XAPP701 (v2.0) March 12, 2007

DDR2 SDRAM Physical Layer Using Direct-Clocking Technique

Author: Tze Yi Yeoh

Summary

This application note describes the DDR2 SDRAM physical layer design using the direct-clocking technique in a Virtex^{TM}-4 device. The direct-clocking technique utilizes some of the architectural features unique to the Virtex-4 family, for example, the 64-tap absolute delay line provided in each I/O block (IOB).

Introduction

The DDR2 SDRAM memory interface is a source-synchronous interface in which the data and clock/strobe that the external memory device transmits is edge aligned. To capture this transmitted data in the Virtex-4 device, either the clock/strobe or the data is delayed.

In the direct-clocking technique of data capture, each data bit is delayed and is center aligned with respect to the internal FPGA clock. The internal FPGA clock captures the transmitted data. Then each data bit is calibrated on a per-bit basis to the internal FPGA clock with a training pattern that the memory sends. The clock/strobe that the memory transmits is not used in any way by the memory controller. As a result, there are no restrictions on the number of data bits associated with a strobe. Because the strobe does not need to be distributed to the associated data bits, no additional clocking resources are required.

Each data line is routed through an IDELAY primitive. A calibration module positions the data eye such that it is centered with respect to the FPGA clock. In addition to accurately positioning the data eye of each data bit in a region that has maximum margin, this technique significantly reduces package skew, clock tree skew, and PCB layout skew. The design and implementation details of the direct-clocking scheme are explained in the following sections.

This application note describes the physical layer of the DDR2 SDRAM memory design. The controller design is described in XAPP702: *DDR2 Controller Using Virtex-4 Devices*.

Calibration Procedure

The calibration procedure begins by executing a write to memory and loading a training pattern in the memory. The training pattern is a continuously oscillating pattern (`...0101...`). At the start of calibration, the controller performs a continuous read from memory. As a result, a continuously oscillating pattern is present on all data channels during the calibration procedure.

Figure 1 shows a timing diagram of the typical case of the calibration procedure.

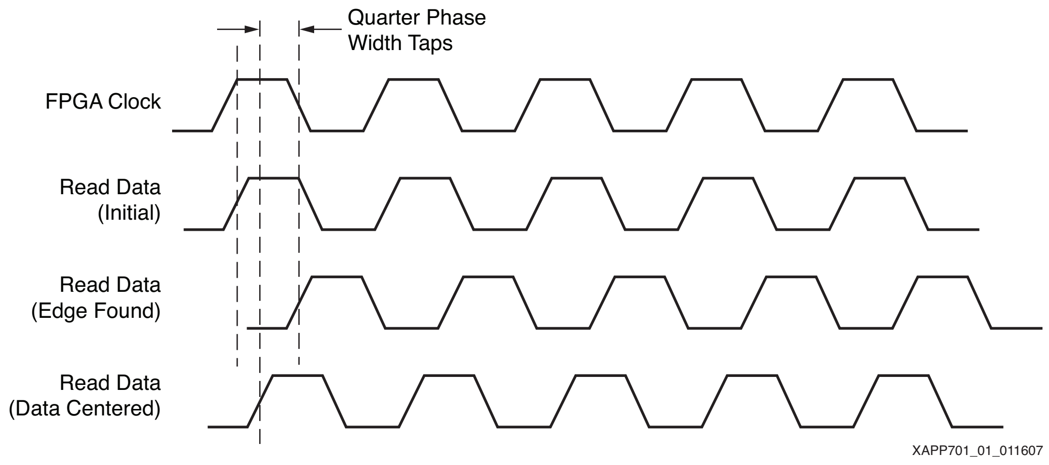

Figure 1: **Timing Diagram Illustrating Typical Case of Calibration Procedure**

In the timing diagram for the typical case shown in Figure 1, the FPGA clock edge falls somewhere within the read data valid window. In this case, the controller increments the IDELAY of the read data channel until an edge is found. The edge detection mechanism is executed by storing the initial value of the read data channel and periodically comparing the current sampling value of the channel. An edge is detected when the current sampling value of the channel does not match the stored value.

When the edge is found, the read data channel IDELAY is decremented by an amount equal to a quarter clock period of the bit time. This amount is a fixed parameter depending on the operating frequency of the memory controller.

Figure 2 shows a timing diagram of a clock edge-straddle case, which is one of two corner cases presented in this document. The clock edge-straddle case occurs when edges of two or more read data bits straddle a clock edge.

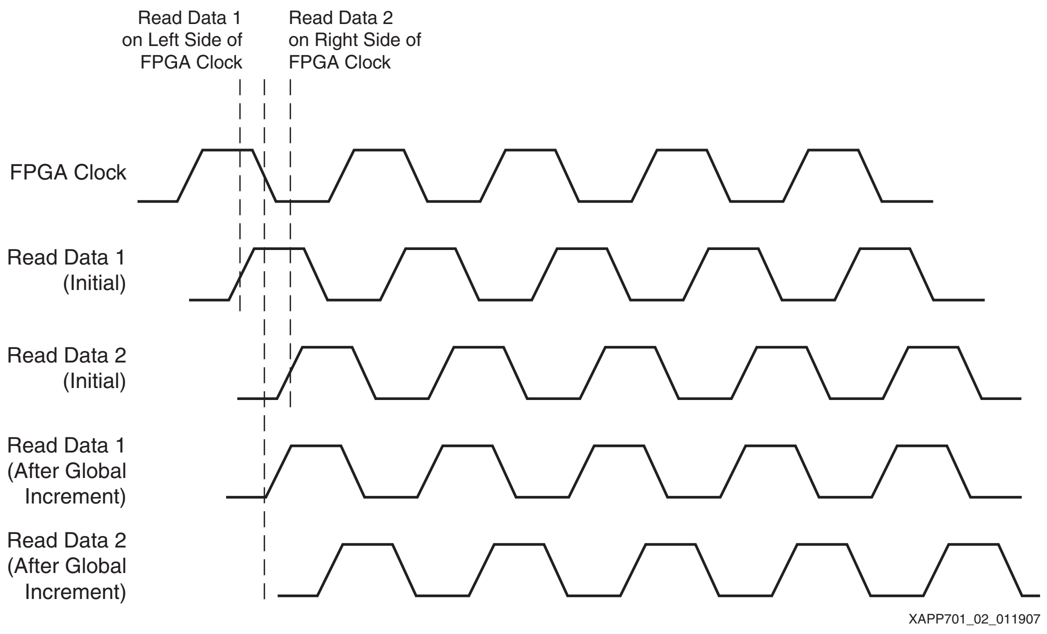
Read Data 1 on Left Side of FPGA Clock
Read Data 2 on Right Side of FPGA Clock
FPGA Clock
Read Data 1 (Initial)
Read Data 2 (Initial)
Read Data 1 (After Global Increment)
Read Data 2 (After Global Increment)
XAPP701_02_011907

Figure 2: **Timing Diagram Illustrating Clock Edge-Straddle Case**

As shown in Figure 2, if no corrective action is taken and the calibration algorithm is allowed to proceed, the result is the read data channels that straddle the clock edge develop one bit time of skew with respect to each other. Because the training pattern is an oscillating pattern, the controller detects an edge straddle condition by checking that all the read data channels are either all zeroes or all ones.

Under real-world conditions, the data and clock edges jitter in time, and the controller monitors for the straddle condition for 10 clock cycles. If a straddle condition is detected, the controller increments the IDELAY modules of all the read data bits until all data bits clear the clock edge. After all data bits clear the clock edge, then the calibration procedure proceeds as in the typical case.

Figure 3 shows a timing diagram of an edge-alignment case, which is a second corner case. The edge-alignment case occurs when the clock and read data edges are edge aligned.

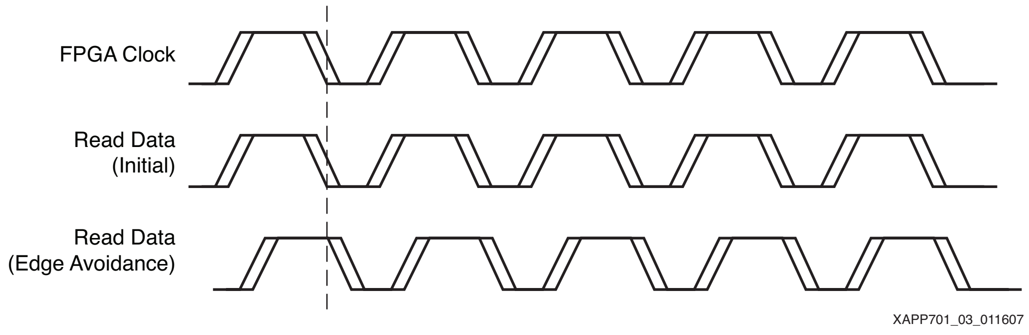

Figure 3: **Timing Diagram Illustrating Edge Alignment Case**

As shown in Figure 3, jitter on the clock and read data edges can disrupt the calibration algorithm. The jitter condition is detected using the same mechanism as the clock edge-straddle case. To get out of edge alignment, all data bits are incremented by 2 taps. After all data bits clear the clock edge, then the calibration proceeds as in the typical case.

At the lower extremes of the operating frequency range (that is,150 MHz and below), the calibration algorithm can use more than the 64-tap limit provided by the IDELAY module. If the IDELAY module is incremented beyond the 64-tap limit, the IDELAY module rolls over to Tap 0. To prevent this rollover from occurring, a tap counter in the calibration module tracks the number of taps incremented and stops the increment when the tap count reaches 55 taps. At this point, the calibration module decrements by a quarter clock period.

The state transition diagram in Figure 4 shows a summary of the calibration algorithm.

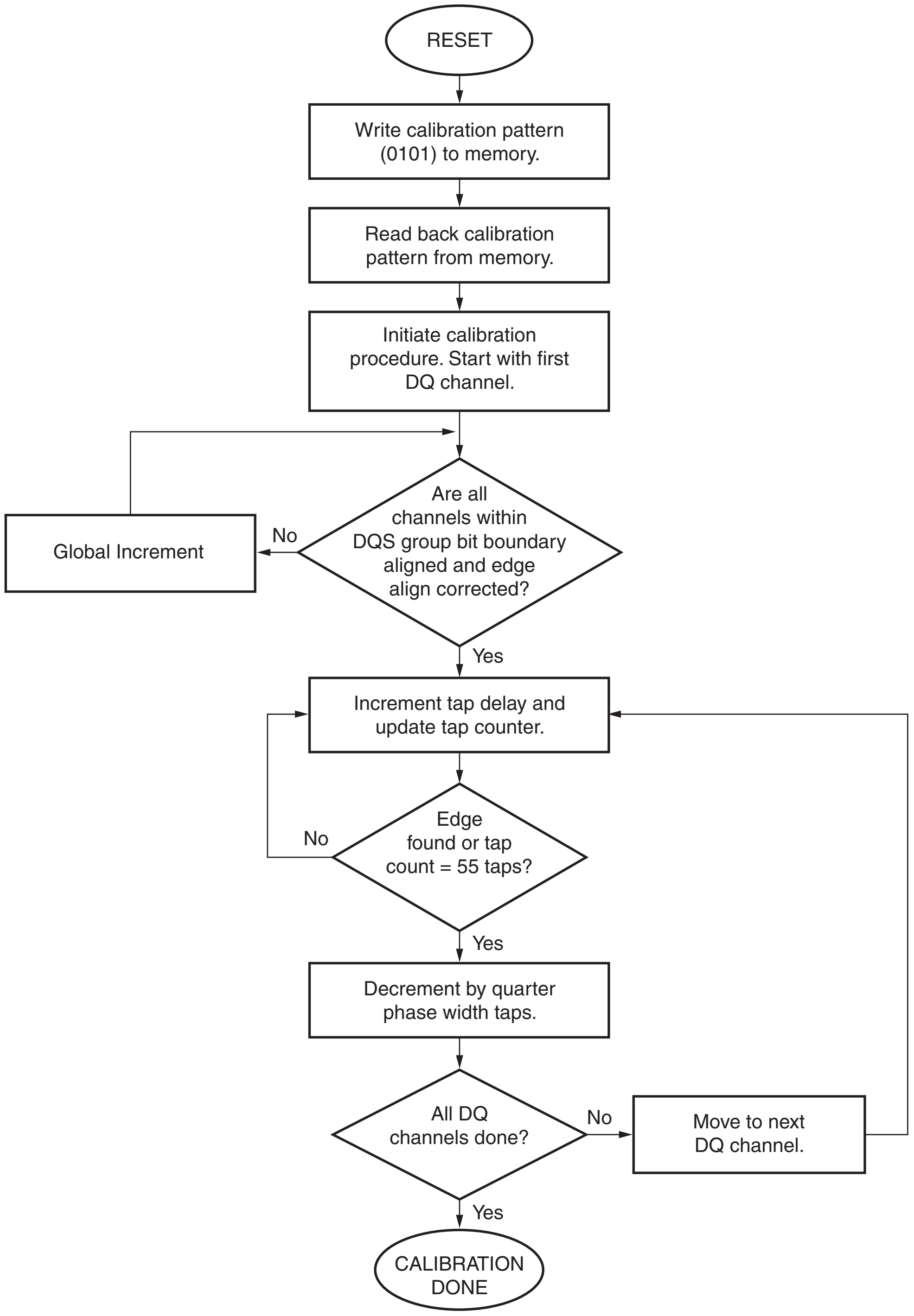
RESET
Write calibration pattern (0101) to memory.
Read back calibration pattern from memory.
Initiate calibration procedure. Start with first DQ channel.
Are all channels within DQS group bit boundary aligned and edge align corrected?
No
Global Increment
Yes
Increment tap delay and update tap counter.
Edge found or tap count = 55 taps?
No
Yes
Decrement by quarter phase width taps.
All DQ channels done?
No
Move to next DQ channel.
Yes
CALIBRATION DONE
x701_04_011907

Figure 4: **State Transition Diagram of the Calibration Algorithm Procedure**

Calibration Logic Implementation

Figure 5 shows a block diagram of the implementation of the calibration logic.

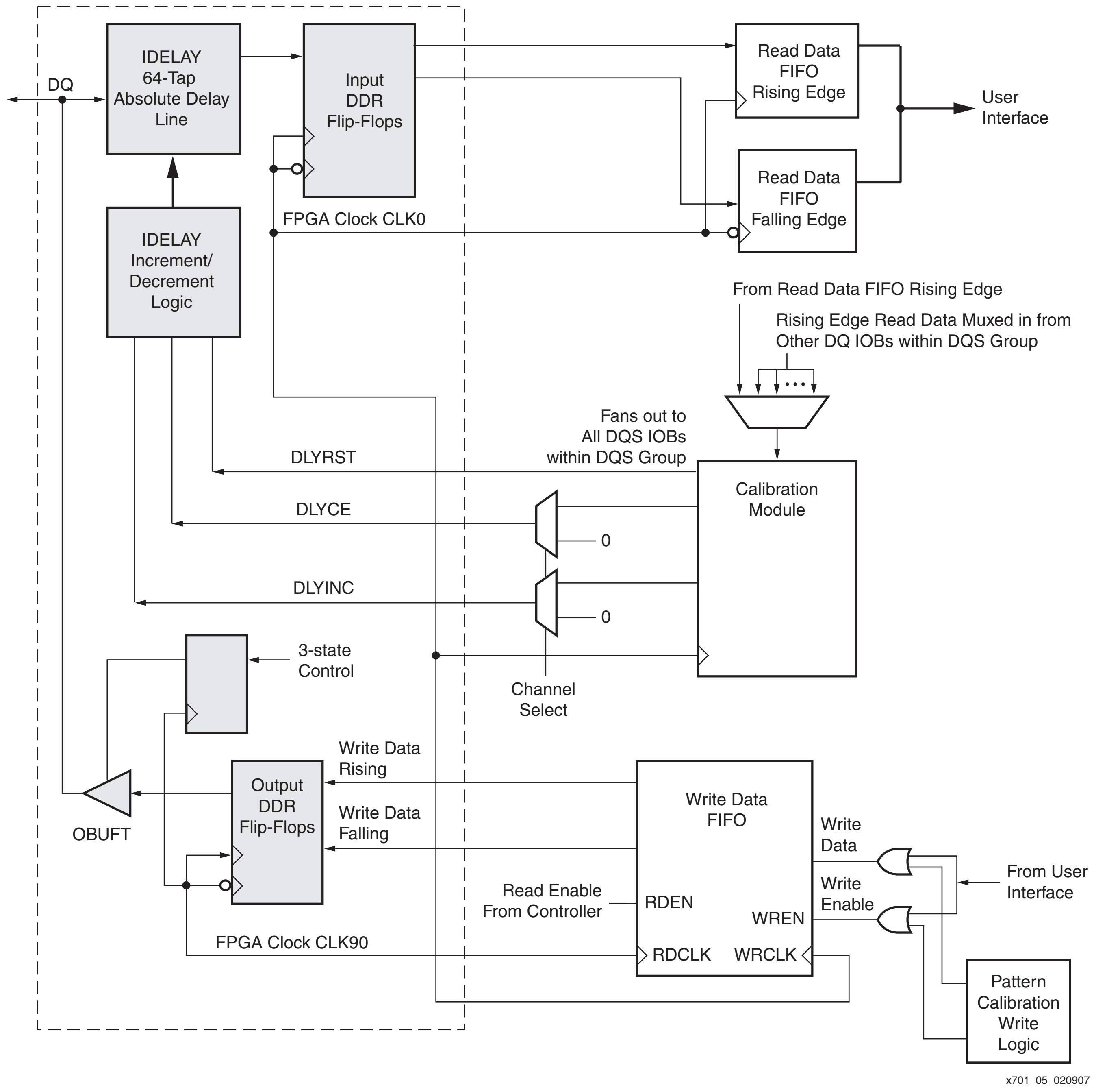

Figure 5: **Calibration Logic Block Diagram**

As shown in Figure 5, the key feature used to implement the calibration procedure is the 64-tap delay line encapsulated in the IDELAY primitive. Each tap in the IDELAY primitive is calibrated to 75 ps and is independent of process, temperature, and voltage. The user can program the amount of delay through the IOB in increments of 75 ps by either incrementing or decrementing the tap delay line. This incrementing and decrementing action is achieved by manipulating the IDELAY primitive control signals, DLYCE and DLYINC. For more information on how to use the IDELAY primitive, see UG070: *Virtex-4 User Guide*.

Immediately after the design comes out of reset and before the calibration procedure begins, the pattern-calibration write logic writes the calibration pattern to the write data FIFO for use later on in the calibration sequence. The following kinds of patterns are written into the FIFO:

- Bit time calibration pattern (…`0101`…)

  The bit time calibration pattern is used to shift the position of the read data on each data line (DQ) such that the FPGA clock edge is centered in the DQ data valid window for maximum margin.

- Read data FIFO write enable calibration pattern (…`A596`…)

  The read data FIFO write-enable calibration calibrates the write-enable signal to the read data FIFOs (rise data and fall data) so that the data that comes out of the read data FIFO is in the correct order.

Some time before calibration begins, the controller issues a write to the memory and writes both calibration patterns to their respective memory addresses. Then the controller issues a continuous read to the memory address that stores the calibration pattern.

The result is that before calibration begins, a continuously oscillating signal is present on all the DQ lines. Also the IDELAY is initialized to a value equal to a quarter clock period of the operating bit time. For example, if the operating frequency is 230 MHz, the IDELAY is initialized to 14 taps (4348 ps bit time / 4).

The calibration procedure is implemented in a time-shared fashion, in which all DQ lines within a DQS group share one calibration module, as illustrated in Figure 6. For example, if the memory used is a 64-bit x8 memory, one calibration module is instantiated for every 8 DQ lines in a DQS group. Since the memory used is a 64-bit-wide memory, there are 8 calibration modules in all.

The calibration procedure begins with the first DQ line in the group. When the procedure completes for that DQ line, the calibration module repeats the calibration procedure for the next DQ line until all DQ lines are calibrated.

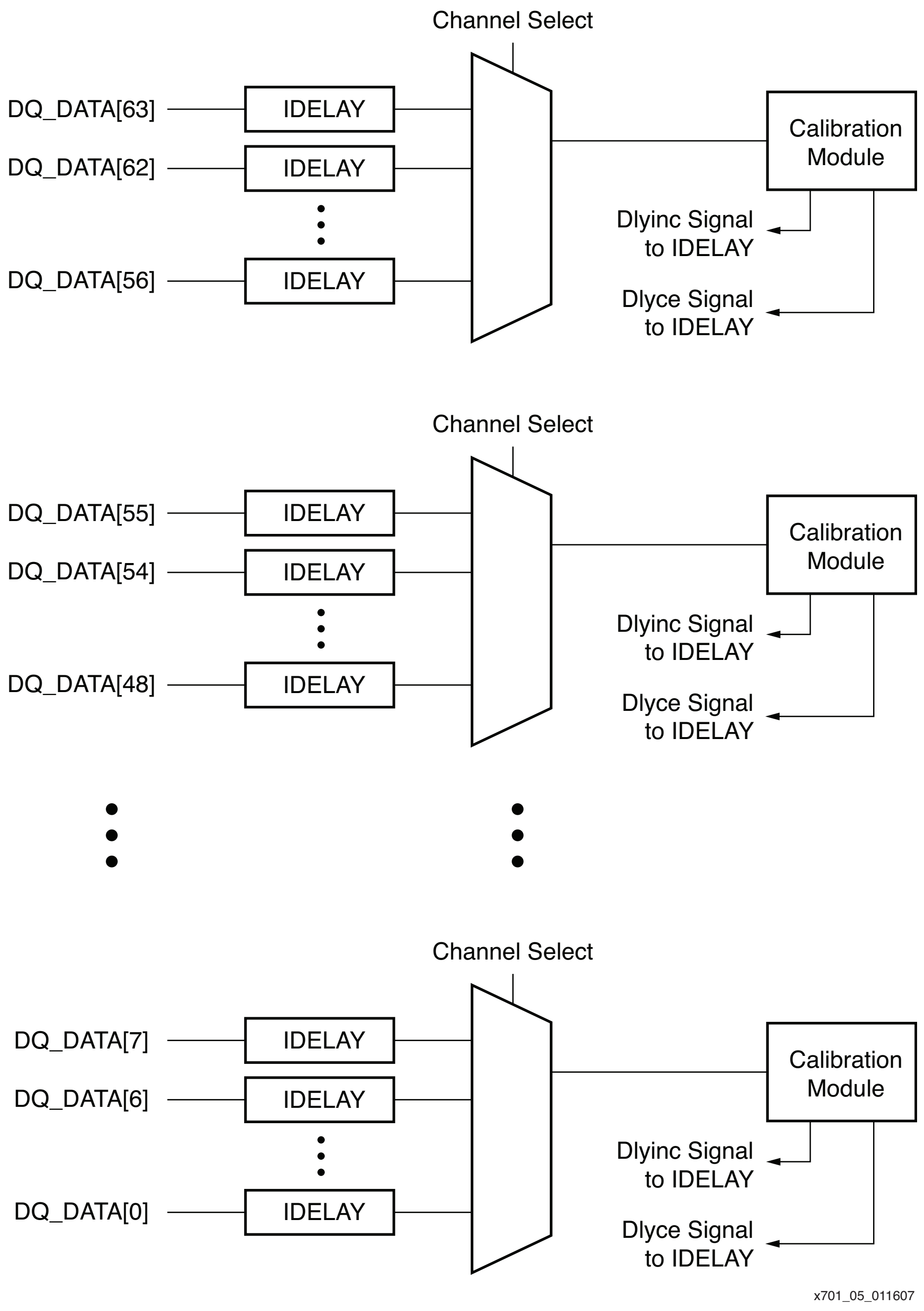

Figure 6: **Time-Sharing of Calibration Modules within DQS Groups**

Data Capture and Recapture

The delayed data is captured in the input DDR flip-flops using the internal FPGA clock as shown in Figure 5, page 6. The output of these flip-flops are then stored in two FIFOs – one for rising edge data and the other for falling edge data. These FIFOs are implemented using LUT RAMs. The write enable for these FIFOs is provided by a read enable signal generated by the controller and aligned to the captured read data based on pattern calibration.

The DDR2 SDRAM devices do not provide a read valid or read enable signal along with read data. Therefore, the controller generates this read enable signal based on the CAS latency and the burst length.

The read enable signal must be asserted during the read preamble and deasserted after the last rising edge of the strobe. The read enable signal must also be aligned to the captured read data at the output of the IDDR flip-flops. For read enable alignment, a known pattern is written to memory after data alignment to the FPGA clock is complete. The known pattern is then read back, and the read enable signal is delayed using shift registers until it is aligned with the captured read data. A read enable signal is generated per byte of data.

Figure 7 shows the timing diagram for data capture and transfer to FIFOs.

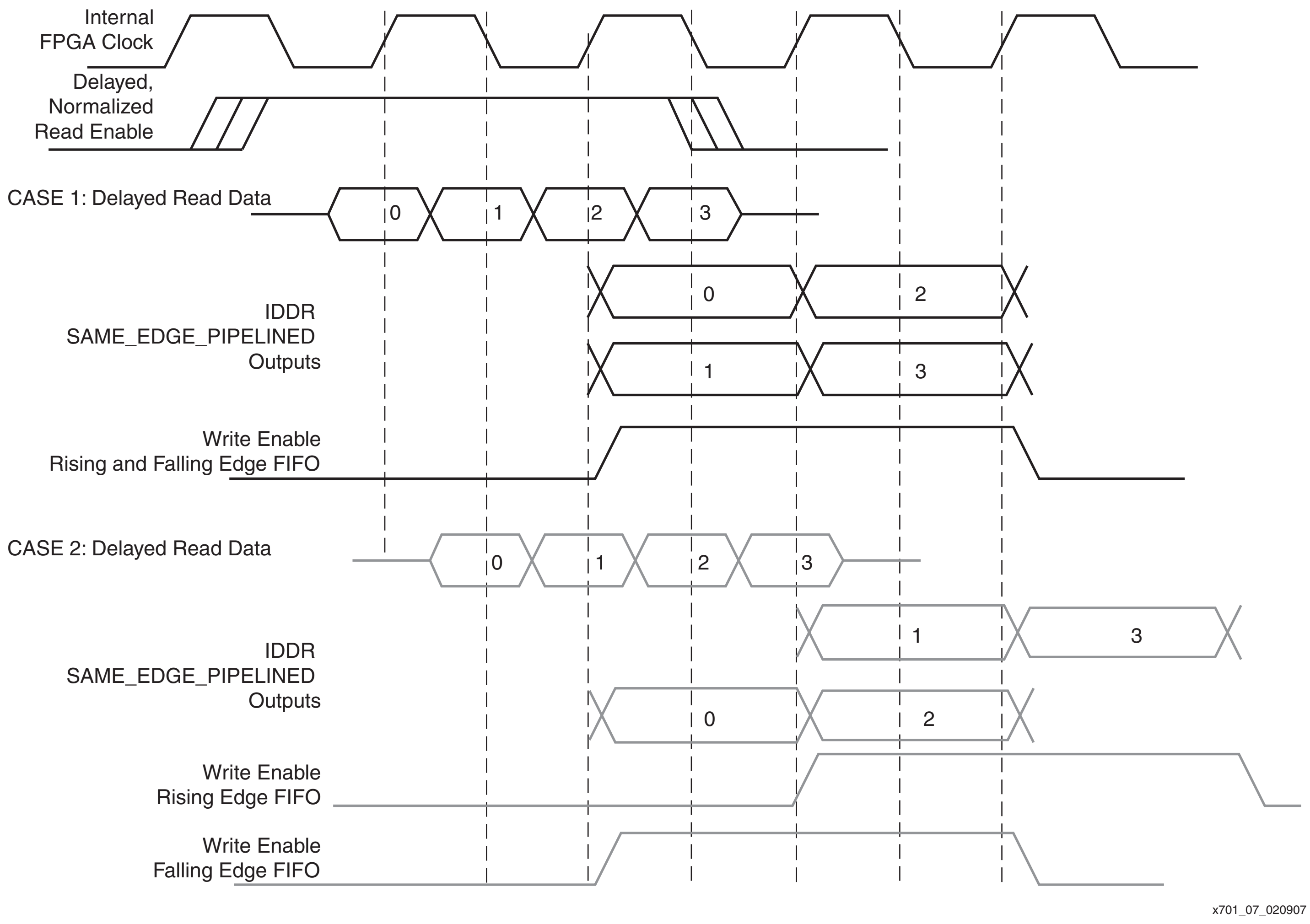
Internal FPGA Clock
Delayed, Normalized Read Enable
CASE 1: Delayed Read Data
0
1
2
3
IDDR SAME_EDGE_PIPELINED Outputs
0
2
1
3
Write Enable Rising and Falling Edge FIFO
CASE 2: Delayed Read Data
0
1
2
3
IDDR SAME_EDGE_PIPELINED Outputs
1
3
0
2
Write Enable Rising Edge FIFO
Write Enable Falling Edge FIFO
x701_07_020907

Figure 7: **Data Capture and Transfer to FIFOs**

Read Timing Analysis

This section presents the read timing analysis of the direct-clocking technique. Read data is captured directly in the FPGA clock domain. As a result, the memory parameter used for the data valid window analysis is the access time (T_{AC}).

For this timing analysis, the external memory parameters included are the following:

- T_{AC} - Access time of read data (DQ) with respect to clock forwarded to memory by FPGA
- T_{DCD} - DCM output duty cycle distortion

Read data (DQ) is captured using the FPGA clock and not the memory clock/strobe (DQS). As a result, T_{AC} (access time of data with respect to clock) is included for this analysis. The DQS-to-DQ memory parameters such as T_{DQSQ} and T_{QHS} are not included in this analysis because T_{AC} overrides them.

T_{AC} is a general parameter that includes process, voltage, and temperature variations. The calibration used in this design eliminates the variation due to process. The effects of process cannot be separated from the effect of voltage and temperature variations from T_{AC}. Therefore, the entire value of T_{AC} is used in the worst-case timing analysis. For more information, contact the appropriate memory vendor.

FPGA parameters considered for this timing analysis are as follows:

- T_{SAMP} - Sampling window specified in the Virtex-4 source synchronous data sheet. For details, see DS302: *Virtex-4 Data Sheet.*
- $T_{PACKAGE\_SKEW}$ - Package skew for a particular device/package.
- $T_{PCB\_LAYOUT\_SKEW}$ - Skew associated with trace length differences between DQ lines on PCB.
- $T_{IDELAYPAT\_JIT}$ - Pattern jitter per IDELAY tap specified in the *Virtex-4 Data Sheet*. $T_{IDELAYPAT\_JIT}$ is pattern-dependent jitter that arises from a random data pattern that traverses through the IDELAY. Pattern jitter is 12 ps.
- $T_{CLOCK\_TREE\text{-}SKEW}$ - Skew on the global clock tree for IOB flip-flops closely placed within a bank.

In a typical case, the maximum number of taps used in the calibration is a half-clock period. However, in a worst-case timing analysis, the corner conditions described in the previous sections must be taken into account. The worst-case conditions must include the maximum skew between two DQ lines within a DQS group given by the specifications, the two extra taps to correct for edge alignment conditions, and any additional skew that arises from package and PCB layout skew. In summary, the worst-case number of taps used in the design is given by the following formula:

$$N_{MAX\_TAPS} = \{(T_{HALF\_CLOCK} + T_{DQSQ} + T_{PACKAGE\_SKEW} + T_{PCB\_LAYOUT\_SKEW}) / T_{IDELAYRESOLUTION}\} + N_{EDGE\_ALIGN\_CORRECT}$$

where:

- T_{HALF_CLOCK} - Half-clock period
- T_{DQSQ} - Maximum skew between DQ lines in a DQS group as defined by DDR2 specifications
- $T_{PACKAGE\_SKEW}$ - Package skew for a particular device/package
- $T_{PCB\_LAYOUT\_SKEW}$ - Skew associated with trace length differences between DQ lines on a PCB
- $T_{IDELAYRESOLUTION}$ - Resolution of a tap in the IDELAY
- $N_{EDGE\_ALIGN\_CORRECT}$ - Number of taps incremented to correct for edge alignment

In the read timing analysis listed in Table 1, for an operating frequency of 230 MHz, the timing values are as follows:

- T_{HALF_CLOCK} is 2173 ps.
- T_{DQSQ} is 300 ps (according to the DDR2 specifications).
- $T_{PACKAGE\_SKEW}$ and $T_{PCB\_LAYOUT\_SKEW}$ are both assumed to be 50 ps.
- $T_{IDELAYRESOLUTION}$ is 75 ps.
- $N_{EDGE\_ALIGN\_CORRECT}$ is 2 taps.

Substituting these timing values into the N_{MAX_TAPS} formula, the maximum number of taps used in this specific case is 36 taps.

Table 1 lists and describes the uncertainty parameters for a read timing analysis at 230 MHz for a -11 device. All parameters are specified in picoseconds. T_{DATA_PERIOD} is half the clock period, minus T_{DCD}. The difference between T_{DATA_PERIOD} and the sum of uncertainties is the valid data window (92 ps). The worst-case timing analysis shows 92 ps of margin at 230 MHz when using a -11 Virtex-4 device.

Table 1: **Read Timing Analysis at 230 MHz for DDR2 Interface in -11 Device**

Uncertainty Parameters	Value (ps)	Description
T_{CLOCK}	4347	Clock period.
T_{DCD}	150	DCM output duty cycle distortion is subtracted from clock phase (equal to half a clock period) to determine T_{DATA_PERIOD}.
T_{DATA_PERIOD}	2024	Data period is half the clock period, with duty cycle distortion subtracted from it.
T_{AC}	1000	Data output access time specified by memory vendor.
T_{SAMP}	500	Sampling window specified in the Virtex-4 source synchronous data sheet.
$T_{IDELAYPAT\_JIT}$	432	At 230 MHz, the total number of taps in the worst case is 36 taps. (See the formula in "Read Timing Analysis".) Therefore, the worst case IDELAY pattern jitter is 36 taps x 12 ps per tap = 432 ps.
$T_{CLOCK\_TREE\_SKEW}$	0	Clock tree skew is eliminated from the worst-case timing analysis because the calibration procedure is performed on a per-bit basis.
$T_{PACKAGE\_SKEW}$	0	Package skew is eliminated from the worst case timing analysis because the calibration procedure is performed on a per-bit basis. However, package skew affects the maximum number of taps used. As a result, package skew is included in the calculations of TIDELAYPAT_JIT. (See the formula in "Read Timing Analysis.")

Table 1: **Read Timing Analysis at 230 MHz for DDR2 Interface in -11 Device *(Continued)***

Uncertainty Parameters	Value (ps)	Description
$T_{PCB\_LAYOUT\_SKEW}$	0	PCB layout skew is eliminated from the worst-case timing analysis because the calibration procedure is performed on a per-bit basis. However, PCB layout skew affects the maximum number of taps used. As a result, layout skew is included in the calculations of TIDELAYPAT_JIT. (See the formula in "Read Timing Analysis.")
Uncertainties	1932	–
Window	92	–

Reference Design

The reference design for the Direct Clocking Data Capture Technique is integrated with the Memory Interface Generator (MIG) tool. This tool has been integrated with the Xilinx CORE Generator^{TM} software. For the latest version of the design, download the IP Update on the Xilinx website at the following URL:

http://www.xilinx.com/xlnx/xil_sw_updates_home.jsp

Conclusion

The Virtex-4 I/O architecture enhances the implementation of source-synchronous memory interface controller designs. The architectural features used in the design include the following:

- IDELAY block – Continuously calibrated delay elements with small tap resolution.
- FIFO16 primitive – Block RAM used as FIFO with no additional CLB resources required for status flag generation.
- Global clocking – High-speed differential global clocking resources provide a better duty cycle. The number of global clock resources required in a design is reduced as a result of differential clocking.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/09/04	1.0	Initial Xilinx release.
11/01/04	1.1	Revised description under "Data Capture and Recapture" section. Revised Figure 5. Reference design is updated on web.
07/11/05	1.2	Revised Table 3, Figure 5, and Figure 7. Revised "Reference Design" links. Added new Table 1.
09/13/05	1.3	Updated "Read Timing Analysis" and "Reference Design" sections.

Date	Version	Revision
10/02/06	1.4	Updated "Strobe Edge Detection Implementation" "Data Capture and Recapture" and Table 3. Removed Figure 7.
03/12/07	2.0	• Revised title of document. • Revised "Summary." and "Introduction." • Deleted "Strobe Edge Detection" section. • Added new section, "Calibration Procedure." • Deleted "Strobe Edge Detection Implementation" section. • Added new section, "Calibration Logic Implementation." • Revised "Data Capture and Recapture" section. • In "Read Timing Analysis" section, revised text and table. • Revised "Conclusion" section.