "QVLD" [6] require some special clocking, and this need is handled by special hardware capability. The input bus registers have two clock inputs. The first, ECLK, is fed by the edge clock, to receive the data at the earliest time, since the edge clock net has less delay and skew to the I/O registers than the primary clock net. But if this data were to be sent directly to a register clocked by the primary clock, the receiving register's input hold time could be violated. Therefore, the input register also takes a second input clock, SCLK, which is fed the primary clock. The register does not output the data until this clock's edge, avoiding any hold time issues. This clock domain transfer mechanism is built-in to the LatticeSC input buffers, thus allowing operation at highest rates of speed.

For the return read data bus “Q” and its accompanying valid signal QVLD, a DLL [3] is employed to dynamically generate a value that determines the proper delay to cause an effective 90° phase shift on CQ’s input buffer [5], so that it too is positioned in the center of the data it captures. This takes advantage of the fact that the DLL and the input buffers contain matching delay blocks, so that the delay selection value generated in the DLL when it generates a 90° shifted clock can also be used in the input buffer to cause the same phase shift. A 9-bit digital bus communicates this delay selection value from the DLL to the “CQ” input buffer.

The read data is then typically transferred from the “CQ” clock domain to the internal clock domain with the assistance of a synchronous FIFO.

For the “Q” data bus and signal QVLD, the delay elements [4] are also used in an “Edge Clock Injection Match” mode. This compensates for the edge clock routing of the “CQ” input, thereby providing an optimal read data edge. Manual changes to the input delay can also be made to each individual “Q” input pin to compensate for differences in board trace delays.

Figure 9. QDRII+ SRAM Memory Interface Clock and Data Paths



PCB Board Trace Matching

- All A, D, WN, RN and K/K# pins must have PCB board trace lengths matched to within 50 psec.
- All Q, QVLD and CQ 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 QDRII/II+ device'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 QDRII/II+ needs to be minimized, since trace

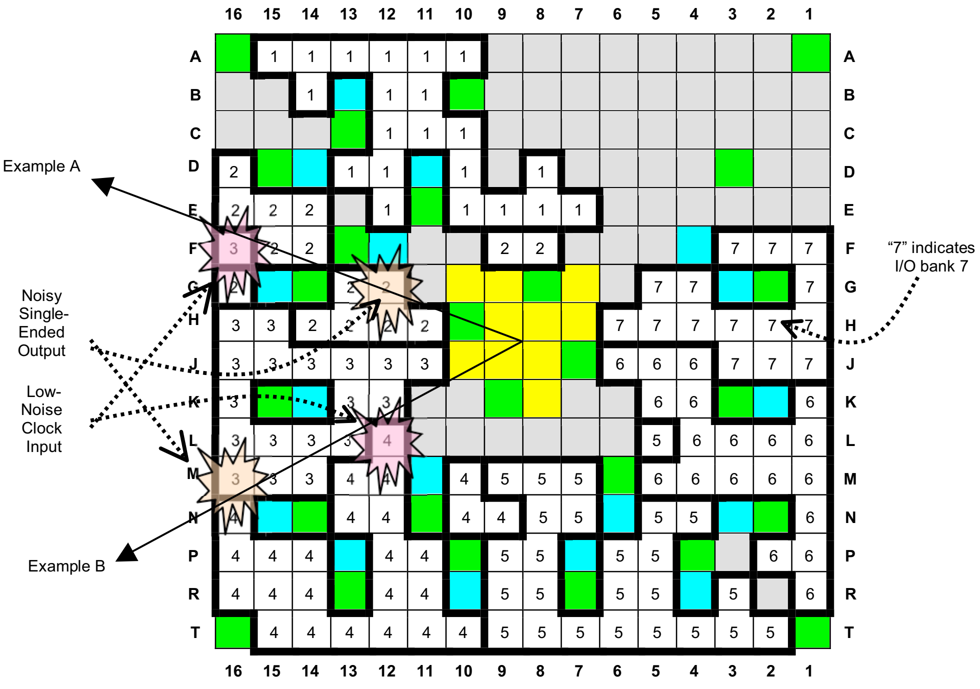
delays, skews and signal degradation will limit overall speed, as previously discussed.

Selecting a Pin That Has Low Jitter Noise

When a signal, such as an input clock or the QDRII/II+ clock K/K# 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 10 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 10 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 10. Selecting a Pin for Low Jitter Noise



	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	🟩	1	1	1	1	1	1									🟩	A
B			1	🟦	1	1	🟩										B
C				🟩	1	1	1										C
D	2	🟩	🟦	1	1	🟦	1		1					🟩			D
E	2	2	2		1	🟩	1	1	1	1							E
F	3	2	2	🟩	🟦			2	2				🟦	7	7	7	F
G	2	🟦	🟩	2	2		🟨	🟨	🟩	🟨		7	7	🟦	🟩	7	G
H	3	3	2	2	2	2	🟩	🟨	🟨	🟨	7	7	7	7	7	7	H
J	3	3	3	3	3	3	🟨	🟨	🟨	🟩	6	6	6	7	7	7	J
K	3	🟩	🟦	3	3			🟩	🟨			6	6	🟩	🟦	6	K
L	3	3	3	3	4							5	6	6	6	6	L
M	3	3	3	4	4	🟦	4	5	5	5	🟩	6	6	6	6	6	M
N	4	🟦	🟩	4	4	🟩	4	4	5	5	🟦	5	5	🟦	🟩	6	N
P	4	4	4	🟦	4	4	🟩	5	5	🟦	5	5	🟩		6	6	P
R	4	4	4	🟩	4	4	🟦	5	5	🟩	5	5	🟦	5		6	R
T	🟩	4	4	4	4	4	4	5	5	5	5	5	5	5	5	🟩	T
	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Optimum Pinout Selection

In order to ensure that the demanding I/O timing requirements of QDRII/II+ 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 11 (for the left-side MACO) and 12 (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 QDRII/II+ 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 also recommends that the LatticeSC device's design be placed and routed before commitment of the board design to manufacture.

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 9 and 10 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 11 and 12 show the designated optimum-performance pins for interfacing the QDRII/II+ Memory Controller to the QDRII/II+ device, for the left-side and right-side MACO respectively.

Table 9. 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 10. 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 11. Preferred Pinout for Left Side Memory Controller

QDR/QDRII Port	Bottom Edge Preferred Pinout				Left Edge Preferred Pinout			
	SC25 900	All 1020	All 1152	All 1152	SC25 900	All 1020	All 1152	All 1152
W_N	AE5	AG28	AJ29	AL34	V4	W25	AA24	AG29
R_N	AJ1	AK32	AN33	AV42	V5	Y26	Y24	AF29
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
A[14]	AE12	AG23	AG23	AM28	R3	T26	V28	AB35
A[15]	AE13	AF22	AG22	AL28	R2	R29	T34	Y38
A[16]	AK4	AG26	AN27	AV35	P2	R30	R34	W38
A[17]	AK5	AG25	AN26	AV34	P3	P31	U30	V42
A[18]	AJ5	AL26	AP27	AY36	N3	P32	T30	U42
A[19]	AJ6	AM26	AP26	AY35	R6	T24	V29	W36

Table 12. Preferred Pinout for Right Side Memory Controller

QDR/QDRII Port	Bottom Edge Preferred Pinout					Left Edge Preferred Pinout				
	SC15 900	**SC25 900**	**All 1020**	**All 1152**	**All 1704**	**SC15 900**	**SC25 900**	**All 1020**	**All 1152**	**All 1704**
W_N	AD25	AH30	AK1	AM2	AU1	Y30	W26	W8	AA11	AG14
R_N	AE26	AG29	AH3	AH6	AL9	AA30	V26	Y7	Y11	AF14
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
A[14]	AK21	AH23	AM5	AL9	AV8	J30	W30	AA1	AG2	AK3
A[15]	AJ19	AH22	AM6	AL10	AV9	M26	Y27	AB6	AC6	AJ9
A[16]	AJ18	AG22	AE12	AP8	AY7	G29	W27	AC6	AD6	AK9
A[17]	AG17	AG21	AD12	AP9	AY8	F29	AA30	AC2	AF4	AK5
A[18]	AH18	AF21	AJ8	AM9	AV10	H28	AA25	AD4	AH2	AL1
A[19]	AH17	AE21	AK8	AM10	AV11	J28	AB25	AD3	AJ2	AM1

Timing Specifications

The timing diagrams in Figures 11 and 12 below show the timing on the QDRII/II+ device interface, and Figures 13 and 14 add the timing on the user interface for command and data.

Figure 11. QDRII/II+ SRAM Interface Timing (4-Word Burst Mode)

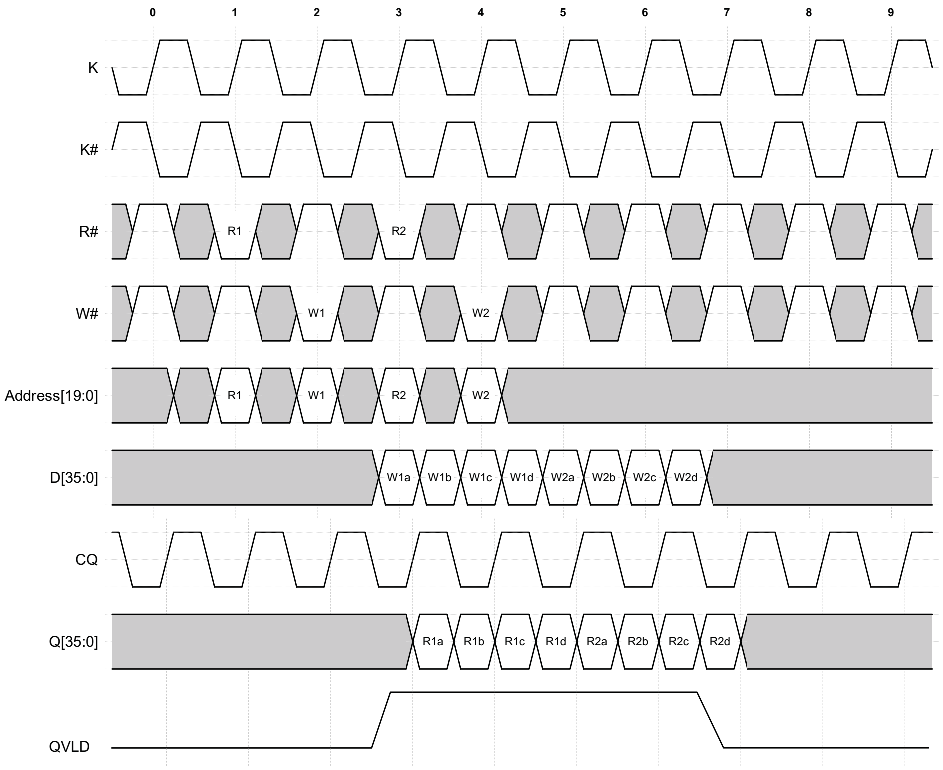


Figure 12. QDRII SRAM Interface Timing (2-Word Burst Mode)

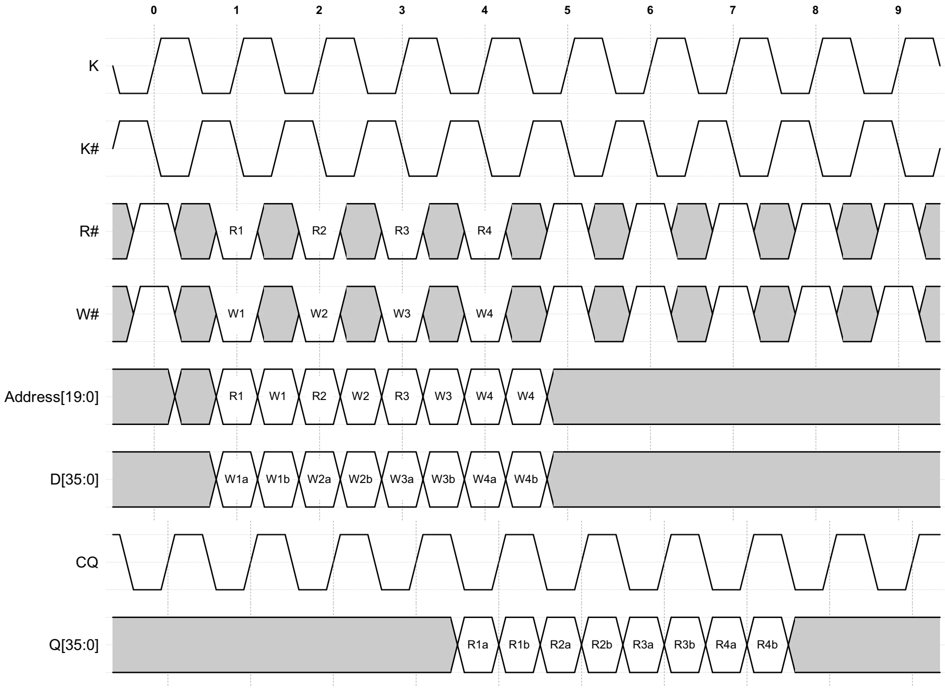


Figure 13. Complete 4-Word Write/Read Sequence, Reading Back Just-Written Data

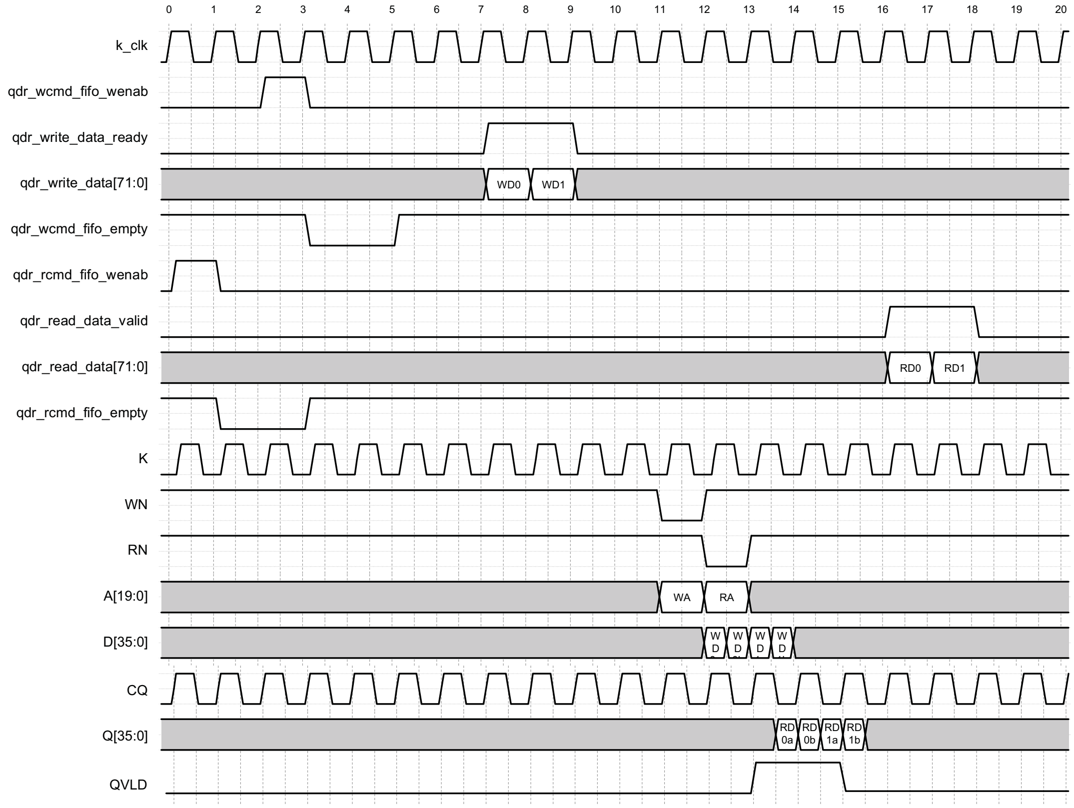
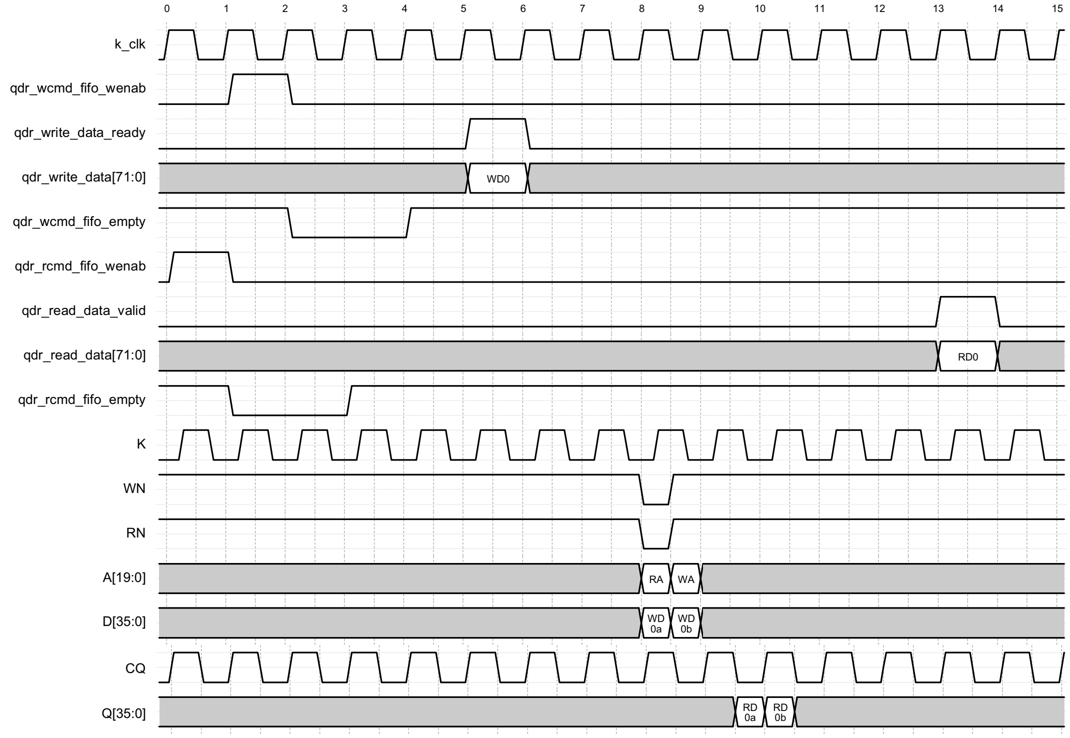


Figure 14. Complete 2-Word Write/Read Sequence, Reading Back Just-Written Data



QDRII/II+ Memory Controller Performance

Table 13 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 13. QDRII/II+ Memory Controller Performance

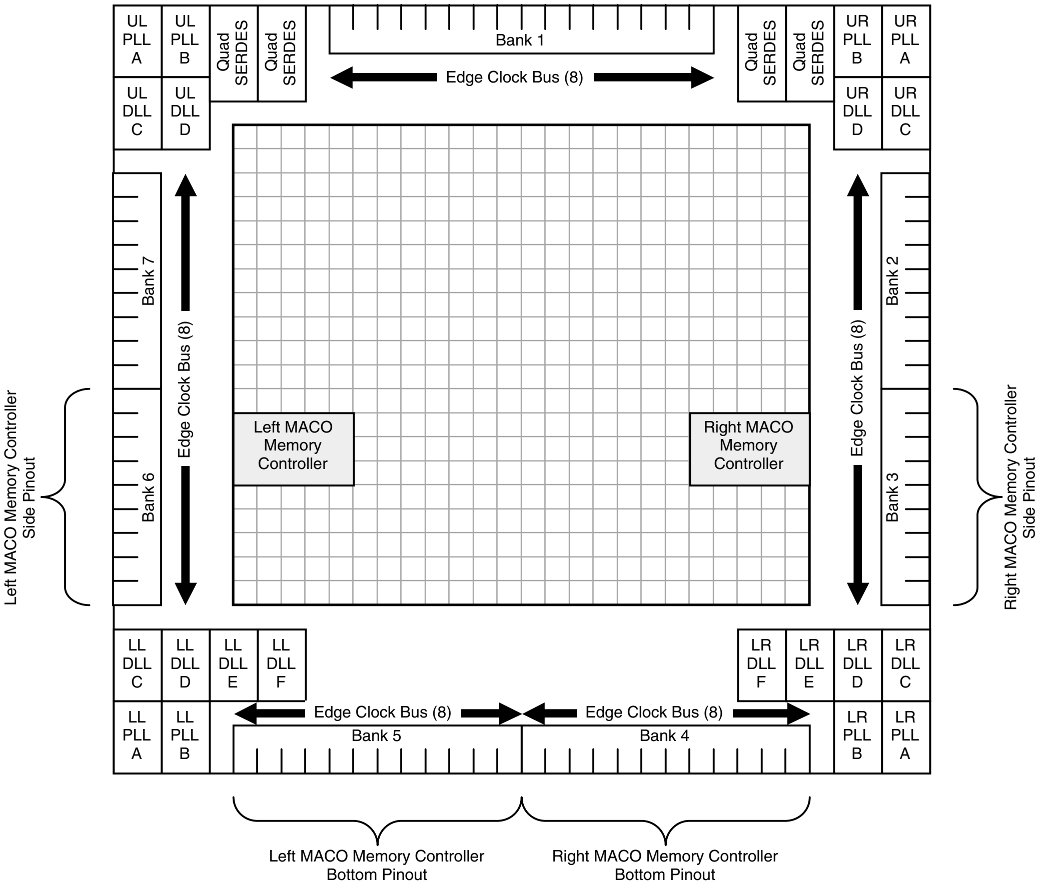
Package	**VCC = 1.0V ±5%**			**VCC = 1.2V ±5%**			**Units**
	-5	**-6**	**-7**	**-5**	**-6**	**-7**	
QDRII	200 400	250 500	250 500	250 500	250 500	250 500	MHz Mbps
QDRII+	275 550	325 650	350 700	325 650	350 700	375 750	MHz Mbps

QDRII/II+ Memory Controller On-Chip Resources

Figure 15 illustrates some of the resources on the LatticeSCM device that are available to the QDRII/II+ 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 15. MACO Memory Controller Resources



Conclusion

Applications using QDRII/II+ SRAM are becoming popular in FPGA designs. LatticeSC MACO devices offer a proven, flexible, and high-performance interface to these SRAM with consistent timing margins to meet your design needs. The ease of integration gives the FPGA designer the freedom to choose different variations of SRAM and reduces the risks of system complexity.

References

- MT54W2MH8J, MT54W1MH18J, MT54W512H36J, 18Mb QDR-II SRAM 4-Word Burst, Micron Technology, Inc., 2003.
- K7R323684B, K7R321884B, 1Mx36 & 2Mx18 QDR-II b4 SRAM, Samsung Electronics Co. LTD., Dec. 2003, Rev 2.0.
- CY7C1411AV18, CY7C1413AV18, CY7C1415AV18, 36-Mbit QDR-II SRAM 4-Word Burst Architecture, Cypress Semiconductor Corp., Feb. 11, 2005.
- QDRII/II+ Evaluation Board Demonstration Design
- Lattice technical note 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
April 2006	01.0	Initial release.
August 2007	01.1	References to LatticeSC changed to LatticeSCM.
September 2007	01.2	Added QDRII+ documentation support.
February 2008	01.3	Updated Features bullets.
March 2008	01.4	Updated GUI Dialog Box for QDRII/II+ Memory Controller Clocks table.
June 2008	01.5	Title changed from "LatticeSCM QDRII/II+ SRAM Controller MACO Core User's Guide" to "QDRII+ SRAM Controller MACO Core User's Guide".
		Updated Features bullets.

Appendix for LatticeSCM FPGAs

Table 14. Performance and Resource Utilization[1]

Configuration									
Type	Data Width	Address Width	Rd/Wr FIFO Depth	Latency	Burst Mode	Slices	LUT4s	Registers	PIOs
QDRII+	18	18	4/4	2.5	4	230	297	233	194
QDRII+	36	18	4/4	2.0	4	342	406	382	194
QDRII	18	18	64/64	1.5	2	453	717	242	194

1. Performance and utilization characteristics are generated using Lattice's ispLEVER® 7.0 software. When using this IP core with different software or in a different speed grade, performance may vary.

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.