



XAPP495 (v1.0) December 13, 2010

Implementing a TMDS Video Interface in the Spartan-6 FPGA

Author: Bob Feng

Summary

Transition Minimized Differential Signaling (TMDS) is a standard used for transmitting video data over the Digital Visual Interface (DVI) and High-Definition Multimedia Interface (HDMI). Both interfaces are popular in a wide range of market segments including consumer electronics, audio/video broadcasting, industrial surveillance, and medical imaging systems. They are commonly seen in flat panel TVs, PC monitors, blue-ray players, video camcorders, and medical displays.

This application note describes a set of reference designs able to transmit and receive DVI and HDMI data streams up to 1080 Mb/s using the native TMDS I/O interface featured by Spartan®-6 FPGAs.

Introduction

The DVI and HDMI protocols use TMDS at the physical layer. The TMDS throughput is a function of the serial data rate of the video screen mode being transmitted. This in turn determines the FPGA speed grade that must be used to support this throughput.

After the Spartan-3A family, Xilinx has offered embedded electrically-compliant TMDS I/O allowing implementation of DVI and HDMI interfaces inside the FPGA. The operation theory for this is detailed in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs* [Ref 1]. The data throughput in that application note was maximized at 666 Mb/s in the fastest speed grade.

The Spartan-6 FPGA on the other hand has made significant speed improvements. [Table 1](#) shows the maximum throughput for each speed grade of the Spartan-6 FPGA.

Table 1: Spartan-6 Family TMDS I/O Throughput

Speed Grade	Throughput (Mb/s)
-4	1080
-3	1050
-2	945
-1L	500

Common video screen modes corresponding to these data rates are listed in [Table 2](#).

Table 2: Common Video Screen Modes

Screen Mode	Pixel Rate (MHz)	Serial Data Rate (Mb/s)	Color Depth
VGA ⁽¹⁾ (640x480 @ 60 Hz)	25	250	24b
480p ⁽²⁾ (720x480 @ 60 Hz)	27	270	24b
SVGA ⁽¹⁾ (800x600 @ 60 Hz)	40	400	24b
XGA ⁽¹⁾ (1024x768 @ 60 Hz)	65	650	24b
HD ⁽¹⁾ (1366x768 @ 60 Hz)	85.5	855	24b
WXGA ⁽¹⁾ (1280x800 @ 60 Hz)	71	710	24b

Table 2: Common Video Screen Modes (Cont'd)

Screen Mode	Pixel Rate (MHz)	Serial Data Rate (Mb/s)	Color Depth
HDTV 720p ⁽²⁾ (1280x720 @ 60 Hz)	74.25	742.5	24b
HDTV 1080i ⁽²⁾ (1920x1080 @ 60 Hz interlaced)	74.25	742.5	24b
HD ⁽¹⁾ (1366x768 @ 60 Hz)	85.5	855	24b
SXGA ⁽¹⁾ (1280x1024 @ 60 Hz)	108	1080	24b

Notes:

1. Refer to the Video Electronics Standards Association (VESA) Coordinated Video Timing Generator [\[Ref 2\]](#).
2. Refer to the Consumer Electronics Association's Consumer Video Timing Specification [\[Ref 3\]](#).

The throughput offered by the fastest Spartan-3A device can be achieved by the slowest speed grade of the Spartan-6 device. For the first time, the most popular 720p and 1080i resolution is achievable in most Spartan-6 devices (except the -1L speed grade).

In addition to the speed enhancement, the Spartan-6 FPGA possesses some unique hardened I/O features ideal for implementing video interfaces. Compared to what is described in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs* [\[Ref 1\]](#) for the Spartan-3A FPGA, this application note focuses on leveraging the new I/O features to achieve higher performance with fewer resources.

Logic Construct

One of the major advancements in the Spartan-6 FPGA is its I/O architecture. This is represented by the addition of new advanced I/O logic and clocking resources, namely, IODELAY2, IOSERDES2, and the phase-locked loop (PLL) and I/O clock distribution network. As detailed in *Spartan-6 FPGA SelectIO™ Resources User Guide* [\[Ref 4\]](#), IODELAY2 provides calibrated delay taps for the TMDS video stream to deskew and phase align on a per-pair basis. IOSERDES2 helps with serialization and deserialization of video pixels using dedicated and hardened circuitry. The PLL synthesizes a high-speed bit rate matching clock, and the I/O distribution routes the clock to designated IODELAY2 and IOSERDES2 elements via a dedicated path with the finest resolution. This not only gives a higher performance improvement over the Spartan-3A family but allows the actual DVI/HDMI transmitter and receiver to be built in a much easier manner.

Transmitter Design

For each of three data pairs in a DVI or HDMI link, the transmitter design is logically divided into two parts: TMDS encoding and 10-bit parallel-to-serial conversion. [Figure 1](#) illustrates the topology of the DVI/HDMI transmitter design in the Spartan-6 FPGA.

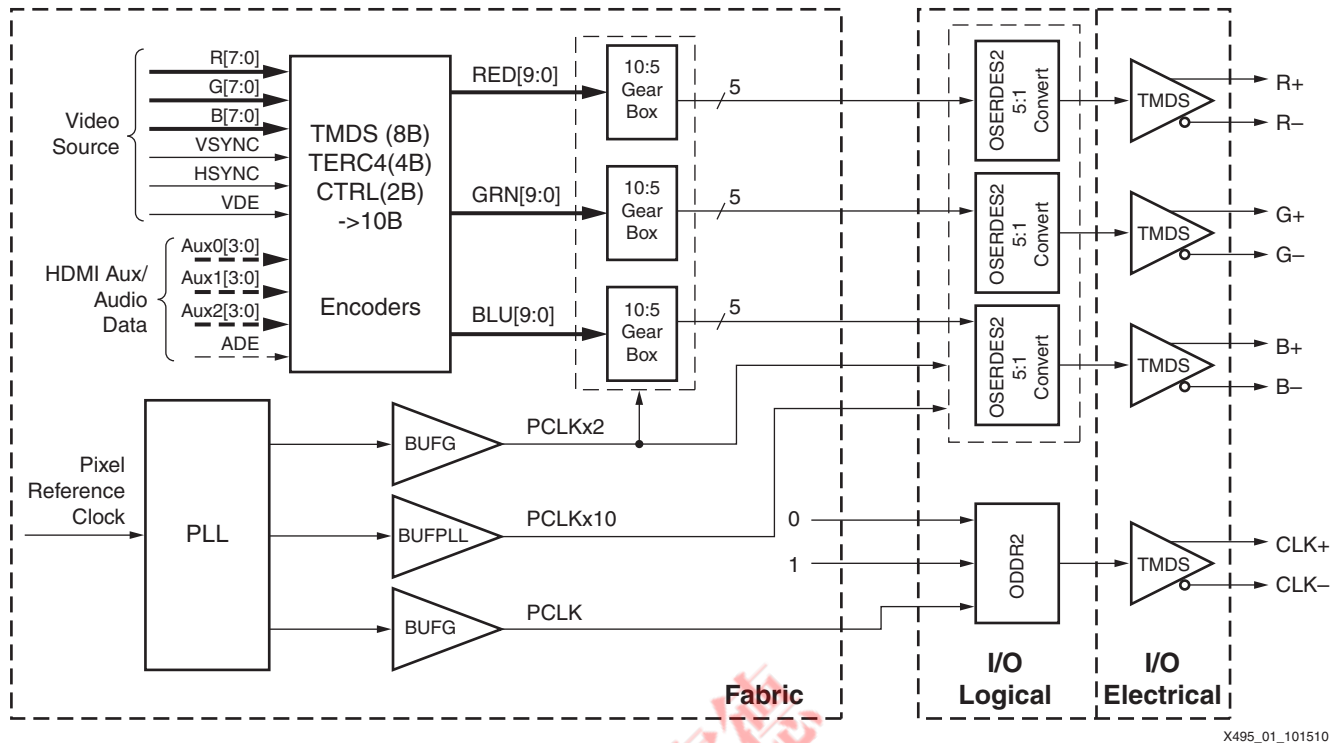


Figure 1: TMDS Transmitter Design

The TMDS encoder in the Spartan-6 FPGA is identical to that in the Spartan-3A FPGA, and is thoroughly discussed in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs* [Ref 1]. However, the serializer and its clocking scheme are significantly different in the Spartan-6 FPGA.

10:1 Serializer

Compared to the Spartan-3A FPGA soft solution [Ref 1], the serializer design in the Spartan-6 FPGA is completely based on hardened circuitry. Each output pin in the Spartan-6 FPGA comes with a built-in OSERDES2 block that converts a maximal 4-bit parallel bus to a 1-bit serial stream. By cascading with another OSERDES2 block within its adjacent output pin, the hardened circuitry is able to perform serialization with a maximal 8:1 ratio. This is typically obtained by default when the output is operating in a differential pair like the TMDS. The OSERDES2 blocks are fully configurable to work in either cascading or non-cascading mode with any ratio ranging from 1 to 8. A detailed description of the OSERDES2 block can be found in *Spartan-6 FPGA SelectIO Resources User Guide* [Ref 4].

To perform the required 10:1 serialization for both DVI and HDMI, two stages of conversion must be made: a 2:1 soft gear box and 5:1 OSERDES2 cascading.

Clocking Scheme

Compared to the transmitter in the Spartan-3A FPGA, the clocking scheme in the Spartan-6 FPGA design is much more simplified. There is no longer the need to implement DDR based high-speed soft multiplexing logic [Ref 1].

Leveraging the built-in high-performance PLL and dedicated I/O clock network, both the soft 2:1 gear box and the hard 5:1 OSERDES2 block work in single data rate (SDR) mode. The PLL takes the pixel rate reference as its input and synthesizes three clocks: a pixel clock, a 10x pixel clock, and a 2x pixel clock.

The 10x pixel clock is used to match the serial data bit rate. It is routed to the OSERDES2 clock through a dedicated BUFPLL driver. This is named as an I/O Clock. The clock runs as fast as

945 MHz in the -2 speed grade, 1050 MHz in the -3 speed grade, and 1080 MHz in the -4 speed grade [Ref 5]. The OSERDES2 block also takes the 2x pixel clock as its 5-bit parallel data input reference.

The 2:1 soft gear box, however, takes both the pixel clock and the 2x pixel clock. It converts the 10-bit TMDS encoded data into a 5-bit data stream. The logic here is extremely insignificant. The only requirement is separate sampling of a 5-bit MSB and 5-bit LSB in the 2x pixel clock domain. The sampling is controlled by a toggle flip-flop generated framing signal.

Timing Analysis and Constraint

Other than resource saving with the help of hardened serialization circuitry, another major advantage of the Spartan-6 FPGA solution is the ease of FPGA logic timing.

The FPGA logic has only two timing domains: pixel clock and x2 pixel clock. By setting the PERIOD constraint on the PLL reference input, the ISE® Design Suite automatically derives appropriate constraints for the 2x and 10x pixel clocks. The latter only applies to hardened clocks and datapaths and has no impact on any of the soft logic running through the FPGA logic. The PERIOD constraint can be set in this manner:

```
NET "pclk" TNM_NET = DVI_CLOCK0;  
TIMESPEC TS_DVI_CLOCK0 = PERIOD "DVI_CLOCK0" 100 MHz HIGH 50%;
```

In the code above, pclk represents the PLL input running at the video pixel rate.

Receiver Design

The receiver typically needs to recover the bit sampling clock using the incoming pixel clock and then apply the bit clock to recover the serial data stream back into 10-bit word aligned symbols. This process is known as clock and data recovery (CDR). The second step involves a channel deskew circuit to remove allowed skews among the three data channels. Finally, the 10-bit symbol is decoded into one of these three formats:

- 8-bit video pixel data through the DVI or HDMI decoder
- 4-bit auxiliary data, i.e., information and audio frames through the HDMI decoder only
- 2-bit control data, e.g., the HSYNC and VSYNC through the DVI or HDMI decoder

The second decoder step in the Spartan-6 FPGA is identical to that in the Spartan-3A FPGA, and is thoroughly discussed in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs* [Ref 1]. The first step CDR and deserialization segment, however, has changed significantly to leverage all the new Spartan-6 FPGA I/O features for higher performance with less resources. Figure 2 illustrates the topology of the DVI/HDMI receiver design in the Spartan-6 FPGA.

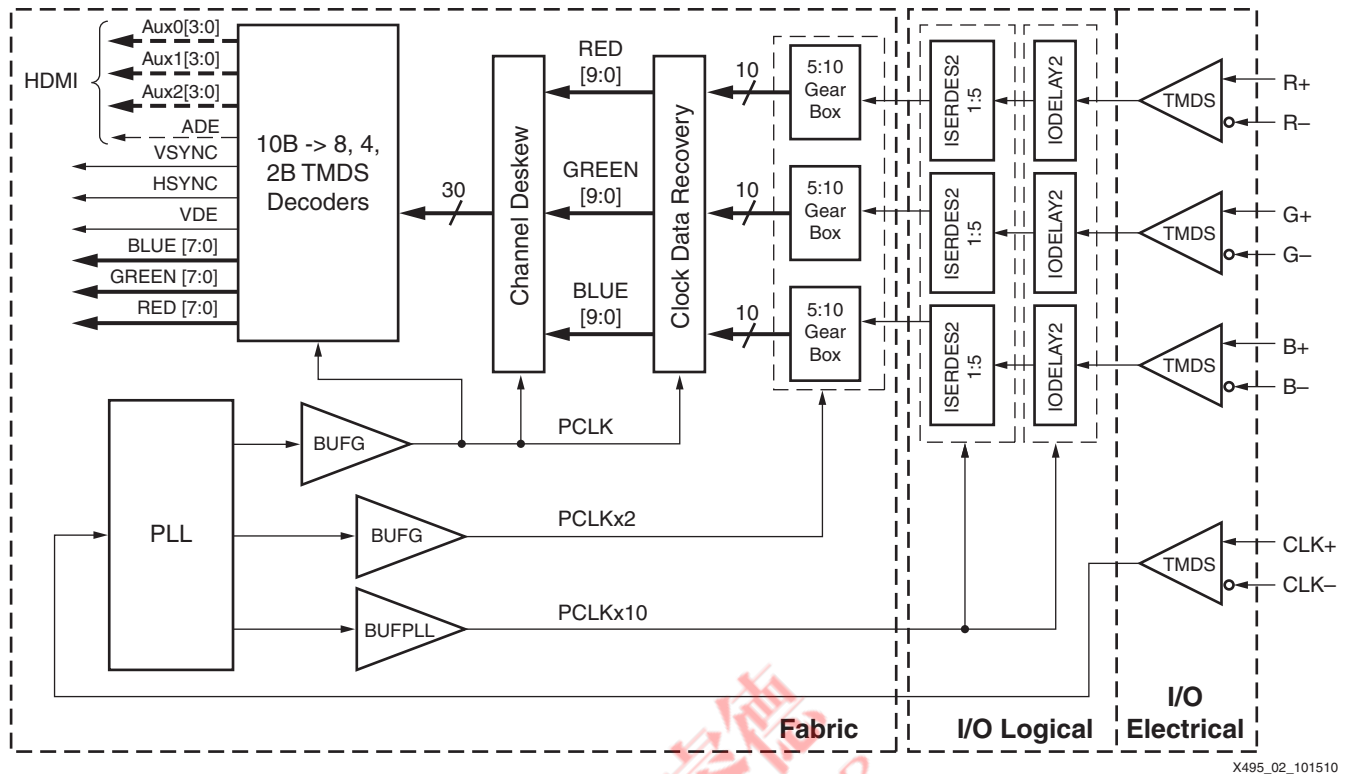


Figure 2: TMDs Receiver Design

1:10 Deserialization

Like the transmitter serialization design, deserialization is also based on hardened circuitry working in SDR mode. Each input pin in the Spartan-6 FPGA comes with a built-in ISERDES2 block that converts a 1-bit serial stream to a maximal 4-bit parallel bus. By cascading with another ISERDES2 block within its adjacent input pin, the hardened circuitry is able to perform serialization with a maximal 1:8 ratio. This is typically obtained by default when the input is operating in a differential pair like the TMDs. The ISERDES2 blocks are fully configurable to work in either cascading or non-cascading mode with any ratio ranging from 1 to 8. A detailed description of the ISERDES2 block can be found in *Spartan-6 FPGA SelectIO™ Resources User Guide* [Ref 4].

Similar to the transmitter serialization design, two stages of conversion must be undertaken to obtain the required 1:10 deserialization: 1:5 ISERDES2 cascading and 1:2 FPGA logic gear box.

Clock and Data Recovery

The TMDs clock channel carries a character rate frequency reference from which the receiver reproduces a bit rate sample clock for the incoming serial streams. The reproduced bit rate clock, however, does not have a guaranteed phase relationship associated with any of the three data lanes. In addition, both DVI [Ref 6] and HDMI specifications [Ref 7] allow certain skew between any two data lanes. As a result, the clock phase must be adjusted individually for each data lane to correctly sample the incoming serial bit. This involves aligning the clock rising edge to the middle sampling window of each data lane. Like the serialization and deserialization steps, the CDR work here is mostly done through hardened circuitry.

The bit sampling clock is reproduced through a PLL using the incoming TMDs pixel clock as a reference. The TMDs clock is multiplied by 10 to match the bit rate and then fed to a BUFPLL that routes the 10x clock to designated ISERDES2 and IODELAY2 blocks. This clock is named IOCLK. Figure 2 illustrates the clock connections.

Other than deserialization of serial bits into a parallel bus, the ISERDES2 block in the Spartan-6 FPGA possesses a unique “phase detector” function when operating in cascading mode. By sampling and comparing the incoming serial data samples using the two ISERDES2 blocks residing in two adjacent differential input pins, the phase detector is able to determine the phase relationship between the current IOCLK rising edge and the serial data transition edge. Subsequent validation and control signals are sent to the FPGA logic for a soft controller state machine to adjust the IODELAY2 accordingly.

The IODELAY2, on the other hand, provides a dynamically adjustable delay line to the incoming serial data bit. Upon receiving the control signals from the controller state machine, the IODELAY2 is able to align the rising edge of IOCLK to the middle of the data sampling window.

A detailed operation theory of the phase detector together with the IODELAY2 and its controller state machine can be found in *Spartan-6 FPGA SelectIO™ Resources User Guide* [Ref 4] and *Source-Synchronous Serialization and Deserialization (up to 1050 Mb/s)* [Ref 8].

To accommodate the unknown clock to data phase relationship, the phase detector and the IODELAY2 together must be able to shift the data window either to the left or right for up to 0.5 UI. This is determined by the DIFF_PHASE_DETECTOR operation mode of the IODELAY2 and its relevant calibration routine. The calibration process:

- Measures the number of taps in one bit time (UI). This becomes the maximum length of the IDELAY line and is stored as a MAX value into an internal register.
- Presets the master IDELAY to the HALF_MAX (half of the MAX value) position.
- Presets the slave IDELAY to the ZERO position.

Five cases can occur depending on the various clock-to-data initial phase positions. Figure 3 to Figure 7 illustrate how the phase detector works together with the IDELAY blocks and moves the data to align with the clock rising edge.

Case 1

The master IDELAY settles around HALF_MAX and the slave IDELAY settles around ZERO. There might be a bounce of ± 2 taps over voltage and temperature changes (Figure 3).

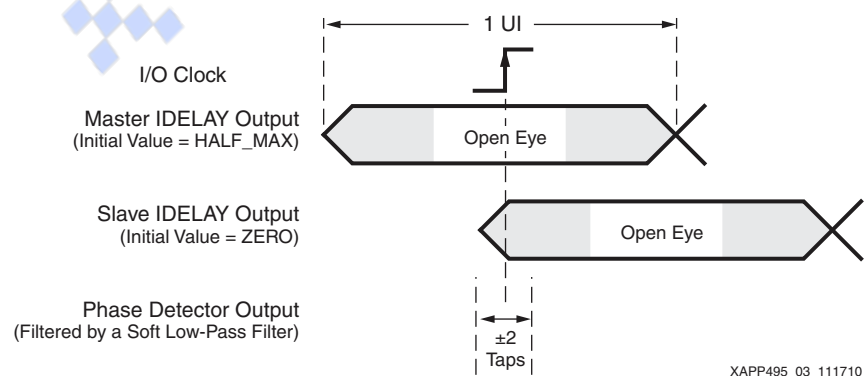


Figure 3: Case 1: IOCLK Initially Falls into the Middle of the Data Window

Case 2

The master IDELAY settles around $\text{HALF_MAX} + n$, and the slave IDELAY settles around n , where $n < \text{HALF_MAX}$ (Figure 4).

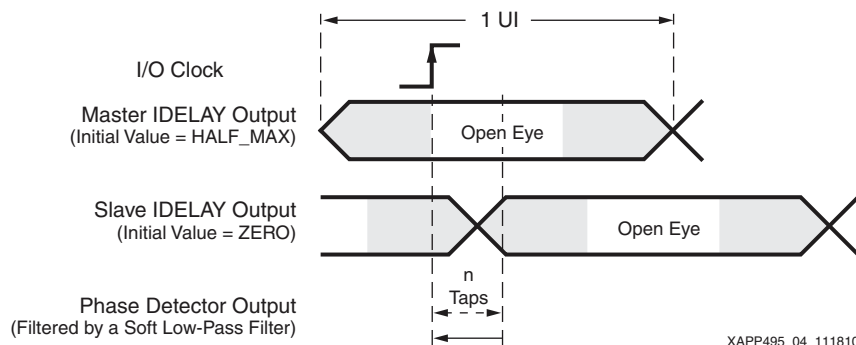


Figure 4: **Case 2: IOCLK Initially Falls into the Left Side of the Data Window**

Case 3

The master IDELAY settles around $\text{HALF_MAX} - n$ and the slave IDELAY settles around $-n$ or $\text{MAX} - n$, where $n < \text{HALF_MAX}$ (Figure 5).

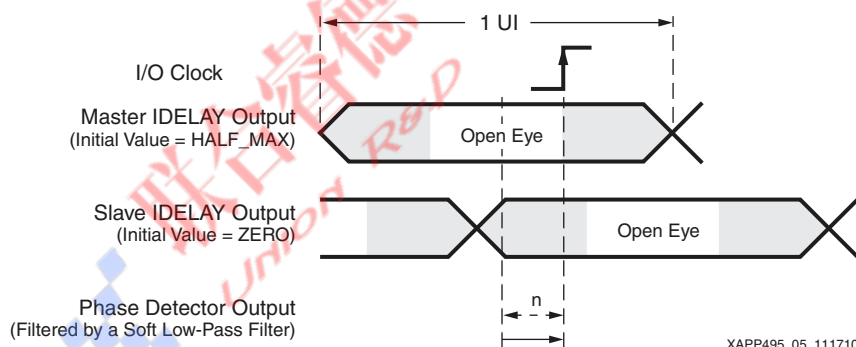


Figure 5: **Case 3: IOCLK Initially Falls into the Right Side of the Data Window**

Case 4

The master IDELAY settles around MAX and the slave IDELAY settles around HALF_MAX (Figure 6).

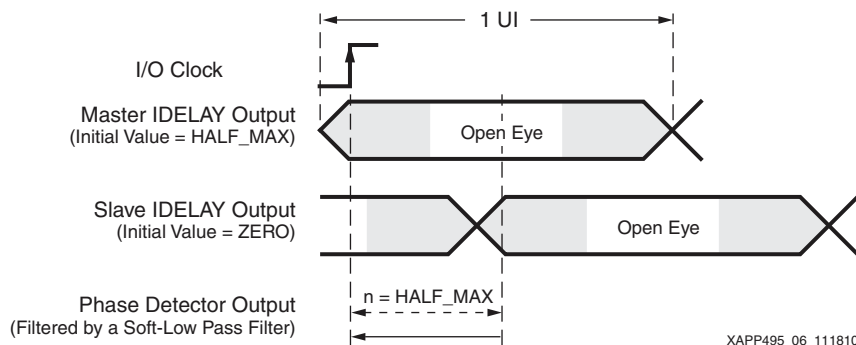


Figure 6: **Case 4: IOCLK Initially Falls by the Left Data Transition Edge**

Case 5

The master IDELAY settles around ZERO and the slave IDELAY settles around HALF_MAX (Figure 7).

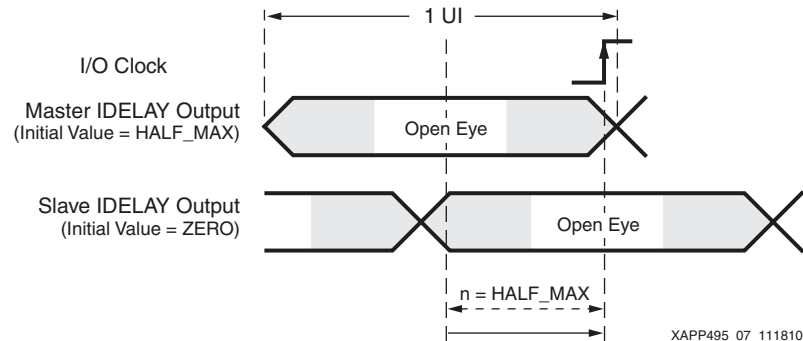


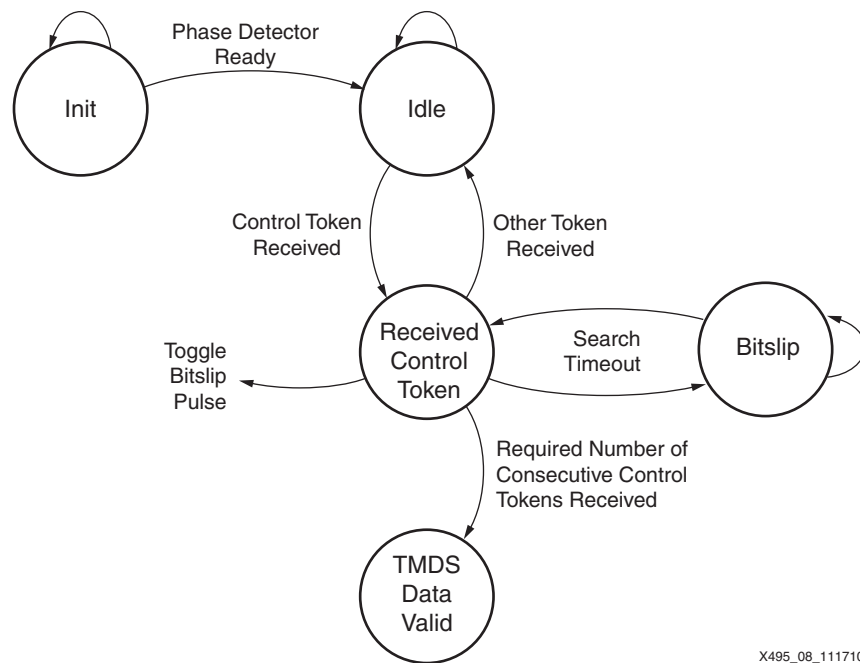
Figure 7: **Case 5: IOCLK Initially Falls by the Right Data Transition Edge**

In Case 4 and Case 5, because the master IDELAY settles either at ZERO or MAX, its IDELAY2 attribute COUNTER_WRAPAROUND must be set to STAY_AT_LIMIT to prevent the IDELAY from underflow or overflow. This is important because the clock and data jitter cause the phase detector to issue extra decrement or increment commands.

This usage is different from that described in *Source-Synchronous Serialization and Deserialization (up to 1050 Mb/s)* [Ref 8], where WRAP_AROUND is suggested when clock and data are roughly pre-aligned. For the cases with unknown initial clock to data phase described in this application note, the use of WRAP_AROUND leads the master sampling window to accidentally draft one bit and subsequently cause bit errors.

Word Boundary Detection

Instead of building a soft barrel shifter in the FPGA logic as described in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs* [Ref 1], another hardened ISERDES2 feature is used in the Spartan-6 FPGA design: the bitslip function. In brief, the ISERDES2 block has the barrel shifter built-in. Upon receiving a one cycle long control pulse, the ISERDES2 block shifts one bit in its deserialized parallel data. *Spartan-6 FPGA SelectIO™ Resources User Guide* [Ref 4] has a detailed description of the bitslip operation. Figure 8 illustrates a possible controller state machine that appropriately issues the bitslip control pulse to determine the word boundary.



X495_08_111710

Figure 8: State Machine to Determine TMDS Word Boundary

Data Lane Deskew

To remove the allowed data lane skew (up to 0.6 pixel time [Ref 6], [Ref 7]), FIFO-based deskew logic is built. Each data lane hosts a FIFO that is capable of holding data before the other two lanes are aligned to the same timing sequence. The detailed operation of the deskew logic is discussed in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs*. The design module in the Spartan-3A FPGA can be dropped into the Spartan-6 FPGA with no modification.

Timing Analysis and Constraint

Like the transmitter design, the receiver also operates in three different clock domains:

- Pixel clock: Applicable to most of the FPGA logic, including the ISERDES2/IODELAY2 controller and TMDS decoder
- Pixel clock x2: Applicable to the 1:2 gear box and ISERDES2/IODELAY2 internal
- Pixel clock x10: Applicable only to the ISERDES2/IODELAY2 internal

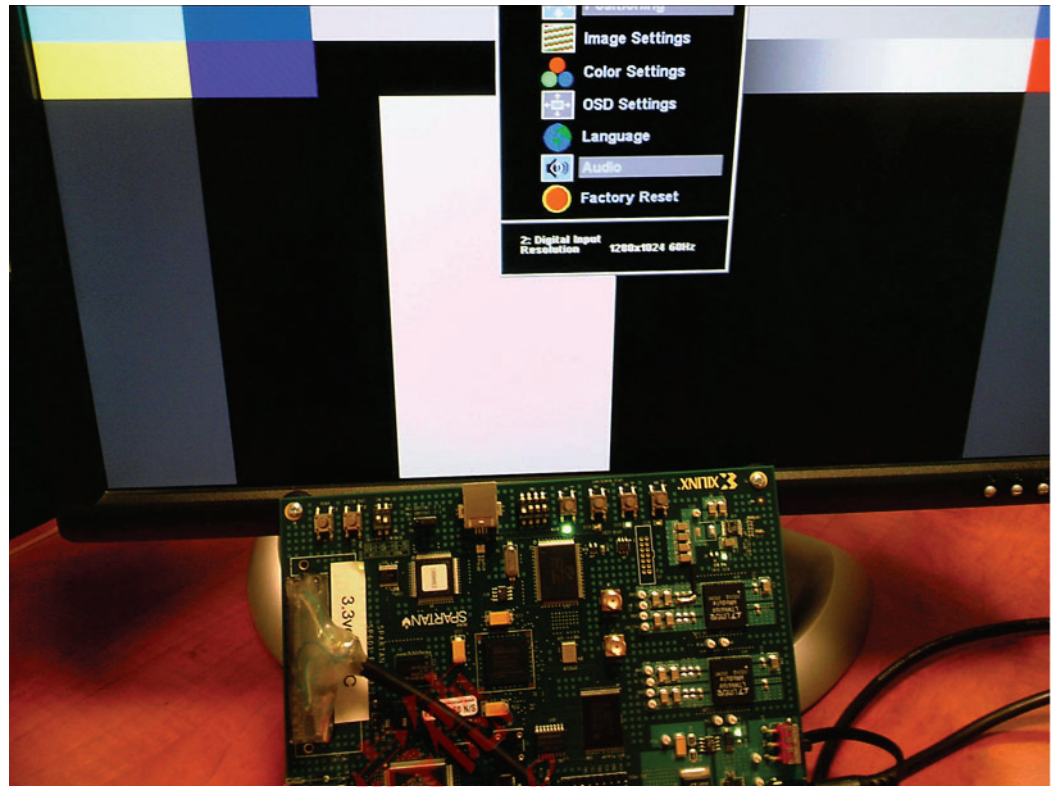
By setting the appropriate PERIOD constraint to the TMDS clock feeding into the PLL input, the ISE software automatically derives constraints for all three clocks. The PERIOD constraint can be set in this manner:

```
NET "TMDS_CLK" TNM_NET = DVI_CLOCK0;
TIMESPEC TS_DVI_CLOCK0 = PERIOD "DVI_CLOCK0" 100 MHz HIGH 50%;
```

In the code above, TMDS_CLK represents the TMDS clock coming from the DVI or HDMI cable.

Implementation

The hardware evaluation of the Spartan-6 FPGA TMDS capability underwent two stages. An initial feasibility study was performed on the Spartan-6 FPGA SP601 Evaluation Kit. A standard DVI cable was cut into half and the conductors soldered to a 68-pin FMC male connector that was mated with the female FMC connector on the SP601 board. Figure 9 demonstrates the connections, together with a color bar successfully displayed on a standard DVI monitor with resolution up to 1280x1024 @ 60 Hz.



X495_09_111710

Figure 9: **Spartan-6 DVI Transmission on SP601 Board**

The success of this proof-of-concept justified the further pursuit of both the transmitter and receiver designs in a more refined hardware platform.

Validation Hardware

Video Connectivity Using TMDS I/O in Spartan-3A FPGAs [Ref 1] has a thorough discussion of the electrical considerations for transmitting and receiving TMDS signals through the use of the TMDS_33 SelectIO™ interface mode in the Spartan-3A FPGA. The Spartan-6 FPGA shares the same concepts:

- TMDS_33 outputs are supported only on banks 0 and 2 (top and bottom).
- TMDS_33 outputs require $V_{CCO} = 3.3V$ and $V_{CCAUX} = 2.5V/3.3V$.
- TMDS_33 inputs require $V_{CCO} = 2.5V/3.3V$ and $V_{CCAUX} = 3.3V$.
- TMDS_33 signals require 50Ω termination to 3.3V at the receiver.

For a simple design that both transmits and receives TMDS signals (such as an evaluation board), it is easiest to use 3.3V for V_{CCAUX} and V_{CCO} on all four banks of the device.

In addition, other considerations described in *Video Connectivity Using TMDS I/O in Spartan-3A FPGAs* [Ref 1] are also applicable here for the Spartan-6 FPGA. These include 100Ω differential impedance matching on PCB trace lines, display data channel (DDC)/hot plug detect (HDP) compatibility, and use of the low-cost TMDS buffer TMDS141 from Texas Instruments to offer additional signal clean-up and ESD protection up to 6 kV. Figure 10 shows an evaluation board named Atlys built with these concepts and considerations in mind.

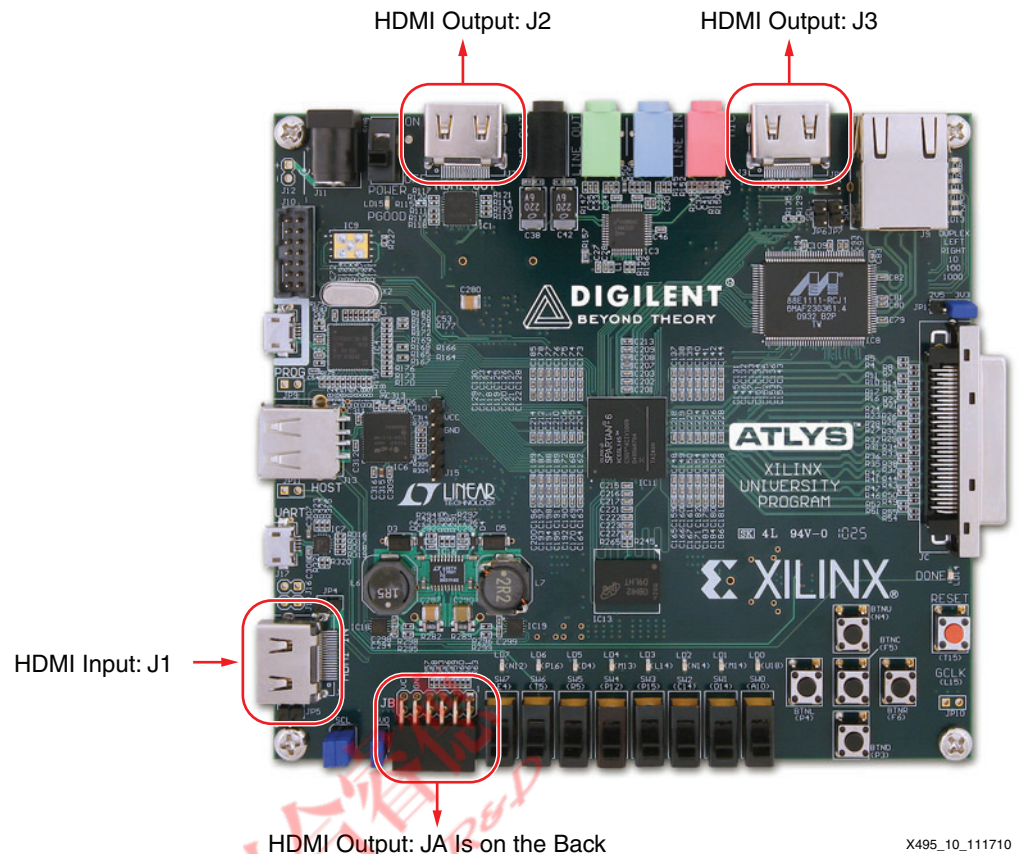


Figure 10: **Atlys TMD5 Evaluation Board**

The Atlys board comes with 2x HDMI input ports and 2x HDMI output ports. The two inputs are both based on standard type-A [Ref 7] HDMI connectors located on J1 and J3. The two outputs, in contrast, are configured into two different types of HDMI connectors: type-A [Ref 7] (located at J2) and type-D [Ref 7] (located at JA). All three type-A ports are buffered through the TI TMDS141 buffers. The signals through the type-D connector, however, are connected directly to the Spartan-6 FPGA. One of the two input ports is connected to FPGA I/O bank 0 and the other is connected to bank 1. The type-A output port is connected to bank 0, and the type-D port is connected to bank 2.

The Atlys board is ideal for showcasing the superior capability of the Spartan-6 FPGA at simultaneously receiving, driving, and processing multiple TMDS HD video streams in a single device. The board is available through the Xilinx University Program or can be ordered from Digilent at <http://www.digilentinc.com>.

Reference Designs

Two reference designs have been built on the Atlys board. For the transmitter alone, a color bar generator is designed to work in various dynamically configurable video timing modes. To demonstrate both the transmitter and receiver, a 2 x 2 DVI matrix is implemented.

Programmable Video Timing Controller

To validate whether the TMDS I/O in the Spartan-6 FPGA is able to transmit video across different screen modes, a color bar generator is built to work with a dynamically configurable pixel clock, as shown in Figure 11. The pixel clock is generated in real time using another new feature in the Spartan-6 FPGA: DCM_CLKGEN. This synthesizes different CLKFX frequencies with variable M/D values using the dynamically configurable system packet interface (SPI)

based on a stable incoming clock reference. *Spartan-6 FPGA Clocking Resources User Guide [Ref 9]* contains a detailed discussion about the DCM_CLKGEN feature.

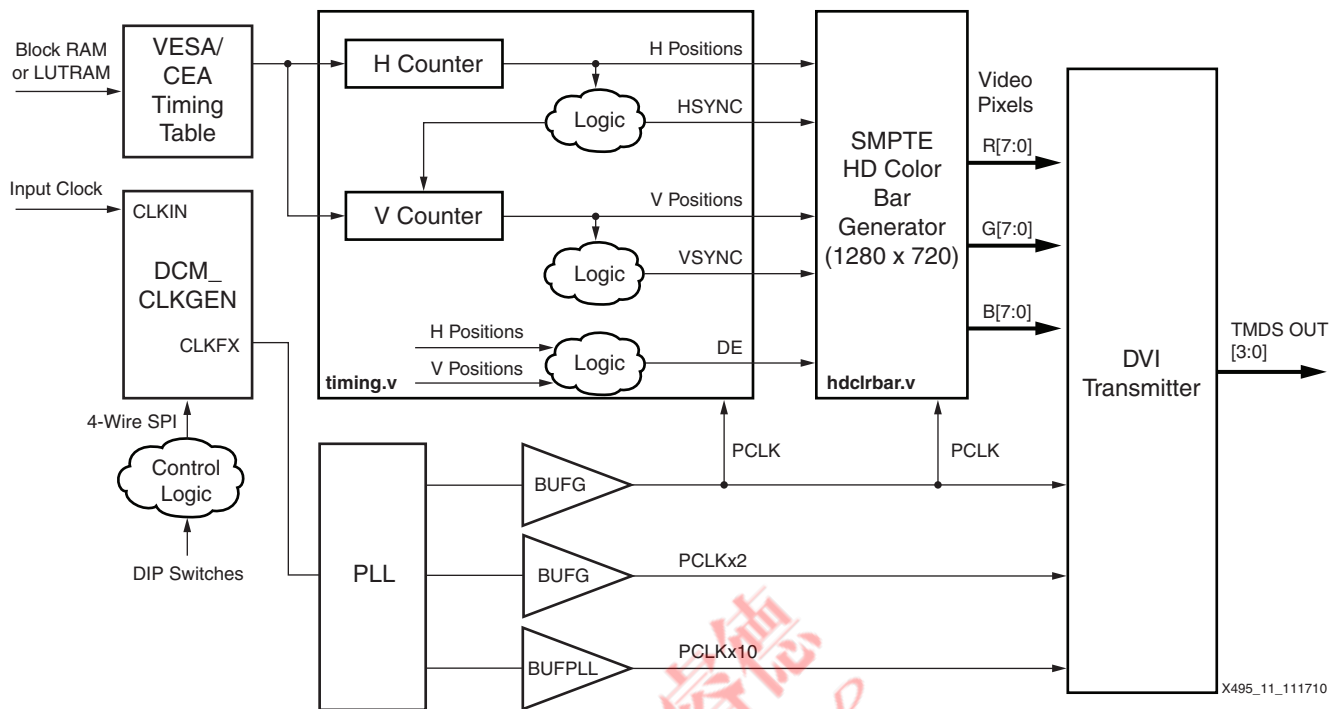


Figure 11: SMPTE HD Color bar Generation with Programmable Video Timing

In this design, using three DIP switches on the board (SW0, SW1, and SW3), the user is able to switch among five different screen modes:

- 640x480 @ 60 Hz
- 800x600 @ 60 Hz
- 1024x768 @ 60 Hz
- 1280x720 @ 60 Hz
- 1280x1024 @ 60 Hz

2 x 2 DVI Matrix

The best way to validate the receiver is to pass the video it receives to an output, based on the idea of what you see is what you get. To demonstrate the multi-port capability of the design, all four ports on the Atlys board are used to provide two inputs and two outputs concurrently.

Figure 12 illustrates the design topology.

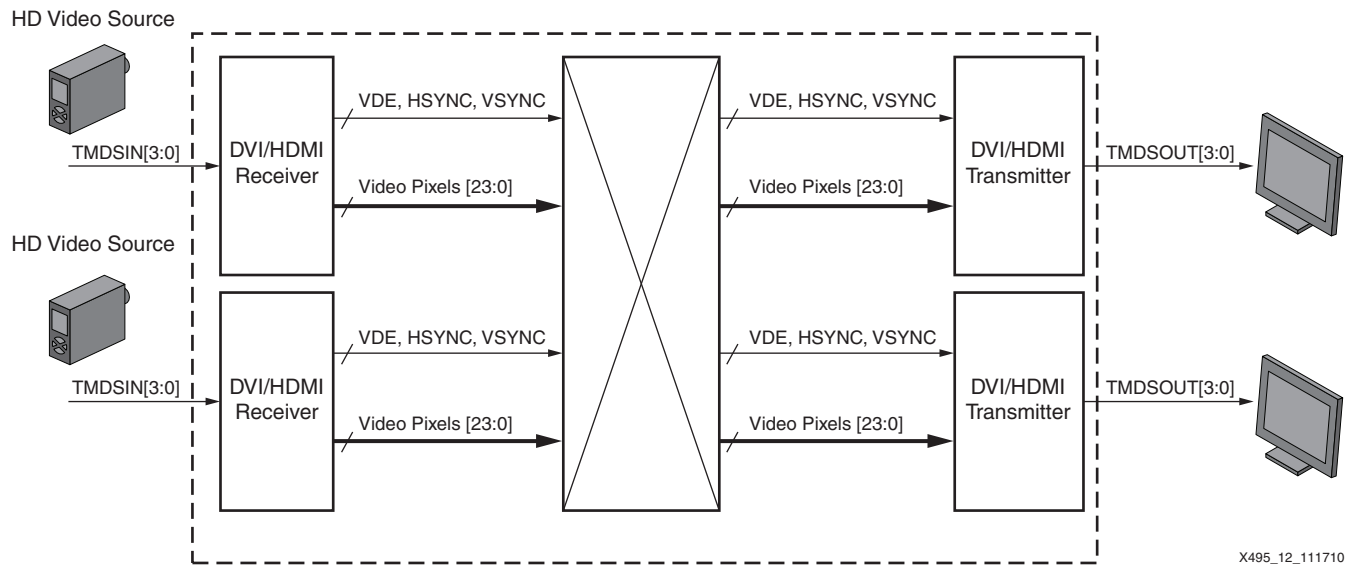


Figure 12: 2 x 2 DVI Matrix

Using two on-board DIP switches (SW0 and SW1), the design is able to route the two video inputs to its outputs with four possible configurations. Figure 13 shows that the two video inputs are routed to the two video outputs at the same time.



Figure 13: 2 x 2 DVI Matrix Operating

Reference Design

Reference Design Files

The reference design files for this application note can be downloaded from:

<https://secure.xilinx.com/webreg/clickthrough.do?cid=154258>.

The ZIP file contains the following design files:

DVI Transmitter

- `dvi_demo/rtl/tx/dvi_encoder.v`: DVI transmitter top module without instantiation of clocking resources
- `dvi_demo/rtl/tx/dvi_encoder_top.v`: DVI transmitter wrapper with instantiation of clocking resources
- `dvi_demo/rtl/tx/encode.v`: DVI encoder
- `dvi_demo/rtl/tx/serdes_n_to_1.v`: Deconfigurable to 5:1 serializer
- `dvi_demo/rtl/tx/convert_30to15_fifo.v`: 30-bit 2:1 gear box
- `dvi_demo/rtl/tx/vtc_demo.v`: Color bar generator with programmable timing controller

DVI Receiver

- `dvi_demo/rtl/tx/dvi_decoder.v`: DVI receiver top wrapper
- `dvi_demo/rtl/tx/decode.v`: DVI decoder instantiating the CDR and channel deskew circuits
- `dvi_demo/rtl/tx/chnlbond.v`: Channel deskew module
- `dvi_demo/rtl/tx/phsaligner.v`: Bit-slip and TMDS data validation state machine
- `dvi_demo/rtl/tx/serdes_1_to_5_diff_data.v`: 1:5 de-serializer

DVI Common Modules

- `dvi_demo/rtl/common/DRAM16XN.v`: Width configurable distributed RAM
- `dvi_demo/rtl/common/timing.v`: Video Timing Controller
- `dvi_demo/rtl/common/hdclrbars.v`: SMPTE HD color bar generator
- `dvi_demo/rtl/common/debounce.v`: DIP switch debouncer
- `dvi_demo/rtl/common/dcmspi.v`: DCM_CLKGEN SPI controller
- `dvi_demo/rtl/common/synchro.v`: Clock boundary synchronizer

DVI Evaluation

- `dvi_demo/rtl/dvi_demo.v`: 2 x 2 DVI Matrix design

The reference design checklist is shown in [Table 3](#).

Table 3: Reference Design Checklist

Parameter	Description
General	
Developer Name	Xilinx
Target Devices (Stepping Level, ES, Production, Speed Grades)	Spartan-6 FPGA
Source Code Provided	Yes
Source Code Format	Verilog

Table 3: Reference Design Checklist (Cont'd)

Parameter	Description
Design Uses Code/IP from Existing Application Note, Reference Designs, Third Party, or CORE Generator™ Software	Yes, <i>Source-Synchronous Serialization and Deserialization (up to 1050 Mb/s)</i> [Ref 8]
Simulation	
Functional Simulation Performed	Yes
Timing Simulation Performed	No
Testbench Used for Functional and Timing Simulations	No
Testbench Format	Verilog
Simulator Software/Version Used	Cadence IUS 8.2
SPICE/IBIS Simulations	No
Implementation	
Synthesis Software Tools/Version Used	XST, version 12.3
Implementation Software Tools/Versions Used	ISE software, version 12.3
Static Timing Analysis Performed	Yes
Hardware Verification	
Hardware Verified	Yes
Hardware Platform Used for Verification	Digilent Spartan-6 FPGA Atlys Board

Table 4 shows the resource utilization for the DVI transmitter alone.

Table 4: Resource Utilization for DVI Transmitter

Resource	Quantity
LUT6	408
Flip-Flops	402
BUFG	2
BUFIO2	1
BUFPLL	1
PLL	1
RAMB	0

Table 5 shows the resource utilization for the DVI receiver alone.

Table 5: Resource Utilization for DVI Receiver

Resource	Quantity
LUT6	478
Flip-Flops	453
BUFG	2
BUFIO2	1
BUFPLL	1

Table 5: Resource Utilization for DVI Receiver (Cont'd)

Resource	Quantity
PLL	1
RAMB	0

The final resource usage is application or design dependent. For instance, in some pass-through designs, the transmitter and receiver might be able to share the same clocking resources for pixel clock, pixel clock x2, and pixel clock x10. This typically happens when the transmitter and receiver are fully synchronized and both use bank 0 or bank 2 TMDs I/Os. Thus, the actual BUFG, BUFPLL, and PLL count for the transmitter is the same as the receiver alone.

Conclusion

This application note successfully demonstrates a high-definition TMDs video connectivity solution featuring the processing of multiple HD channels in parallel. The design demonstrates the Spartan-6 FPGA in DVI or HDMI applications (including the 480p, 1080i, and 720p video formats).

References

1. [XAPP460](#), *Video Connectivity Using TMDs I/O in Spartan-3A FPGAs*.
2. *VESA Coordinated Video Timing Generator, Revision 1.1*, Video Electronics Standards Association, <http://www.vesa.org>.
3. CEA-861-D, *A DTV Profile for Uncompressed High Speed Digital Interfaces*, Consumer Electronics Association, <http://www.ce.org>.
4. [UG381](#), *Spartan-6 FPGA SelectIO Resources User Guide*.
5. [DS162](#), *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.
6. *Digital Visual Interface, Revision 1.0*, Digital Display Working Group, <http://www.ddwg.org>.
7. *High-Definition Multimedia Interface Specification Version 1.3a*, HDMI Licensing, LLC, <http://www.hdmi.org>.
8. [XAPP1064](#), *Source-Synchronous Serialization and Deserialization (up to 1050 Mb/s)*.
9. [UG382](#), *Spartan-6 FPGA Clocking Resources User Guide*.

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
12/13/10	1.0	Initial Xilinx release.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you "AS-IS" with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.