Hardware User Guide

ZIPcores Xilinx® Spartan-6 HD-Video Development Board

ZIP-HDV-001 Rev. B
February 2014

Table of Contents

Overview

Introduction

The Zipcores HD-Video development board is a flexible video-processing platform designed for use in both High Definition (HD) and Standard Definition (SD) video applications. The board is based on a Xilinx Spartan-6 FPGA with a number of supporting ICs for the reception and transmission of different video formats. On the input side, the board features 2 x DVI (HDMI), 2 x 3G/HD-SDI, 2 x CVBS (PAL/NTSC) and 2 x Analogue (RGB or YPbPr) ports. On the output side the board features 2 x DVI (HDMI) and 2 x 3G/HD-SDI ports. In addition the board has a set of LVDS or general purpose differential pins that can be used for interfacing to LVDS-style monitors, cameras and displays. Figure (1) shows the general board layout and distribution of board components.

Board layout

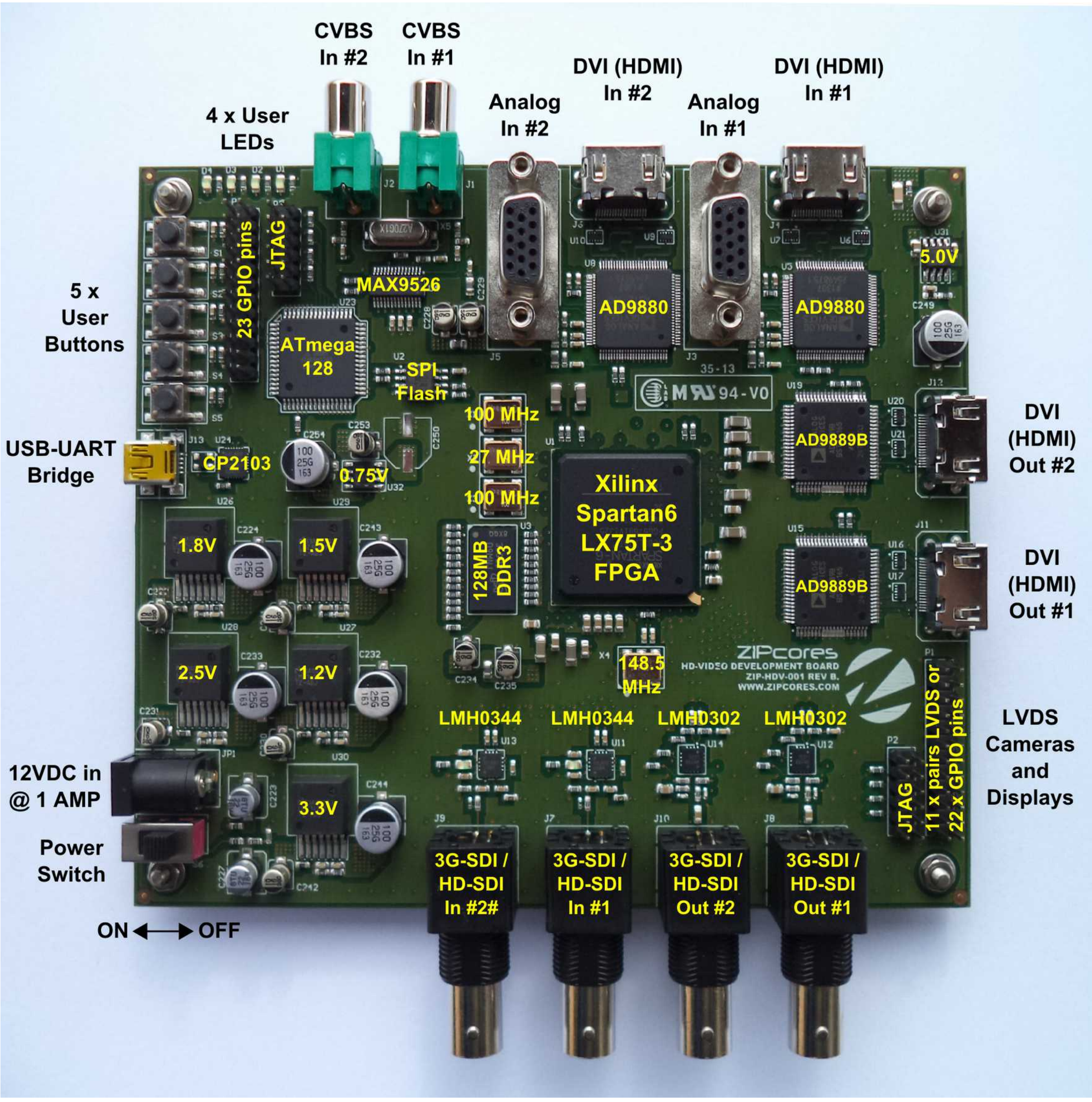


Figure 1: ZIP-HDV-001 Rev B. board

Key features

- Xilinx® Spartan-6 XC6SLX75T-3FGG676 FPGA
- Compatible with free Xilinx ISE WebPACK™ software
- 2 x DVI video inputs (HDMI type-A connector)
- 2 x 3G or HD-SDI serial video inputs (75Ω BNC)
- 2 x Analogue (RGB or YPbPr) inputs (15-pin D-SUB)
- 2 x CVBS (PAL/NTSC) analogue inputs (RCA connector)
- 2 x DVI video outputs (HDMI type-A connector)
- 2 x 3G or HD-SDI serial video outputs (75Ω BNC)
- Support for VESA and CEA-861-D video modes
- Resolutions up to 1920x1080 @ 60 Hz refresh
- 128M x 16-bit DDR3 Memory @ 400MHz (12.8 Gb/s B/W)
- 32MB SPI flash memory for FPGA configuration
- FPGA JTAG programming header
- 24-pin header with 11 pairs of LVDS I/O or 22 GPIO pins
- On-board ATmega128 8-bit AVR Micro-controller
- Micro-controller JTAG programming header
- 24-pin header for ATmega128 GPIO
- USB-UART bridge for simple PC-based control
- 2 x 100 MHz, 1 x 27 MHz and 1 x 148.5 MHz oscillators
- 4 x general purpose LEDs
- 5 x general purpose push-buttons
- Robust SPDT power switch
- 12V DC universal power supply included
- 4 x mounting holes (size M5)

Block diagram

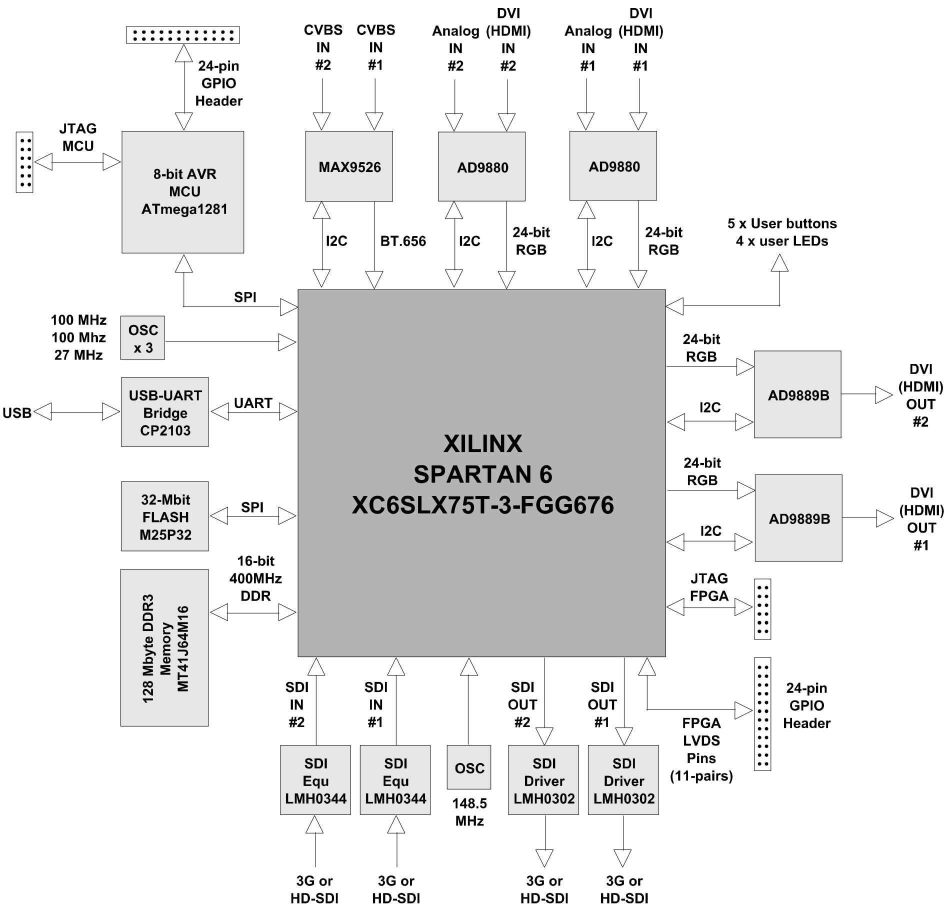


Figure 2: Block diagram showing FPGA connectivity with main board components

Detailed description

Spartan-6 FPGA

The main FPGA is a Xilinx® Spartan-6 XC6SLX75T, FGG676, -3 speed grade device. It is a low-power (1.2V core-voltage) FPGA optimized for applications requiring DSP, embedded memory and high-speed serial connectivity. Full details can be found from the Xilinx website. A brief summary of the device resources is listed below:

- Slices 11,662
- Logic cells 74,637
- Flip-flops 93,296
- Distributed RAM (kbits) 692
- Block RAM (18 kbits each) 172
- DSP blocks (DSP48A1) 132
- Digital Clock Managers (DCM) 12
- Phase Locked Loops (PLL_ADV) 6
- Memory Controller Blocks (MCB) 4
- GTP Transceivers (GTPA1_DUAL) 8
- Maximum single-ended pins 348
- Maximum differential pairs 174
- Configuration memory (Mbit) 19.6

General purpose user I/O and LVDS pins

The board features a header (P1) with 24 pins connected to a set of LVDS pins on the FPGA. These pins are arranged as LVDS pairs that can be configured as either inputs or outputs. They can also be used for general purpose user I/O. Note that the 100Ω termination resistors for the LVDS inputs are not included on the board. If an external 100Ω termination is not used, then a termination resistor can be specified in the FPGA pads using the 'DIFF_TERM' attribute in the UCF constraints file. Note that as well as the LVDS 2.5V standard, the Spartan-6 FPGA can support many other differential and single-ended I/O formats. Please refer to the Spartan-6 I/O user guide for more details.

P1

(G13)	**LVDS_33N**	1	2	**LVDS_33P**	**(H12)**
(D13)	**LVDS_34N**	3	4	**LVDS_34P**	**(E13)**
(A13)	**LVDS_35N**	5	6	**LVDS_35P**	**(C13)**
(A12)	**LVDS_36N**	7	8	**LVDS_36P**	**(B12)**
(A14)	**LVDS_37N**	9	10	**LVDS_37P**	**(B14)**
(J12)	**LVDS_38N**	11	12	**LVDS_38P**	**(K12)**
(H15)	**LVDS_43N**	13	14	**LVDS_43P**	**(J15)**
(J17)	**LVDS_48N**	15	16	**LVDS_48P**	**(J16)**
(F17)	**LVDS_56N**	17	18	**LVDS_56P**	**(G16)**
(E20)	**LVDS_57N**	19	20	**LVDS_57P**	**(F20)**
(G17)	**LVDS_58N**	21	22	**LVDS_58P**	**(H17)**
	GND	23	24	**GND**	

Figure 3: FPGA LVDS and/or GPIO pins

General purpose user buttons and LEDs

There are five general purpose user buttons labelled S1 to S5 on the board. Each one is a single push-to-make switch with a 4K7 pull-up resistor to the 3.3V supply. Button S1 is designated as a main system reset with the other buttons designed for general purpose user input. All inputs are active low into the FPGA.

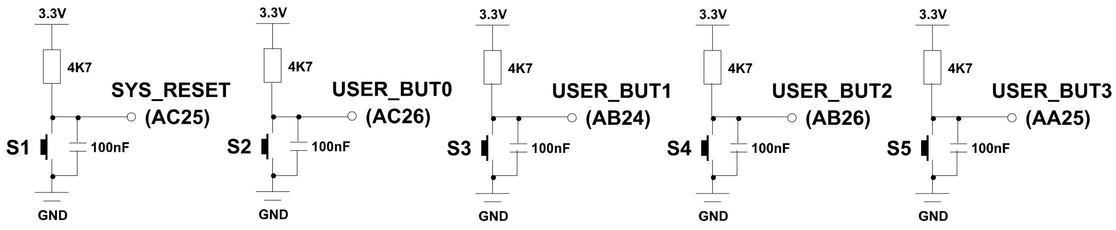


Figure 4: User buttons

The board also features four green surface mount LEDs. These LEDs are connected to the FPGA via a 330Ω resistor in series to ground.

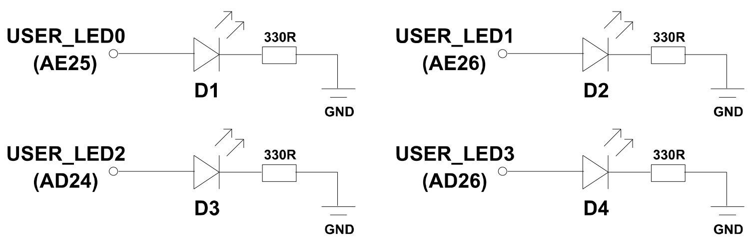


Figure 5: User LEDs

User clocks and oscillators

In total there are 4 oscillators that provide the clock inputs into the FPGA. Of these, 3 clocks are single-ended clocks at LVCMOS 3.3V voltage levels. The other is a 148.5 MHz LVDS clock that is discussed in the SDI clocking section. The single-ended oscillators provide 2 x 100 MHz and 1 x 27 MHz sources that may be used for general purpose clocking. The 27 MHz source is especially useful for video applications because it can be used to generate most pixel clock frequencies. This is typically done using the DCM and/or PLL resources inside the FPGA. The 400MHz clock for the external memory may also be generated in a similar way by multiplying the 100MHz DDR3 external clock source.

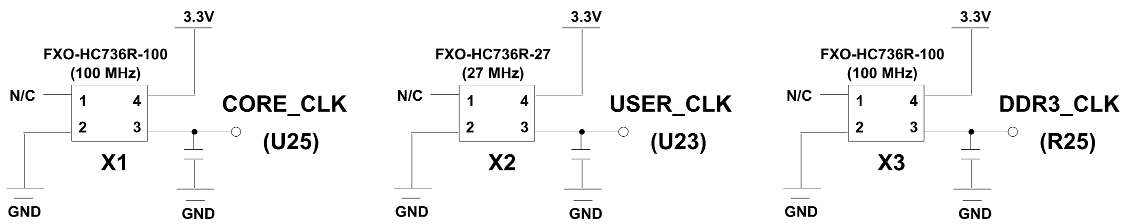


Figure 6: Single-ended user clocks

USB-UART bridge

A USB port is provided for serial connectivity with the FPGA at speeds of up to ~1Mbit/s. The USB port is actually a USB-UART bridge and is implemented using the CP2103 component from Silicon Labs. This component effectively makes the USB port look like a standard UART adapter to the host interface. For example, if this were a PC then the USB port would look like a serial COM port device.

In order to communicate with the CP2013 via USB, then the host computer must first install the appropriate driver software from Silicon Labs. This software is available for both Windows, Linux and MAC OSX and can be found on the Silicon Labs website at: www.silabs.com. The UART_RX and UART_TX signals connect directly to the FPGA pins and operate at 3.3V voltage levels.

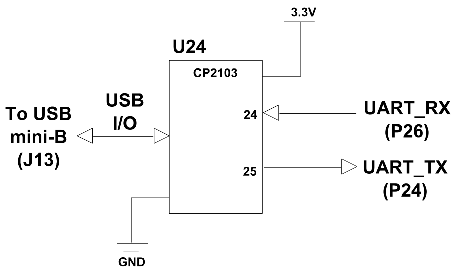


Figure 7: USB-UART bridge circuit

SPI flash memory

The SPI flash memory is a 32-Mbit M25P32 device from Numonyx. It features a 75 MHz SPI bus interface and is used as a non-volatile store for the FPGA configuration program. On power-up, the FPGA will load the program from the SPI flash memory by default.

Programming of the SPI memory is done via the JTAG programming port. More details on how to do this are given in the JTAG configuration section and also in Appendix A at the end of the document. Figure (8) shows the basic circuit with the dedicated SPI signals that connect to the the FPGA.

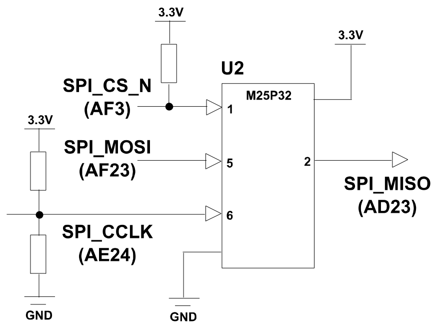


Figure 8: SPI flash memory circuit

DDR3 SDRAM memory

The board features a Micron DDR3 SDRAM memory part number: MT41J64M16JT-15E. The main function of the DDR3 memory is to provide a high-bandwidth external store for a video frame buffer. The memory is organized as 8M x 16-bit x 8 banks giving a total capacity of 128Mbytes. The memory interface with the FPGA has a nominal clock frequency of 400MHz permitting a peak data transfer rate of 400MHz x 16-bit x 2 or 12.8 Gbits/s.

The Spartan-6 FPGA features an embedded Memory Controller Block (MCB) that is compatible with the DDR3 memory component. In addition, Xilinx ISE comes with a Memory Interface Generator (MIG) tool that offers a simple platform for generating the MCB source code and associated design constraints. Appendix B gives a worked example of how to implement the memory interface using the Xilinx MIG tool.

ATmega128 general purpose Micro-controller

The ATmega128 is a versatile, general purpose, 8-bit AVR micro-controller with 128kbytes of embedded flash memory for user-program and data. Most of the MCU pins are connected to header P4 as shown in Figure (9) below. These pins can be used as general purpose digital I/O, or alternatively, they can be used for a variety of dedicated functions such as: UART, I2C, SPI, Analogue-to-digital conversion, external interrupts, counters and PWM outputs. In addition, the MCU also has an SPI interface with the FPGA which allows for serial communications at rates of up to 8Mbits/s. All pins operate at 3.3V levels.

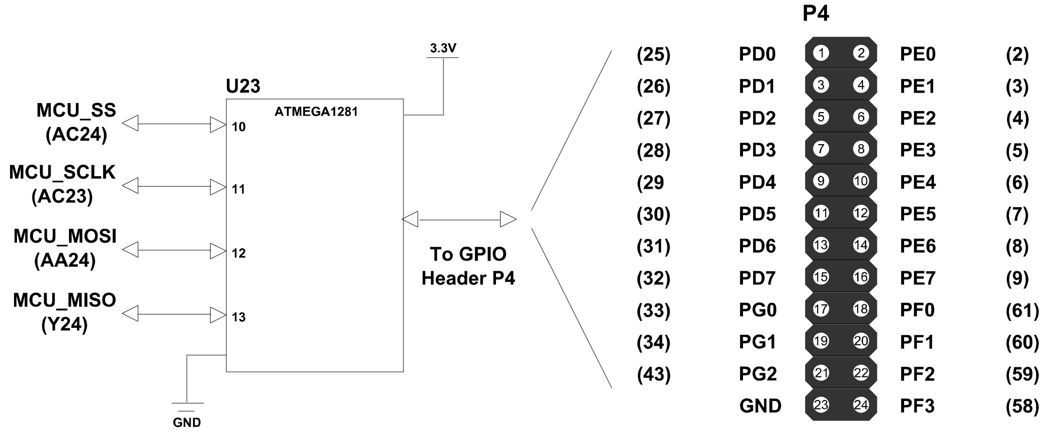


Figure 9: ATmega128 GPIO pins and SPI connectivity with the FPGA

CVBS (PAL/NTSC) video inputs

Two multiplexed CVBS inputs are are provided by the MAX9526 video decoder IC. The IC is capable of decoding standard PAL and NTSC video formats automatically without any user intervention. In default operation, the decoder will lock to video input #1. The I2C configuration registers may be programmed to select input video #2 or automatically lock to whichever input is present.

On detection of a valid video input, the IC generates a standard 10-bit BT.656 video stream that is synchronous with the 27 MHz pixel clock. The MAX9526 has various configuration registers that may be accessed via a dedicated I2C port. The I2C register address is 0x42 for a write and 0x43 for a register read. Figure (10) below describes the basic CVBS decoder circuit and connectivity with the FPGA.

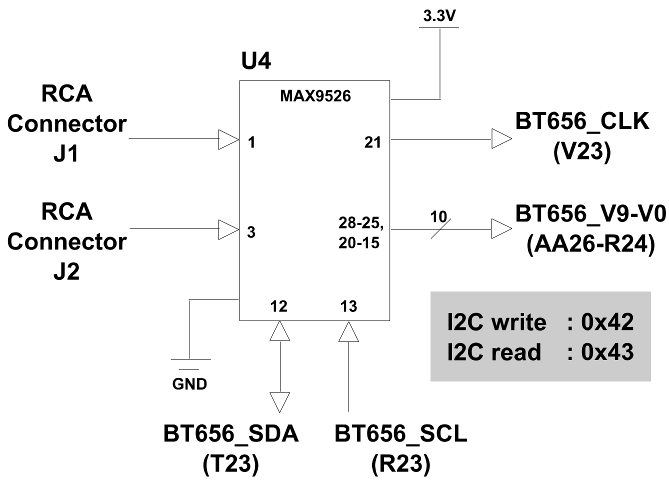


Figure 10: MAX9526 video decoder

Analogue (RGB/YPbPr) video inputs

The analogue video inputs enter the board via a pair of 15-pin D-SUB 'VGA' style connectors. Both RGB graphics and YPbPr component video formats are supported by the AD9880 receiver IC. The receiver has a high-precision internal PLL which allows it to lock to the incoming video signal. This PLL must be configured using the I2C registers to correspond to the desired video mode. The output video timing parameters must also be correctly programmed using I2C in order to position the video frame for display. In addition, the AD9880 provides a wide range of registers to adjust the colours, brightness and general appearance of the sampled video. Analogue video modes with pixel clock frequencies of up to 135 MHz are supported. This generally equates to a maximum resolution of 1280 x 1024 pixels at 75Hz refresh according to the VESA specification.

The digitally sampled video is output to the FPGA via a 24-bit RGB port. This data is synchronous with the output clock and is accompanied by a standard VSYNC, HSYNC, DE and (optional) FIELD signal. The FIELD signal is only active for interlaced video modes. Figure (11) below shows the connectivity between the AD9880, the video inputs and the FPGA.

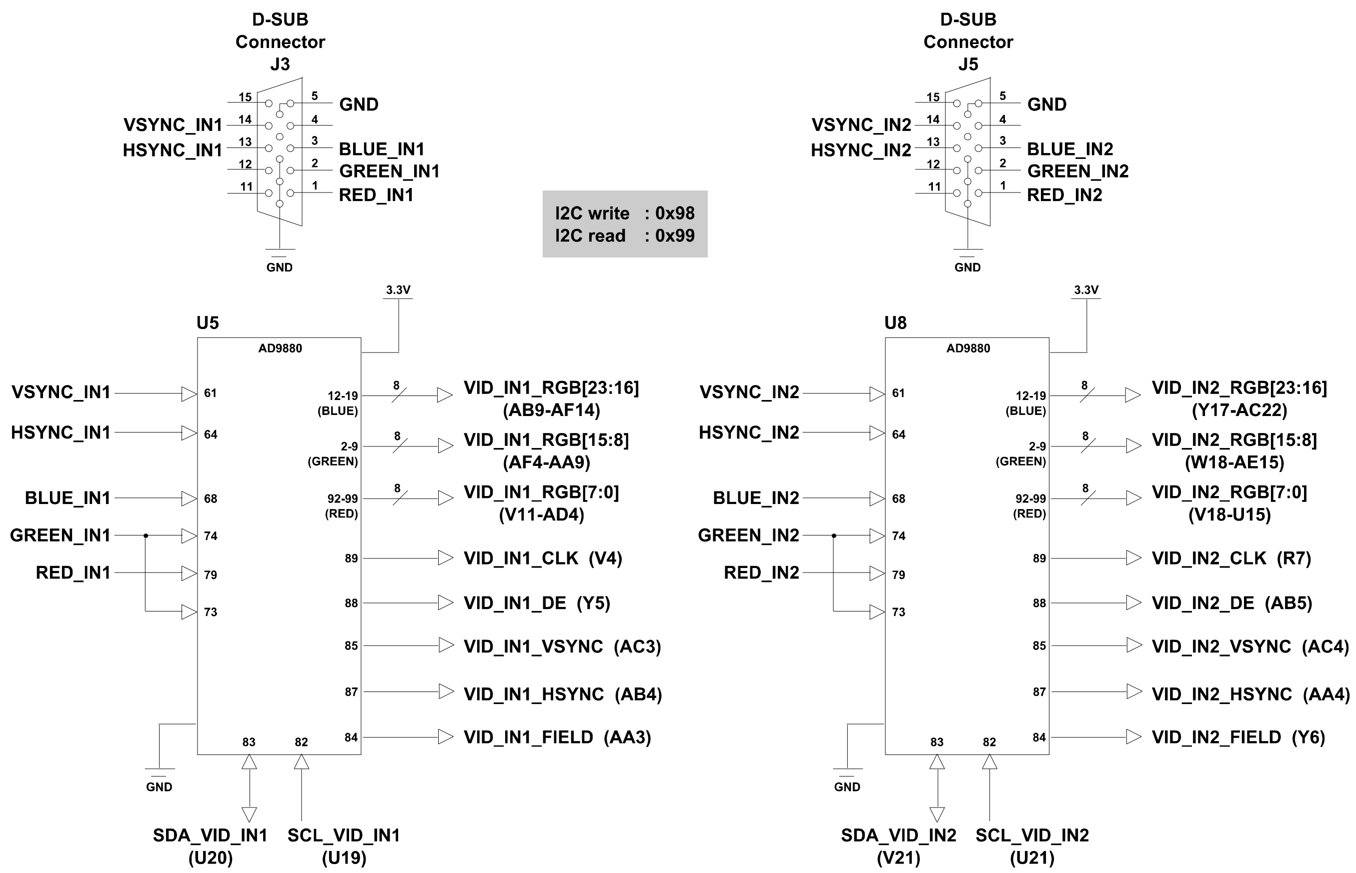


Figure 11: AD9880 Analogue receivers

DVI (HDMI) video inputs

The digital video inputs enter the board via a pair of HDMI type-A connectors and pass directly to a pair of AD9880 receiver ICs. The input video format supported is DVI or *non-encrypted*[1] HDMI with a maximum pixel clock frequency of up to 150 MHz. As a general rule, most standard VESA and CEA-861-D modes are accommodated up to a maximum resolution of 1920 x 1080 pixels and a refresh rate of 60Hz[2]. A full list of supported formats is provided in the AD9880 documentation.

In order to receive a digital video signal, the AD9880 TMDS PLL must first be configured correctly for the desired video mode. An example I2C register configuration is given in Appendix D of this document.

As with the analogue video modes, the video outputs from the AD9880 follow a similar format. The video data is output to the FPGA via a 24-bit RGB port. This data is synchronous with the output clock and is accompanied by a standard VSYNC, HSYNC, DE and FIELD signal. The FIELD signal is only active for interlaced video. The connectivity between the HDMI connectors, the AD9880 and the FPGA is shown in Figure 12 below.

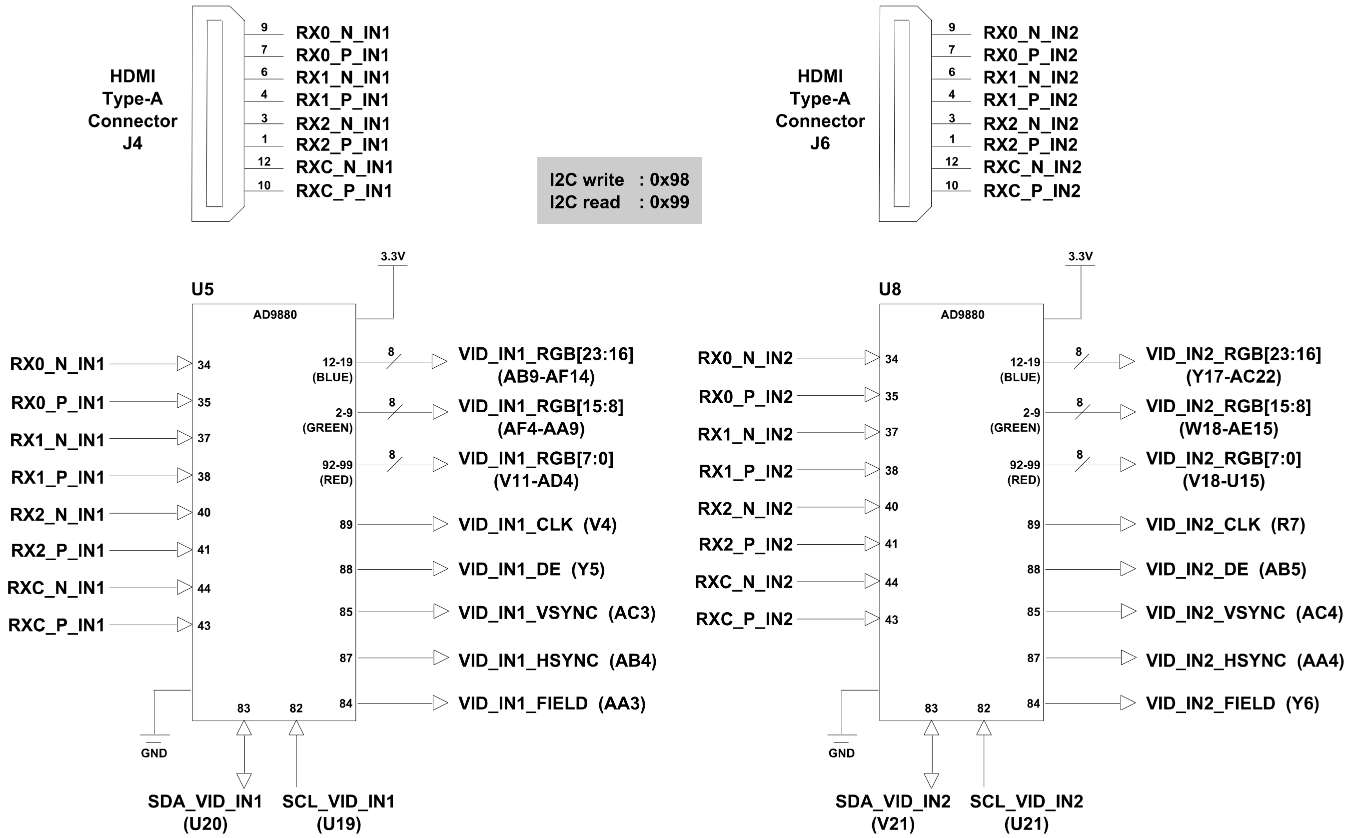


Figure 12: AD9880 HDMI receivers

1 Note that the development board does not implement High-Bandwidth Digital Content Protection (HDCP) and therefore does not support the reception (or transmission) of encrypted video sources.

2 The maximum attainable pixel clock frequency will also be determined by the maximum frequency of the FPGA circuit. As such, the static timing of the FPGA core logic and I/O must also be taken into consideration when determining the highest possible video display resolution and frame refresh rate.

DVI (HDMI) video outputs

The digital video outputs are provided by a pair of AD9889B transmitter ICs that are routed to a pair of type-A HDMI output connectors. The FPGA provides a standard 24-bit RGB data path to the transmitter with accompanying VSYNC, HSYNC, DE and CLK signals. By default, the AD9889B assumes 24-bit RGB pixels that are sampled on the rising edge of the pixel clock.

The AD9889B is capable of generating an output video signal with very little register set-up. All the video timing information is automatically derived from the input VSYNC, HSYNC and DE signals. Unless the user wishes to change any of the default settings, the only requirement is to power up the IC and set the drive strength of the outputs depending on the output pixel clock frequency. An example of how to do this is shown in Appendix E at the end of the document.

As with the digital video inputs, the output video format supported is DVI or *non-encrypted* HDMI up to 1920 x 1080 pixels resolution at 60Hz refresh rate. Again, most standard VESA and CEA-861-D modes are accommodated but a full list of modes is given in the AD9889B documentation.

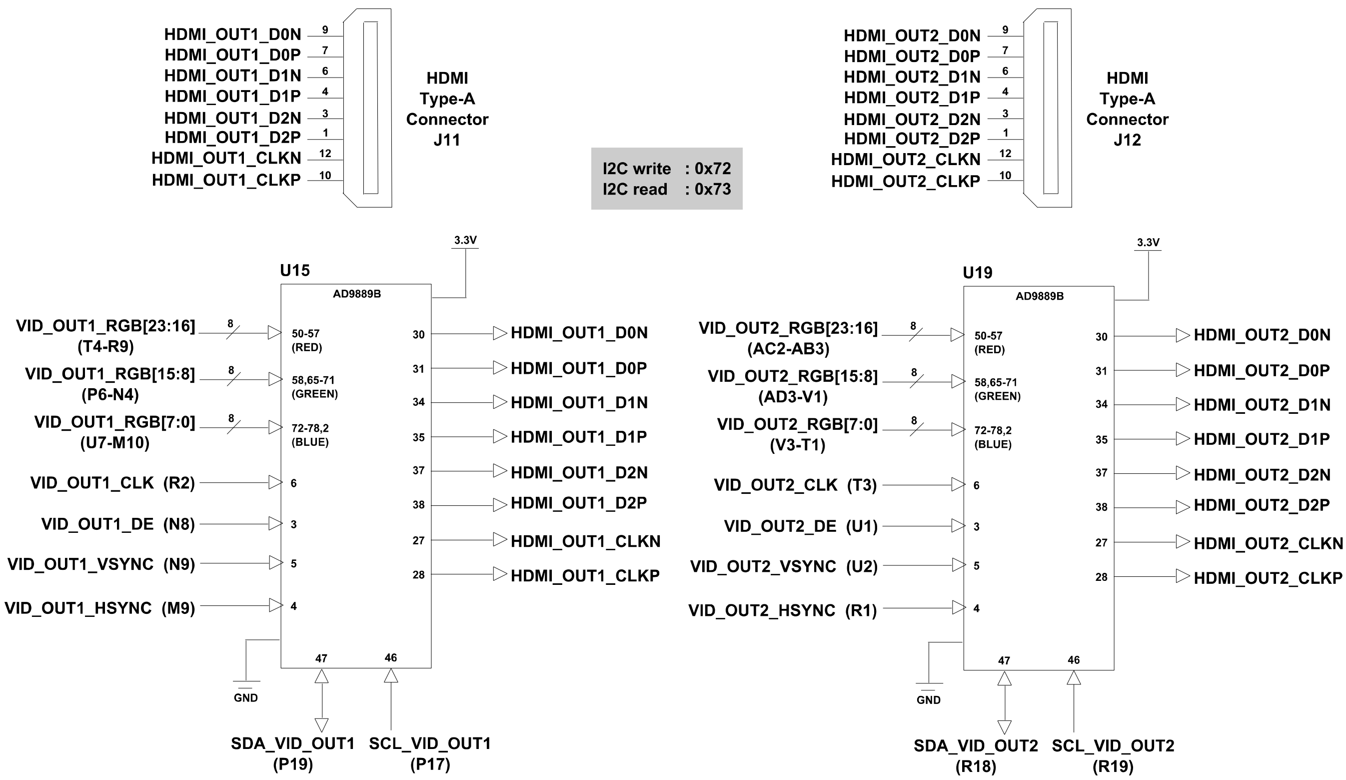


Figure 13: AD9889B HDMI transmitters

3G/HD-SDI serial video inputs

The SDI serial inputs enter the board via a pair of 75Ω BNC connectors. The SDI signals then enter a pair of LMH0344 adaptive cable equalizers that ensure the best quality signal into the FPGA. The SDI serial bitstream is routed via a differential pair to the FPGA where it is then de-serialized into 20-bit parallel video data. The SDI circuit supports data rates of up to 3Gbits/s which permits video resolutions of up to 1920 x 1080 pixels at 60Hz refresh rates. The SMPTE specifications covered are: 424M (3G-SDI), 292M (HD-SDI) and 259M (SD-SDI). The minimum data rate supported is approximately 125 Mbits/s. On reception of a valid SDI input stream, the LMH0244 ICs will assert the 'Carrier Detect' line low. This is indicated by LEDs D5 and D6 being illuminated on the board.

The Spartan-6 FPGA features a number of embedded dual GTP transceivers that permit high-speed de-serialization and serialization of the SDI bitstream. Specifically, the SDI inputs on the board are routed to the GTP transceiver located at site 'X0Y1' on the FPGA. These GTP signals are given the names 'MGT****101' in the Xilinx documentation. In order to successfully de-serialize the bitstream, it's important to configure the GTP transceivers correctly. Please refer to Appendix C for a worked example of how to do this using the Xilinx tools.

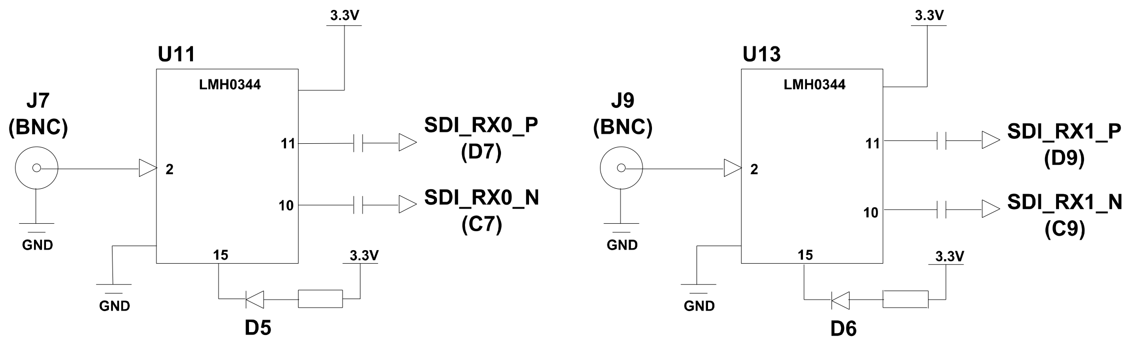


Figure 14: SDI input circuit

3G/HD-SDI serial video outputs

In board also features a pair of SDI video outputs that are connected to the same dual GTP Transceiver as the SDI inputs (site X0Y1). The transceiver permits the serialization of a 20-bit parallel video data signal into a differential bitstream. This bitstream is in turn passed to the LMH0302 line driver ICs for output to the 75Ω BNC connectors. Again, both 3G, HD and SD SMPTE specifications are supported by the SDI outputs. As with the SDI inputs, Appendix C gives a worked example of how to configure the GTP transceivers for 3G-SDI operation.

In addition, there are two separate control signals from the FPGA to the line drivers. These are the TX_EN signal and the TX_SEL signals. The TX_EN line is active high (low by default) and enables the driver outputs. The signal TX_SEL selects between HD and SD modes. By default the signal is pulled low which signifies HD operation.

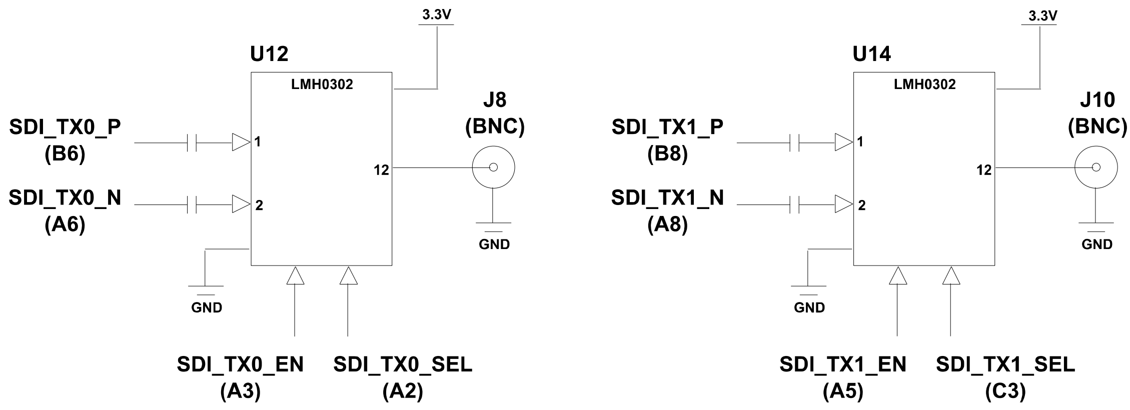


Figure 15: SDI output circuit

SDI clocking

The GTP transceivers inside the FPGA require a precise external clock source for the PLL to successfully lock to the incoming bitstream. To satisfy this requirement, the board comes with an external 148.5MHz LVDS clock source that connects to the MGTREFCLK0N_101 and MGTREFCLK0P_101 clock inputs of the FPGA. Figure (16) shows the LVDS clock circuit in more detail.

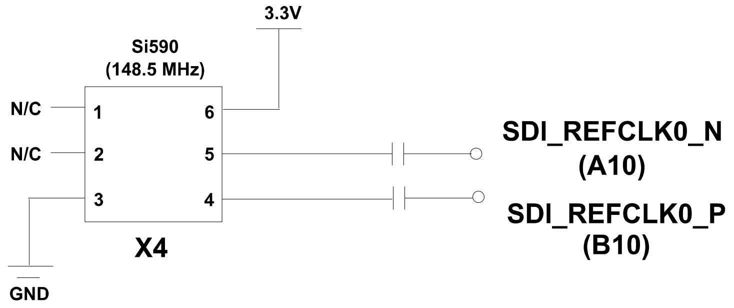


Figure 16: 148.5 MHz LVDS clock

JTAG Configuration

FPGA configuration

The configuration of the Spartan-6 FPGA is done via JTAG programming header P2. Programming of the FPGA is done using the 'Xilinx Platform Cable USB II' and the Impact programming software that comes free with Xilinx ISE WebPACK™. Xilinx WebPACK may be downloaded from the Xilinx website at www.xilinx.com. The pins on the programming cable should be connected to the board as described below in Figure (17). Appendix A and F show additional examples of how to program the FPGA with the user generated bitfile. It also demonstrates how to program the SPI configuration flash. Note that before programming, make sure that the board is powered up and the programming voltage is present, indicated by a green status LED on the USB programmer.

P2

	2.5V	1	2	**GND**
(F21)	**FPGA_TMS**	3	4	**GND**
(A24)	**FPGA_TCK**	5	6	**GND**
(G21)	**FPGA_TDO**	7	8	**GND**
(C23)	**FPGA_TDI**	9	10	**GND**
	GND	11	12	**GND**

Figure 17: FPGA programming set-up using the JTAG header

ATmega128 configuration

The configuration of the ATmega128 micro-controller is done using the dedicated JTAG header P3. There are various JTAG-capable programmers and debuggers available from Atmel. The example shown uses the JTAGICE mkII debugger although at the time of writing, JTAGICE3 is a more up-to-date version. Both programmers/debuggers come with a 'squid' cable that can be wired to the JTAG header as shown in Figure (18) below. Programming the micro-controller is done using the Atmel Studio software that is freely available from the Atmel website at www.atmel.com.

P3

	3.3V	1	2	**GND**
(56)	**AT_TMS**	3	4	**GND**
(57)	**AT_TCK**	5	6	**GND**
(55)	**AT_TDO**	7	8	**GND**
(54)	**AT_TDI**	9	10	**GND**
	GND	11	12	**GND**

Figure 18: ATmega programming set-up using the JTAG header

(Note: The ATmega may also be programmed indirectly via SPI using the AVRISP programmer. However, the ATmega SPI pins: MISO, MOSI, SCK are connected to the FPGA. The FPGA must therefore be configured such that these pins are routed to the external GPIO pins on header P1. In this way the user has 'indirect' access to the ATmega SPI pins for programming).

Power supplies

12V DC mains adapter

The board is shipped with a 12V DC (1.67A) mains adapter part number: PSAA20R-120-R. The adapter comes with an interchangeable AC mains plug that is suitable for USA, UK, EUR and AUST operation. The 12V DC plug into the board is a standard 'Wall Wart' type of 2.1 mm diameter. Figure (19) shows the adapter and interchangeable parts. In order to power the board, insert the 12V DC plug into the socket and slide the power switch to the 'ON' position as shown by the arrow in Figure (1).

Figure 19: Mains plug adapter

Board power supplies

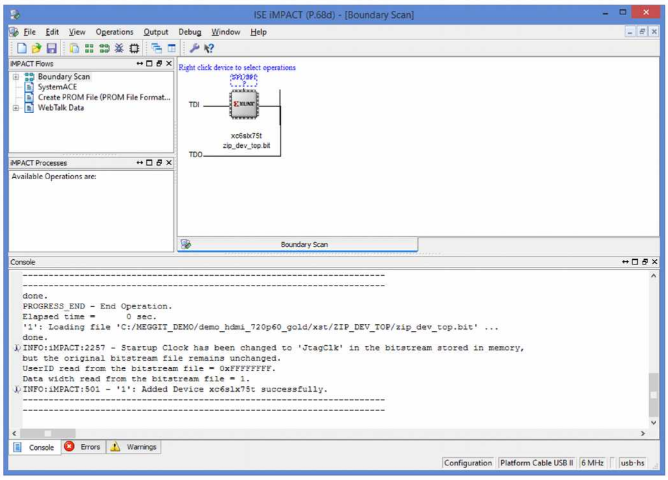
In total there are 7 different power supplies on the board that provide power to the FPGA and external ICs. A description of these power supplies is given in the table below:

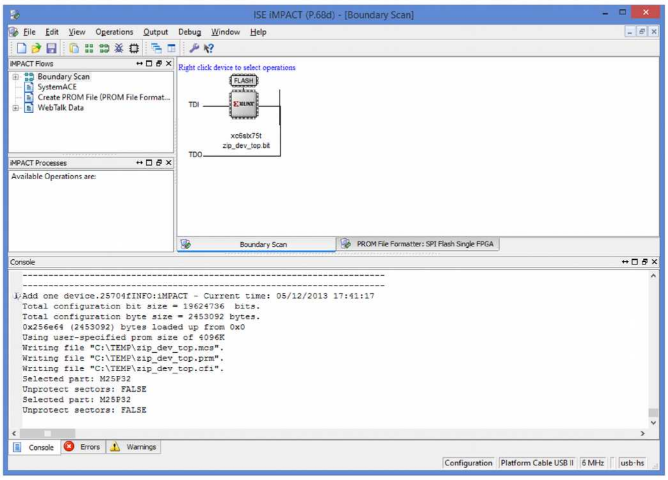
ID	***Part number***	***Supply voltage***	***Used for***
U26	LMZ14201	1.8V @ 1A	MAX9526, AD9880, AD9889B
U27	LMZ14201	1.2V @ 1A	FPGA (VCCINT)
U28	LMZ14201	2.5V @ 1A	FPGA (VCCAUX)
U29	LMZ14201	1.5V @ 1A	FPGA (VCCO), DDR3
U30	LMZ14201	3.3V @ 1A	FPGA (VCCO), Oscillators, SPI Flash, MAX9526, AD9880, LMH0344, LMH0302, ATMEGA1281, CP2103
U31	LP3878	5.0V @ 800mA	HDMI output connector reference voltage
U32	LP2998	0.75V @ 400mA	DDR3 reference voltage

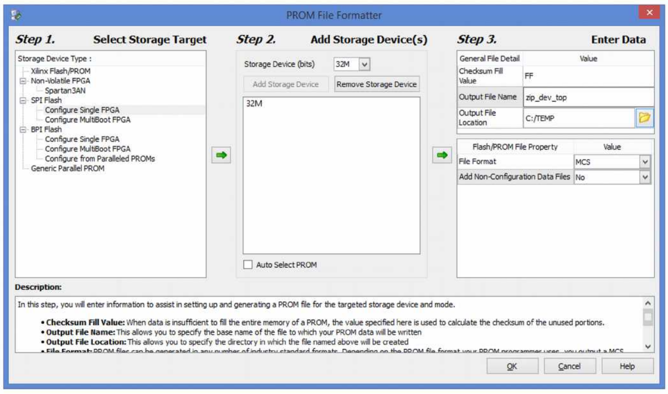
During operation of the board, the peak current drain is between 500-700 mA depending on which ICs are active.

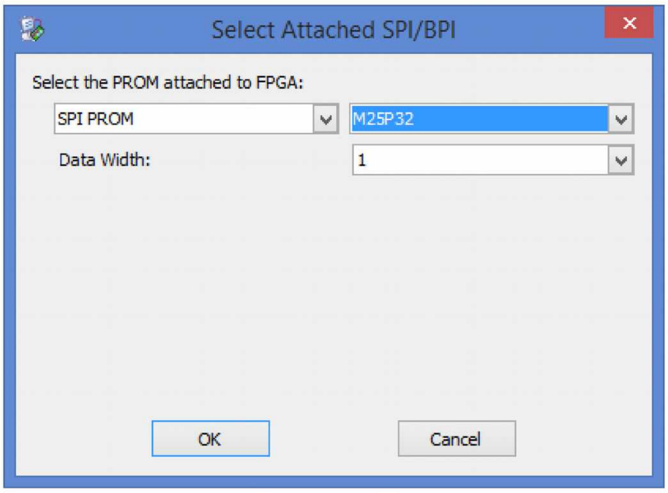
Appendices

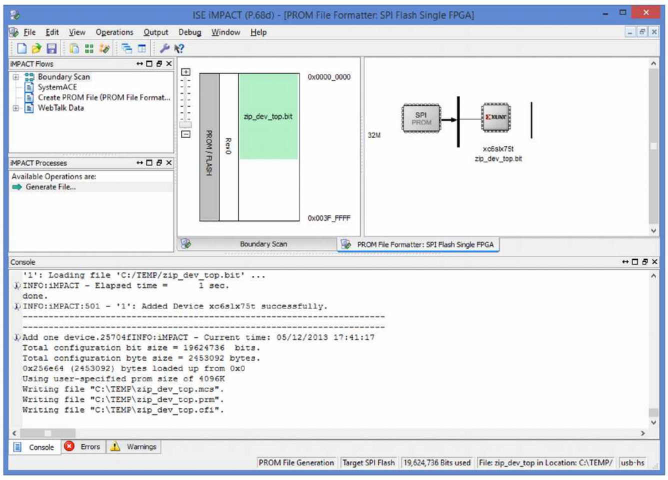
Appendix A: JTAG configuration of the FPGA using the Xilinx® Impact tool



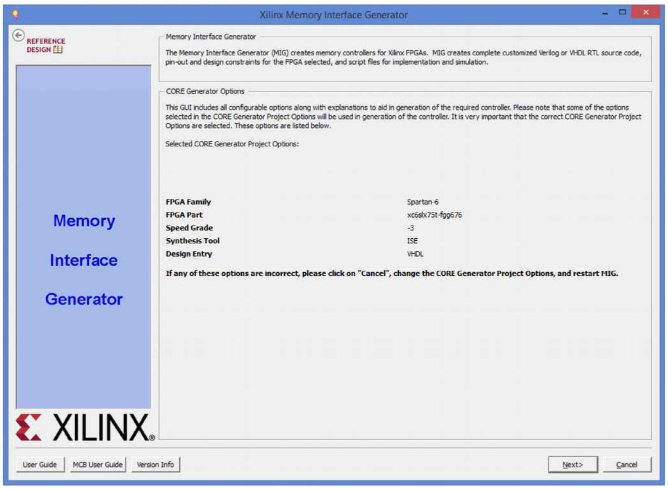


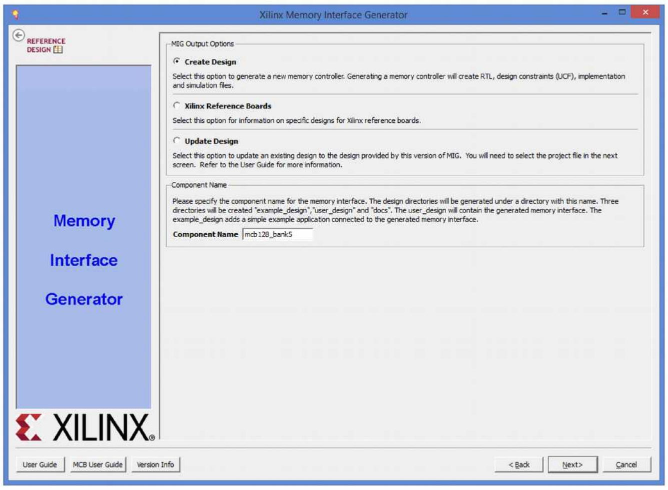


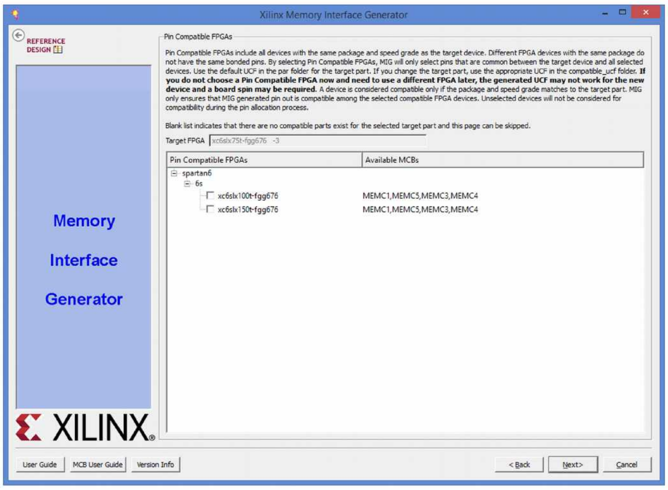


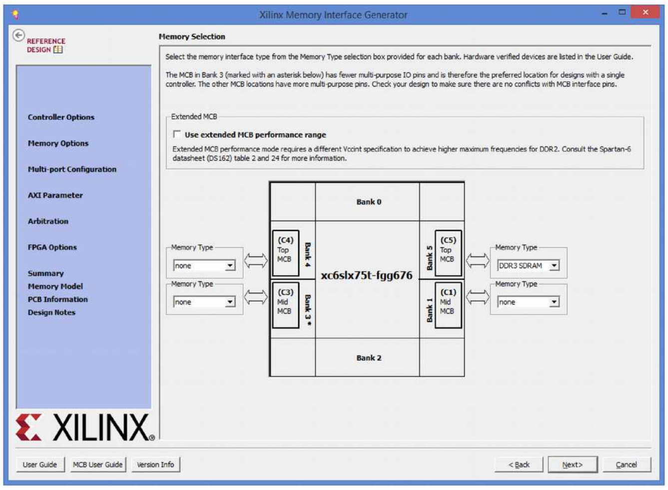


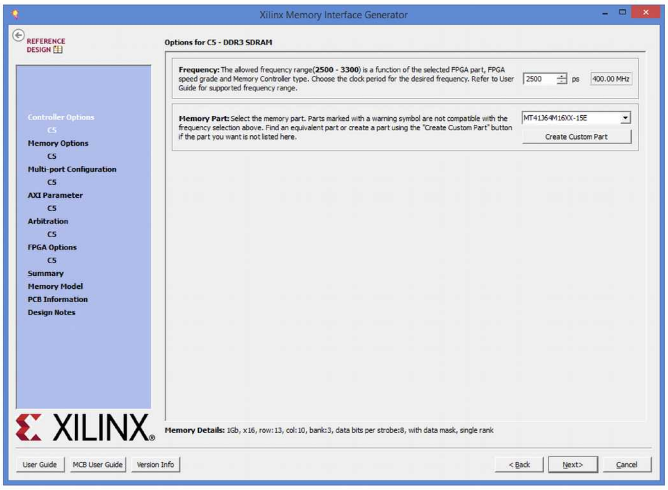
Appendix B: DDR3 memory implementation using the Xilinx® MIG tool

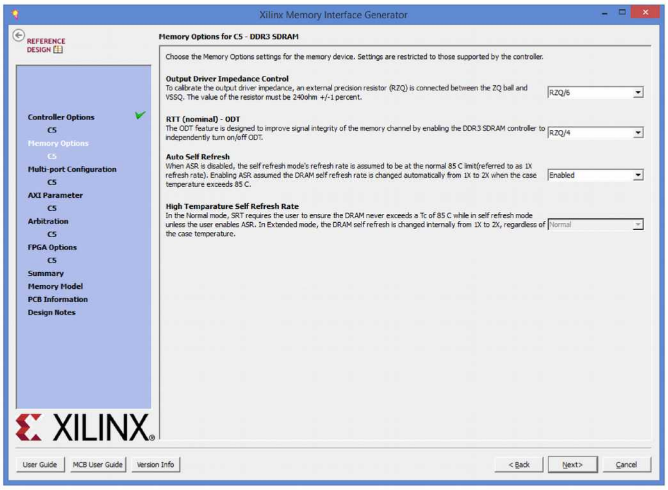


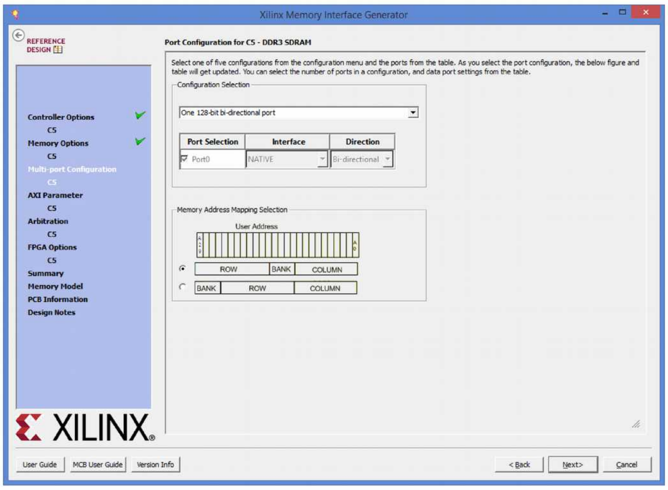


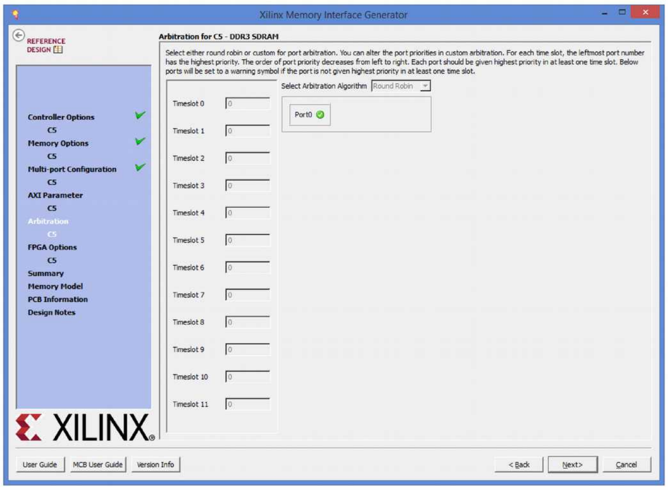


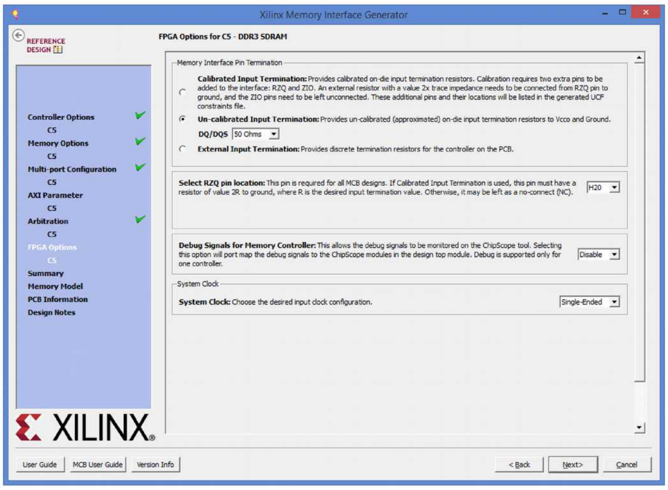


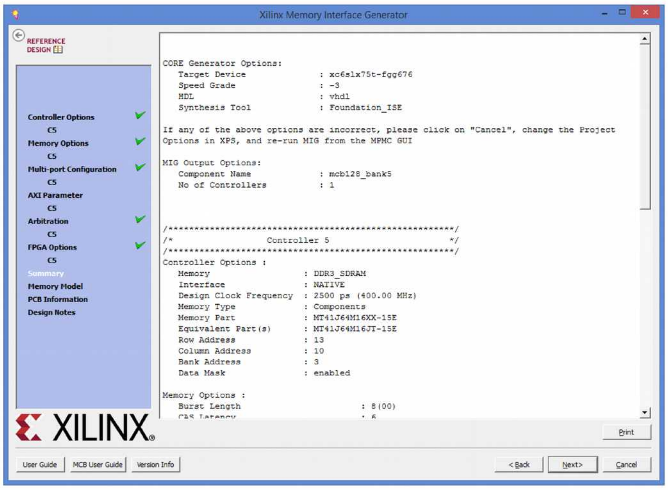


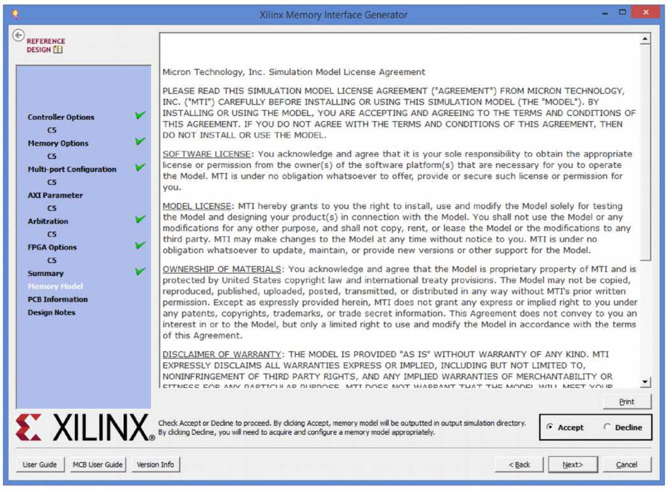


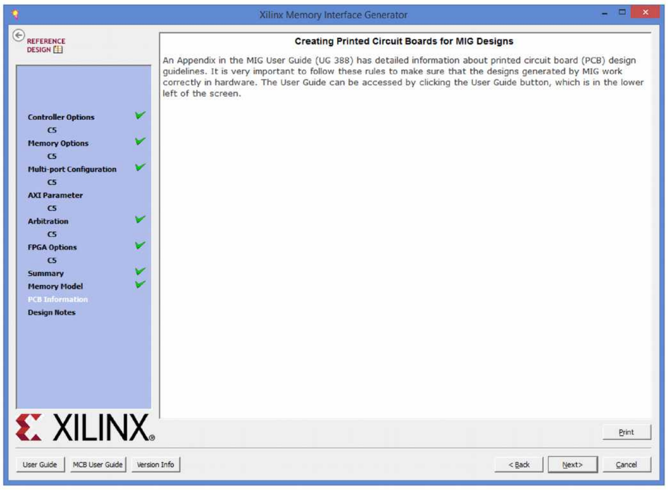

Xilinx Memory Interface Generator
Port Configuration for C5 - DDR3 SDRAM


Xilinx Memory Interface Generator
Arbitration for C5 - DDR3 SDRAM


Xilinx Memory Interface Generator
FPGA Options for C5 - DDR3 SDRAM


Xilinx Memory Interface Generator


Xilinx Memory Interface Generator
Micron Technology, Inc. Simulation Model License Agreement

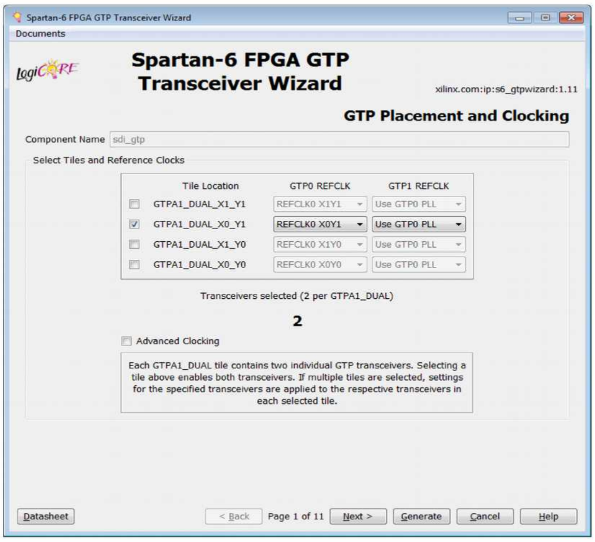

Xilinx Memory Interface Generator
Creating Printed Circuit Boards for MIG Designs

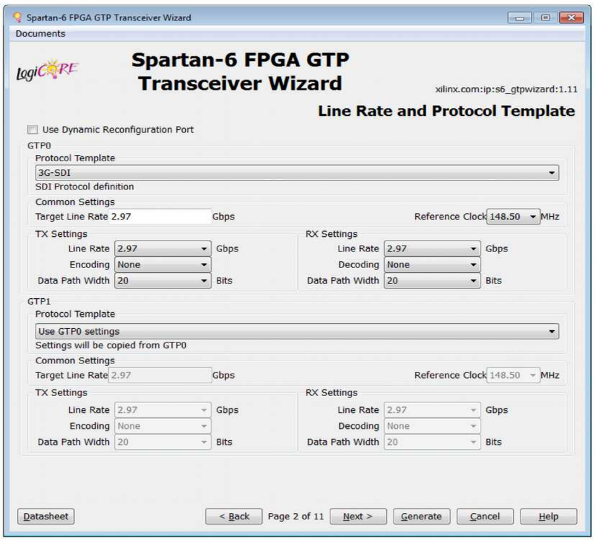
Note: The Xilinx MIG tool assumes that the input DDR3 clock is 400MHz. However, the DDR3 clock on the board is 100 MHz. For this reason, the generic settings in the example generated 'mcb128_bank5.vhd' component must be modified as shown below:

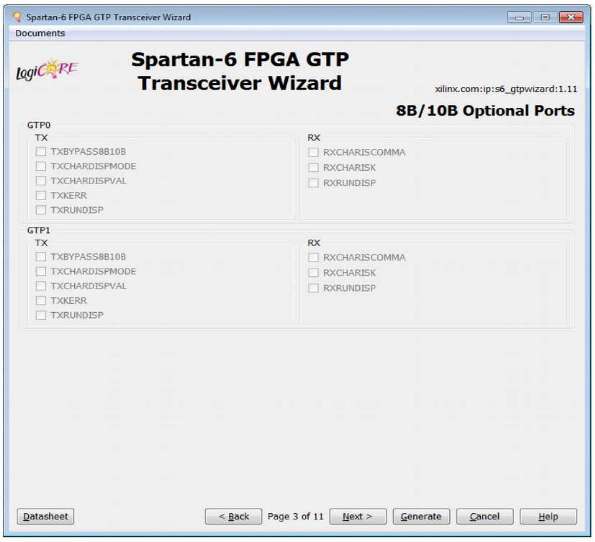
```
-----------------------------------------------------------------------
-- Original code generated by MIG tool assumes a 400 MHz input clock --
-----------------------------------------------------------------------

constant C5_CLKOUT0_DIVIDE       : integer := 1;
constant C5_CLKOUT1_DIVIDE       : integer := 1;
constant C5_CLKOUT2_DIVIDE       : integer := 16;
constant C5_CLKOUT3_DIVIDE       : integer := 8;
constant C5_CLKFBOUT_MULT        : integer := 2;
constant C5_DIVCLK_DIVIDE        : integer := 1;
constant C5_INCLK_PERIOD         : integer := ((C5_MEMCLK_PERIOD * C5_CLKFBOUT_MULT) /
(C5_DIVCLK_DIVIDE * C5_CLKOUT0_DIVIDE * 2));
```

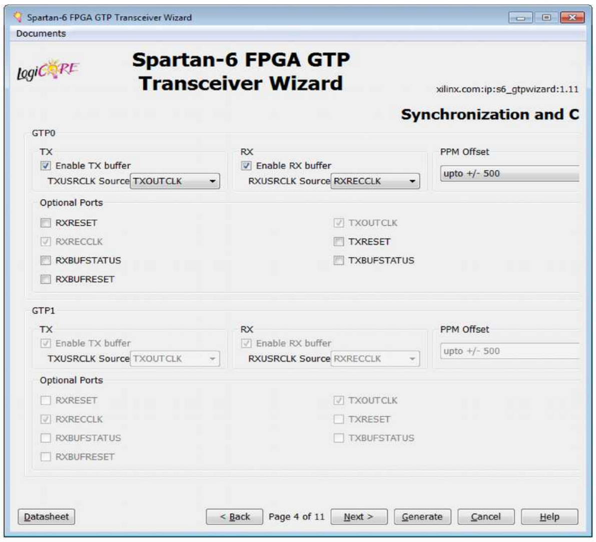
```
---------------------------------------------
-- Modified code for a 100 MHz input clock --
---------------------------------------------

constant C5_CLKOUT0_DIVIDE       : integer := 1;
constant C5_CLKOUT1_DIVIDE       : integer := 1;
constant C5_CLKOUT2_DIVIDE       : integer := 16;
constant C5_CLKOUT3_DIVIDE       : integer := 8;
constant C5_CLKFBOUT_MULT        : integer := 8;
constant C5_DIVCLK_DIVIDE        : integer := 1;
constant C5_INCLK_PERIOD         : integer := ((C5_MEMCLK_PERIOD * C5_CLKFBOUT_MULT) /
(C5_DIVCLK_DIVIDE * C5_CLKOUT0_DIVIDE * 2));
```

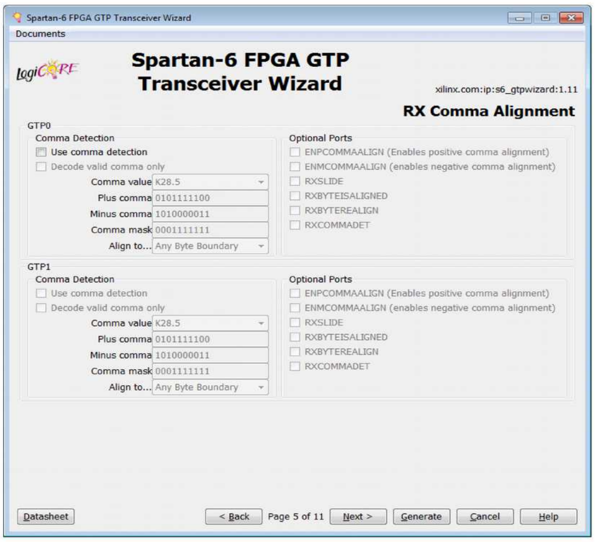
Appendix C: 3G/HD-SDI transceiver implementation using the Xilinx® GTP Transceiver Wizard

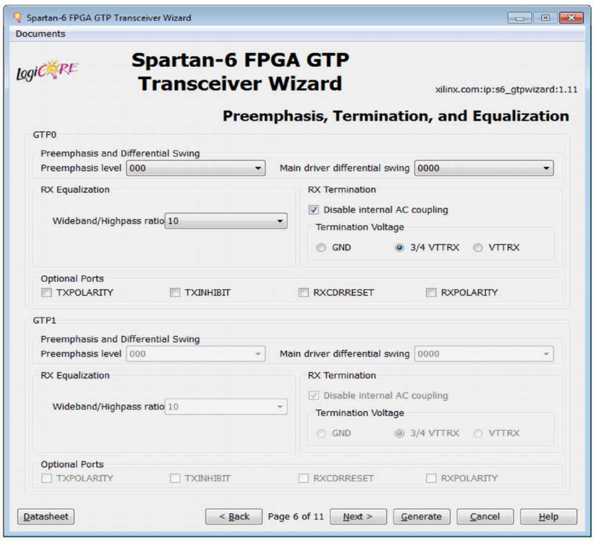


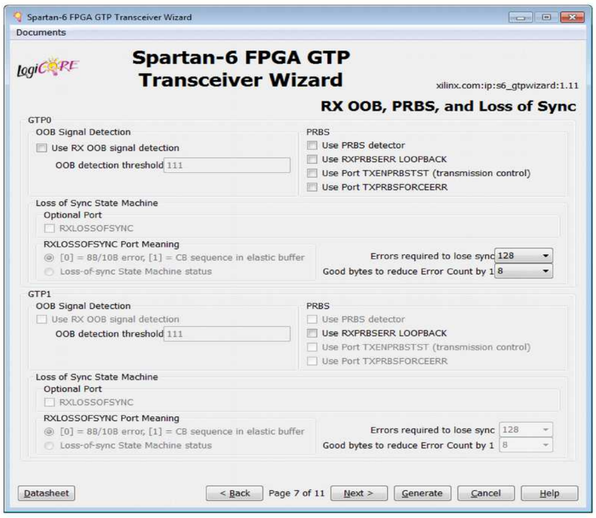


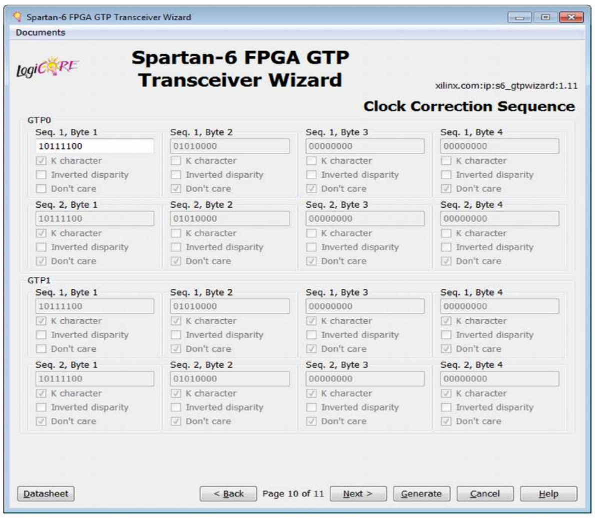












Note: For HD-SDI transceiver implementation then the set up is identical to 3G-SDI with the exception that the 'Protocol Template' must be set to 'SDI' and the target line rate for TX and RX must be set to 1.485 Gbps instead of 2.97 Gbps.

Appendix D: Example AD9880 I²C register configuration

This shows an example (minimum) register configuration for receiving input video at HD720p (1280x720p60) resolution using the DVI (HDMI) input video ports. The video format is assumed to be 24-bit RGB with separate syncs and one pixel per clock. Pixels are captured on the rising edge of the pixel clock.

Reg address	***Value***	***Reg name***	***Description***
0x01	0x67	PLL divider MSBs	PLL feedback divider MSBs
0x02	0x20	PLL divider LSBs	PLL feedback divider LSBs (N.B. the PLL divider must be set to 0x672 for 720p60 operation split over registers 0x01 and 0x02)
0x03	0xA8	VCO range Charge pump External clock enable	VCO range set to 2 Charge pump current set to 5 External clock disabled (use internal PLL clock).
0x4D	0x3B	TMDS PLL control 1	Sets the loop filter and pump current values
0x4E	0x6D	TMDS PLL control 2	Sets the loop filter and pump current values
0x4F	0x54	TMDS PLL control 3	Sets the loop filter and pump current values
0x50	0x20	VCO gear control	Must be set to 0x20 for correct operation
0x53	0x3F	Phase recovery loop control	Must be set to 0x3F for correct operation

Please refer to the AD9880 datasheet for a full description of the configuration registers. Datasheets can be downloaded from the Analog Devices website at: www.analog.com.

Appendix E: Example AD9889B I^2C register configuration

This shows an example (minimum) register configuration for generating output video at HD720p (1280x720p60) resolution using the DVI (HDMI) output video ports. The video format is assumed to be 24-bit RGB with separate syncs and one pixel per clock. Pixels are captured on the rising edge of the pixel clock.

Reg address	***Value***	***Reg name***	***Description***
0x41	0x10	System PD	Powers up/down the IC. Default state is power down. Writing 0x10 powers up the device
0xA2	0x87	TMDS data output drive strength	Sets the output drive strength of the TMDS data lines. Value 0x87 sets to high drive strength
0xA3	0x87	TMDS clock output drive strength	Sets the output drive strength of the TMDS clock line. Value 0x87 sets to high drive strength

Please refer to the AD9889B datasheet for a full description of the configuration registers. Datasheets can be downloaded from the Analog Devices website at: www.analog.com.

Appendix F: Running the example demo

A simple demo is provided to get you started with the development board. The demo generates a pair test pattern outputs on the HDMI output ports at 1280x720p60 resolution. The test patterns generated are a set of colour-bars and a yellow 'bouncing square' on a magenta background as shown in Figure (20) below. Vertical and horizontal blanking regions are shown in black. Figure (21) shows a photo of the board set-up and the demo in operation. Note that when switching HDMI output connectors the FPGA circuit must be reset by pressing button 'S1'.

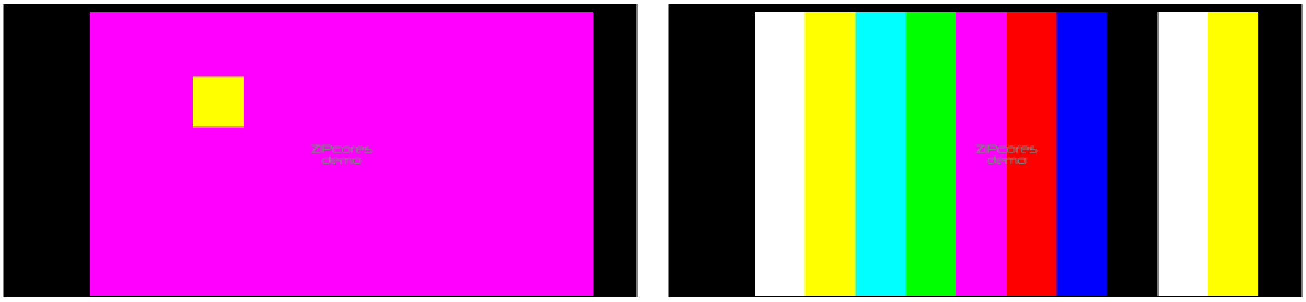


Figure 20: Test patterns generated on HDMI ports 1 & 2

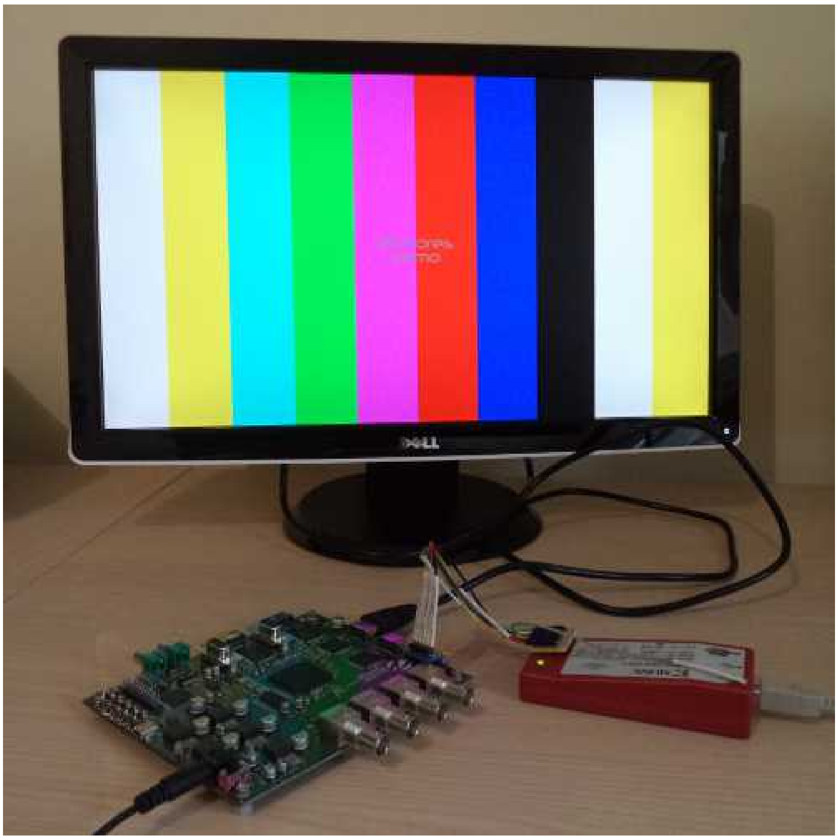


Figure 21: Photo showing working demo set-up

In order to program the FPGA for the demo, the Xilinx programming cable must first be connected to the JTAG programming pins 'P2' as described in the JTAG configuration section of this document. Connect one of the HDMI outputs to a suitable display using an HDMI cable (a standard computer monitor capable of 1280x720p60 resolution is ideal). Plug in the mains adapter and power-up the board. Then, program the FPGA by taking the following steps:

1. Invoke the Xilinx Impact programming software (Figure 22)
2. Click on the 'Boundary Scan' icon in the 'Impact Flows' window
3. Initialize the JTAG chain by clicking on the initialize chain icon

   (The FPGA should be identified in the JTAG chain and displayed in the main Impact window)
4. Right-click on the FPGA icon and choose 'Assign new configuration file'
5. Select the 'zip_dev_top.bit' bitfile provided with the demo
6. Right-click again on the FPGA icon and select 'Program'
7. The FPGA will now be programmed and the board should generate the video test patterns correctly

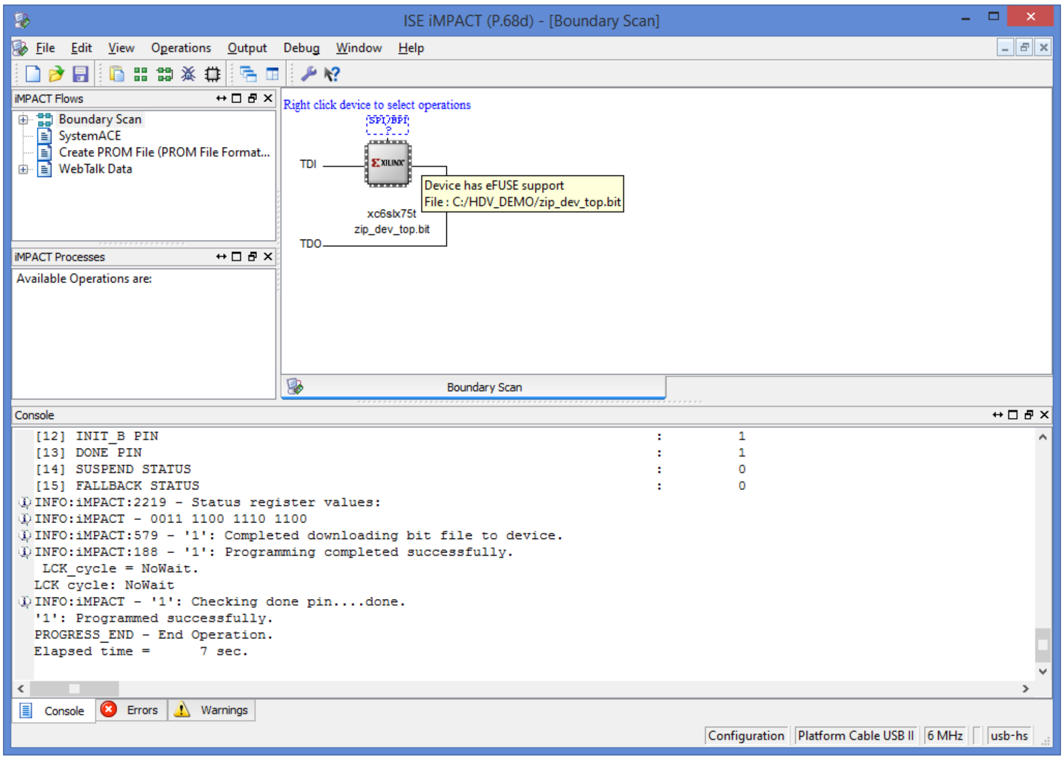


Figure 22: Programming the demo bitfile using the Xilinx Impact programming tool

Appendix G: List of supporting design files

The HD-video development board has a number of supporting design files and documents that may be downloaded from the Zipcores website at: www.zipcores.com/downloads.html.

A list of these files and a brief description is given below:

Filename	***Description***
docs/ zip_hdv_user_guide.pdf master.ucf	Folder containing various design docs. HD-video development board hardware user guide (this document). Example master 'UCF' file that defines all the top-level pinouts and design constrains for the Spartan-6 FPGA. This file may be used as a basis for compiling your FPGA designs using the Xilinx ISE software.
ise/ ZIP_DEV_TOP/ZIP_DEV_TOP.ise ZIP_DEV_TOP/zip_dev_top.bit ZIP_DEV_TOP/zip_dev_top.ucf	This folder contains the Xilinx ISE project environment for the demo. ISE project setup file (double click to invoke project). Bitfile for programming the FPGA demo as per Appendix F. UCF constraints file for FPGA demo.
vhdl/	This folder contains the top-level VHDL source code files for the demo example. The top-level synthesizable component is called zip_dev_top.vhd. The top-level testbench for the VHDL simulation is called zip_dev_top_bench.vhd. *(Note: some of the source-code is commercially sensitive and for this reason has been encrypted. Please contact Zipcores to obtain the decryption password)*
modelsim/ ZIP_DEV_TOP.mpf	This folder contains the Modelsim simulation environment in order to run a VHDL hardware simulation of the demo. You will need to obtain a copy of Mentor Graphics Modelsim in order to use these files. Modelsim project setup file (double click to invoke project).
perl/	This folder contains a perl script for parsing and generating image files from the simulation output. A free copy of PERL can be found at: http://www.activestate.com/activeperl.
misc/	This folder contains various data sheets and design notes for the components on the board.

Zipcores offers a wide range of video IP cores and custom solutions for the HD-video development board. If you have a specific requirement or simply want to discuss a potential solution then please get in touch. Further details may be found by visiting our website or contacting us at: www.zipcores.com/help.php.