Interfacing FPGAs to DDR3 SDRAM memories

Paul Evans - November 08, 2007

DDR3 SDRAM memory architectures support higher bandwidths with bus rates of 600 Mbps to 1.6 Gbps (300 to 800 MHz), 1.5V operation for lower power, and higher densities of 2 Gbits on a 90-nm process. While this architecture is undoubtedly faster, larger, and lower power per bit, how is the interface between a DDR3 SDRAM DIMM to an FPGA accomplished?

The key word: leveling.

Without the leveling feature designed directly into the FPGA I/O structure, interfacing anything to a DDR3 SDRAM DIMM is complicated, costly, and involves numerous external components including delay lines and associated controls.

What is leveling and why is it so important?
To improve signal integrity when supporting higher frequencies the, JEDEC committee defined a fly-by termination schemes used with the clocks and the command/address bus signals to improve overall signal integrity in support of higher performance. Fly-by topology reduces the simultaneous switching noise (SSN) by deliberately causing flight-time skew between clock and data/strobes at every DRAM as the clock and address/command signals traverse the DIMM, as shown in Figure 1.

Figure 1. DDR3 SDRAM DIMMs: Flight-Time Skew Reduces SSN, Data must be Leveled up to Two Clock Cycles at the Controller
Flight time skew can be up to 0.8 tCK, which is a wide enough spread not to know in which of two clock cycles the data may return. Therefore, the "leveling" feature was defined for DDR3 memories by JEDEC to enable controllers to compensate for this skew by adjusting the timing per byte lane.

Current generation FPGAs offer many features that interface with double data rate SDRAM memories for a wide range of applications such as desktops, servers, storage, LCD displays, and networking and communication equipment. However, to work with the newest DDR3 SDRAM technology, a robust leveling scheme is required.

**FPGA I/O structure**

FPGAs, such as the high performance Altera Stratix III device family, provide I/Os capable of speeds up to 400 MHz (800 Mbps) with higher frequencies expected soon and greater flexibility to support existing and emerging external memory standards such as DDR3.

**Read Leveling**

![Diagram 1](image1)

![Diagram 2](image2)

During a read operation, the memory controller side must compensate for the delays introduced by the fly-by memory topology that impact the read cycle. Leveling should be thought of as more than just I/O delay that appears in the data path. 1T (a register used to hold data for one complete double data rate cycle) and neg-edge registers are also required to level or align all the data. Each DQS requires a separate phase shift of the resync clock position (process, voltage and temperature [PVT])
Initially, each separate DQS is phase-shifted a nominal 90° and the DQ data associated with its group is captured. Then a free-running resynchronization clock (at the same frequency and phase of the DQS) is used to move the data from its capture domain into the leveling circuit—shown by the pink and orange links in the figure. At this stage, each DQS group has a separate resynchronization clock.

Next the DQ data is passed to the 1T registers.

Note that in this example, the 1T registers are not required in the lower channel. This process begins aligning the upper channel to the lower channel. Whether the 1T registers are required or not for any given channel is decided automatically as part of the calibration scheme in free PHY IP core.
Both DQS groups are then passed on the neg-edge registers. Again, optional registers are switched in or out at start up by the automatic calibration process, if required. The final stage is to align both the upper and lower channels back onto the same resynchronization clock, thus creating a source synchronous interface that passes a fully-aligned, or leveled, single data rate (SDR) data to the FPGA fabric.

**Write leveling**

Similar to read leveling but in reverse, DQS groups are launched at separate times to coincide with a clock arriving at devices on the DIMM, and must meet the tDQSS parameters of ±0.25 tCK. The controller must adjust the DQS-to-CK relationship by creating a feedback loop where by the controller writes to the DRAM and reads back sweeping through sequential phases until it find the end points of the write window. The data launch point is then set in the middle of the good window for the best set up and hold margin.

**Other FPGA I/O innovations**

High-end FPGAs have a host of other innovative I/O features that allow simple and robust interfacing to a range of memory interfaces such as dynamic on-chip termination (OCT), variable I/O delay, and half data rate (HDR) capability, as shown in

![Figure 3. Useful I/O Features for DDR3 SRAM Memory Interfaces](image-url)
This remainder of this article follows these features, examining each step in detail from left to right as they are shown in the diagrams.

*Dynamic OCT*
Parallel and serial OCT provide the appropriate line termination and impedance matching for both the read and write busses. This removes the need for external resistors at the FPGA and saves on external component costs, board space, and routing complexity. It also significantly reduces power consumption, because the parallel termination is effectively out of circuit on a write operation.

Figure 4. Dynamic OCT - Read and Write Operations

shows the terminations for both read and write operations.

Variable Delay for DQ Deskew
Variable input and output delay (shown in Figure 5) is used for trace length mismatch and electrical deskew. The fine input and output delay resolution (i.e., 50-picosecond [ps] steps) are used for finer inter-DQS deskew (separate to the leveling function) which are caused either by mismatch in board length or variations in I/O buffers of the FPGA and memory devices, as shown in Table 1. Ultimately, this increases the capture margin for each DQS group.

<table>
<thead>
<tr>
<th>Table 1 Resolution and Absolute Value Pending Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td>750 ps</td>
</tr>
<tr>
<td>50 ps</td>
</tr>
<tr>
<td><strong>Output</strong></td>
</tr>
<tr>
<td>1100 ps</td>
</tr>
<tr>
<td>50 ps</td>
</tr>
</tbody>
</table>
The delay elements are reached from the FPGA fabric at run time to implement automatic DDR3 deskew algorithms as part of the start-up calibration process.

Figure 6. Conceptual DQ Deskew within a DQS Group centered around 90-Degree Phase-Shifted DQS

shows an illustration of how DQ data can be deskewed and centered around DQS for extra capture margin. The output delay can also be used to insert a small amount of skew into the output path to intentionally reduce the number of I/Os being switched simultaneously.

Reliable Capture

Figure 7. DQ Capture Circuit
The DQS signals serve as the input strobes and must be shifted to an optimal position to capture read transactions. The phase-shift circuitry (shown in Figure 7) can shift the incoming DQS signals by 0°, 22.5°, 30°, 36°, 45°, 60°, 67.5°, 72°, 90°, 108°, 120°, 135°, 144°, or 180°, depending on the DLL frequency mode. The shifted DQS signals are then used as clocks at the I/O element input registers.

Figure 7. DQ Capture Circuit

Figure 8. DLL and DQS Phase-Shift Circuitry
The delay-locked loop (DLL) shown in Figure 7 maintains the phase shift in a fixed location across PVT.

![Figure 7. DQ Capture Circuit](image)

Figure 8. DLL and DQS Phase-Shift Circuitry shows the relationship between the DLL and phase-shift circuitry. The phase comparator of the DLL blocks works to keep the phase difference between the two inputs to zero. This is accomplished by updating the delays (10 – 16) in the DLL blocks equally. The control signal that is used to update one of the delay blocks in the DLL is also sent to the delay blocks in the DQS input path. For example, 90° can be achieved by using all 16 delays in the DLL and the 4th delay tap in the DQS phase shift input path:

\[
\frac{360}{16} \times 4 = 90°
\]

or 36° can be achieved by selecting 10 delays in the DLL and tap 4 of the DQS phase shift input path:

\[
\frac{360}{10} \times 1 = 36°
\]

or 120°:

\[
\frac{360}{12} \times 4 = 120°
\]

The DLL uses a frequency reference to generate control signals dynamically for the delay chains in each of the DQS pins, allowing it to compensate for PVT variations. There are four DLLs in this FPGA, each located in a corner of the device, which allows each DLL to reach two sides of the device, thus allowing support for multiple DDR3 SDRAM memory interfaces on all sides of the device.
High-speed data rate domain crossing and design simplification

DDR capture registers and HDR registers allow safe transfer of data from the double data rate domain (data on both edges of the clock), down to the SDR domain (data on single positive edge of clock at the same frequency, but at twice the data width), down to HDR domain (data on the positive edge of clock, but the frequency is now half that of the SDR and the data width is again doubled), making the internal design timing much easier to achieve.

Die, package, and digital signal integrity enhancements

Figure 9. Typical Block Diagram of Input Path Registers

shows how the DQ data moves through these various data rate domains.
The design of an FPGA die and package must provide robust signal integrity for the high-performance memory interfaces (i.e., having an 8:1:1 user I/O to ground and power ratio and optimized signal return paths as shown in Figure 10. Eight User I/Os to each Power and Ground.
In addition, the FPGA should provide dynamic OCP and variable slew rate to manage signal rise and fall times as well as programmable drive strength to match your desired standard (i.e. SSTL 1.5 Class II).

**Conclusion**

High-performance FPGAs complement high-performance DDR3 SDRAM DIMMs by providing high-memory bandwidth, improved timing margin, and great flexibility in system design. With DDR3 expected to soon surpass DDR2 in usage, the lower cost, higher performance, higher density and superior signal integrity provided by high-end FPGAs must provide JEDEC-compliant read/write leveling functionality to interface to high performance DDR3 SDRAM DIMMs. The combination of FPGAs with DDR3 SDRAM supports the high-throughput requirements of today's and next generation communication, networking, and digital signal processing systems.

**Author’s Biography**

Paul Evans is the product marketing engineer responsible for Altera's Stratix III FPGAs. He joined Altera in 2000 as a senior application engineer. Prior to joining Altera, Mr. Evans was a technical services manager for Ometron Ltd. where he established Ometron’s technical services department. Mr. Evans has also held engineering positions at Image Automation Inc. and Smiths Industries Ltd. He holds a BEng in digital electronic engineering from the University of Kent at Canterbury in England.