1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

DVI-to-RGB (Sink) 1.4 IP Core User Guide

Revised April 22, 2015; Author Elod Gyorgy

1 Introduction

This user guide describes the Digilent DVI-to-RGB Video Decoder Intellectual Property. This IP interfaces directly to raw transition-minimized differential signaling (TMDS) clock and data channel inputs as defined in DVI 1.0 specs for Sink devices. It decodes the video stream and outputs 24-bit RGB video data along with the pixel clock and synchronization signals recovered from the TMDS link.

IP quick facts	
Supported device families	Zynq®-7000, 7 series
Supported user interfaces	Xilinx®: IIC, vid_io Digilent: TMDS
Provided with core	
Design files	VHDL
Simulation model	VHDL Behavioral
Constraints file	XDC
Software driver	N/A
Tested design flows	
Design entry	Vivado™ Design Suite 2014.3
Synthesis	Vivado Synthesis 2014.3

2 Features

- Connects directly to top-level digital visual interface (DVI) port
- 24-bit video (clocked parallel video data with synchronization signals) output
- Display Data Channel interface with built-in EDID ROM
- Resolutions supported: 1920x1080/60Hz down to 800x600/60Hz (148.5 MHz – 40 MHz)
- Xilinx interfaces used: IIC, vid_io
- Digilent interfaces used: TMDS

3 Performance

The IP constrains TMDS_Clk to 165 MHz, the maximum frequency outlined in DVI 1.0 specifications. However, depending on the actual FPGA part or speed grade, the maximum supported frequency might be lower. If the top-level design fails timing on pulse-width checks inside the IP instance, TMDS_Clk needs to be (re)constrained to the maximum frequency supported on the project part. Check the part datasheet for F_{MAX_BUFIO}, which is the most likely reason for failed timing. TMDS_Clk should be constrained for $F_{MAX\_BUFIO}/5$. Consequently, this is the maximum pixel clock frequency supported on that FPGA family and speed grade.

4 Overview

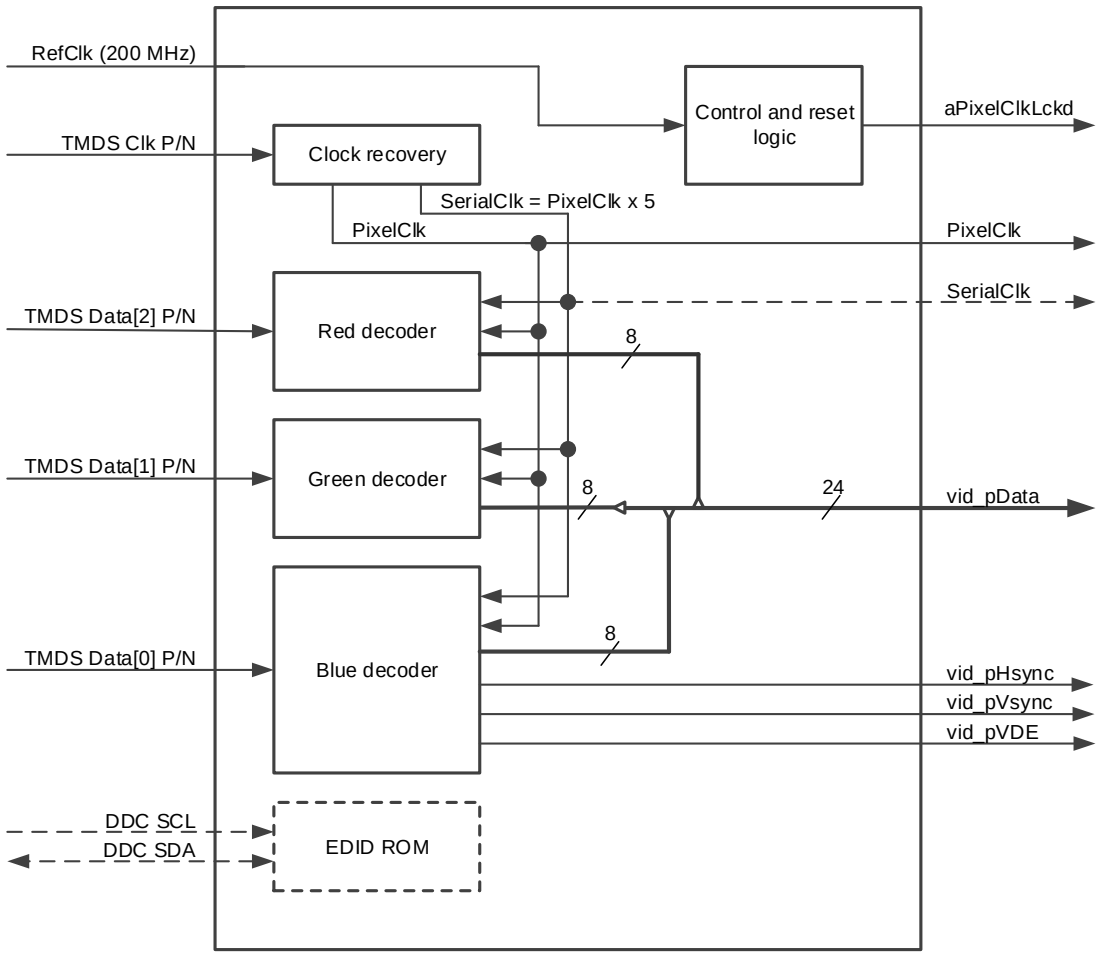

Figure 1. DVI to VGA converter block diagram.

The IP is built from multiple blocks: one clock recovery block, one data decoder block for each data channel (see [3], [4]), one optional DDC (Display Data Channel) block and one control/reset block.

4.1 Clock recovery

The clock channel carries a character-rate frequency reference. One character (or 10 bits) are transmitted every period on each data channel. Dedicated deserializer primitives, which require a fast serial clock, will be used to sample the serial data stream. The clock recovery block generates a serial clock and a pixel clock from the clock channel. The frequency ratio between the two clocks is 5:1.

Since the clock frequencies are relatively high and the recovered clocks have tight phase requirements, dedicated clocking primitives are instantiated inside this block. These can be seen in Figure 2.

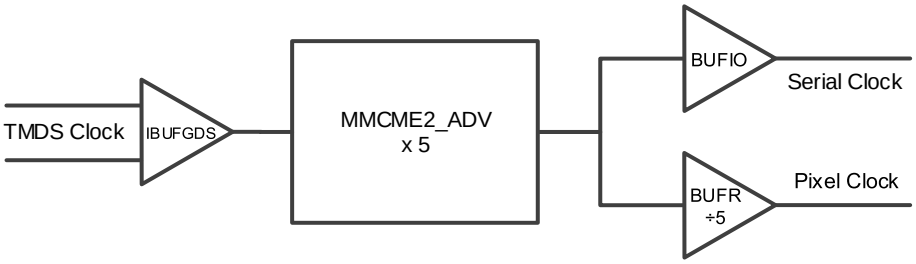

Figure 2. Clock network overview.

4.2 Data Decoder

There is no phase relationship between the recovered pixel clock or serial clock and the data channels. Furthermore, the data channels themselves are allowed a considerable skew between them. So the decoder block needs to align the phase of the serial clock with the data bit and find the character boundary in each data stream. Since this has to be done independently on each of the three data channels, it makes sense to vary the phase of the data streams while keeping the clock phase fixed.

4.2.1 Synchronization

To help with synchronization, the DVI protocol specifies period cues (control tokens) to be sent. These control tokens are sufficiently different from the rest of the data that their succession can be used for synchronization. Synchronization is automatically (re)started when a stable clock is detected and recovered. The time it takes for a lock to be achieved depends on the phase relationship of the clock and data streams. It should not last more than 1 minute.

DVI characters are 10-bits long, so a 10:1 deserialization of the data stream is needed. This can be achieved with two ISERDESE2 primitives in a cascaded DDR configuration. In this configuration, the master and slave ISERDESE2 take the serial data stream and sample it on both edges of a serial clock. Thus, for every five serial clock periods, ten data bits are sampled. This 10-bit data is then output synchronously with a divided clock, which is our pixel clock from the clock recovery block. Although this recovers 10-bit words from the data stream at a frequency which can be passed on to general logic inside the FPGA, it does guarantee the word actually starts at a character boundary or if the data stream is sampled when data is stable.

To find the best moment to sample the data stream (i.e., the middle of an open eye), an IDELAYE2 primitive is inserted in front of ISERDESE2. This primitive is capable of delaying the data signal in tap increments. In this IP, a 78ps increment is used for a total of 32 increments. For the highest pixel clock frequency supported (165 MHz), one bit period is covered in 7 tap increments. The goal is to find the tap delay value that shifts the data enough so that it gets sampled in the middle of its stable zone. The phase alignment module compares the 10-bit words with the four special control tokens. If a succession of tokens is not recognized in a timeout period, we are in the jitter zone and it increments the tap delay. This is done repeatedly until control tokens are reliably recognized and the algorithm settles on the middle of the stable bit period (open eye).

However, the IDELAYE2 primitive only provides a fine phase adjustment on the bit level, not covering the whole character. To find the character boundary, a coarse phase adjustment is needed. This is achieved by the “bitslip” feature of the ISERDESE2 primitive. If all the tap increments have been tried and control tokens are still not detected, it is assumed that we are not at the right character boundary. In this case, invoking “bitslip” causes either a shift right by one while bit or a shift left by three in the 10-bit word. After “bitslip” completes, phase alignment begins again, looping through the tap increments until tokens are found.

Phase alignment is considered completed when a succession of control tokens are reliably detected on all data channels. At this moment all three data channel are considered valid.

However, since inter-pair channel skew is not negligible and channels are aligned independently, the recovered data streams might have different delays on them. To eliminate this skew, the channels are bonded by buffering them in FIFO memories and holding them back independently until the video blanking period starts at the same time on all three. At this stage, all three data channels are valid and in-sync.

4.2.2 Decoding

The TMDS standard encodes data so that the serial data stream contains few transitions (0-to-1 or 1-to-0) and a DC balance (the same number of zeros and ones over a long time period). Every 10-bit character actually encapsulates 8 bits of useful data. The exception to this are the control tokens, which encapsulate 2 bits of control data. The data decoder block applies the decoding algorithm as specified in the DVI 1.0 specifications. After decoding we are left with control data in blanking periods or pixel data in active periods. Since each data channel carries one color, a 24-bit RGB pixel bus is output from the IP.

4.3 EDID ROM (Display Data Channel)

The DDC block emulates a read-only memory containing a default Digilent-branded extended display identification data (EDID). The EDID is defined in the dgl_dvi_edid.txt file included with the IP (see section 6.3). Upon synthesis, this file gets incorporated in the netlist. Modifying this file is allowed.

5 Port Descriptions

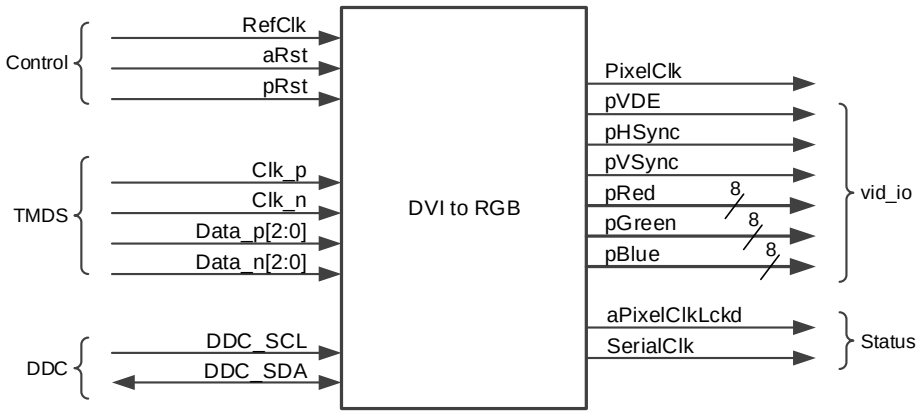

Figure 3. IP top-level diagram.

The signals of the DVI to VGA Core are listed and described in Table 1.

Signal Name	Interface	Signal Type	Init State	Description
RefClk	-	I	N/A	200 MHz reference clock.
aRst(_n)	-	I	N/A	Asynchronous reset of configurable polarity. Assert, if RefClk is not within spec.
pRst(_n)	-	I	N/A	Active-high reset synchronous with PixelClk. Configurable polarity.
Clk_p/Clk_n	tmds	I	N/A	DVI Clock Channel.
Data_p[2:0]/Data_n[2:0]	tmds	I	N/A	DVI Data Channel 0.
SCL	iic	I	N/A	Display Data Channel clock signal. Optional.
SDA	iic	IO	HiZ	Display Data Channel data signal. Optional.
PixelClk	rgb	O	N/A	Pixel clock recovered from the TMDS clock channel. Driven by BUFR.
pVDE	rgb	O	0	Video data valid: • 1 = Active video. • 0 = Blanking period.
pHSync	rgb	O	0	Horizontal synchronization video timing signal.
pVSync	rgb	O	0	Vertical synchronization video timing signal.
pRed (7:0)	rgb	O	zeros	Video red color data.
pGreen (7:0)	rgb	O	zeros	Video green color data.

pBlue (7:0)	rgb	O	zeros	Video blue color data.
aPixelClkLckd	-	O	0	Active-high asynchronous locked signal for PixelClk. When low, PixelClk is lost or not within specs.
SerialClk	-	O	N/A	Fast clock, toggling at five times the frequency of PixelClk. It is used internally for deserialization. Provided optionally as output for advanced use, like clocking an RGB2DVI core. Driven by BUFIO.

Table 1. Port descriptions.

6 Designing with the core

6.1 Customization

The IP provides three customizable parameters: the polarity of reset signals, the availability of the DDC channel, and the serial clock output.

Enabling the DDC channel and serial clock will add the respective ports to the IP and are available to user logic.

6.2 Using SerialClk

While the fast serial clock is normally only used internally for deserialization, it is available and useful in other limited circumstances. The limitations arise from the fact that this clock has a frequency usually too large (five-times the PixelClk frequency) to clock user logic. For example, 1080p will result in a 742.5 MHz SerialClk. A BUFIO primitive drives this clock net to accommodate the large frequencies, which can only clock the I/O column in the same bank/clock region.

One case would be to use this clock to drive the Digilent RGB2DVI core, sharing the clocking logic between the two cores.

6.3 Default EDID

The file named dgl_div_edid.txt contains 128 bytes of EDI data in a format readable by Vivado synthesis. Each line has exactly one byte in binary format. Byte 0 is the first in the file. The following table summarizes the EDID in a human-readable format.

Monitor	
Model name	Digilent DVI
Manufacturer	DGL
Plug and Play ID	DGL0000
Serial number	n/a
Manufacture date	2014, ISO week 5
Filter driver	None
EDID revision	1.4
Input signal type	Digital (DVI)
Color bit depth	8 bits per primary color
Color encoding formats	RGB 4:4:4
Screen size	510 x 290 mm (23.1 in)
Power management	Not supported
Extension blocs	None
DDC/CI	n/a

Color characteristics	
Default color space	Non-sRGB
Display gamma	2.20
Red chromaticity	Rx 0.640 - Ry 0.330
Green chromaticity	Gx 0.276 - Gy 0.594
Blue chromaticity	Bx 0.145 - By 0.060
White point (default)	Wx 0.283 - Wy 0.297
Additional descriptors	None

Timing characteristics	
GTF standard	Not supported
Additional descriptors	None
Preferred timing	No
Detailed timing #1	1280x1024p at 60Hz (5:4)
Standard timings	640 x 480p at 60Hz - IBM VGA 800 x 600p at 60Hz - VESA 1024 x 768p at 60Hz - VESA 1680 x 1050p at 60Hz - VESA STD 1920 x 1080p at 60Hz - VESA STD 1280 x 1024p at 60Hz - VESA STD

Table 2. Default EDID.

7 References

The following documents provide additional information on the subjects discussed:

1. Xilinx Inc., *UG471: 7 Series FPGAs SelectIO Resources*, v1.3, October 31, 2012.
2. Xilinx Inc., *UG472: 7 Series FPGAs Clocking Resources*, v1.6, October 2, 2012.
3. Xilinx Inc., *XAPP460: Video Connectivity Using TMDS I/O in Spartan-3A FPGAs*, V1.1, June 24, 2011.
4. Xilinx Inc., *XAPP495: Implementing a TMDS Video Interface in the Spartan-6 FPGA*, v1.0, December 13, 2010.
5. Xilinx Inc., *WP249: SPI-4.2 Dynamic Phase Alignment*, v1.3, July 6, 2011.
6. DDWG: *Digital Visual Interface DVI*, Revision 1.0, April 2, 1999.