



WP383 (v1.0) March 1, 2011

Achieving High Performance DDR3 Data Rates in Virtex-7 and Kintex-7 FPGAs

By: Adrian Cosoroaba

FPGA-based systems frequently require an external memory interface to buffer data that exceeds the capacity of the FPGA's internal memory. This memory interface can often dictate overall system performance because it must provide sufficient read/write bandwidth to keep up with the flow of data into and out of the FPGA.

This I/O performance requirement translates into the need for higher memory interface bandwidths with each new generation of FPGA products. In addition to higher performance, the memory interface solution also needs to be flexible and easy to implement. In many ways, the memory controller and interface design determines the capability of the overall system.

This white paper describes various memory interface and controller design challenges and the 7 series FPGA high-performance solution that achieves a 1.866 Gb/s DDR3 data rate for Virtex®-7 and Kintex™-7 FPGAs.

Memory Interface Trends and Xilinx Solutions

With each generation of double data rate (DDR) SDRAMs, the data rate per pin has increased to satisfy higher system performance requirements.

FPGA-based memory controller implementations have been challenged to keep up with these trends. There are two important aspects of the interface and controller implementation that present challenges in conjunction with these higher data rates:

- Read data capture in the FPGA physical layer (PHY)
- Controller implementation in the FPGA logic fabric

Read Data Capture

Read data capture becomes challenging when data rates increase because the read data valid window gets smaller and timing margins shrink. The FPGA I/O block and clocking elements need to provide a precise capture mechanism, and the data capture clock has to be maintained within the data valid window, even in an environment where voltages and temperatures are inconstant. See [Figure 1](#).

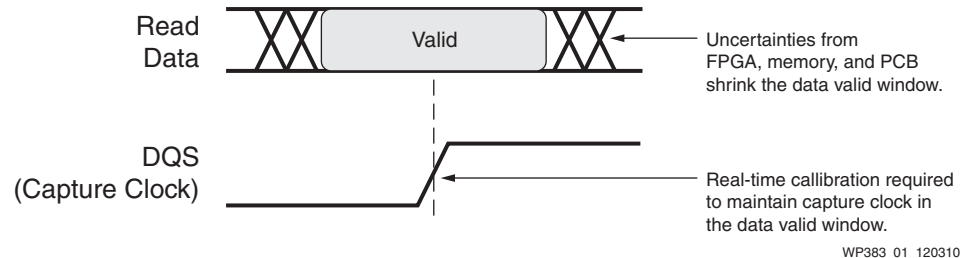
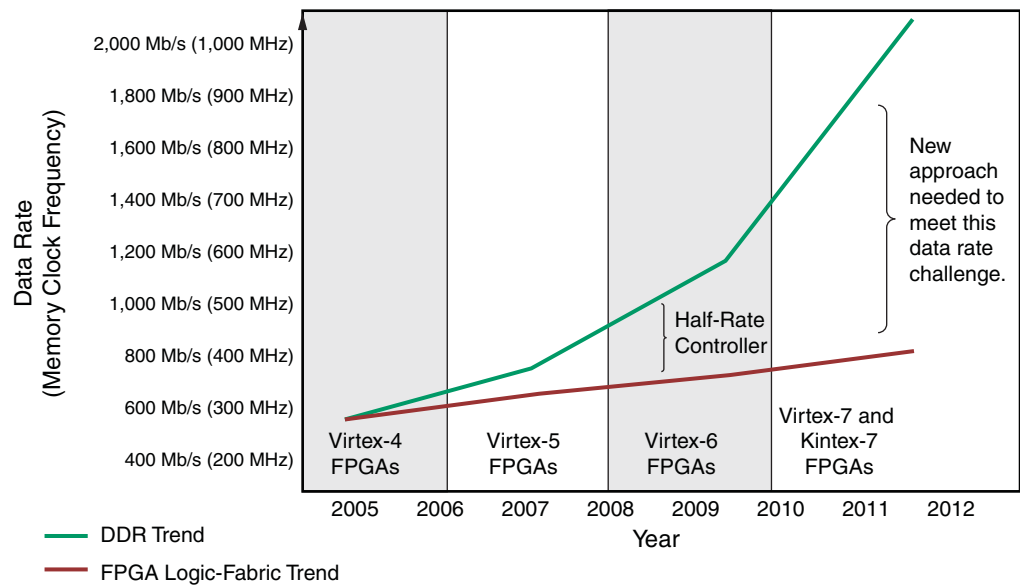


Figure 1: **Clock to Data and the Real-Time Calibration Challenge**

Therefore, increasing data rates and shrinking timing margins create a requirement for a more precise clocking and calibration circuit. Such a circuit needs to provide a robust mechanism for real-time calibration, ensuring that the capture clock (DQS) is maintained within the data valid window with sufficient setup and hold margins under varying system conditions. In the 7 series FPGAs, there is a new clocking architecture element called the Phaser that provides this real-time calibration capability at 1.866 Gb/s data rates for Virtex-7 and Kintex-7 FPGAs.

Controller Implementation

A second and equally challenging aspect of the design is the logic-fabric-based implementation of the memory controller. With increasing DDR memory data rates (2X the memory clock rate), the FPGA logic-fabric-based controller that manages memory commands and data flow must run at higher rates as well. With each successive FPGA generation, the memory clock rates for each DDR architecture generation have increased at a faster pace than logic fabric performance, creating a gap between required memory clock rates and the rates at which logic-fabric-based controllers can be clocked (see [Figure 2](#)). To address these challenges, Xilinx has created innovative solutions.



WP383_02_122010

Figure 2: Data Rate Challenge for FPGA Logic-Fabric-Based Controllers

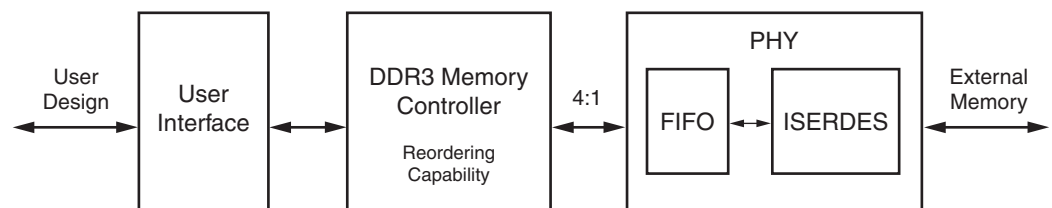
In previous-generation products (such as Virtex-6 FPGAs), a DDR3 controller running at half the rate of the memory clock is implemented to meet the logic fabric timing limitations. This implementation provides a good match between the data rate capabilities of the I/O and the logic fabric clock rate capabilities needed for the controller implementation.

In the 7 series FPGAs, however, a new architecture bridges the wider gap between DDR3 data rates of up to 1.866 Gb/s (needed in high-performance applications) and the logic fabric clock rate.

7 Series FPGA Memory Interface Architecture

With the introduction of the 7 series FPGAs, Xilinx has improved the architecture of the PHY layer memory interface and controllers to achieve data rates of 1.866 Gb/s and has also improved the efficiency of the DDR3 controller design to achieve higher effective bandwidth.

As shown in Figure 3, the architecture of the memory controller and interface incorporates three functional modules: the PHY, the DDR3 memory controller, and the user interface.



WP383_03_120610

Figure 3: DDR3 Controller and Interface Architecture

For DDR3 interfaces that run at a data rate of 1.866 Gb/s, the clock rate of the logic-fabric-based controller needs to be 1/4 the memory clock rate to implement controller state machines in the logic fabric that can keep up with the 933 MHz DDR3

memory clock rate. The improved PHY architecture makes this possible with a dedicated FIFO that provides the gearbox capability to decouple the memory clock rate from the logic-fabric-based controller clock rate at an appropriate ratio. This ratio can be 4:1, or for lower memory clock rates, 2:1. DDR3 interfaces that require clock rates of over 800 MHz (1,600 Mb/s) are suitable for 4:1 decoupling ratios, while other interfaces like QDR-II+ or RLDRAM II running at rates below 550 MHz can be better matched with a 2:1 decoupling ratio.

Optimizing the Physical Layer for Higher Data Rates

The PHY is responsible for capturing read data and for transferring write data from the controller to the external memory devices. The I/O drivers must provide sufficient drive strength and switching speed for data rates up to 1.866 Gb/s supported by the highest speed grade DDR3 devices. An innovative I/O architecture ensures drive capability with excellent signal integrity at these high rates. A 2V pre-driver voltage option (V_{CCAUX_IO}) was needed to ensure sufficient signal gain to meet the drive requirements for the 1.866 Gb/s rates. See [Figure 4](#).

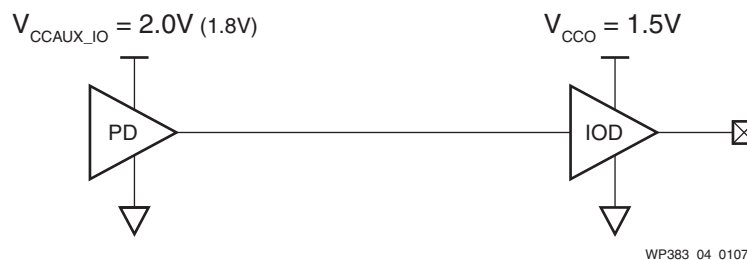


Figure 4: 2.0V Pre-driver (V_{CCAUX_IO}) Voltage Supply Option

In addition to enabling the capabilities of the I/O driver and receiver to switch at these high data rates, the read data capture and real-time calibration require a dedicated clocking circuit that can manage the initial calibration of the clock to the data valid window and maintain this relationship over changes in voltage and temperature during system operation. 7 series FPGAs incorporate a new clocking structure as part of the clock management tile (CMT) called the Phaser that has all the built-in capabilities to control and maintain the clock-to-read-data timing with up to 7 ps precision.

[Figure 5](#) outlines the I/O block and CMT-to-logic fabric signal relationship that constitutes the basis of the PHY architecture. The system clock is driving the phase-locked loop (PLL) in the CMT, and that in turn drives the clock through the BUFG for the memory controller. A PLL output also drives the Phaser control and the PHASER_IN and PHASER_OUT blocks.

The PHASER_IN performs two primary functions: it delays the DQS, which is then used to capture the data (DQ in [Figure 5](#)) in the ISERDES; and it controls the transfer of the data from the ISERDES to the IN_FIFO block.

The FIFO has two modes, 1:2 and 1:1. The 1:2 mode further decouples the high data rates of the I/O from the logic fabric that needs to run at a lower rate. With the 1:4 ISERDES and 1:2 FIFO transfer ratios, the data is effectively running single data rate at $\frac{1}{4}$ of the memory clock rate. This is a necessity when implementing data rates higher than 1,600 Mb/s or 800 MHz. The memory controller needs to run only at $\frac{1}{4}$ of the 800 MHz rate. This 1:4 decoupling of the two clock systems (memory and controller) provides the benefit of easier timing closure. For 1.866 Gb/s data rates or 933 MHz clock rates, the memory controller runs at only 233 MHz. The logic-fabric-based controller state machine can meet timing if the clock rates are kept in this range.

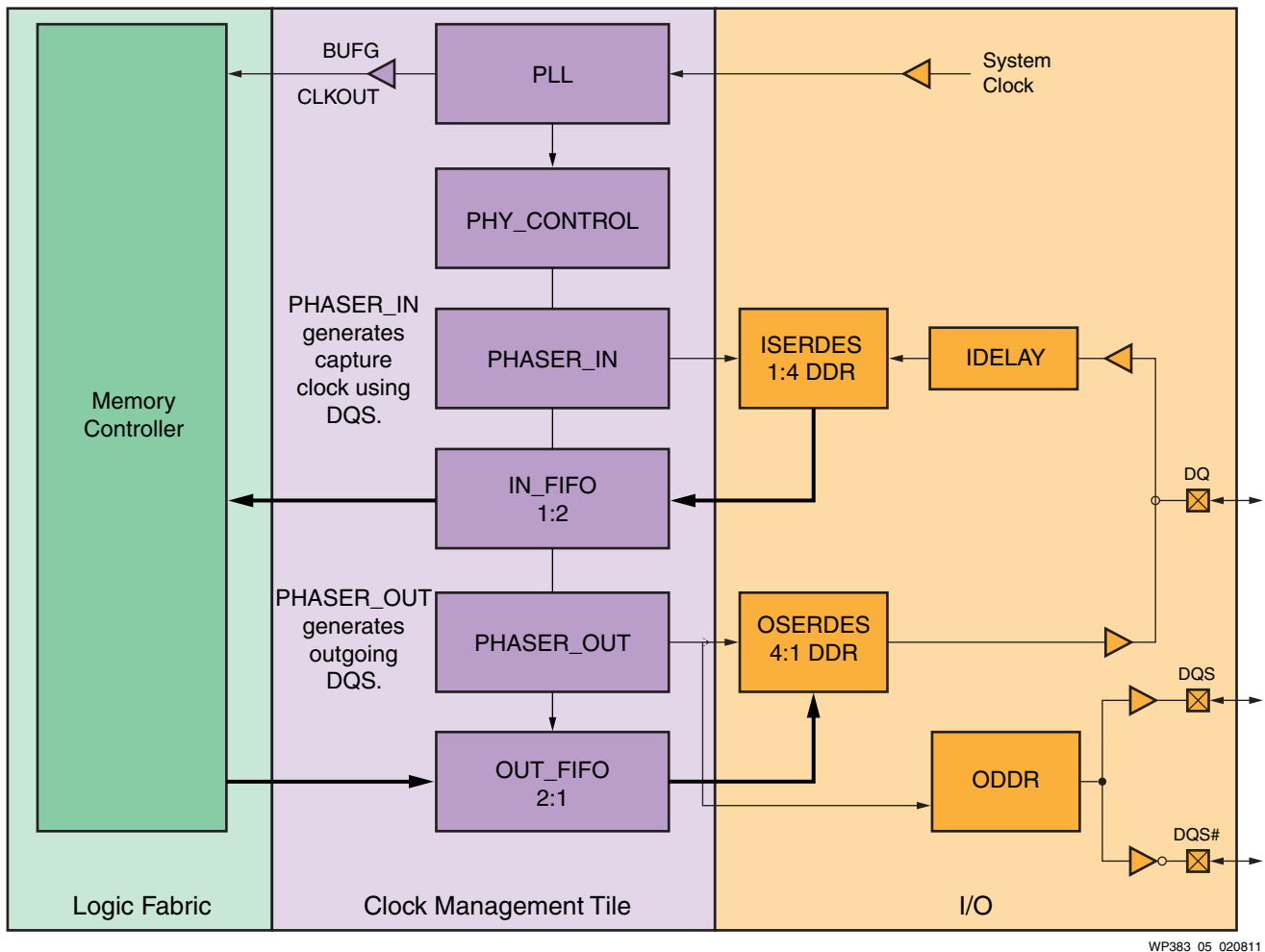


Figure 5: PHY Architecture and Data Transfer to the Logic-Fabric-Based Memory Controller

The output datapath and clocking are similar to the input path, but the PHASER_OUT and OUT_FIFO are used to clock and control the data out transfer. The PHASER_OUT also controls the generation of the outgoing DQS signal.

Achieving Higher Performance with the DDR3 SDRAM Controller

The maximum data rate per pin is just one factor used to determine the performance of the memory interface. The effective or average bandwidth that determines the overall performance is a factor of data rate per pin, efficiency, and bus width.

Higher efficiency comes from enabling the reordering function of the DDR3 controller. It reorders reads and groups the reads and writes to more effectively use the data bus. Efficiency can be defined as the number of clock cycles transferring data divided by the total number of clock cycles. Or:

$$\text{Efficiency (\%)} = \frac{\text{Clock cycles transferring data}}{\text{Total clock cycles}} \times 100 \quad \text{Equation 1}$$

This is very important because designers might not always be able to simply increase the clock frequency to obtain more bandwidth. Implementing high clock frequency systems is sometimes costly and increases power consumption; failing to take advantage of the maximum bandwidth attainable for a particular clock frequency is not an optimal way to design. In some systems, if the controller is not effectively using

the data bus, the actual bandwidth can drop to as low as 10% of the maximum capability. If a string of RWRW commands are placed on the bus, the turnaround time introduces a lot of dead cycles or gaps. Additionally, if the read or write commands are not accessing existing open rows, the pre-charge time can also add dead cycles. These dead cycles or gaps in the data bus are inherent to the way DDR3 SDRAMs operate and can be mitigated by a reordering controller.

To get maximum bandwidth, the controller needs to minimize RWRW accesses to the DDR3 SDRAM as well as maximize accesses to open rows. The Xilinx 7 series FPGA DDR3 controller leverages the Virtex-6 FPGA DDR3 controller’s reordering architecture, which has already been successfully proven in mass-production systems. With an incoming command queue that possesses the added intelligence to reorder the user commands, the controller can do the following:

- Reorder alternating RWRW accesses by grouping reads and writes to minimize bus turnaround time
- Reorder reads to minimize page misses, thereby causing more commands to access open rows

Additionally, the user has the option to turn off the reordering controller mode and implement a strict order if needed.

Testing the Reordering DDR3 Controller

To quantify these efficiency improvements, several test patterns were implemented with the DDR3 controller for the Virtex-6 FPGA and then compared to the nearest competitive device that does not use a reordering algorithm in their DDR3 controller.

Figure 6 shows how random and alternating read/write patterns can reduce the data bus efficiency and how the Xilinx DDR3 reordering controller can improve the bandwidth by an average of 2X over in-order controller implementations. The DDR3 reordering controller used in the benchmarks has a queue of up to eight commands or “bank machines.” The same DDR3 reordering controller options implemented in Virtex-6 FPGAs have been ported to the 7 series FPGAs.

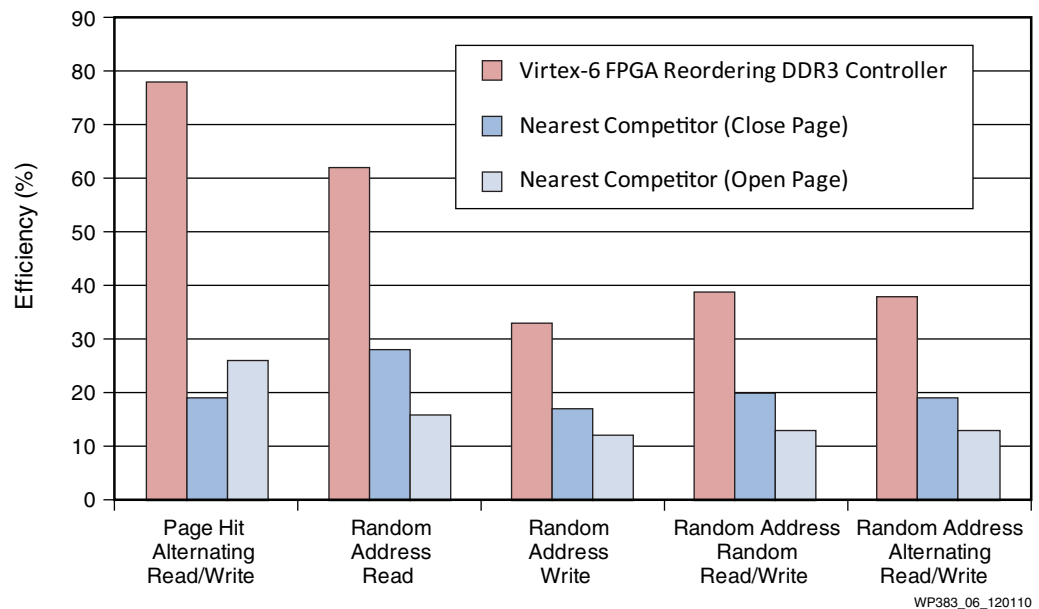
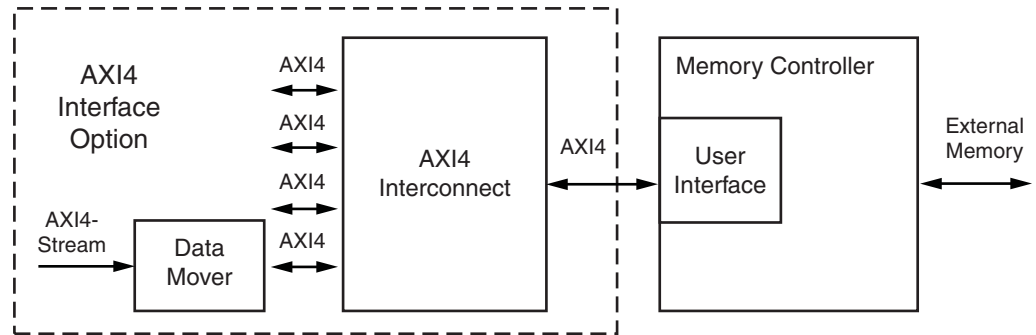


Figure 6: Bandwidth Efficiency and Improvements with Xilinx Reordering Controller

User Interface Options for Higher Performance and Flexibility

Interfacing to the memory controller is done through a user interface that runs synchronously with the memory controller clock. With 7 series FPGAs, Xilinx offers two options for the user interface to provide added flexibility and higher performance for various applications, as illustrated in [Figure 7](#).



WP383_07_121410

Figure 7: Memory Controller User Interface Options in 7 Series FPGA Implementation

In applications where the external memory interface determines system performance, a single user port is typically preferred. One option offered in the 7 series FPGAs is the simple FIFO-based user interface, similar to the Virtex-6 FPGA implementation, running at the same clock rate as the controller.

The Advanced Microcontroller Bus Architecture (AMBA®4) AXI4 is a second option for applications where multiport or AXI4 compatibility is needed, such as in video or embedded applications. The AXI4 multiport function enables different users to share the memory controller through the AXI4 protocol. The AXI4 Interconnect block provides the connectivity required to support multiple ports or users accessing the controller or seamlessly connecting to other AXI4 IP. Furthermore, a data mover block can provide additional functionality to connect via AXI4-Stream. This AXI4 compatibility option is offered with comparable performance to the simple single-port user interface, but it can achieve much higher performance in embedded applications when compared to previous multi-port memory controller (MPMC) implementations. Refer to [WP379](#), *AXI4 Interconnect Paves the Way to Plug-and-Play IP* for further information on the AXI4 interconnect benefits.

Higher Productivity with the Memory Interface Generator (MIG)

A complete memory controller and interface design can be generated with the MIG wizard, a software tool freely available from Xilinx as part of the CORE Generator™ tool. The MIG design flow is similar to the traditional FPGA design flow. The benefit of the MIG tool is that there is no need to generate the RTL code from scratch for the PHY interface or the memory controller. The MIG wizard generates the RTL and UCF files, which are the HDL code and constraints files, respectively. These files are based on a library of hardware-verified IP, with modifications coming from the user's inputs. MIG produces the pre-verified IP with the necessary constraints that facilitate higher performance.

The designer has complete flexibility to further modify the RTL code or the constraints file. Unlike other solutions that offer black-box implementations, the MIG tool outputs unencrypted code, providing complete flexibility to change or further customize a design.

The MIG output files are categorized in modules that apply to different building blocks of the design: user interface, PHY, controller state machine, etc. The user has the option to also connect a different controller to the PHY generated by the MIG. Additionally, the MIG generates a synthesizable testbench with memory checker capability. The testbench is a design example used in the functional simulation and the hardware verification of the Xilinx base design. The testbench issues a series of writes and reads back to the memory controller. It can also be used as a template to generate a custom testbench, estimate the bandwidth efficiency, and therefore verify the expected performance for different memory access patterns.

Hardware Verification and Characterization to Ensure Higher Performance

Hardware verification of memory interface and controller IP for 7 series FPGAs is an important final step to ensure a robust and reliable high-performance solution. Xilinx has verified memory interface designs and characterized them for several generations of FPGA products using a thorough characterization methodology with real system testing procedures. The characterization process is based on a multitude of tests that ensure process, voltage, and temperature (PVT) corners are tested in a real system environment that reaches stringent conditions similar to or surpassing actual user systems.

Conclusion

Achieving higher performance for memory interfaces is a design process that starts from the ground up, with silicon features that enable higher maximum data rates and with memory controller features that enable higher efficiency with higher sustainable data rates. With 7 series FPGAs, Xilinx has developed the highest performance memory interface solution in the industry by maximizing the DDR3 data rate capability to 1.866 Gb/s as well as improving the efficiency of the controller to sustain these high data rates in more demanding applications. Xilinx is also continuing to provide easy-to-use software tools, like MIG, that enable faster customization of the core IP. Xilinx memory interface solutions are based on extensive hardware characterization to ensure that high performance is sustainable in typical systems under changing voltage and temperature conditions.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
03/01/11	1.0	Initial Xilinx release.

Notice of Disclaimer

The information disclosed to you hereunder (the "Information") is provided "AS-IS" with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.