



Implementing High-Speed DDR3 Memory Controllers in a Mid-Range FPGA

A Lattice Semiconductor White Paper

March 2010

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Introduction

As system bandwidths continue to increase, memory technologies have been optimized for higher speeds and performance. The next generation family of Double Data Rate (DDR) SDRAMs is the DDR3 SDRAM. DDR3 SDRAMs offer numerous advantages compared to DDR2. These devices are lower power, they operate at higher speeds, offer higher performance (2x the bandwidth), and come in larger densities. DDR3 devices provide a 30% reduction in power consumption compared to DDR2, primarily due to smaller die sizes and the lower supply voltage (1.5V for DDR3 vs. 1.8V for DDR2). DDR3 devices also offer other power conservation modes like partial refresh. Another significant advantage for DDR3 is the higher performance/bandwidth compared to DDR2 devices due to the wider pre-fetch buffers (8-bits wide for DDR3 compared to 4-bits for DDR2), and the higher operating clock frequencies. However, designing to the DDR3 memory interfaces also becomes more challenging. Implementing a high-speed, high-efficiency DDR3 memory controller in a FPGA is a formidable task. Until recently, only a few high-end (read: expensive) FPGAs supported the building blocks needed to interface reliably to high speed DDR3 memory devices. However, a new generation of mid-range FPGAs now provides the building blocks, a high-speed FPGA fabric, clock management resources and the I/O structures needed to implement the next generation DDR3 memory controllers. This white paper examines the design challenges, and how one particular FPGA family, the LatticeECP3, can facilitate DDR3 memory controller design.

DDR3 Memory Controller Challenges

DDR3 devices present a host of challenges for the memory controller. The operating frequencies for the DDR3 begin at the higher end operating frequencies of DDR2, and then go much higher. DDR3 memory interfaces require clock speeds in excess of 400 MHz. This is a major challenge in FPGA architectures. The fly-by architecture and the Read and Write leveling have introduced an additional level of complexity for the DDR3 memory controller architecture.

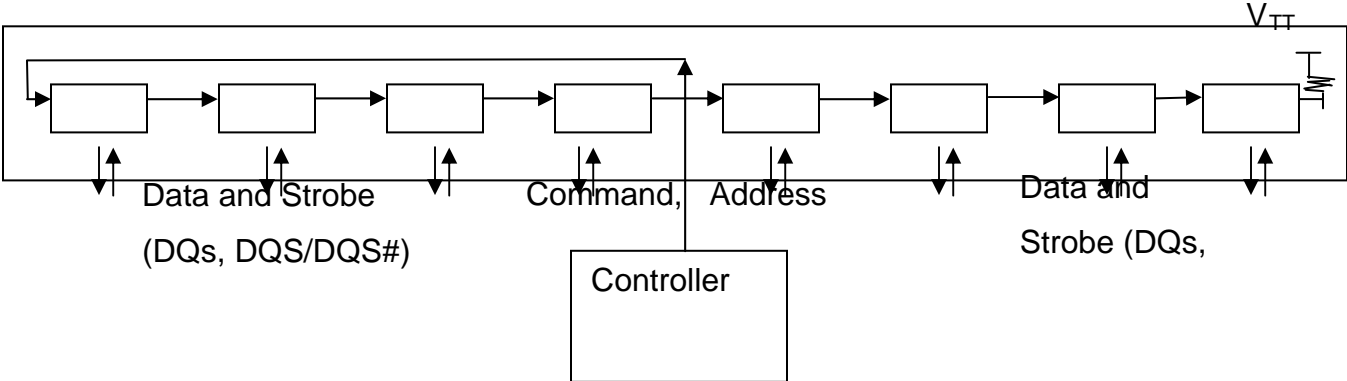


Figure 1 - Fly-by Architecture for DDR3

Unlike DDR2 which uses a T-branch topology, DDR3 adopts a fly-by topology that provides better signal integrity at higher speeds. The fly-by signals are the command, address, control and clock signals. As shown in Figure 1, these signals from the memory controller are connected in series to each DRAM device. This improves signal integrity by reducing the number of stubs and the stub lengths. However, this creates another issue since the delay for each memory component is different, depending on where it is in the sequence. This delay difference is compensated for by using the Read Leveling and Write Leveling techniques as defined by the DDR3 specification. The fly-by topology requires that the memory system be calibrated each time at power ON. This requires additional intelligence in the DDR3 memory controller that allows the calibration to be done automatically at start-up.

Read and Write Leveling

During Write Leveling, the memory controller needs to compensate for the additional flight time skew (difference in the signal delay to each memory device) introduced by the fly-by topology with respect to strobe and clock. As shown in Figure 2, the source CK and DQS signals are delayed in getting to the destination. This delay can be different for each memory component on the memory module and has to be adjusted on a chip-by-chip basis and even on a byte basis if the chip has more than one byte of data. The diagram illustrates one memory component. The memory controller delays DQS, one step at a time, until a transition from a zero to a one is detected on the destination CK signal. This will realign DQS and CK so that the destination data on the DQ bus can be captured reliably. Since this is done automatically by the DDR3 memory controller, the board designer need not worry about the details of the implementation. The designer benefits from the additional margin created by the Write Leveling feature in the DDR3 memory controller.

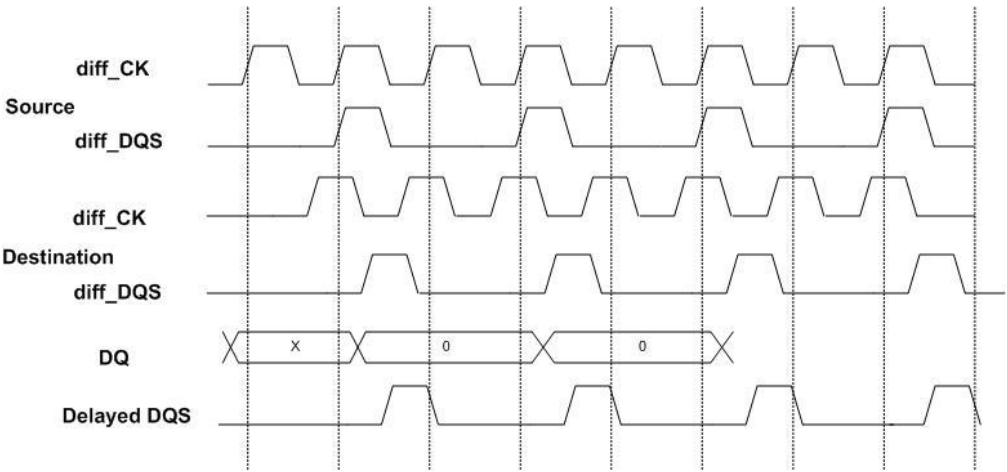


Figure 2 - Timing Diagram for Write Leveling

DDR3 Memory Clocking Resources and Interface Blocks

The LatticeECP3 FPGA’s I/Os have dedicated circuitry to support high-speed memory interfaces, including DDR, DDR2 and DDR3 SDRAM memory interfaces. As shown in Figure 3 below, the ECP3 family also contains dedicated clocking resources to support the next generation high-speed memory controllers like DDR3. The Edge clocks (ECLK1, ECLK2) are high-speed, low-skew clocks used to clock high-speed data in and out of the device. The DQS lanes provide clock inputs (DQS) and up to 10 input data bits on that clock. Each DQS lane is serviced by a DQSBUF to control clock access and delay. The DQS lanes are supported by the DQSDLLs (one each on the right and left sides of the device). The DQSDLLs are dedicated DLLs for creating a 90° clock delay.

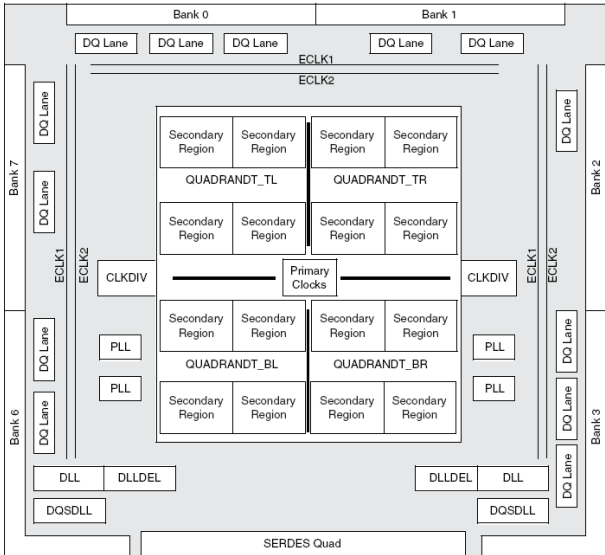


Figure 3 - LatticeECP3 DDR Memory Clocking Resources

The Lattice DQS circuitry includes an automatic clock transfer circuitry that simplifies the memory interface design and ensures robust operation. Additionally, the DQS Delay block provides the required clock alignment for DDR memory interfaces. The DQS signal feeds from the PAD through a DQS delay element to a dedicated DQS routing resource. Temperature, voltage and process variations of the dedicated DQS delay block are compensated for by a set of calibration (7-bit delay control) signals from

two DQSDLLs on opposite sides of the device. Each DQSDLL compensates DQS delays in its half of the device. The DLL loop is compensated for by the system clock and dedicated feedback loop.

A LatticeECP3 FPGA PLL is used to generate the clocks for the DDR3 memory interface. For example, for a 400 MHz DDR3 interface, a general-purpose PLL is used to generate three clocks: a 400 MHz clock, a 90° shifted version of this 400 MHz clock, and a 200 MHz clock. The 90° shifted version of the 400 MHz clock is used to generate the DQ and DQS outputs. The non-shifted version of the 400 MHz clock is used to generate the clocks (CLKP and CLKN) to the DDR3 memory. The 200 MHz clock is used to generate the address and command (ADDR/CMD) signals. The clocking implementation is transparent to the customer and is implemented automatically by the Lattice design tools.

Write leveling required for DDR3 is achieved by using the dynamic delay input to a module dedicated for DDR3 memory interfaces, called the DQSBUFD module. The DQSBUFD module includes the DQS delay block, the clock polarity control logic, and the data valid module. Write leveling required for DDR3 is achieved using the dynamic delay (DYNDELAY) input to the DQSBUFD module. This can delay the outputs of each DQS Group based on the write leveling requirements.

For the DDR3 memory read interface, the DQS Transition Detect block detects the DQS transition when the memory device drives DQS low and generates the read clocks used to transfer data into the FPGA.

The Lattice IPexpress tool can be used to generate the DDR3 memory interface blocks described above. These pre-engineered blocks in the LatticeECP3 reduce complexity for the designer by providing the appropriate building blocks required to interface to high-speed DDR3 memory devices.

DDR3 Memory Controller

Lattice provides a full-featured DDR3 Memory Controller IP core to interface to industry standard DDR3 components and DIMMs. A block diagram of the Lattice DDR3 memory

controller is shown in Figure 4. The Lattice memory controller connects to the LatticeECP3 DDR3 memory interface blocks (IO modules) and the clocking circuitry to provide customers with an out-of-the-box solution for interfacing to DDR3 memory components and DIMMs. The controller implements a number of features to improve overall throughput. For example, command pipelines are implemented to improve overall throughput. The IP uses efficient bank management techniques to manage multiple banks in parallel. This minimizes access delays and helps improve the memory bandwidth.

The LatticeECP3 DDR3 memory controller can be generated using the Lattice IPexpress tool. The GUI-based tool allows designers to specify the memory controller parameters (clock rate, data bus width, configuration, etc.) to generate the DDR3 memory controller IP core. Designers can customize the parameters through the GUI. For example, the GUI allows users to customize the memory timing parameters and regenerate the memory controller with fresh timing values. Along with the DDR3 memory controller IP core, simulation models and test benches are also provided so that designers can test their design before they take it to a board.

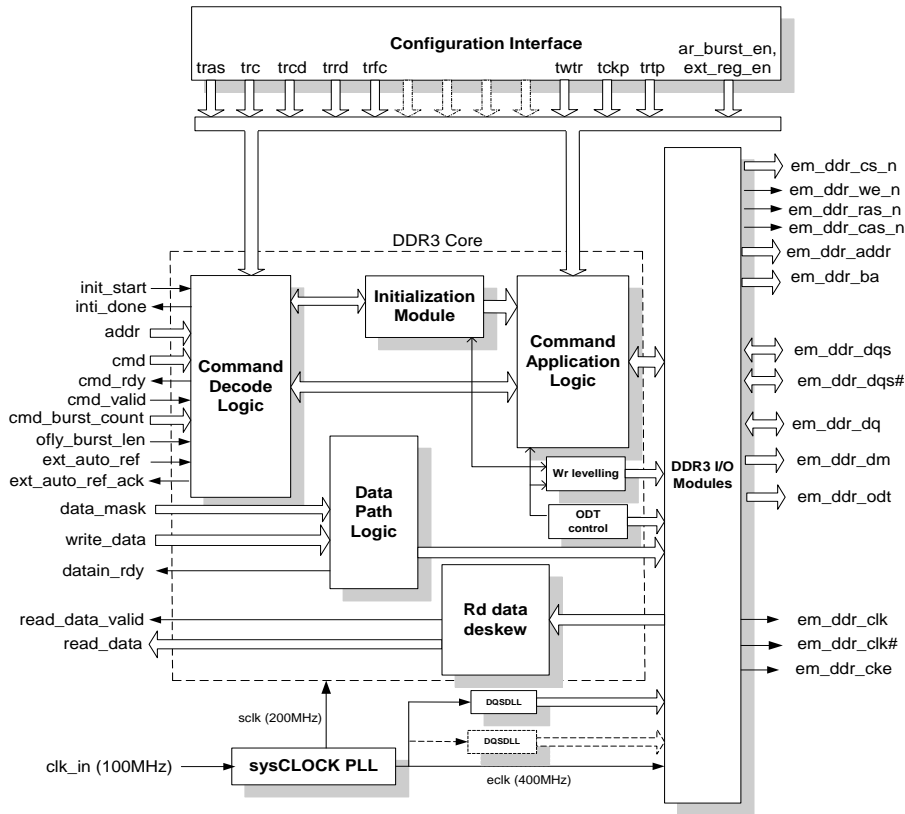


Figure 4 - Lattice DDR3 Memory Controller Block Diagram

The LatticeECP3 DDR3 memory controller has been fully validated with both DDR3 memory components and DIMMs. Lattice also provides multiple hardware evaluation boards that customers can use to check the LatticeECP3 DDR3 memory controller operation, interfacing to either DDR3 components or DIMMs. The LatticeECP3 family is the industry's only mid-range FPGAs supporting DDR3 memory interface, thus providing designers with a low-cost, low-power solution for next generation system designs.

Conclusion

System bandwidth requirements continue to grow exponentially. As prices for DDR3 SDRAMs fall, DDR3 SDRAMs will be more widely adopted in networking applications. These increasing system bandwidth requirements are pushing memory interface

speeds while costs continue to be driven down. Facilitating the design of robust high-speed memory interfaces in a mid-range FPGA was a principal goal of the Lattice ECP3 family of FPGAs. The dedicated yet flexible DDR capabilities of the ECP3 mean that designers now have a cost-effective solution for next-generation memory controller needs. The LatticeECP3 DDR3 primitives, coupled with the Lattice DDR3 memory controller IP core, significantly reduce the complexity of DDR3 memory interfaces and facilitate quicker time-to-market for next generation system designs implementing DDR3.

###