



XAPP700 (v1.2.1) April 5, 2007

Dynamic Phase Alignment for Networking Applications

Author: Tze Yi Yeoh

Summary

This application note describes a dynamic phase alignment (DPA) application for networking interfaces in a Virtex™-4 device. The reference design performs bit alignment, word alignment, and real-time window monitoring.

Introduction

Data recovery and bus deskew is essential in many source-synchronous networking interfaces. Data may arrive at the FPGA with channel-to-channel skew due to layout constraints resulting in trace length differences. To deskew the channels and properly align the bus in the proper word boundary, networking protocols such as SPI 4.2 require the transmitter to send a training pattern to the receiver during initialization.

Using the Virtex-4 SelectIO™ logic resources, a dynamic phase alignment module is easily implemented for the receiver in the FPGA to effectively remove skew and position the forwarded clock in the center of the data eye with maximum margin, as illustrated in Figure 1. This module also keeps the clock centered to the data eye by continuous monitoring and adjustment.

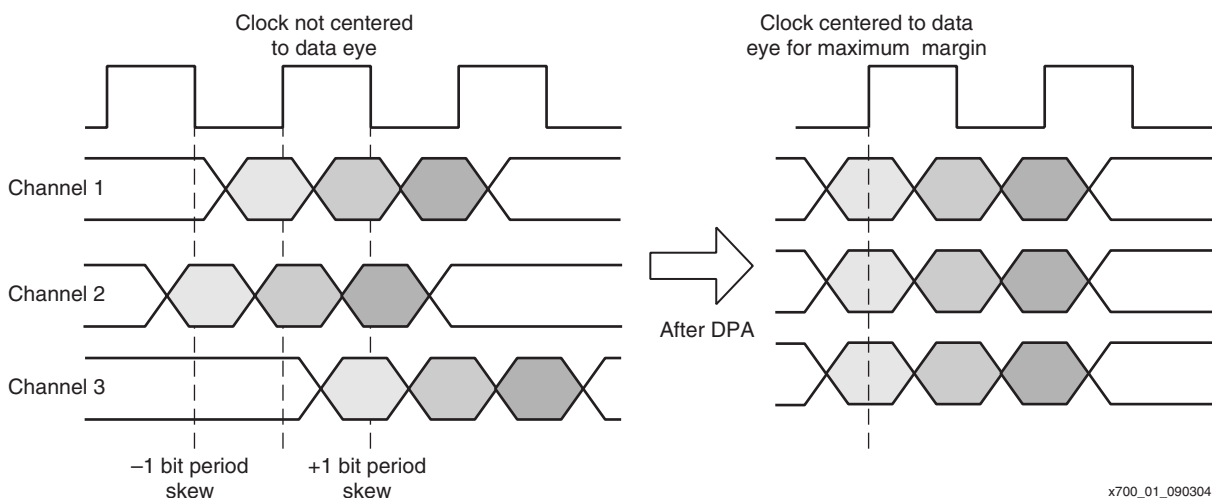


Figure 1: Effects of Dynamic Phase Alignment

Reference Design Details

The reference design uses the SPI 4.2 training protocol to illustrate a method of dynamic phase alignment design. This 20-bit training pattern has 10 zeros and 10 ones (0000_0000_0011_1111_1111) and is sent repeatedly by the transmitter on every channel as long as it is in training mode. The target interface is a source-synchronous bus interface with 16 LVDS data channels and a forwarded clock. The data is deserialized internally by a factor of 1:4.

This is *not* the design used in the Xilinx SPI4.2 IP core.

© 2004–2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Table 1 lists the DPA reference design performance, resource utilization, and implementation requirements.

Table 1: DPA Reference Design

| | |
|---|---|
| Device used for implementation | XC4VLX25 |
| Resource utilization (per receiver channel) | 324 slices One each: BUFG, BUFIO, BUFR 16 ISERDES |
| HDL used | VHDL and Verilog |
| Synthesis Tool | XST |
| Implementation tool | Xilinx ISE 6.3i |
| Simulation tool | Modelsim 5.8 |

Design Overview

The reference design performs three functions:

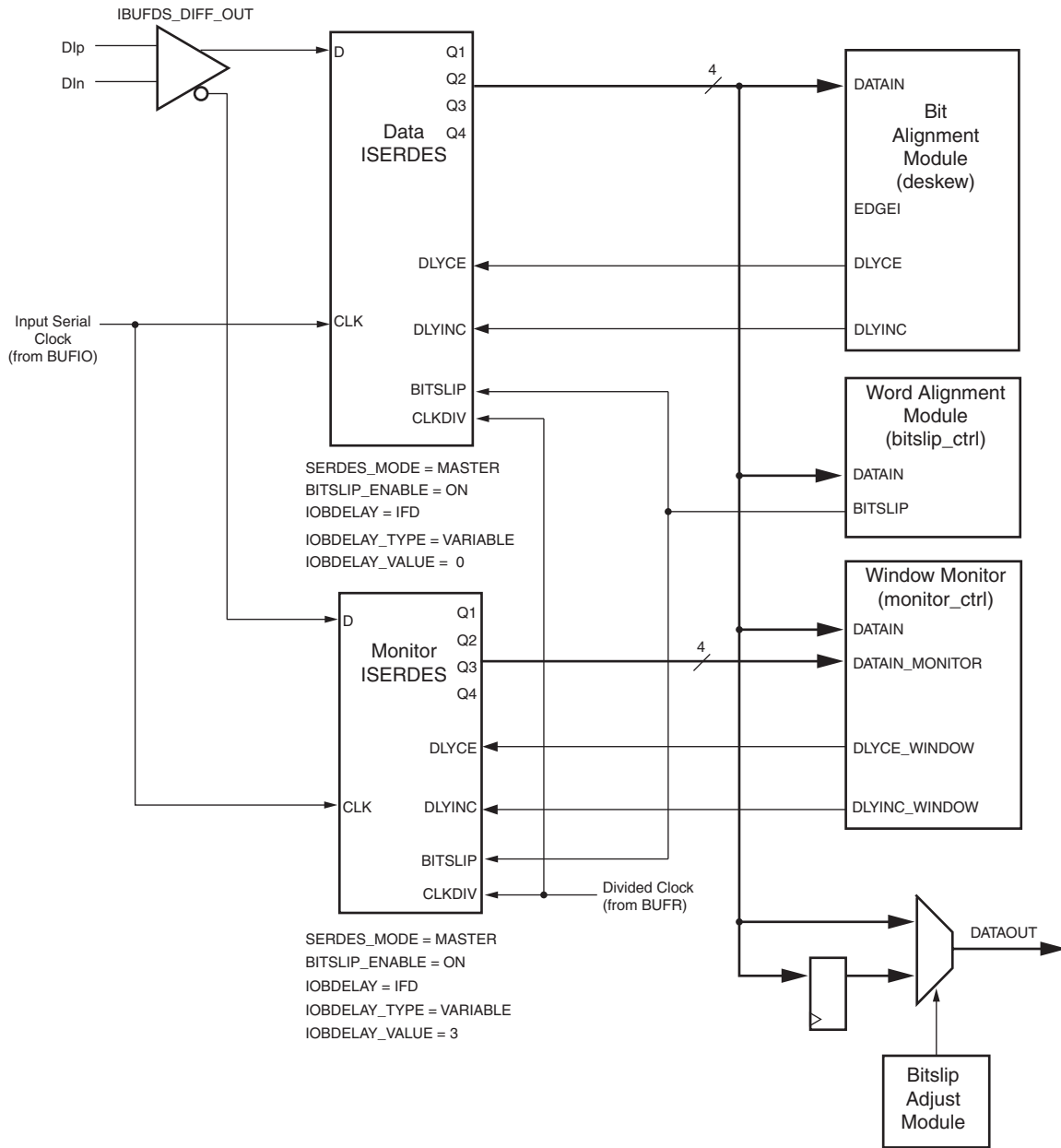
- Bit alignment
- Word alignment
- Real-time window monitoring

The training function comprises the bit alignment and word alignment functions and occurs during the initialization phase. Bit alignment corrects for data skew of less than one bit period by positioning the clock edge at the center of the data eye. Word alignment corrects for data skew greater than one bit period by aligning the incoming pattern to the pre-specified training pattern. After the training phase is complete, the training module goes into monitor mode, continually scanning the incoming data stream to ensure the clock is always positioned at the center of the data eye. The training algorithm uses the Virtex-4 ChipSync™ features including:

- Dedicated serial-to-parallel converter
- Bit-slip sub-module
- Programmable 64-tap delay line

The reference design also uses the Virtex-4 regional clocking resources.

An overview block diagram of the training function is shown in Figure 2. Only one channel is shown. Some of the interconnection are simplified to avoid redundancy.



x700_02_120104

Figure 2: Block Diagram for One Channel

Each ISERDES module can deserialize data up to six bits wide. In this design, data is deserialized to a 4-bit parallel word. Since LVDS is used as the signaling standard, two IOBs and therefore two ISERDES modules are available per channel. One ISERDES is configured as the data ISERDES and the other as the monitor ISERDES. Both ISERDES modules are configured in master mode. The IBUFDS_DIFF_OUT primitive connects the input data stream to both ISERDES modules. The output of the IBUFDS_DIFF_OUT is differential. The negative output is connected to the input of the monitor ISERDES.

The BITSLIP_ENABLE attribute is set to ON for both ISERDES modules to use the Bitslip submodule. The state machines in the fabric implement the bit alignment, word alignment, and window monitoring algorithm, and are time-shared across all 16 channels to conserve FPGA resources (Figure 3).

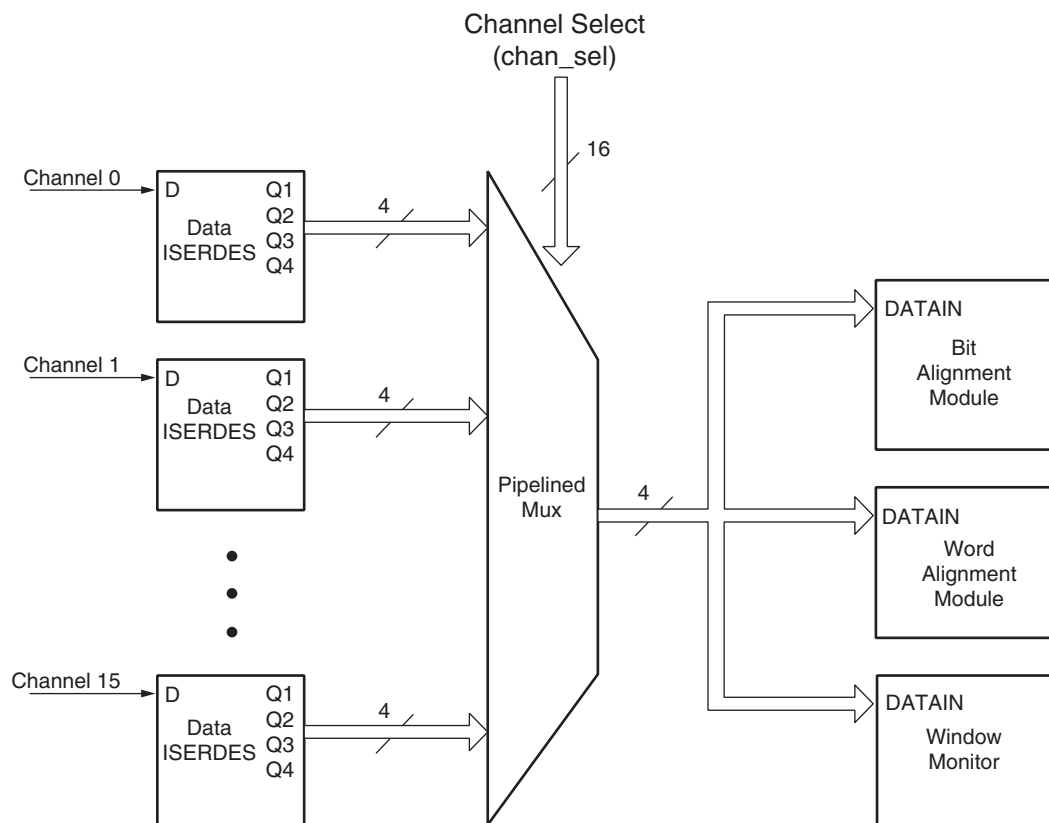


Diagram shown is for Data ISERDES only.
The configuration for the monitor ISERDES is identical.

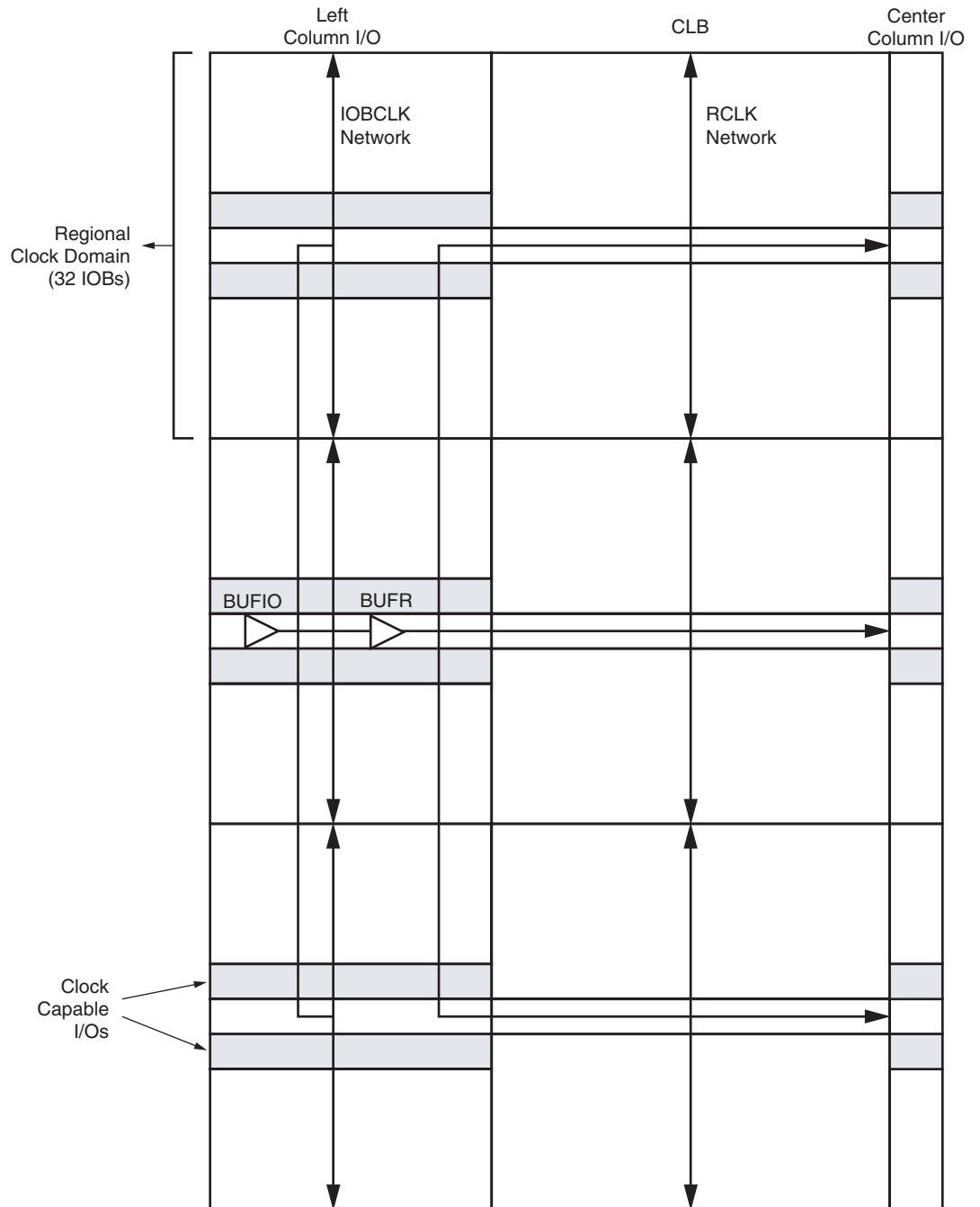
x700_03_120104

Figure 3: Timesharing of Training Control Modules Across 16 Channels

The value of the tap delay line of the data ISERDES module (IOBDELAY_VALUE attribute) is initialized at zero. The tap delay line of the monitor ISERDES is initialized to a value three taps greater than the data ISERDES due to the requirements of the window monitoring feature (see "Window Monitoring"). The instantiation of the IDELAYCTRL module is not shown in Figure 2. The IDELAYCTRL module provides an absolute reference voltage to the tap delay line. This sets the tap delay to an absolute value independent of process, voltage, or temperature. Since the tap delay line is used in VARIABLE mode, one IDELAYCTRL module per regional clock domain using the tap delay line must be instantiated. A 200 MHz reference clock (REFCLK) must be supplied to the IDELAYCTRL module through a global clock buffer. The IDELAYCTRL RDY signal indicates a valid REFCLK signal. Only one reference clock is needed per Virtex-4 device. Initially, one instance of an IDELAYCTRL module can be instantiated without a LOC constraint. The MAP tool replicates the first instantiation for every IDELAYCTRL location on the Virtex-4 device. If the RDY signal is used, the MAP tool generates an AND gate and connects all the RDY signals to the AND gate. Instantiating an IDELAYCTRL module without a LOC constraint consumes one global clock line per regional clock domain. There are eight global clock lines per regional clock domain. Routing resources for the RST and RDY signals are also consumed. To free up global clock resources and routing resources after the initial place and route, go back and instantiate the correct number of IDELAYCTRL modules with LOC constraints. Further details on the IDELAYCTRL module are discussed in Chapter 7 of the *Virtex-4 User Guide*.

Placement of Clock and Data IOBs

Figure 4 shows three regional clock domains on the left side of a Virtex-4 device.



x700_02_120204

Figure 4: Regional Clock Domains

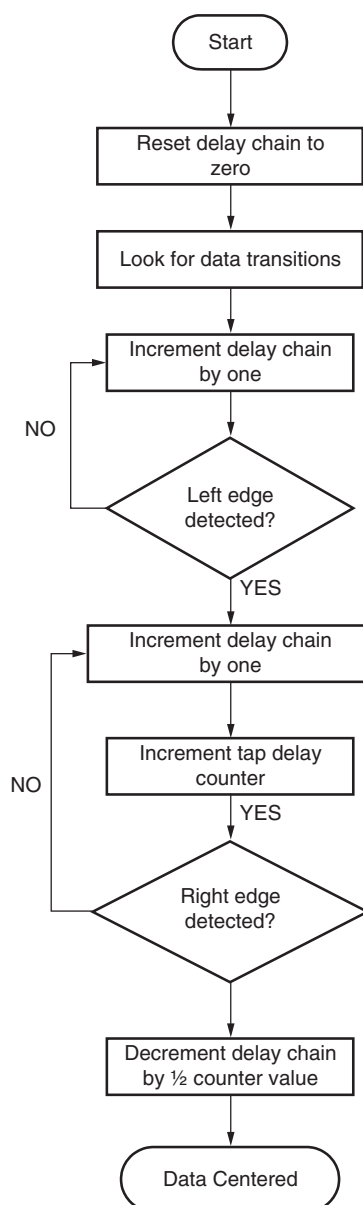
In Figure 4, the forwarded clock is routed into a clock capable I/O location. The design uses the regional clock buffer resources BUFIO and BUFR. BUFIO, the high-speed clock buffer, distributes the input clock to the CLK input of the ISERDES on the IOCLK network. The IOCLK network is a full differential clock line designed to manage very high switching speeds. BUFR, the regional clock buffer, divides the input clock according to the BUFR_DIVIDE attribute and distributes the slow rate clock to the parallel output of the ISERDES and to the rest of the regional clock domain. The BUFR primitive has a built-in clock divider controlled by the BUFR_DIVIDE attribute. In a DDR design with a 1:4 deserialization factor, BUFR_DIVIDE is set to two. The output of BUFR connects to the CLKDIV input of the ISERDES through the RCLK

network. Both the BUFIO and BUFR primitives can span up to three regional clock domains; its own region plus one region directly above and below it. Details on the operation of the BUFIO and BUFR primitives are discussed in the *Virtex-4 User Guide, Chapter 1: Clocking Resources*.

The reference design uses a total of 17 LVDS I/O pairs (16 data and one forwarded clock) for a total of 34 IOBs. Since the number of IOBs in one regional clock domain is limited to 32, the implementation tools place the design in two regional clock domains. Due to the multi-region spanning capability of the regional clock resources, only one instance each of BUFIO and BUFR needs to be instantiated. The implementation tools automatically places the IOBs and the clock resources in adjacent regional clock domains.

Bit Alignment

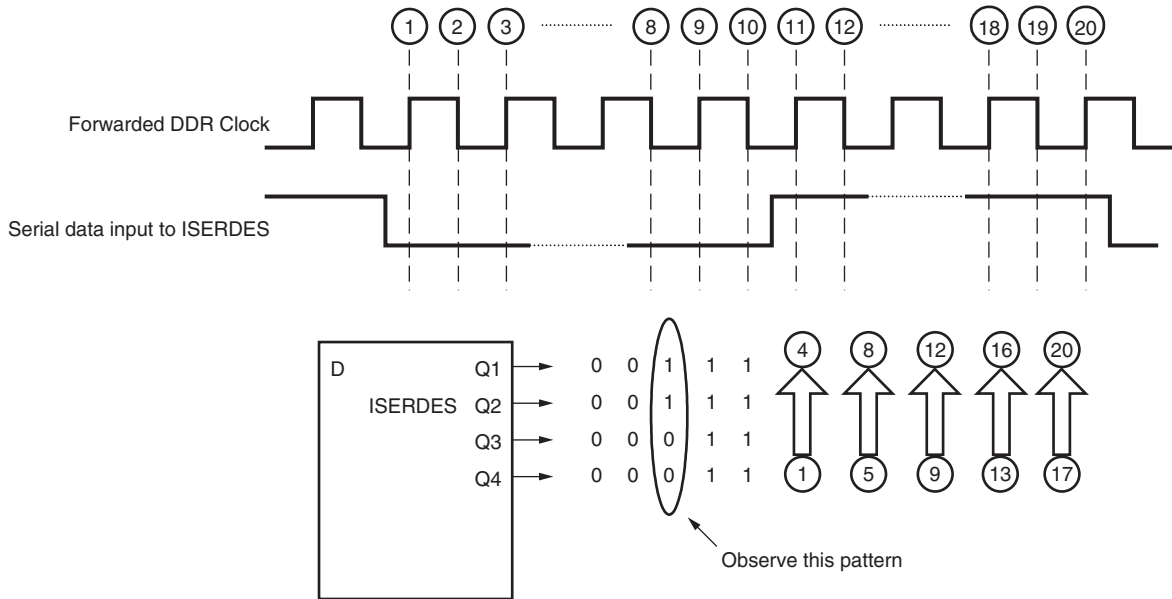
The goal of the bit-alignment procedure is to position the captured clock edge in the center of the data eye to provide maximum margin. The bit-alignment procedure uses the tap delay line feature of the ISERDES. The *Virtex-4 User Guide* (Chapter 8) contains information about using the tap delay line feature. The algorithm delays the data channel with respect to the clock. [Figure 5](#) shows the state transition diagram for the bit-alignment algorithm.



x700_04_120504

Figure 5: Bit Alignment State Transition Diagram

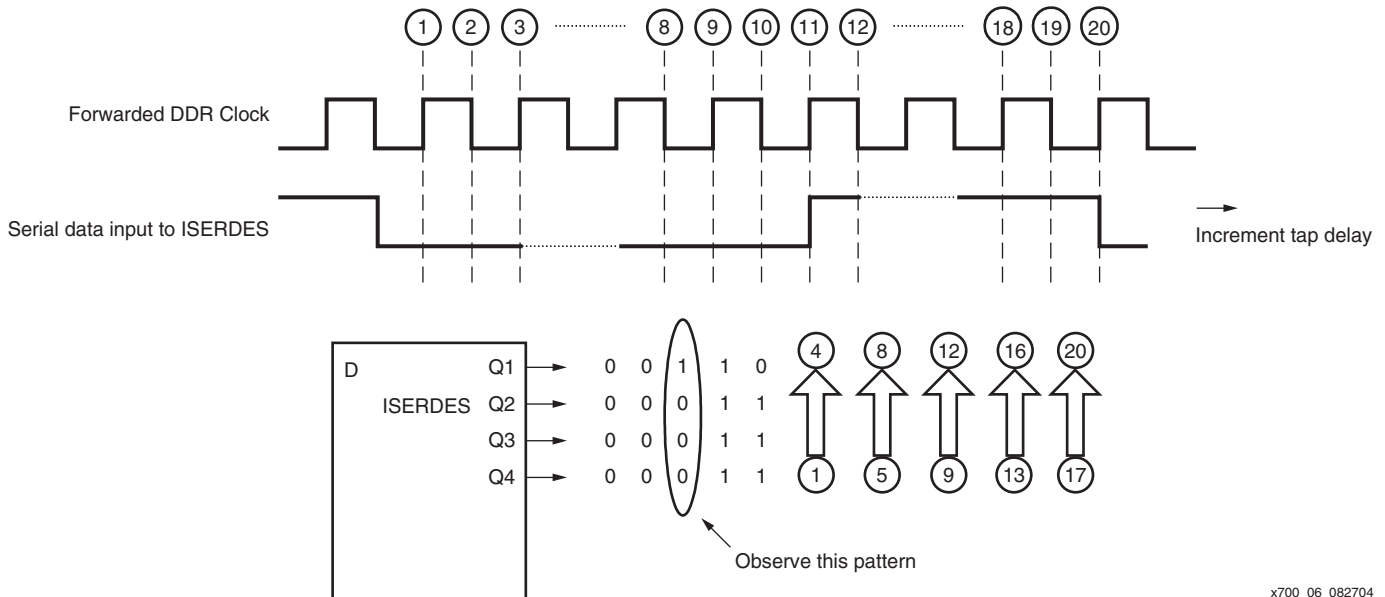
Bit alignment begins by parsing the output of the ISERDES until it finds a word containing data transitions (Figure 6). The data transition detection mechanism is achieved by bit-wise XORing the output of the ISERDES.



x700_04_081304

Figure 6: State of ISERDES Outputs at the Beginning of Bit-Alignment

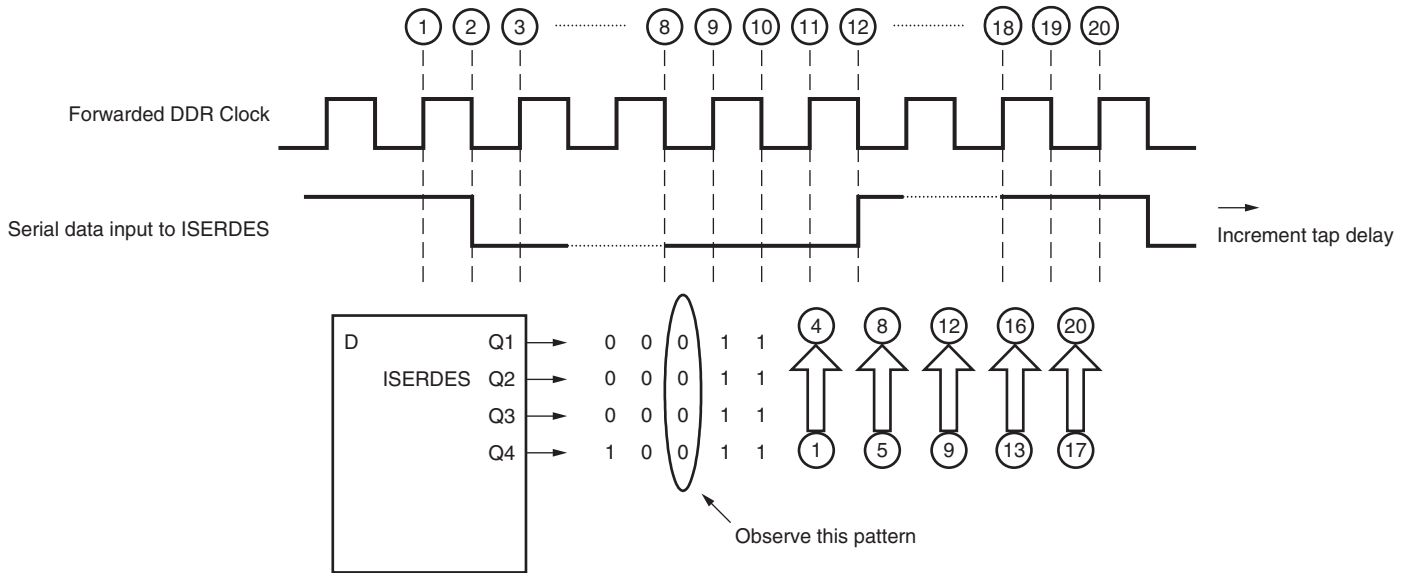
Once a word containing data transitions is detected, the module increments the tap delay of the data channel until the correct edge is detected. This condition is illustrated in Figure 7.



x700_06_082704

Figure 7: First Edge Detected

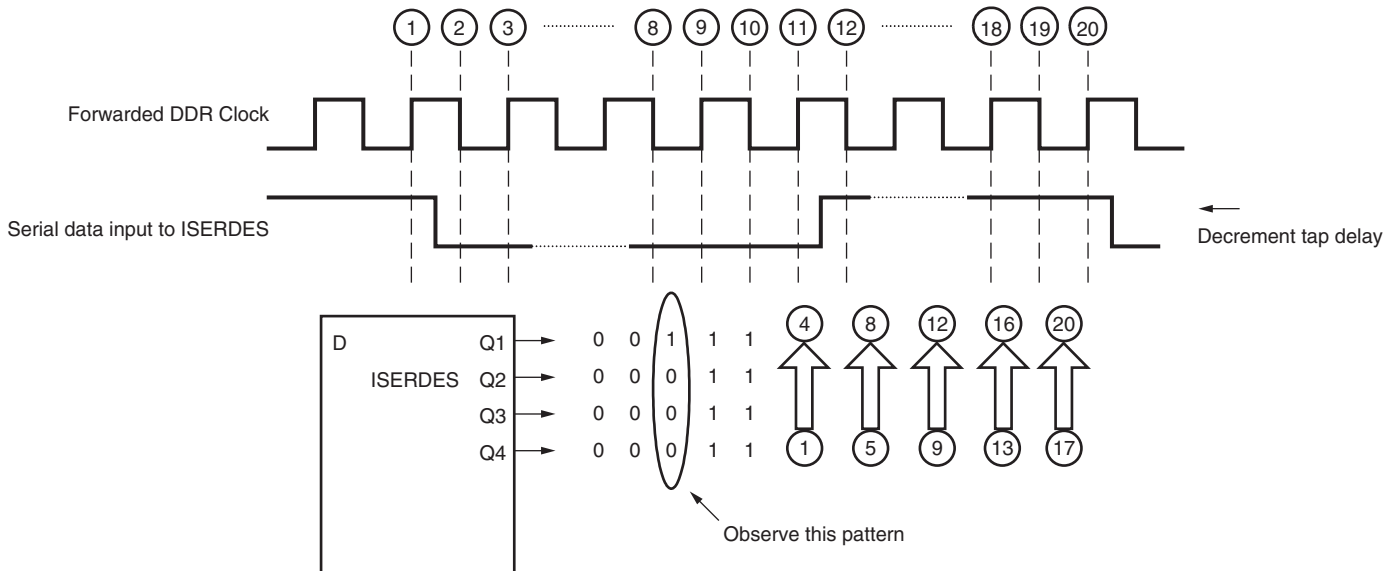
After the first edge is detected, the tap counter starts counting the number of tap increments as the state machine increments the tap delay until the left edge is found. This condition is illustrated in Figure 8.



x700_07_120104

Figure 8: Second Edge Detected

After the second edge is detected, the state machine decrements the tap delay line by half the amount of the tap counter to position the clock edge at the center of the data eye. The final result is shown in Figure 9.



x700_08_120104

Figure 9: Data Centered

With only one bit-alignment state machine, time-shared across all 16 data channels, the bit-alignment procedure is circular starting with Channel 0 and ending with Channel 15. A 16-bit register, called `chan_sel` is used as a scheduler to monitor the enabled channel.

Word Alignment

The word alignment procedure aligns the output pattern from the ISERDES to a specific training pattern. This procedure effectively removes word skew and aligns all channels to a specific word boundary. The word alignment unit primarily uses the Bitslip sub-module of the ISERDES. The Bitslip sub-module matches the output of the serial-to-parallel converter to a specific pattern. It accomplishes this by effectively shifting the output one bit at a time until the pattern is found. The Bitslip sub-module is activated by asserting the BITSLLIP port of the ISERDES for one CLKDIV cycle. In DDR mode, the result of a Bitslip operation is guaranteed to be valid only after two CLKDIV cycles. Therefore, the appropriate number of wait states needs to be accounted for in the word alignment algorithm.

Figure 10 shows the state transition diagram for the word alignment algorithm.

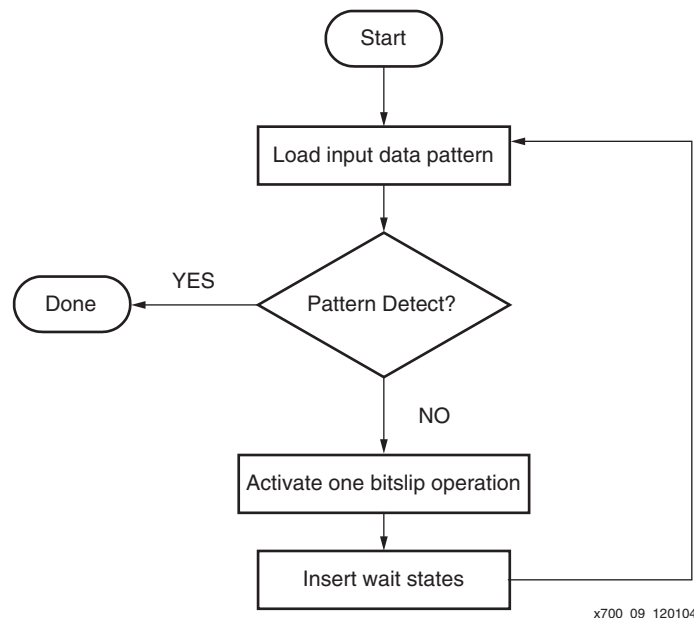


Figure 10: Word Alignment Algorithm State Transition Diagram

The word alignment algorithm matches the output of the ISERDES to a certain pattern. Initially, the training pattern is 0000_0000_0011_1111_1111. The pattern 0011 occurs only once in the training pattern and is chosen as the pattern to match. The algorithm begins by loading in the parallel (4-bit) output of the ISERDES. It checks the word against the pattern it is trying to match (0011). As long as there is not a match with the intended pattern (0011) it asserts the Bitslip sub-module. At the end of the Bitslip algorithm, the output (per channel) of the ISERDES can be delayed by up to one CLKDIV cycle (with respect to each other) depending on the behavior of the Bitslip sub-module. Therefore, at the end of a Bitslip operation, a Bitslip adjust procedure corrects the one CLKDIV cycle delay by adding an additional pipeline stage to the output.

With only one bit-alignment state machine, time-shared across all 16 data channels, the bit-alignment procedure is circular starting with Channel 0 and ending with Channel 15. A 16-bit register, called chan_sel is used as a scheduler to monitor the enabled channel.

Bitslip Adjust Procedure

At the end of the word alignment procedure, the parallel output of the ISERDES can be delayed by up to a maximum of one CLKDIV cycle (with respect to each other). This is due to the behavior of the Bitslip sub-module. A Bitslip adjust procedure is called at the end of the word alignment procedure to correct for this behavior. At the end of the word alignment algorithm, Channel 15 is the last channel to be word aligned. The Bitslip adjust algorithm starts with Channel 15 and notes when the 0011 pattern was last detected. Making this a reference point, the Bitslip adjust circuit scans through the ISERDES outputs of all 16 channels and notes when

the outputs are delayed or advanced with respect to the reference point (Channel 15). At the end of the scan, the circuit determines whether a pipeline stage should be inserted to the data path to correct for any delay insertion by the Bitflip sub-module.

Window Monitoring

After the initialization stage where the training procedure is called, the channels are assumed to remain trained throughout normal operation. However, due to operating conditions (voltage, or temperature), the data valid window can shift. The window monitoring unit continuously monitors the data valid window during normal operation and adjusts the sampling point as necessary to provide maximum margin.

The algorithm uses the spare ISERDES module resulting from using differential I/O. Only one ISERDES module is needed for deserialization. The monitor ISERDES is configured in master mode and the differential in / differential out input buffer (IBUFDS_DIFF_OUT) is used to feed the input serial stream to both ISERDES modules. Due to the differential nature of the buffer outputs, the input serial stream to the monitor ISERDES is inverted. In the window monitoring algorithm, the window monitoring control unit makes the appropriate adjustment when comparing the outputs of the monitor ISERDES with the data ISERDES.

The window monitoring procedure primarily uses the tap delay feature of the ISERDES. The algorithm is set up to delay the data channel with respect to the clock. [Figure 11](#) shows the state transition diagram for the bit alignment algorithm.

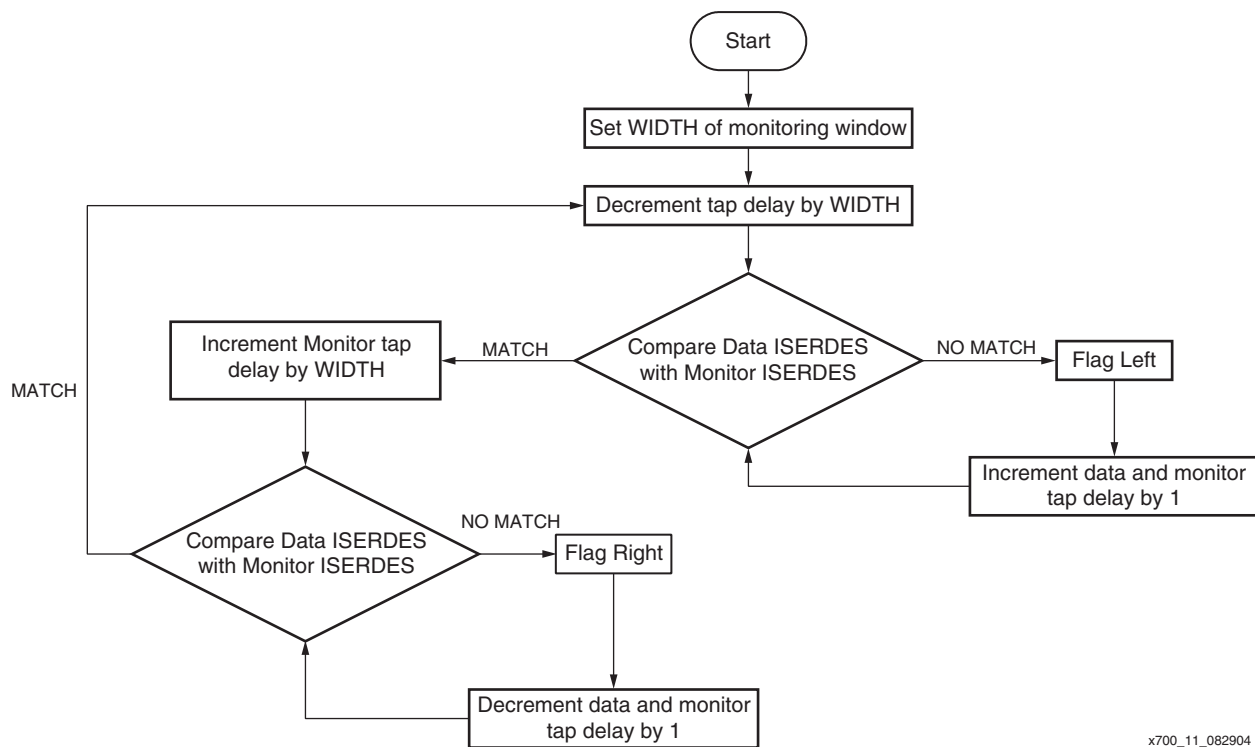


Figure 11: Window Monitoring State Transition Diagram

The window monitoring procedure begins by determining the size of the data valid window to be monitored. Since the tap delay is an absolute value of 78 ps, the width of the data valid window is translated directly into number of taps. At the start of the initialization procedure, the value of the tap delay of the monitor ISERDES is set to a value three taps greater than the data ISERDES. This is due to an arbitrary setting to six of the data valid window width of 1 Gb/s bit time ($6 \times 78 \text{ ps} = 468 \text{ ps}$). Therefore, the monitor ISERDES is initialized to the right edge of the data valid window. The procedure decrements the monitor tap delay until it reaches the left edge of the data valid window. Check to see if the monitor output corresponds to the data output. If it does not, it flags a mismatch and increments the tap delay for both the monitor and

data ISERDES modules. It checks again for a mismatch. It keeps checking until the data matches. Then, the procedure increments the monitor tap delay until it reaches the right edge of the data valid window. It performs the same checks and in the case of a mismatch, decrements the tap delay for BOTH monitor and data ISERDES until a match is found. This procedure repeats indefinitely under normal operation keeping the clock centered on the data window over voltage, and temperature variations.

With only one bit-alignment state machine, time-shared across all 16 data channels, the bit-alignment procedure is circular starting with Channel 0 and ending with Channel 15. A 16-bit register, called `chan_sel` is used as a scheduler to monitor the enabled channel.

Using the Reference Design

Reset Sequence

When resetting the system, reset the IDELAYCTRL module first, allowing the IDELAYCTRL ready signal to pulse Low before activating the system reset. The reset guidelines for IDELAYCTRL must be strictly followed, see the [Virtex-4 User Guide, Chapter 7 "SelectIO Logic Resources"](#). The reset sequence is shown in [Figure 12](#).

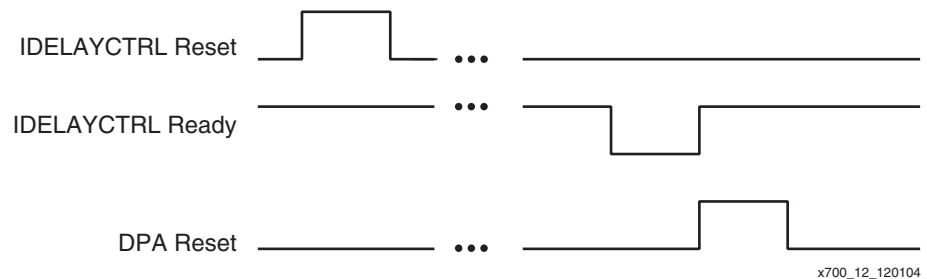


Figure 12: Reset Sequence

Asserting Train Enable Signal

After deasserting the system reset, wait at least 16 CLKDIV cycles before asserting the train enable signal (`train_en`). This allows for data latency through the ISERDES. Valid data must appear on the outputs of the ISERDES before enabling the bit alignment algorithm. The train enable sequence is illustrated in [Figure 13](#).

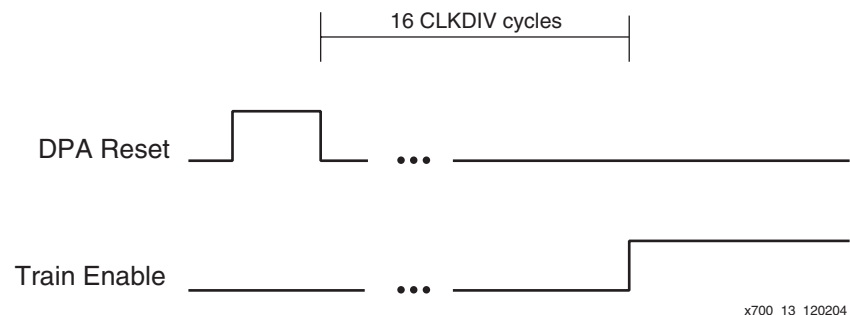


Figure 13: Train Enable Sequence

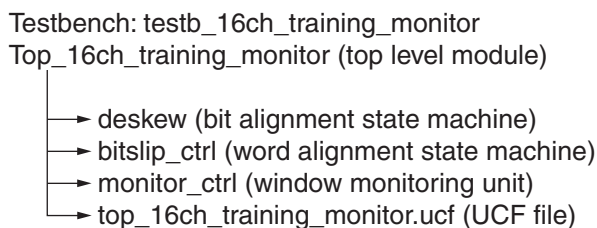
Once train enable is asserted, it should remain High until train enable (`train_done`) is asserted.

Implementation Without LOC Constraints

The reference design can be implemented as-is without LOC constraints. The ISE implementation tools automatically places all the data channels within the multi-region covered by the regional clock buffer resources (BUFIO and BUFR). Also, one instance of IDELAYCTRL is instantiated in the reference design. Since there are no LOC constraints associated with this IDELAYCTRL module, the MAP tool replicates this instantiation throughout the entire device.

The MAP tool also creates an AND gate and ties the RDY signals from all the IDELAYCTRL modules to that AND gate. Further details on the IDELAYCTRL module are discussed in *Chapter 7 of the Virtex-4 User Guide*.

Figure 14 shows the reference design hierarchy.



x700_14_120104

Figure 14: Reference Design without LOC Constraints Hierarchy

Conclusion

This application note describes an implementation of a dynamic phase alignment design for SPI 4.2, a networking protocol with a training pattern. The reference design has bit alignment, word alignment, and real-time window monitoring. It uses the advanced ChipSync I/O features available in Virtex-4 devices including the dedicated serial-deserializer, the Bitslip sub-module, and the variable tap delay.

The reference design is available on the Xilinx web site at:

<http://www.xilinx.com/bvdocs/appnotes/xapp700.zip>

Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|----------|---------|---|
| 09/09/04 | 1.0 | Initial Xilinx release. |
| 12/07/04 | 1.1 | Revisions to match the new reference design file. |
| 07/21/05 | 1.2 | Revisions to correct text. |
| 04/05/07 | 1.2.1 | Deleted Appendix A. |