

# Implementing High-Performance Memory Interfaces with Virtex-4 FPGAs

You can center-align clock-to-read data at “run time” with ChipSync technology.

by Adrian Cosoroaba  
Marketing Manager  
Xilinx, Inc.  
[adrian.cosoroaba@xilinx.com](mailto:adrian.cosoroaba@xilinx.com)

As designers of high-performance systems labor to achieve higher bandwidth while meeting critical timing margins, one consistently vexing performance bottleneck is the memory interface. Whether you are designing for an ASIC, ASSP, or FPGA, capturing source-synchronous read data at transfer rates exceeding 500 Mbps may well be the toughest challenge.

## Source-Synchronous Memory Interfaces

Double-data rate (DDR) SDRAM and quad-data-rate (QDR) SRAM memories utilize source-synchronous interfaces through which the data and clock (or strobe) are sent from the transmitter to the receiver. The clock is used within the receiver interface to latch the data. This eliminates interface control issues such as the time of signal flight between the memory and the FPGA, but raises new challenges that you must address.

One of these issues is how to meet the various read data capture requirements to implement a high-speed source-synchronous interface. For instance, the receiver must ensure that the clock or strobe is routed to all data loads while meeting the required input setup and hold timing. But source-synchronous devices often limit the loading of the forwarded clock. Also, as the data-valid window becomes smaller at higher frequencies, it becomes more important (and simultaneously more challenging) to align the received clock with the center of the data.

### Traditional Read Data Capture Method

Source-synchronous clocking requirements are typically more difficult to meet when reading from memory compared with writing to memory. This is because the DDR and DDR2 SDRAM devices send the data edge aligned with a non-continuous strobe signal instead of a continuous clock. For low-frequency interfaces up to 100 MHz, DCM phase-shifted outputs can be used to capture read data.

Capturing read data becomes more challenging at higher frequencies. Read data can be captured into configurable logic blocks (CLBs) using the memory read strobe, but the strobe must first be delayed so that its edge coincides with the center of the data valid window. Finding the correct phase-shift value is further complicated by process, voltage, and temperature (PVT) variations. The delayed strobe must also be routed onto low-skew FPGA clock resources to maintain the accuracy of the delay.

The traditional method used by FPGA, ASIC, and ASSP controller-based designs employs a phase-locked loop (PLL) or delay-locked loop (DLL) circuit that guarantees a fixed phase shift or delay between the source clock and the clock used for capturing data (Figure 1). You can insert this phase shift to accommodate estimated process, voltage, and temperature variations. The obvious drawback with this method is that it fixes the delay to a single value predetermined during the design phase. Thus, hard-to-predict variations within the system itself – caused by different routing to different memory devices, variations between FPGA or ASIC devices, and ambient system condi-

tions (voltage, temperature) – can easily create skew whereby the predetermined phase shift is ineffectual.

These techniques have allowed FPGA designers to implement DDR SDRAM memory interfaces. But very high-speed 267

also cause data and address timing problems at the input to the RAM and the FPGA's I/O blocks (IOB) flip-flop. Furthermore, as a bidirectional and non-free-running signal, the data strobe has an increased jitter component, unlike the clock signal.

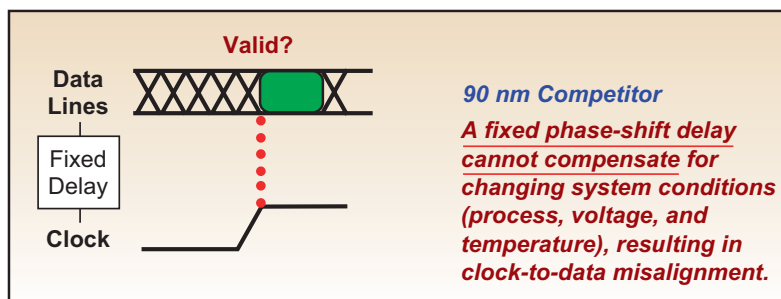


Figure 1 – Traditional fixed-delay read data capture method

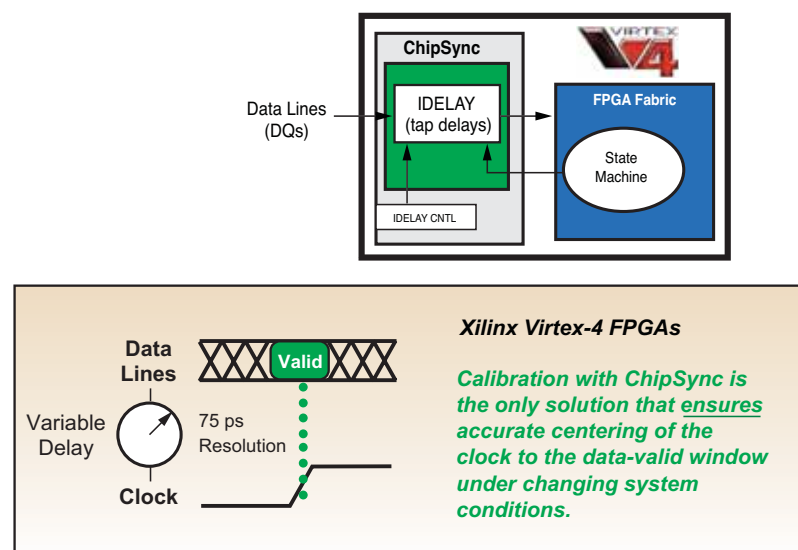


Figure 2 – Clock-to-data centering using ChipSync tap delays

MHz DDR2 SDRAM and 300 MHz QDR II SRAM interfaces demand much tighter control over the clock or strobe delay.

System timing issues associated with setup (leading edge) and hold (trailing edge) uncertainties further minimize the valid window available for reliable read data capture. For example, 267 MHz (533 Mbps) DDR2 read interface timings require FPGA clock alignment within a .33 ns window.

Other issues also demand your attention, including chip-to-chip signal integrity, simultaneous switching constraints, and board layout constraints. Pulse-width distortion and jitter on clock or data strobe signals

### Clock-to-Data Centering Built into Every I/O

Xilinx® Virtex™-4 FPGAs with dedicated delay and clocking resources in the I/O blocks – called ChipSync™ technology – answer these challenges. These devices make memory interface design significantly easier and free up the FPGA fabric for other purposes. Moreover, Xilinx offers a reference design for memory interface solutions that center-aligns the clock to the read data at “run time” upon system initialization. This proven methodology ensures optimum performance, reduces engineering costs, and increases design reliability.

# ChipSync features are built into every I/O. This capability provides additional flexibility if you are looking to alleviate board layout constraints and improve signal integrity.

ChipSync technology enables clock-to-data centering without consuming CLB resources. Designers can use the memory read strobe purely to determine the phase relationship between the FPGA's own DCM clock output and the read data. The read data is then delayed to center-align the

determine the phase relationship between the FPGA clock and the read data received at the FPGA. This is done using the memory read strobe. Based on this phase relationship, the next step is to delay read data to center it with respect to the FPGA clock. The delayed read data is then captured

transition (second-edge taps) in the FPGA clock domain.

Having determined the values for first-edge taps and second-edge taps, the state machine logic can compute the required data delay. The pulse center is computed with these recorded values as  $(\text{second-edge taps} - \text{first-edge taps})/2$ . The required data delay is the sum of the first-edge taps and the pulse center. Using this delay value, the data-valid window is centered with respect to the FPGA clock.

ChipSync features are built into every I/O. This capability provides additional flexibility if you are looking to alleviate board layout constraints and improve signal integrity.

Each I/O also has input DDR flip-flops required for read data capture either in the delayed memory read strobe domain or in the system (FPGA) clock domain. With these modes you can achieve higher design performance by avoiding half-clock-cycle data paths in the FPGA fabric.

Instead of capturing the data into a CLB-configured FIFO, the architecture provides dedicated 500 MHz block RAM with built-in FIFO functionality. These enable a reduction in design size, while leaving the CLB resources free for other functions.

## Clock-to-Data Phase Alignment for Writes

Although the read operations are the most challenging part of memory interface design, the same level of precision is required in write interface implementation. During a write to the external memory device, the clock/strobe must be transmitted center-aligned with respect to data. In the Virtex-4 FPGA I/O, the clock/strobe is generated using the output DDR registers clocked by a DCM clock output (CLK0) on the global clock network. The write data is transmitted using the output DDR registers clocked by a

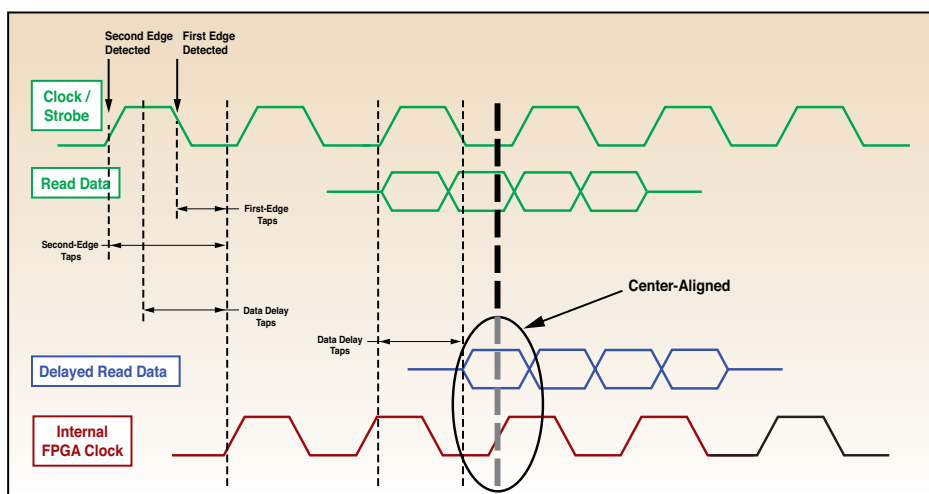


Figure 3 – Clock-to-data centering at “run time”

FPGA clock in the read data window for data capture. In the Virtex-4 FPGA architecture, the ChipSync I/O block includes a precision delay block known as IDELAY that can be used to generate the tap delays necessary to align the FPGA clock to the center of the read data (Figure 2).

Memory read strobe edge-detection logic uses this precision delay to detect the edges of the memory read strobe from which the pulse center can be calculated in terms of the number of delay taps counted between the first and second edges. Delaying the data by this number of taps aligns the center of the data window with the edge of the FPGA DCM output. The tap delays generated by this precision delay block allow alignment of the data and clock to within 75 ps resolution.

The first step in this technique is to

directly in input DDR flip-flops in the FPGA clock domain.

The phase detection is performed at run time by issuing dummy read commands after memory initialization. This is done to receive an uninterrupted strobe from the memory (Figure 3).

The goal is to detect two edges or transitions of the memory read strobe in the FPGA clock domain. To do this, you must input the strobe to the 64-tap IDELAY block that has a resolution of 75 ps. Then, starting at the 0-tap setting, IDELAY is incremented one tap at a time until it detects the first transition in the FPGA clock domain. After recording the number of taps it took to detect the first edge (first-edge taps), the state machine logic continues incrementing the taps one tap at a time until it detects the second

DCM clock output that is phase-offset 90 degrees (CLK270) with respect to the clock used to generate clock/strobe. This phase shift meets the memory vendor specification of centering the clock/strobe in the data window.

Another innovative feature of the output DDR registers is the SAME\_EDGE mode of operation. In this mode, a third register clocked by a rising edge is placed on the input of the falling-edge register. Using this mode, both rising-edge and falling-edge data can be presented to the output DDR registers on the same clock edge (CLK270), thereby allowing higher DDR performance with minimal register-to-register delay.

### Signal Integrity Challenge

One challenge that all chip-to-chip, high-speed interfaces need to overcome is signal integrity. Having control of cross-talk, ground bounce, ringing, noise margins, impedance matching, and decoupling is now critical to any successful design.

The Xilinx column-based ASMBL architecture enables I/O, clock, and power and ground pins to be located anywhere on the silicon chip, not just along the periphery. This architecture alleviates the problems associated with I/O and array dependency, power and ground distribution, and hard-IP scaling. Special FPGA packaging technology known as SparseChevron enables distribution of power and ground pins evenly across the package. The benefit to board designers is improved signal integrity.

The pin-out diagram in Figure 4 shows how Virtex-4 FPGAs compare with a competing Altera Stratix-II device that has many regions devoid of returns.

The SparseChevron layout is a major reason why Virtex-4 FPGAs exhibit unmatched simultaneous switching output (SSO) performance. As demonstrated by signal integrity expert Howard Johnson, Ph.D., these domain-optimized FPGA devices have seven times less SSO noise and crosstalk when compared to alternative FPGA devices (Figure 5).

Meeting I/O placement requirements and enabling better routing on a board requires unrestricted I/O placements for

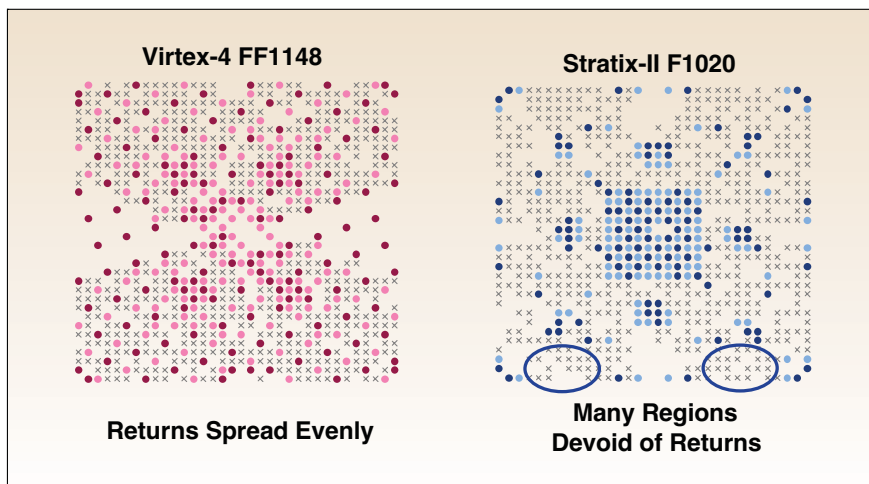


Figure 4 – Pin-out comparison between Virtex-4 and Stratix-II FPGAs

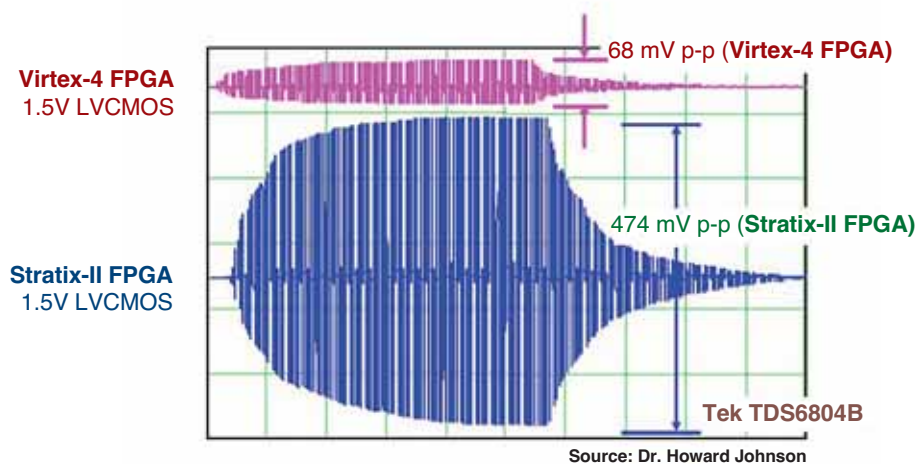


Figure 5 – Signal integrity comparison using the accumulated test pattern

an FPGA design. Unlike competing solutions that restrict I/O placements to the top and bottom banks of the FPGA and functionally designate I/Os with respect to address, data and clock, Virtex-4 FPGAs provide unrestricted I/O bank placements.

Finally, Virtex-4 devices offer a differential DCM clock output that delivers the extremely low jitter performance necessary for very small data-valid windows and diminishing timing margins, ensuring a robust memory interface design.

These built-in silicon features enable high-performance synchronous interfaces for both memory and data communications in single or differential mode. The ChipSync technology enables data rates

greater than 1 Gbps for differential I/O and more than 600 Mbps for single-ended I/O.

### Conclusion

As with most FPGA designs, having the right silicon features solves only part of the challenge. Xilinx also provides complete memory interface reference designs that are hardware-verified and highly customizable. The Memory Interface Generator, a free tool offered by Xilinx, can generate all of the FPGA design files (.rtl, .ucf) required for a memory interface through an interactive GUI and a library of hardware-verified designs.

For more information, visit [www.xilinx.com/memory](http://www.xilinx.com/memory).