

DesignCon 2007

Calibration Techniques for High-Bandwidth Source-Synchronous Interfaces

Manoj Roge, Altera Corporation

Andy Bellis, Altera Corporation

Phil Clarke, Altera Corporation

Joseph Huang, Altera Corporation

Mike Chu, Altera Corporation

Yan Chong, Altera Corporation

Abstract

This paper describes the calibration techniques that can be used to maximize the link performance of parallel source-synchronous interfaces used by all mainstream DRAM and SRAM memories. The focus of memory vendors is to provide their lowest cost memories to the marketplace in the timeliest manner. Even though memory performances double with every generation, the memory uncertainties do not decrease at the same rate. This places a significant burden on memory controllers in optimizing the link performance. This paper will demonstrate by using timing analysis that beyond 800 Mbps, dynamic calibration techniques need to be implemented to track changes due to process, voltage, and temperature (PVT) variations. Results from timing analysis and simulations demonstrate that self-calibration and skew compensation improves the timing margin in source-synchronous parallel interfaces and can be implemented with reasonable die size overhead.

Author Biographies

Manoj Roge

Manoj Roge is a system architect in Altera Corporation's Product Planning Group, with responsibility for defining the FPGA features and roadmap for future products. He has 13 years experience working in the semiconductor industry in various roles such as test and design engineering, business development, and marketing. Mr. Roge has published technical articles in magazines such as EE Times and Design Wave, has four patents granted, and several invention disclosures. Additionally, Mr. Roge has played an active role in standards definition in the Network Processor Forum (NPF), JEDEC, QDR Consortium, and CellularRAM working group. He holds an MSEE from the University of Texas and an MBA from Santa Clara University.

Andy Bellis

Andy Bellis is the team leader of the Memory Architect Group in Altera's Intellectual Property (IP) division. Andy's role is defining memory interface architectures for current and future FPGA families. Andy has 10 years experience in the semiconductor industry in various roles such as ASIC design and IP development, and has three patents granted.

Phil Clarke

Phil Clarke is a senior design engineer working for Altera Corporation. He has four years experience designing FPGAs, and now works as part of the team defining and developing the Altera® memory interface product offerings. He holds a MEng from Imperial College London.

Joseph Huang

Joseph Huang received BS and MS degrees in electrical engineering from San Jose State University in California, in 1987 and 1989, respectively. He joined the Altera Corporation in 1990, where he has worked on circuit design, system performance, and chip integration. Currently a manager in the IC Design department, he has been involved in the development of FLEX® 8K, FLEX 6000, FLEX10K®, APEX™, APEX II, Stratix®, and Stratix II FPGAs.

Michael Chu

Michael Chu is currently the supervising member of technical staff in Altera Corporation's IC Design group with responsibility for defining and designing FPGA products. He has worked in the area of designing SRAMs, EEPROMs, and FPGAs for over 15 years in various technical and management roles. Mr. Chu has published papers in various technical magazines and holds several patents and invention disclosures. He holds an MSEE from Pennsylvania State University.

Yan Chong

Yan Chong received the B.S and M.S. degrees in physics from The University of Science and Technology of China in 1992 and 1995, respectively, and MSEE from University of Florida, Gainesville, in 1998. She joined Altera in 1999 and has been working on circuit design of DLLs, external memory interface, IO buffers, PLLs, and embedded memories.

Introduction

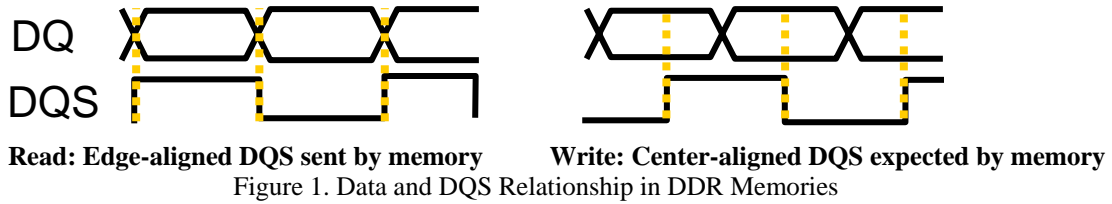
As on-chip computation speed has increased exponentially, I/O bandwidth has become the bottleneck for achieving higher system performance. Off-chip memories (DRAMs as well as SRAMs) have managed to keep pace, but as memory interface speeds increase and the bit period shrinks correspondingly, the memory and system uncertainties do not decrease by the same amount, thus becoming a larger portion of the bit period. In effect, the data valid window becomes a much smaller percentage of the bit period. Most mainstream memories such as DDR, DDR2, DDR3, and RLDRAM II DRAMs, and QDR II SRAMs use a source-synchronous interface where data strobe or output data clock sent by the memories are well aligned with the data used by memory controllers to capture the data. The old method of setting the data strobe in the center of the data valid window once at power-up will not guarantee robust design as voltage (V) and temperature (T) vary over time. With data rates beyond 800 Mbps (bit period 1.25 ns), the data valid window at the memory controller during the read operation goes down to the 100-200 ps range. This margin does not provide enough slack to account for VT variations that happen over time, which may cause the data valid window to shift in either direction and at the same time shrink as a percent of the bit period. Hence there is a need to continuously calibrate the placement of the strobe, such that the strobe is always in the center of the data valid window. This self-calibration maximizes link performance over all PVT conditions. This paper describes the implementation of self-calibration techniques used in designing Altera FPGAs at the 65-nm process node.

There is a need to widen the data valid window by reducing some of the uncertainties. This paper also describes the skew compensation techniques used to reduce some of the memory and system uncertainties. Finally, this paper presents a timing analysis with and without self-calibration and skew compensation to justify these features despite the slight overhead in die size and complexity. Even though this paper focuses on memory interfaces using FPGAs, these techniques can be applied to any source-synchronous interface implemented in an ASIC.

Overview of Source-Synchronous Memory Interfaces

Mainstream memory interfaces such as DDR, DDR2, DDR3, RLDRAM II, and QDR II are source-synchronous, where a bidirectional data strobe (DQS in DDR, DDR2, and DDR3, QK in RLDRAM II, and CQ in QDR II) is used for both read and write operations. However, handling timing is different in each case. The data strobe is edge aligned with the data during a read operation and center aligned during a write operation (see Figure 1). One data strobe signal is associated with a number of data bits, usually eight, but can vary from 4 to 36 bits. Within the memory, the data strobe path closely matches the data path, hence the access times with respect to data strobe are significantly better than the access times with respect to clock.

If data strobe is used from the memory to capture the data sent by memory, maximum performance is realized. The challenge for the controller is to transfer the data captured in the memory domain to the system domain. This can either be done with resynchronization FIFOs or registers.



Timing Margin Challenge

Memory vendors have met the performance requirements of processors with several innovations in the architecture. Unfortunately, as shown in Table 1, the DRAM uncertainty as a percent of the bit period goes up with every technology generation. Similarly, the board uncertainties do not decrease as the data rates go up. As a result, the data valid window that the controller sees continues to shrink with every technology generation.

Memory Parameters (ps)	DDR-400	DDR2-800	DDR3-800
t_{CK} Clock Period	5000	2500	2500
Bit Time	2500	1250	1250
t_{DQSQ} DQS to DQ Skew	400	200	200
t_{QH} Data Output Hold Time From DQS	2000	950	900
Data Eye at DRAM	1600	750	700
% bit Period	64%	60%	56%
DRAM Uncertainties	36%	40%	44%

Source: JEDEC (DDR3 specs are preliminary)

Table 1. Memory Specifications for Different DRAM Technologies

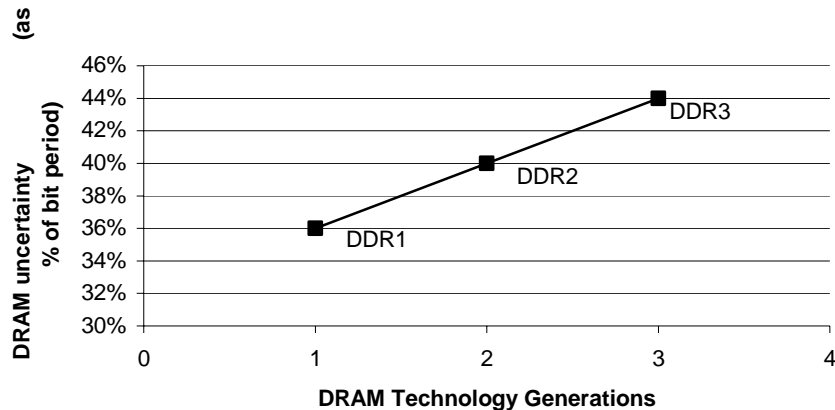


Figure 2. DRAM Uncertainty Trend With Technology Generation

Furthermore, the data valid window shifts due to VT variations over time, as shown in Figure 3. Relying on static timing, the safe valid window over PVT conditions would be very small or even nonexistent.

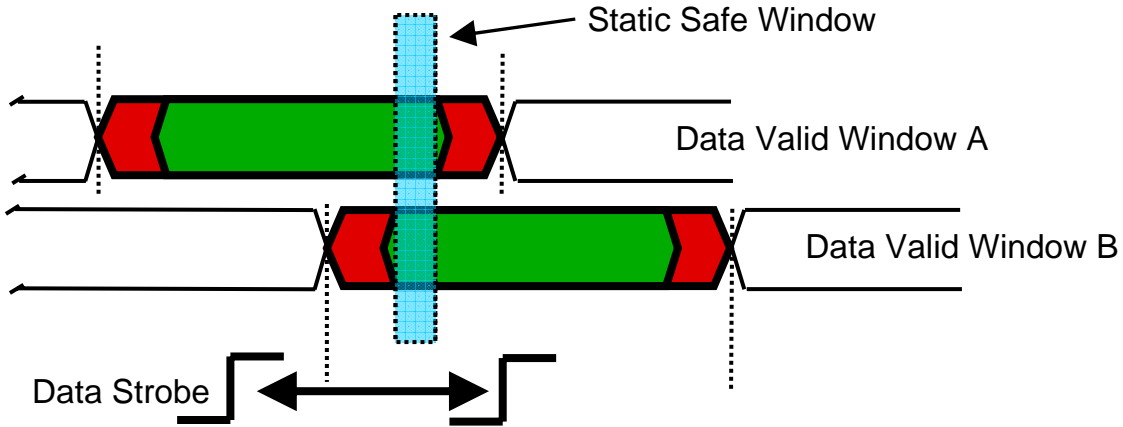


Figure 3. Data Valid Window Shifts Due to VT Variations

Need for Calibration Techniques

The previous section presented an overview of how, as applications move to higher frequencies and newer technology generations, memory and board uncertainties are increasing as a percentage of the bit period. The need to place the strobe in the center of the data valid window is most important, but keeping it centered is equally important as voltage (V) and temperature (T) vary over time. Additionally, there is a need to reduce some of these uncertainties and improve the margin on the controller side to achieve higher performance. Lastly, one of the most difficult tasks in designing a memory subsystem is accurately estimating these uncertainties. Timing analysis is typically done by assuming worst case numbers for all parameters. With calibration, the internal timing is set up in real time in a real system environment to achieve optimum timing, highest performance, and a robust design.

Auto-Calibration Description

The aim of data calibration is to determine and setup the optimum sampling phase of the capture and resynchronization clock. The capture register shown in Figure 4 captures the data from external memory. The resync register is used to transfer the data from memory domain to FPGA/ASIC system domain.

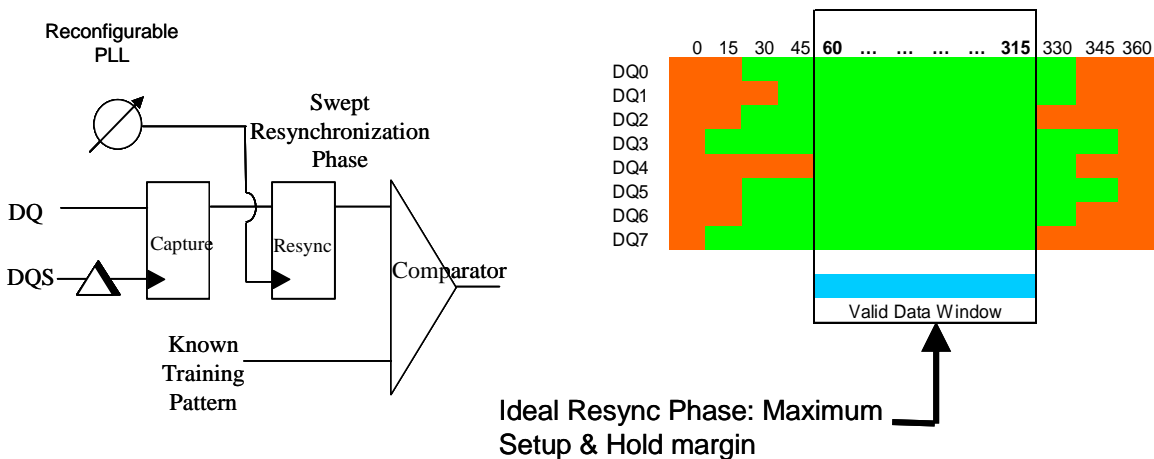


Figure 4. Auto-Calibration System Block Diagram

Figure 5 shows an overview of the two distinct calibration-process phases: startup calibration and tracking.

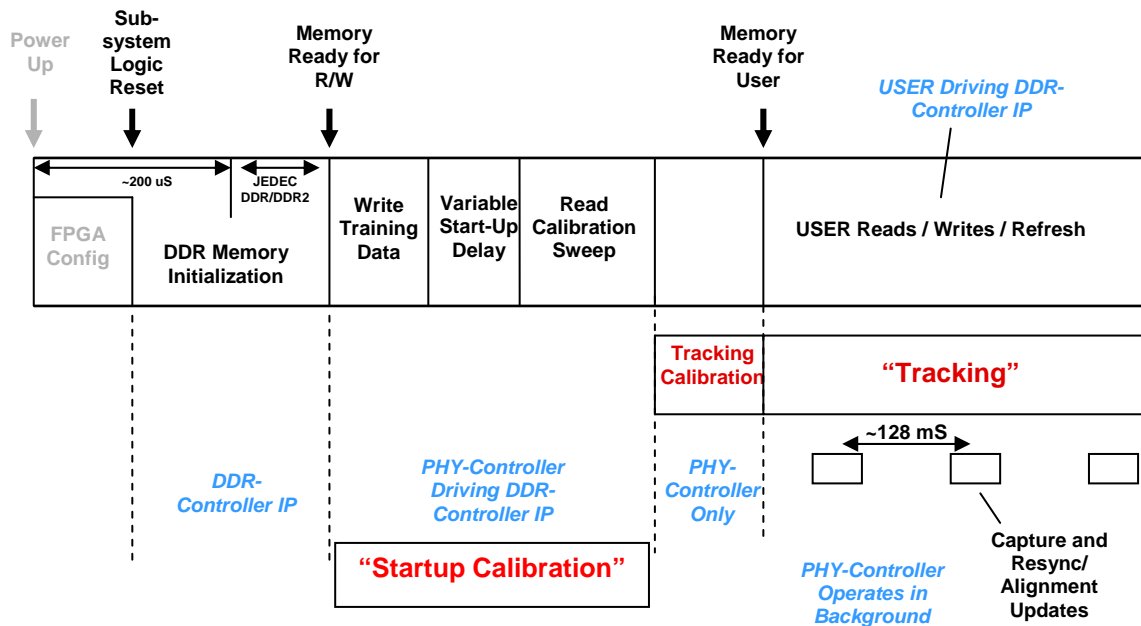


Figure 5. Calibration Sequence Diagram

Startup Calibration

Startup calibration is performed only once at power-up. Once the memory initialization is completed, the calibration sequencer writes a training pattern to the attached memory. Following completion of the training pattern writes, the sequencer reads the training pattern back from the memory. The read data is captured, resynchronized, and written into the internal memory, then the read data is compared against the training pattern written to the memory after power-up. If the read data matches the training pattern, the comparison logic block signals a “pass” to the calibration sequencer. If the read data does not match the training pattern, the comparison logic block signals a “fail” to the calibration sequencer.

Once a training pattern is read, the comparison operation has been completed, and the pass/fail result has been stored, the calibration sequencer uses the phase-locked loop (PLL) reconfiguration logic to shift the resynchronization clock phase by one voltage-controlled oscillator (VCO) phase step. When the PLL reconfiguration has been completed, the calibration sequencer performs another training pattern read and comparison operation. This sequence continues until the PLL VCO phase taps have been swept through two complete cycles. Two cycles are used to enable the correct cycle to be chosen for the full-rate to half-rate demultiplexing as well as for finding the correct resynchronization clock phase.

The training pattern read comparison is done on a pin-by-pin basis in order to minimize logic. Therefore, for a 64-bit DDR2 interface using a single read data demultiplex phase, the calibration sequencer will perform 64 training pattern reads and comparisons before all pass/fail results have been stored for the memory interface pins for a given PLL phase tap. Once all the pass/fail results have been stored, the calibration sequencer calculates the optimum resynchronization clock phase, which places the resynchronization clock in the center of the data valid window.

Mimic Path

Mimic path is used to track delay variation due to VT changes during the memory read and write transactions without interrupting the memory operation. Since the echo clocks (data strobes) in QDR II and RLDRAM II are free running, the echo clock paths can be monitored to track the VT changes on the data bus. However, in the case of DDR DRAMs, the DQS is not free running. Hence a special mimic path is created that mimics the delay of the clock output to the memory as far as the pads of the FPGA and the delay from the input DQS pads to the capture register.

During this tracking operation, the calibration sequencer measures the delay of the mimic path by varying the phase of the measure clock. Any change in the delay of the mimic path indicates a corresponding change in the round-trip delay, and a corresponding adjustment is made to the phase of the resynchronization clock.

The assumption made about the mimic path is that the VT variation on the round-trip delay path that resides outside of the FPGA is negligible, hence there is no variation of the data valid window at the capture registers. Any VT variation in the memory devices is accounted for by timing analysis.

Tracking Calibration

Tracking calibration takes place after the start-up calibration phase has been completed. During initial calibration, the mimic path is sampled using the measure clock. The sampled value is then stored by the calibration sequencer. Once a sample value has been stored, the calibration sequencer uses the PLL reconfiguration logic to change the phase of the measure clock by one VCO phase tap. The calibration sequencer then stores the sampled value for the new mimic path clock phase. This sequence continues until all mimic path clock phase steps have been swept. Once the calibration sequencer has stored all the mimic path sample values, it calculates the optimum sampling phase for the mimic path that is stored as the reference mimic path-sampling phase. This optimum sampling phase is used during the VT tracking phase.

Tracking

Tracking is a continuous process that is transparent to the user and is used to maintain a near-optimum resynchronization clock phase as VT varies. In user mode, the calibration sequencer periodically performs a tracking operation as defined in the tracking calibration description. At the end of the tracking calibration operation, the calibration sequencer compares the most recent optimum tracking phase against the reference

sampling phase. If the sampling phases do not match, this indicates that the mimic path delays have changed due to voltage and temperature.

When the calibration sequencer detects that the mimic path reference and most recent sampling phases do not match, the calibration sequencer uses the PLL reconfiguration logic to change the phase of the resynchronization clock by the same number of VCO taps as calculated during the tracking operation. The most recent mimic path sampling phase is then set as the mimic path reference sampling phase. This allows the tracking process to maintain the near-optimum capture clock-phase setup as voltage and temperature vary over time

Altera provides the memory PHY as a megafunction through the Quartus® II development software. The memory PHY can be integrated with any memory controller as long as the interface and handshake requirements are met.

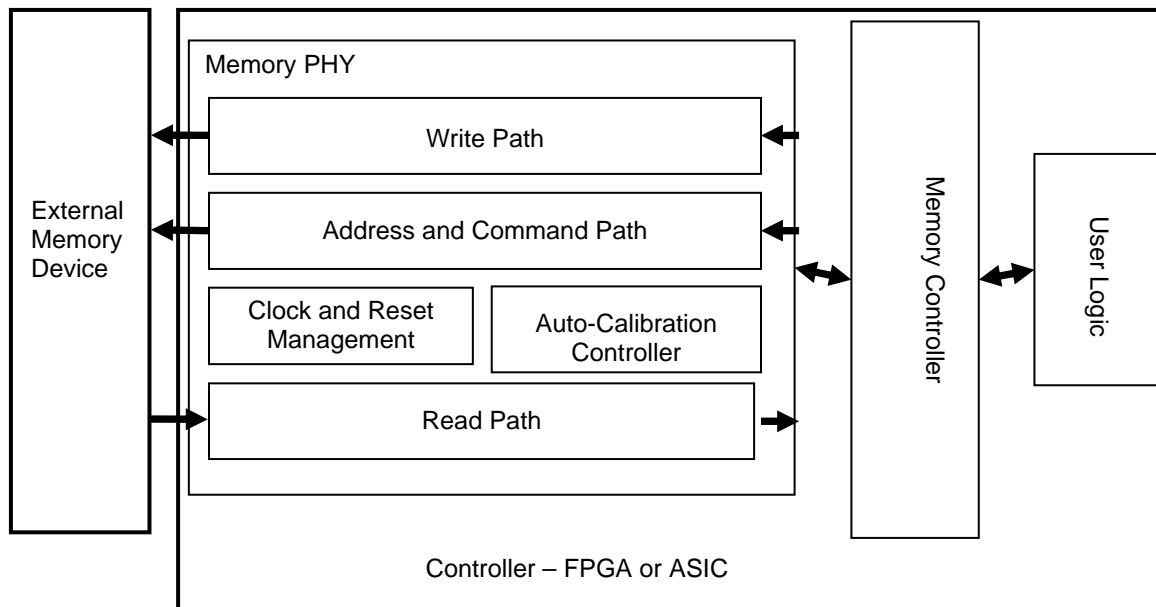


Figure 6. Memory Interface System

Deskew Compensation

In order to further improve the performance of the memory subsystem along with calibration techniques mentioned in the section above, skew between the data pins, including memory device skew, board skew, and controller package skew, can be minimized. Skew improvement can also help in reducing the cost of the system by using lower cost PCBs or connectors. Deskew compensation can be applied for memory interfaces as well as for source-synchronous chip-to-chip interconnects (see Figure 7).

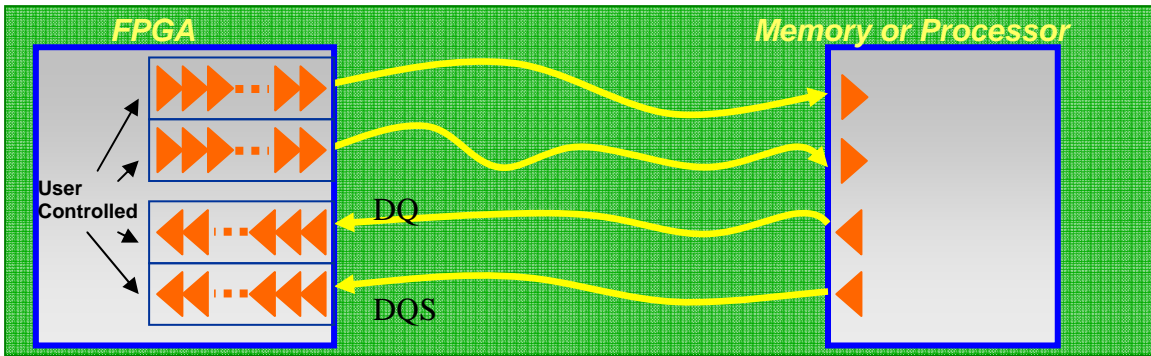


Figure 7. DQ-DQS Skew Components

Deskew calibration will minimize the DQ to DQ and DQ to DQS skew inside the FPGA, on the board, and from the memory device. Deskew compensation must be performed for both read and write paths. By matching the DQ paths and aligning with DQS path, the access time from the memory can be reduced during the read operation. Similarly by matching the DQ and DQS paths, the setup and hold times required by the memory can be reduced during the write operation thus enabling higher performance. Since deskew calibration is done only on power-up, the variation in the DQ and DQS paths due to VT variations cannot be compensated. Timing analysis shows that even though the skews are reduced they are not completely eliminated. The delay setting can be controlled by the calibration sequencer or by configuring the FPGA, using configuration RAM (CRAM) bits as shown in Figures 9 and 10.

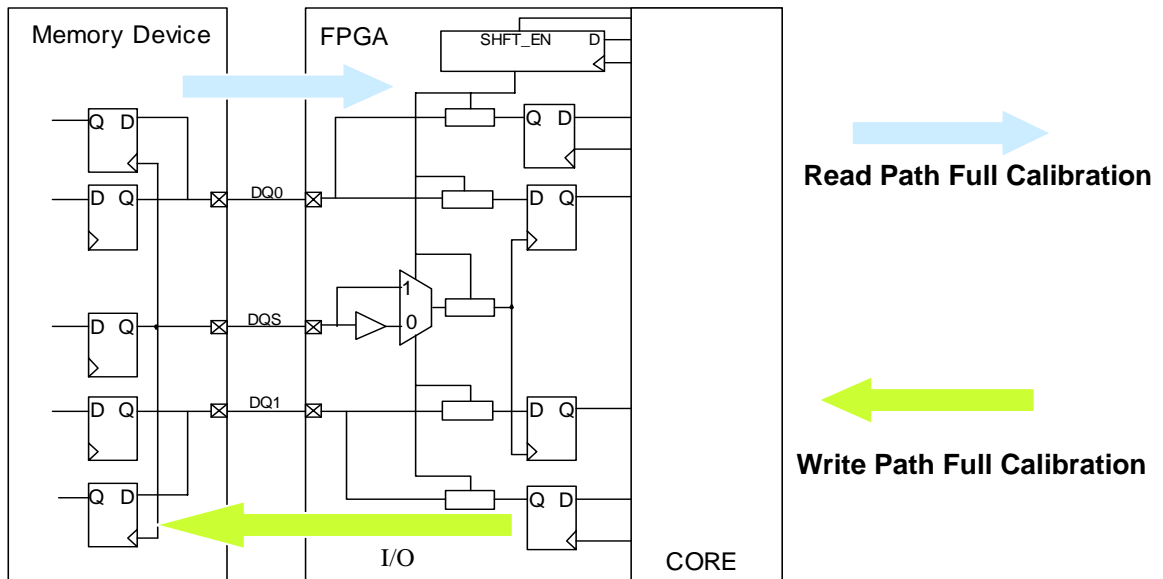


Figure 8. Full Path Calibration With Skew Adjustment Circuits

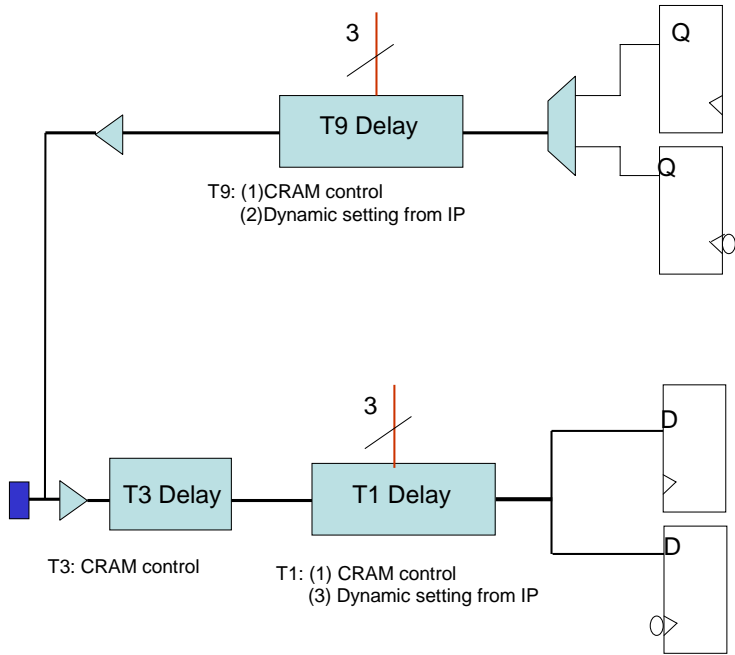


Figure 9. Delay Elements for Deskew Circuits in DQ Paths of the Controller

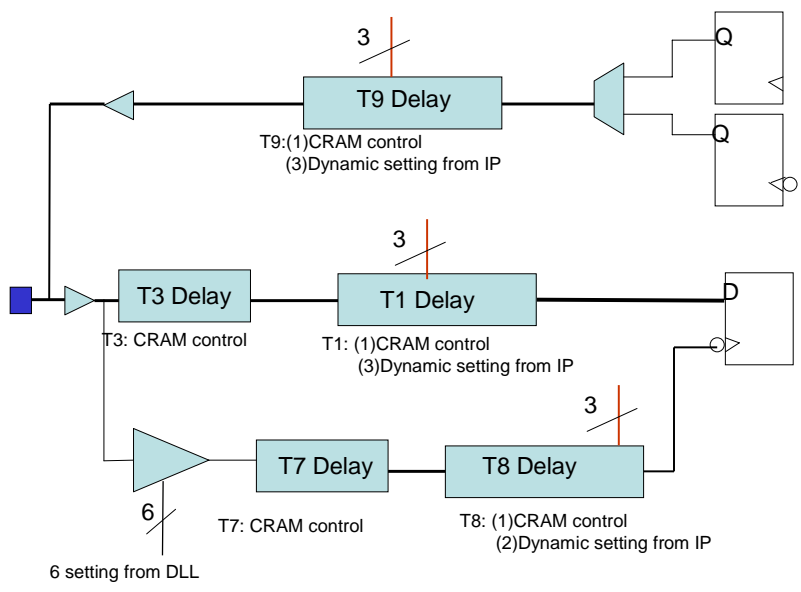


Figure 10. Delay Elements for Deskew Circuits in DQS Path of the Controller

The read path starts from the output driver of the memory device, through the memory package, through the board trace, and ending at the input register of the controller. The procedure for calibrating the read-path deskew is:

- Write the test pattern into the memory device. To ensure that the pattern can be written correctly, the frequency may be lowered during the write.
- Read back the test pattern. If a predefined data pattern is available from the memory, such as the DDR3 MPR (multipurpose register), the write pattern step can be skipped.
- Align all the data paths by adjusting the delay chain on the individual DQ paths to the longest DQ path.
- Pick the DQS delay setting that aligns DQS in the middle of the data valid window.

The simulation results of the deskew calibration algorithm are shown in Figure 8. Note that these delays have a resolution of 50 ps, hence the worst-case skew can be expected among the data bus to be 50 ps.

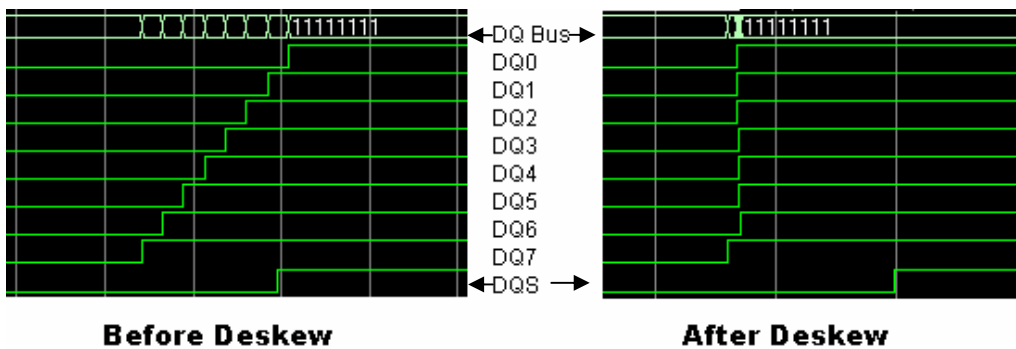


Figure 11. Simulation Results of Deskew Calibration

The write path starts from the output driver of the controller, goes through the controller package parasitics and the board traces, ending at the input register of the memory device. Similar to the read path, the skew between the data lines can be reduced to allow the memory to recognize maximum setup and hold times for its input registers.

Calibration Sequencer

To bring all these individual processes together, a sequencer is designed to control and manage the ordering of these calibration processes, and removing any interdependencies between the calibration processes (read needed for write deskew and write needed for read deskew in the case of bidirectional buses). At the end of the calibration processes, the sequencer applies relevant settings into the I/O that allows the next process to occur. After a complete interface is calibrated, the sequencer passes control of the interface to the memory controller. The calibration sequencer will be offered as soft IP for Altera FPGAs.

Timing Analysis

Parameter	Timing Without Calibration (ps)*	Timing With Calibration (ps)*	Description
t_{HP}	1250	1250	Ideal half period time
DRAM Uncertainties	500	200	DQ-DQ and DQS-DQ skew are reduced with de-skew calibration.
t_{DCD}	125	125	FPGA output clock duty cycle distortion ($\pm 5\%$)
FPGA + Board Uncertainties	550	300	FPGA uncertainties include DLL jitter, setup hold times of IOE registers, DQS clock tree skew and rise/fall mismatch, and SSI jitter
Margin	75	625	Read timing margin

Table 2. Timing Improvements With Calibration for 400-MHz DDR3 Read Operation

Parameter	Timing Without Calibration (ps)*	Timing With Calibration (ps)*	Description
t_{HP}	1250	1250	Ideal half period time
DRAM Uncertainties	425	200	DQ-DQ and DQS-DQ skew are reduced with de-skew calibration.
t_{DCD}	125	125	FPGA output clock duty cycle distortion ($\pm 5\%$)
FPGA + Board Uncertainties	575	300	FPGA uncertainties include PLL jitter, Clock network skew, PRBS data pattern jitter, rise/fall mismatch, and SSO pushout
Margin	125	625	Write timing margin

Table 3. Timing Improvements With Calibration for 400-MHz DDR3 Write Operation

**Timing numbers are preliminary and based on simulation data.*

As the timing analysis shows, timing margin is improved significantly with deskew and auto-calibration. The deskew calibration scheme has been verified with simulations and the auto-calibration scheme has been validated with proof-of-concept experiments using Stratix-II FPGAs. Figures 12 and 13 show the detailed eye diagram captured for DDR2 implementation of 667-Mbps and 800-Mbps respectively using Stratix II devices (Altera's 90-nm FPGAs). Auto-calibration significantly simplifies memory interface design and makes the interface very robust over PVT conditions. These calibration features will be built in Altera's 65-nm FPGA families.

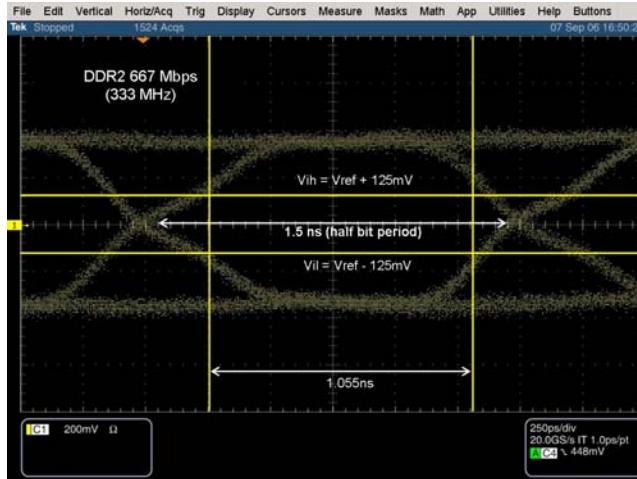


Figure 12. Eye Diagram of DDR2 667-Mbps Interface on Stratix II FPGA With Auto-Calibration

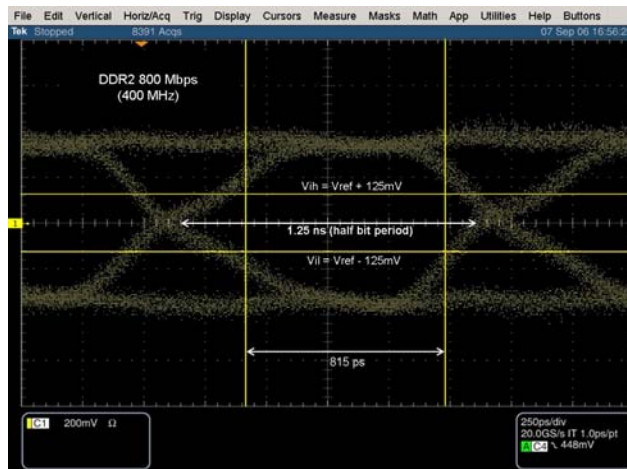


Figure 13. Eye Diagram of DDR2 800-Mbps Interface on Stratix II FPGA With Auto-Calibration

Die-Size Impact

To fairly compare the overhead based on the calibration, the deskew circuits were calculated as a percentage of the I/O periphery circuits. The overhead is less than 5 percent. From the deskew timing analysis, the improvement in margin is due to both calibration and deskew, which can be over 40 percent of the bit period.

Calibration Circuit Overhead

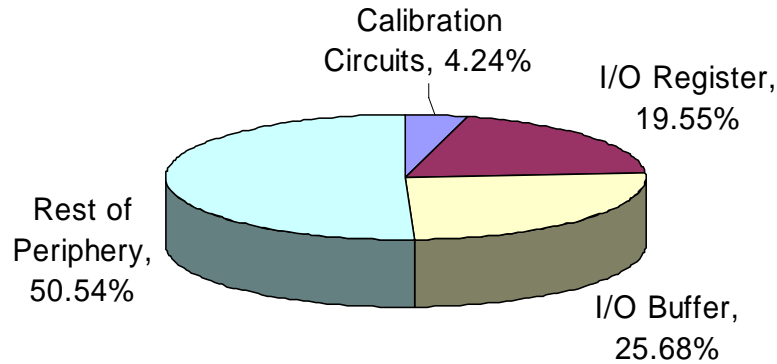


Figure 14. Calibration Circuit Overhead

Summary

Inaccurate timing errors based on usage of the static timing approach can limit the performance of the memory interfaces and affect robustness. Simulations and proof-of-concept testing has shown that the calibration techniques discussed in this paper improve timing margins for high-performance source-synchronous interfaces and can be implemented with a reasonable cost overhead. More importantly, these calibration techniques significantly simplify timing closure for memory interfaces that enable system designers to focus on their logic timing closure and gain a higher confidence in creating functioning memory interfaces with push-button-like capability.



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.