White Paper

Benefits of Altera’s High-Speed DDR2 SDRAM Memory Interface Solution

Introduction

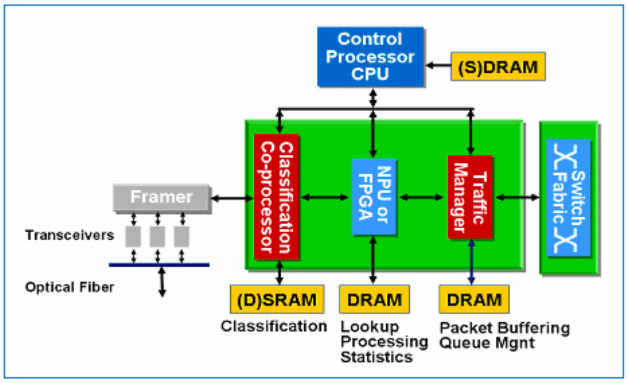
This white paper provides a general overview of the Double Data Rate 2 (DDR2) SDRAM interface, discusses some of the design challenges in DDR2 SDRAM, and details Altera’s solution used to implement 534 megabits per second (267 MHz) DDR2 interfaces using Stratix™ II FPGAs.

The Market Need for High Speed Memories

SDRAMs have traditionally been used in personal computers (PCs). As processor core speeds exceeded 2 GHz, revolutionary changes in memory speed, efficiency, size and costs were required to support these processor enhancements. DDR SDRAMs were introduced as a cost-effective path for upgrading data bandwidth to memory and have quickly become the memory of choice in PC and server markets. DDR SDRAMs have seen a drastic drop in price in the last couple of years, bringing them to price parity with conventional SDRAMs.[1]

This drop in price has not gone un-noticed in other applications such as the networking market. Ever increasing end-performance requirements in networking have resulted in greatly increased data bandwidths, which in turn have resulted in the need for faster memories. Memory requirements for networking applications fall into several categories, with DRAMs being primarily used for packet buffer memory (where large amounts of packet memory are required to store entire packets while network processors process the packet headers, see Figure 1).

Figure 1. Networking Application Requiring High-Speed Memories



[1] *Denali Memory Report (February 2003) – Volume 2, Issue 2*

Next generation memories are required to support processor core speeds beyond 4 GHz, as well as the networking infrastructure required to connect these computers. DDR2 SDRAM is designed to help drive these processor and network advancements.

General Overview of the DDR2 SDRAM Interface

Traditional single data rate (SDR) SDRAM architectures are performance-limited in their interfaces; DDR SDRAMs, therefore, were introduced as an enhancement. DDR2 SDRAM is the next evolutionary step from DDR memory. Most of the addressing and command control interface between SDR and DDR2 is identical; the fundamental difference is in the data interface. Double Data Rate (DDR) is achieved by using both the rising and falling edges of the clock to transfer data.

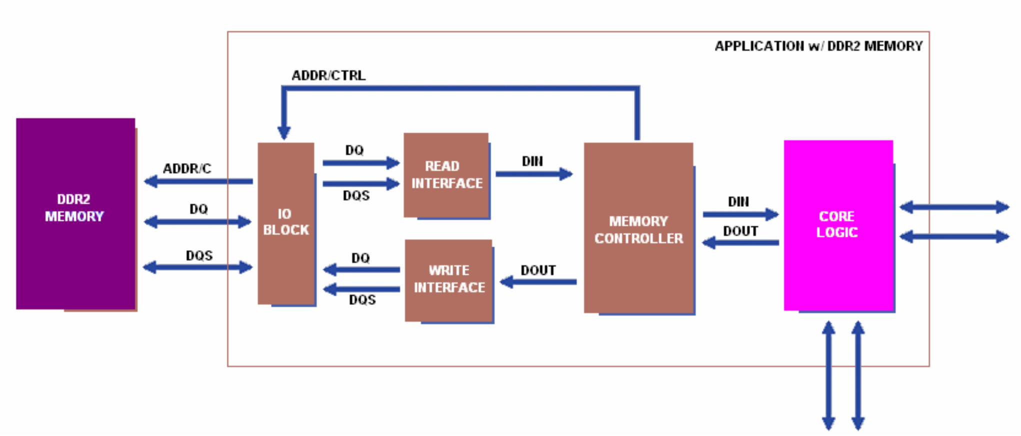
The DDR2 SDRAM architecture employs a 4*n*-prefetch architecture, where the internal data bus is four times the width of the external data bus. A single read or write cycle involves a single 4*n*-bit wide, one-clock-cycle data transfer at the core of the memory, and four corresponding *n*-bit wide, one-half-clock-cycle data transfers at the I/O (two external clock cycles, corresponding to two rising edges and two falling edges). This enables high-speed operation, as the internal column accesses are one quarter the frequency of the external data transfer rate.

The data interface is designed to transfer two *n*-bit wide words per clock cycle. However, if the data transfers were based on a free-running system clock, the maximum frequency would be attained as soon as the total output access and flight time equaled the bit time. Additionally, in such a scheme, the data does not track the clock with changes in temperature and loading, cutting down the effective data valid window and further limiting the maximum attainable frequency. To alleviate these limitations, DDR2 SDRAMs use a byte-wide, bidirectional differential or single-ended data strobe (DQS) that is transmitted externally, along with data (DQ) for data capture. DQS is transmitted edge-aligned by the DDR SDRAM during reads, and center-aligned by the controller during writes to the memory. The DDR SDRAM utilizes on-chip delay-locked loops (DLLs) to clock out DQS and corresponding DQs, ensuring that they are well matched and track each other with changes in voltage and temperature.

DDR2 SDRAMs feature differential clock inputs (`CK` and `CK#`), helping to mitigate the effects of duty-cycle variation on the clock inputs. As with SDR and DDR SDRAMs, DDR2 SDRAMs also support the use of data mask (DM) signals to mask data bits during write cycles. All inputs and outputs are compliant with the JEDEC standard for SSTL-18.

A DDR2 SDRAM memory implementation consists of a number of design blocks such as a memory control block, a read physical block, and a write physical block (Figure 2). The memory control block is responsible for efficiently issuing accesses to/from memory from/to the application specific core logic. The read physical block handles the external signal timing that captures the data during read cycles. Similarly the write physical block handles the issuing of clock and data with the appropriate external signal timing.

Figure 2. DDR2 SDRAM Implementation Blocks

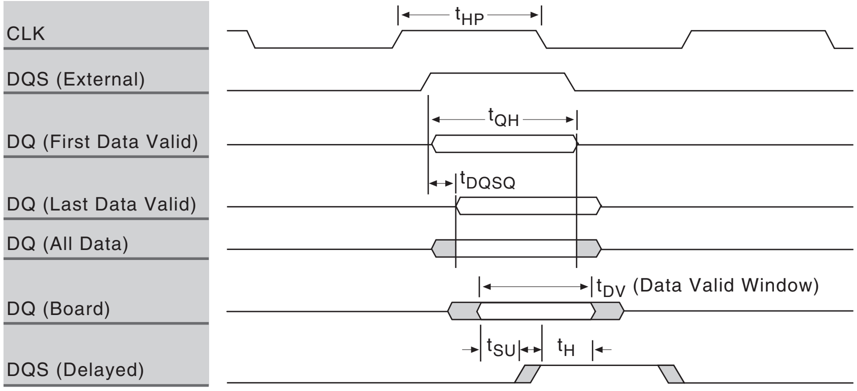


The major challenge in designing a high-speed DDR2 external interface is to reliably capture the DQ data with the DQS strobe in the during reads, and to properly drive DQ data with DQS during writes. The following sections provide a general overview of the read and write design issues, focusing on the DQS-DQ relationship. DDR2 has an added challenge over other high speed memory interfaces in that the DQS signal is a bidirectional strobe (compared to a unidirectional clock) – the strobe means that PLLs cannot be used in the front end portion of the read path, and bidirectional introduces issues in cleanly tristating the signal at high speed.

Read Cycle

During a read cycle, DQS and the corresponding DQ group are clocked out by the memory using the same internal clock. An on-chip DLL is used to edge-align DQ and DQS transitions to CK. Even though a DLL is used to clock out DQS and DQ, a finite mismatch occurs between DQS and its related byte-wide DQ group. This mismatch is represented in the datasheet by the t_{DQSQ} parameter, shown in Figure 3 as DQ (Last data valid). Therefore, a data word is valid from the point of the latest switching signal in a group to the earliest switching signal (off the following DQS edge) in the group. Furthermore, variations in clock duty cycle eventually result in loss of the data valid window. The effective data valid window is shown in Figure 3 as DQ (All data).

Figure 3. Data Valid Window at the Memory Output



The data valid window is further reduced by the time it takes the signals to arrive at the controller pins, primarily due to a combination of factors such as connector and board-induced skews, shown in Figure 3 as DQ (Board). DQS can then be used for data capture at the controller after appropriately positioning the DQS in relation to the data valid window.

The amount of time that DQS must be delayed is governed by board-induced skew between the DQS and DQ group, the resulting data valid window at the controller, and the sampling window requirements at the controller capture registers. The minimum time that DQS must be delayed with respect to DQ should account for t_{DQSQ} and any board-induced skews. Similarly, the maximum amount of time that DQS should be delayed is calculated from t_{QH} and board-induced skew. Nominal delay would imply DQS is center-aligned with the DQ data valid window, as shown in Figure 3 as DQS (Delayed). Additional board induced effects such as crosstalk and inter-symbol interference must also be taken into consideration, these effects will further decrease the side of the data valid window.

The controller designers can use several approaches to align the DQS to the center of the data valid window: board trace delay on DQS, on-chip trace delay on DQS, on-chip DLLs, or on-chip PLLs.

Board Trace Delay on DQS

Board trace delay on DQS is a more traditional approach used for aligning DQS and a related DQ group. While this can suffice for low-frequency systems and for relatively simple topologies, it proves to be a performance barrier for more sophisticated systems for the following reasons:

- A simple first order calculation at 400 Mbps puts the nominal DQS delay with respect to DQ at 1.08ns (with a min/max range of +/- 0.62ns). To achieve this delay, approximately 6 to 7 inches of trace length[2] must be added to the DQS line. Not only does this further complicate board layout, it also can result in increased board cost. Especially, when interfacing with DIMMs, routing the additional length required for each of the DQS signals can be difficult.

[2] *This length estimation is based on an approximate delay of 160 ps/inch for a FR4 laminate Microstrip with a 50-Ω characteristic impedance.*

- The required delay and resulting trace length must be accurately pre-determined. This locks the interface to a specific frequency, and does not leave a controller designer with much flexibility. Any changes in interface frequency would require laying out the board again.
- Increased DQS trace length leads to a greater margin of error due to increased susceptibility to noise and skew injection, and delay variance with changing voltage and temperature.
- Increased trace length also results in higher load capacitance on the DQS line. Thus, rise/fall times are further compromised, limiting the maximum attainable frequency.

On-Chip Trace Delay on DQS

DQS is routed to the capture registers through a line different from that of the DQ signals. Delay implemented on this dedicated DQS line either through increased length, or through use of a line with different characteristics, results in a delayed version of DQS reaching the registers. While this approach is similar in theory to the board trace delay approach, it is primarily used to fine-tune the delays achieved in the prior approach. Therefore, limitations of the board trace delay approach are not addressed. This approach continues to prevent higher performance.

On-Chip Delay Elements

This approach utilizes a number of delay elements connected in series to achieve a pre-determined delay. The delay and corresponding number of delay elements must be calculated based on frequency of operation and the right number of elements picked for each frequency bin. A designer can then implement varying design techniques to use a combination of coarse and fine delay to further fine-tune to the desired delay. However, delay elements are inherently susceptible to process, voltage, and temperature (PVT) variations, with variations seen up to ±40%. This variation in delay increases the effective sampling window requirements of the controller, and does not scale with frequency. Thus, the limitation of this approach makes it useful for lower frequencies.

On-Chip DLLs & PLLs

On-chip DLLs and PLLs are used primarily in conjunction with on-chip delay elements to introduce delay onto the DQS lines. By using a reference clock that is at the desired interface frequency and basing the required delay as percent of that clock period, the DLLs or PLLs can then pick the right number of delay elements to achieve the desired delay. As such, the previously stated limitations of delay elements are addressed; the DLLs or PLLs help compensate for PVT variations, allowing for much higher frequencies to be attained. While this approach does involve the use of relatively complicated structures (DLLs and PLLs), the benefits of higher frequencies and system bandwidths offset the cost of implementation.

Write Cycle

During a write cycle, DQS must arrive at the memory pins center-aligned with data. Here, the controller designer can adopt a couple of different approaches: automatic 90° DQS phase offset at the controller outputs or use of board trace delays. The memory device then uses internally-matched routing for the DQS and data to allow for data capture with the DQS signals. The approach adopted by a controller designer must take into account effects on the read-cycle timing. For example, if board trace delays were used, limitations (that have been discussed in prior sections) with this approach would restrict the maximum attainable frequency.

DQS Preamble/Postamble

DQS in DDR2 SDRAM is a bidirectional strobe (an output of the FPGA during writes and an input during reads), where the DQS signal must be driven low before going to or coming from a tristate. The state where DQS is low just after a tristate is called the preamble and the state where DQS is low just before it goes to tristate is called the postamble.

There are preamble and postamble specifications for both read and write operations in DDR2 SDRAM the controller designer must take into account these specifications and design the logic appropriately; otherwise glitches can occur on the DQS signal and clock incorrect data.

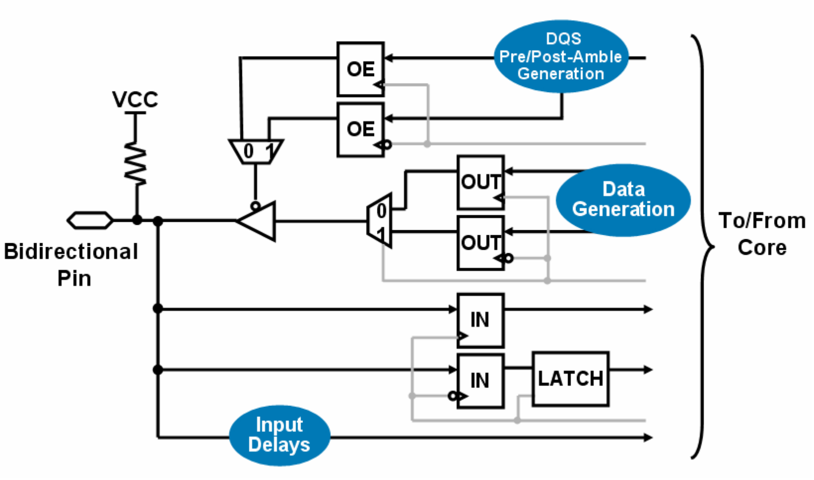
Memory Controller Implementation in Stratix II

This section will concentrate on some of the hardware features that have been implemented in the Altera Stratix II device that addresses the system performance bottlenecks discussed in prior sections. In particular, features that enable FPGA DDR2 SDRAM interface frequencies up to 534 Mbps are discussed.

I/O Elements

Figure 4 shows an I/O element (IOE) for Stratix II. Each IOE contains six registers and one latch. Two registers and a latch are used for input, two registers are used for output, and two registers are used for output-enable control. Input and output registers are independent of each other, enabling a bidirectional DDR I/O path to be implemented entirely in the IOE. When the input path is active, the output enable disables the tri-state buffer, which prevents data from being sent out on the output path.

Figure 4: DDR I/O Element



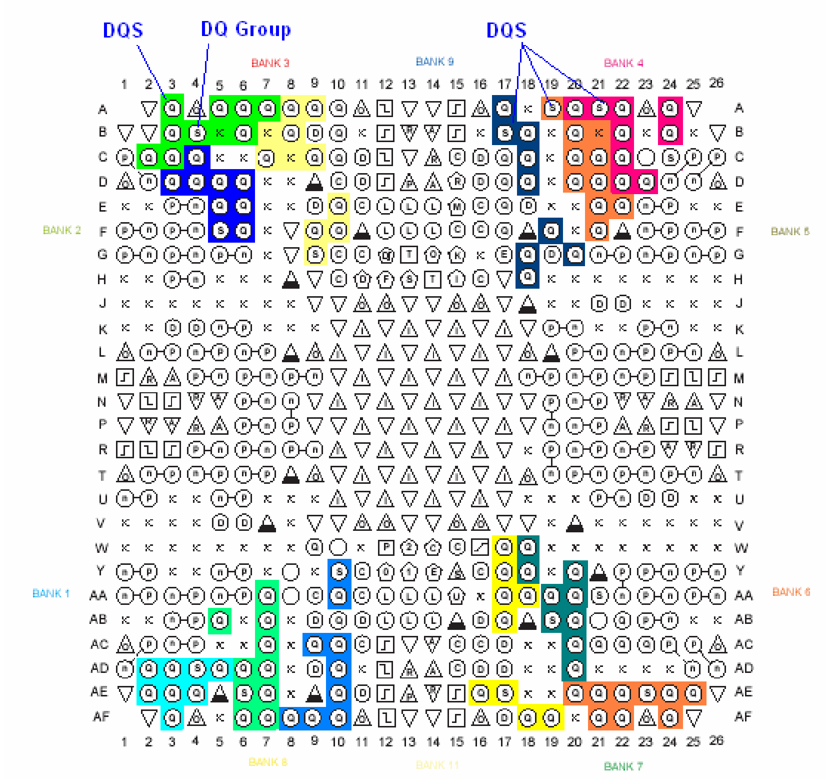
The advantage of having these registers in the IOE allows for minimum clock and data skew, which minimizes the degradation of the sampling window – this effectively allows Stratix II's DDR2 implementation to achieve high data rates with high reliability.

Dedicated DQS-DQ Groups

Stratix II devices feature dedicated DQS-DQ groups at the top and bottom of the die. When not interfacing with external memory, designers can use these pins as general-purpose I/O pins. However, when interfacing with external memories such as DDR2 SDRAMs, dedicated pins must be used for DQS. These in turn are associated with their respective DQ groups. Stratix II DQS-DQ groups can be configured in a number of ratios from 36 groups of one DQS for every 4 data bits (x4 mode) to four groups of one DQS for every 36 data bits (x36 mode). These ratios are dependent on the device and package. Reference the Stratix II datasheet for details on the ratios with respect to the package.

Figure 5 represents the pin layout of the DQS and DQ pins when in x8 mode. Each different block represents one DQ group. Within each group is a dedicated DQS signal.

Figure 5. DQ-DQS Groups



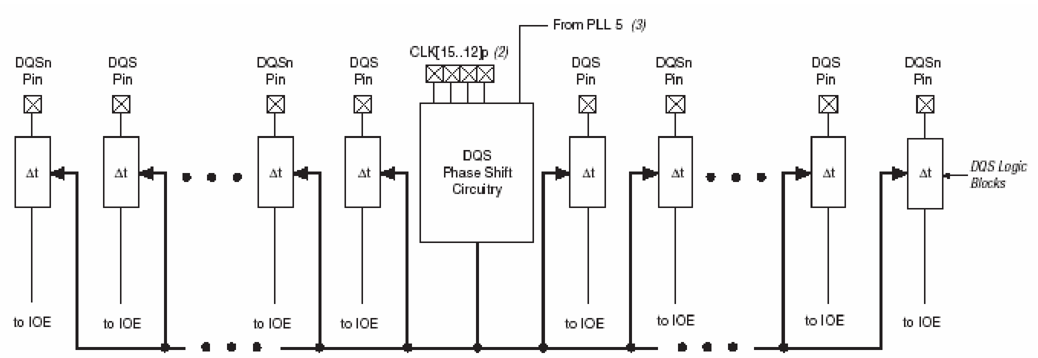
This figure is for illustration purposes only and does not represent any particular device in any package.

The dedicated DQS pins tie internally to a set of delay elements before being routed to the I/O input registers. The cumulative delay of the delay elements is controlled by a DQS phase shift circuitry, and is discussed further in the proceeding section. The dedicated DQS approach does take away some layout flexibility, but the benefits realized with this approach offset the limitations. Use of dedicated pins ensures tighter sampling window requirements at the controller pins. In addition, limiting the use of delay elements to a reduced number of I/O pins results in decreased die area consumption, leading to die cost savings.

Dedicated DQS IO

Stratix II devices have dedicated DQS phase shift circuitry (see Figure 6) to enable automatic, on-chip delay insertion of incoming DQS signals. This DQS phase shift circuitry uses a frequency reference to generate control signals for the delay elements on each of the dedicated DQS pins, allowing it to compensate for PVT variations. This frequency reference can be an external reference clock or an internal PLL output.

Figure 6: Dedicated DQS Phase Shift Circuitry



The approach of implementing a dedicated phase shift circuitry has several significant advantages:

- DQS can be treated as a 9th “DQ” (within an 8-bit DQ group) and thus can be laid out identical to DQ signals on a board. In addition, designers can treat each DQS-DQ group separately and route accordingly (without forgetting that eventually all groups have to re-synchronize to a common system clock in the controller). This simplifies board layout and reduces board costs.
- Any board-induced skew effects and board impedance variance with voltage and temperature will equally affect both DQS and DQ, ensuring a tighter timing relationship between DQS and DQ at the controller pins.
- As the on-chip phase-shift circuitry uses a frequency reference, the delay can be tuned to match a desired frequency without having to re-layout the board. Additionally, active PVT compensation ensures the tight DQS-DQ timing relationship is maintained all the way to the capture registers.

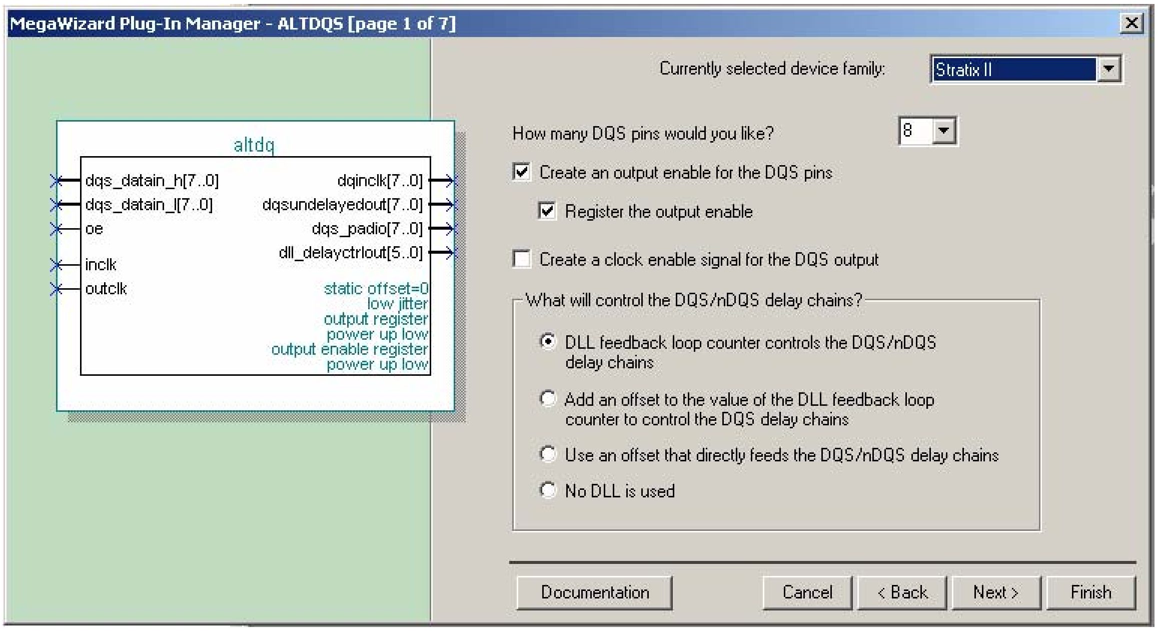
The Stratix II architecture allows the phase shifted DQS signal to go into the logic array. This allows for further optimization of the memory implementation, if required.

Stratix II also has hardware features to handle the DQS postamble circuitry. This simplifies the logic in the memory controller by adding dedicated resources. These resources ensure data is not lost or corrupted when there is noise on the DQS line at the end of a read.

Software Support

Altera’s Quartus II software provides Megafunctions which instantiate RTL blocks based on the DDR2 memory configuration, see Figure 7. The software analyzes the DQS input frequency to determines the optimal resynchronization clock phase and provides Tcl commands for extracting timing analysis results. A simulation model is also available to accurately verify the DLL and DQS delay behavior.

Figure 7: ALTDQ and ALTDQS Megafunctions in the Quartus II software.



MegaWizard Plug-In Manager - ALTDQS [page 1 of 7]

Currently selected device family: Stratix II

How many DQS pins would you like? 8

☑ Create an output enable for the DQS pins

☑ Register the output enable

□ Create a clock enable signal for the DQS output

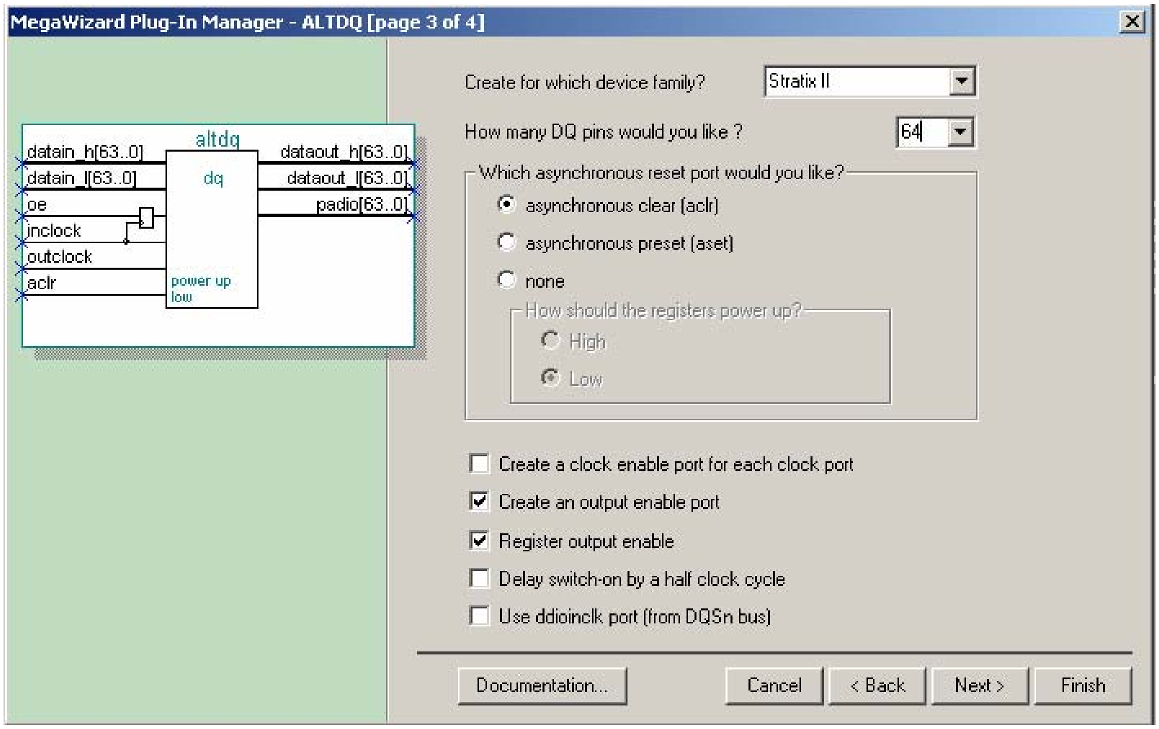
What will control the DQS/nDQS delay chains?

● DLL feedback loop counter controls the DQS/nDQS delay chains

○ Add an offset to the value of the DLL feedback loop counter to control the DQS delay chains

○ Use an offset that directly feeds the DQS/nDQS delay chains

○ No DLL is used



MegaWizard Plug-In Manager - ALTDQ [page 3 of 4]

Create for which device family? Stratix II

How many DQ pins would you like ? 64

Which asynchronous reset port would you like?

● asynchronous clear (aclr)

○ asynchronous preset (aset)

○ none

How should the registers power up?

○ High

● Low

□ Create a clock enable port for each clock port

☑ Create an output enable port

☑ Register output enable

□ Delay switch-on by a half clock cycle

□ Use ddioinclk port (from DQSn bus)

Stratix II DDR2 SDRAM System-Level Solution

Altera designed the Stratix II FPGA with high-speed memory issues in mind, in order to provide a clean Stratix II DDR2 SDRAM system solution. For information on Stratix II DDR2 SDRAM such as intellectual property, reference designs, demonstration boards, software support, timing analysis, simulation and documentation visit the Altera memory solutions website at **www.altera.com/memory**

Conclusion

DDR2 SDRAMs is the next generation DRAM technology of choice for PCs and server applications, and are experiencing ever-increasing adoption in the networking market. At the same time, increasing system bandwidth requirements are pushing memory interface speeds. The burden of designing a robust interface eventually transfers to the memory system designer. While interface problems limit performance, designers do have solutions available to them to tackle these issues.

Altera's Stratix II FPGAs include dedicated hardware (such as dedicated DQS phase shift circuitry) to allow FPGA-to-DDR2 SDRAM interface performance to run up to 534 Mbps (267 MHz). In addition to these hardware features, Altera also offers software support, intellectual property and hardware reference platform to allow FPGA designers to efficiently develop their products using DDR2 SDRAM.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com