DVI, VGA, and Component Video Demonstration

User Guide

UG248 (v1.1) October 11, 2006

XILINX®

Revision History

DVI, VGA, and Component Video Demonstration UG248 (v1.1) October 11, 2006

The following table shows the revision history for this document.

	Version	Revision
01/25/06	1.0	Initial Xilinx release.
10/11/06	1.1	Edits and additions to Chapter 4, “How to Make the Demo.”

Contents

Schedule of Figures

Schedule of Tables

Chapter 1: DVI/VGA to DVI/VGA Loop-Through Design

Chapter 2: SD/HD Video to SD/HD Video Loop-Through Design

Chapter 3: Control and Initialization Interface

Chapter 4: How to Make the Demo

About This Guide

This guide describes the VIODC Verilog loop-through demo designs, how to run them, and how to make them. These designs serve as starting points for creating standalone VIODC applications using Verilog. The guide also describes how the PicoBlaze™ is used to initialize the I2C interface on each device.

Guide Contents

This manual contains the following chapters:

- Chapter 1, "DVI/VGA to DVI/VGA Loop-Through Design"
- Chapter 2, "SD/HD Video to SD/HD Video Loop-Through Design"
- Chapter 3, "Control and Initialization Interface"
- Chapter 4, "How to Make the Demo"

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/literature/index.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*

Convention	Meaning or Use	Example
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *Development System Reference Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** [*option_name*] *design_name*
Braces { }	A list of items from which you must choose one or more	**`lowpwr =`**{**`on`**|**`off`**}
Vertical bar |	Separates items in a list of choices	**`lowpwr =`**{**`on`**|**`off`**}

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis . . .	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Handbook.*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Chapter 1

DVI/VGA to DVI/VGA Loop-Through Design

Description

For this demonstration, the Video Input Output Daughter Card (VIODC) receives video data in either DVI or analog VGA forms, and then retransmits the data in DVI and analog forms (Figure 1-1). The receive functionality is achieved using the AD9887A, and the transmit is performed using the TFP410 for DVI and ADV7123 for analog.

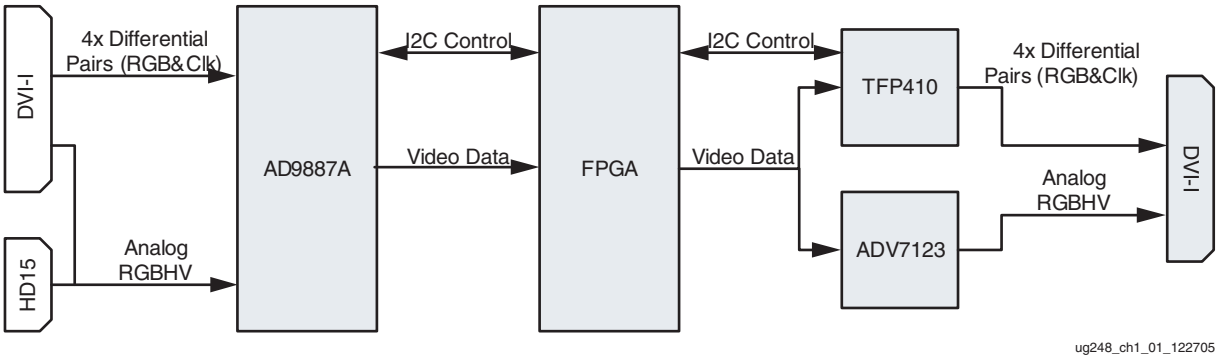

Figure 1-1: **DVI/VGA to DVI/VGA Loop-Through Design**

Input Interface

The input module `VIODC_AD9887_IN.v` captures data from the AD9887A and outputs the data in a single-pixel-per-clock format with corresponding syncs. This module is configured to take data from the AD9887A in a dual-pixel format, meaning that there are two pixels per clock cycle. Thus, the clock, IN_DATACLK, is at half the pixel rate. To convert to a single clock per cycle, the frequency of DATACLK must be doubled. A digital clock manager (DCM) is used to do this. The DCM_RST signal allows the DCM to be reset when the input clock frequency changes.

The input flip flops run off the 1X clock based on DATACLK to capture the data raw data signals. These signals are then transferred to the OUT_CLK (2x IN_DATACLK) domain with another set of flip flops. This path allows for ¾ DATACLK cycles, or 1.5 OUT_CLK

cycles. This second set of flip flops also serializes the odd and even pixels, so that the video stream is one pixel.

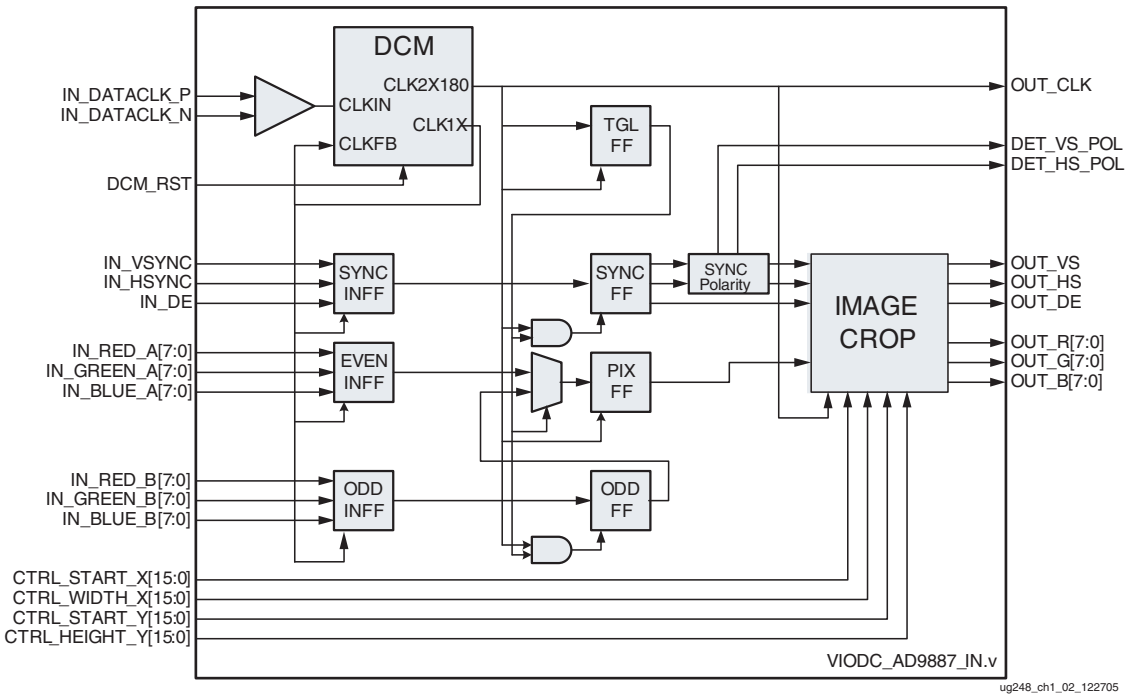

Figure 1-2: **Input Module `VIODC_AD9887_IN.v`**

The sync polarity block detects the incoming sync polarity and reports these detected polarities, available to the DET_xx_POL ports. The sync signals from this block are always active High.

When in analog mode, the AD9887A does not output an active Input Data Enable (IN_DE) signal. In this case, the last block (the image crop module) is used to generate the IN_DE signal. If IN_DE is active, the cropping is disabled and the input IN_DE is used.

Initialization Tables

To run various input modes, the AD9887A must be configured as well as the control input to the `VIODC_AD9887_IN.v` module. All I2C communications to the AD9887A are at address 0x9A/9B.

Analog SVGA60 Mode

Table 1-1: **Analog SVGA60 Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x0059	Start X coordinate of data.
CTRL_WIDTH_X[15:0]	0x0320	Width, in pixels.
CTRL_START_Y[15:0]	0x001B	Start Y coordinate of data.
CTRL_HEIGHT_Y[15:0]	0x0258	Height, in lines.

Table 1-2: **Analog SVGA60 AD9887A Registers**

Register Name	Register Address	Register Value	Description
Active Interface	0x12	0x81	Force selection of analog input
PLL Div MSB	0x01	0x41	PLL divider value. VGA60 has 1056 cycles per HSYNC period. 1056 -1 = 0x41F
PLL Div LSB	0x02	0xF0	
VCO/CPMP	0x03	0x8C	VCORNGE = 00, CURRENT = 011
Phase Adjust	0x04	0x80	Default phase = T/2
Clamp Placement	0x05	0x24	36 cycles after HSYNC
Clamp Duration	0x06	0x24	36 cycles in duration
HSOUT Pulse Width	0x07	0x80	128 cycles in HSYNC

Analog XGA60 Mode

Table 1-3: **Analog XGA60 Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x00A1	Start X coordinate of data.
CTRL_WIDTH_X[15:0]	0x0400	Width, in pixels
CTRL_START_Y[15:0]	0x0024	Start Y coordinate of data
CTRL_HEIGHT_Y[15:0]	0x0300	Height, in lines

Table 1-4: **Analog XGA60 AD9887A Registers**

Register Name	Register Address	Register Value	Description
Active Interface	0x12	0x81	Force selection of analog input
PLL Div MSB	0x01	0x53	PLL divider value. XGA60 has 1344 cycles per HSYNC period. 1344 -1 = 0x53F
PLL Div LSB	0x02	0xF0	
VCO/CPMP	0x03	0xB4	VCORNGE = 01, CURRENT = 101
Phase Adjust	0x04	0x80	Default phase = T/2
Clamp Placement	0x05	0x40	64 cycles after HSYNC
Clamp Duration	0x06	0x40	64 cycles in duration
HSOUT Pulse Width	0x07	0x88	136 cycles in HSYNC

Analog SXGA60 Mode

Table 1-5: **Analog SXGA60 Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x00F9	Start X coordinate of data.
CTRL_WIDTH_X[15:0]	0x0500	Width, in pixels
CTRL_START_Y[15:0]	0x002A	Start Y coordinate of data
CTRL_HEIGHT_Y[15:0]	0x0400	Height, in lines

Table 1-6: **Analog SXGA60 AD9887A Registers**

Register Name	Register Address	Register Value	Description
Active Interface	0x12	0x81	Force selection of analog input
PLL Div MSB	0x01	0x69	PLL divider value. SXGA60 has 1688 cycles per HSYNC period. 1688 -1 = 0x697
PLL Div LSB	0x02	0x70	
VCO/CPMP	0x03	0xD0	VCORNGE = 10, CURRENT = 100
Phase Adjust	0x04	0x80	Default phase = T/2
Clamp Placement	0x05	0x64	100 cycles after HSYNC
Clamp Duration	0x06	0x64	100 cycles in duration
HSOUT Pulse Width	0x07	0x70	112 cycles in HSYNC

Analog UXGA60 Mode

Table 1-7: **Analog UXGA60 Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x0031	Start X coordinate of data.
CTRL_WIDTH_X[15:0]	0x0640	Width, in pixels
CTRL_START_Y[15:0]	0x0032	Start Y coordinate of data
CTRL_HEIGHT_Y[15:0]	0x04B0	Height, in lines

Table 1-8: **Analog UXGA60 AD9887A Registers**

Register Name	Register Address	Register Value	Description
Active Interface	0x12	0x81	Force selection of analog input

Table 1-8: **Analog UXGA60 AD9887A Registers *(Continued)***

Register Name	Register Address	Register Value	Description
PLL Div MSB	0x01	0x86	PLL divider value. UXGA60 has 2160 cycles per HSYNC period. 2160 -1 = 0x86F
PLL Div LSB	0x02	0xF0	
VCO/CPMP	0x03	0xD4	VCORNGE = 10, CURRENT = 101
Phase Adjust	0x04	0x80	Default phase = T/2
Clamp Placement	0x05	0x40	64 cycles after HSYNC
Clamp Duration	0x06	0x40	64 cycles in duration
HSOUT Pulse Width	0x07	0xC0	192 cycles in HSYNC

DVI Mode

DVI does not require different settings for different resolutions. Nor does DVI require any cropping.

Table 1-9: **DVI TFP410 Registers**

Register Name	Register Address	Register Value	Description
Active Interface	0x12	0xC1	Force selection of DVI input
Mode Control 2	0x10	0x74	Set bus to 2 pixels/clock

Output Interface

The output module `VIODC_DVI_VGA_OUT.v` takes a pixel stream with sync signals and drives the signals necessary to interface to the TFP410 DVI transmitter and ADV7123 video DAC (Figure 1-3). To drive the clocks to these parts, DDR output flip flops are used. This ensures a static timing relationship between the clocks and the data.

The only real logic component of this module is the sync generation. The raw input syncs, IN_VS and IN_HS are expected to be active High. These sync signals are used to generate the output syncs, based on the CTRL_SYNC_MODE and CTRL_xx_POL.

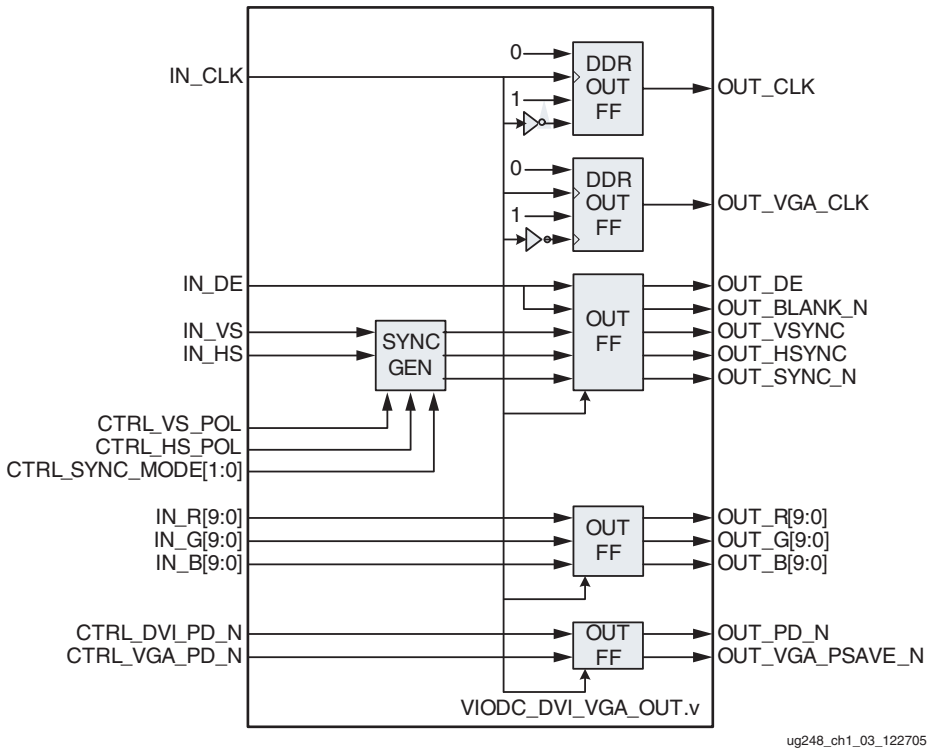

Figure 1-3: **Output Module `VIODC_DVI_VGA_OUT.v`**

CTRL_SYNC_MODE selects the following modes shown in Table 1-10.

Table 1-10: **Output Sync Modes**

Mode	Sync	Description
0	Separate	IN_VS and IN_HS are output as OUT_VSYNC and OUT_HSYNC with polarity selected by CTRL_VS_POL and CTRL_HS_POL. OUT_SYNC_N is not used and held Low. This mode is the most standard and should be used with DVI.
1	Composite	IN_VS and IN_HS are combined into a single signal and output on OUT_HSYNC. OUT_VSYNC and OUT_SYNC_N are not used and held Low.
2	On Green	IN_VS and IN_HS are combined and output on OUT_SYNC_N to be combined with the green channel. OUT_VSYNC and OUT_HSYNC are not used and held Low.
3	Separate Blocked	Identical to mode 0 except that there are no horizontal syncs during VSYNC. OUT_SYNC_N is not used and held Low.

Initialization Tables

The only initialization required on the output side is the TFP410 DVI transmitter. All I2C communications to the TFP410 is at address 0x70/71. The TFP410 does not require different settings for different resolutions.

Table 1-11: **TFP410 DVI Transmitter Registers**

Register Name	Register Address	Register Value	Description
CTL_1_MODE	0x08	0xBF	Enable with 24-bit interface mode.
CTL_2_MODE	0x09	0x00	Default OK.
CTL_3_MODE	0x0A	0x80	Default OK.

Running the Demo

DIP Switch Settings

The DIP switches on the VIODC are used to control the demonstration. Only the right four switches are used. These switches select the operating mode. The four corresponding LEDs reflect the mode selected by these switches. Table 1-12 indicates the DVI/VGA loopthrough modes.

Table 1-12: **DIV/VGA Loopthrough Modes**

Mode ID	VIODC DIP Switch Settings	Mode Name	Description
0	XXXX:0000	DVI Digital Passthrough	Passes any Digital DVI input resolution out as both DVI and analog VGA
1	XXXX:0001	Analog SVGA60	Analog 800x600@60Hz input, output as both DVI and analog VGA
2	XXXX:0010	Analog XGA60	Analog 1024x768@60Hz input, output as both DVI and analog VGA
3	XXXX:0011	Analog SXGA60	Analog 1280x1024@60Hz input, output as both DVI and analog VGA
4	XXXX:0100	Analog UXGA60	Analog 1600x1200@60Hz input, output as both DVI and analog VGA

Chapter 2

SD/HD Video to SD/HD Video Loop-Through Design

Description

For this demonstration, VIODC receives video data and then retransmits the data. The input and output interfaces include Composite, S-Video, or Component. The Composite and S-Video interfaces support NTSC (525i) timings. The Component interface supports NTSC (525i) 525p, 720p, and 1080i. The ADA4412 filter is used to condition the inputs before the ADV7403 is used to digitize the signals. The AVD7321A video encoder is used to generate the analog signals and the ADA4410 filter conditions these signals for output.

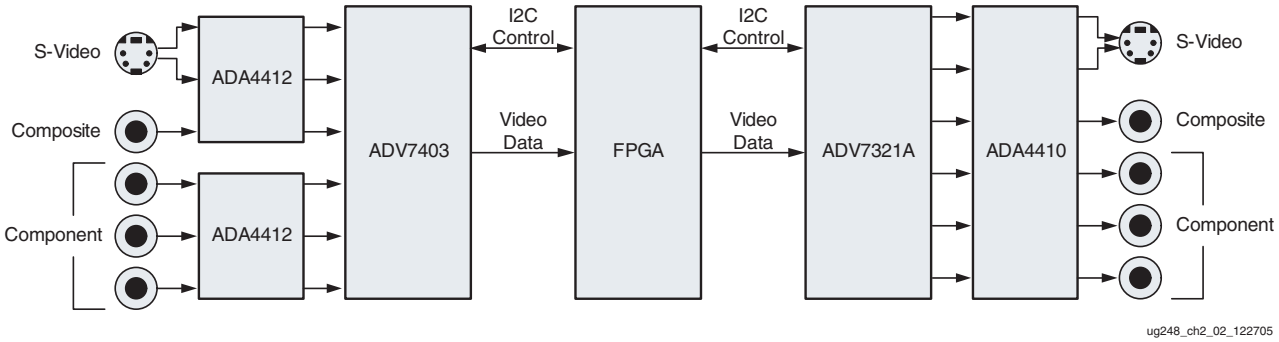

Figure 2-1: **SD/HD Video to SD/HD Video Loop-Through Design**

Input Interface

The input module `VIODC_ADV7403_IN.v` captures data from the ADV7403 and outputs the data as YPrPb with corresponding syncs. This module is configured to take data from the ADV7403 in a 30-bit 4:4:4 format.

The sync polarity block detects the incoming sync polarity and reports these detected polarities, available to the DET_xx_POL ports. The sync signals from this block are always active High.

The last block, the image crop module, is used to generate a DE signal.

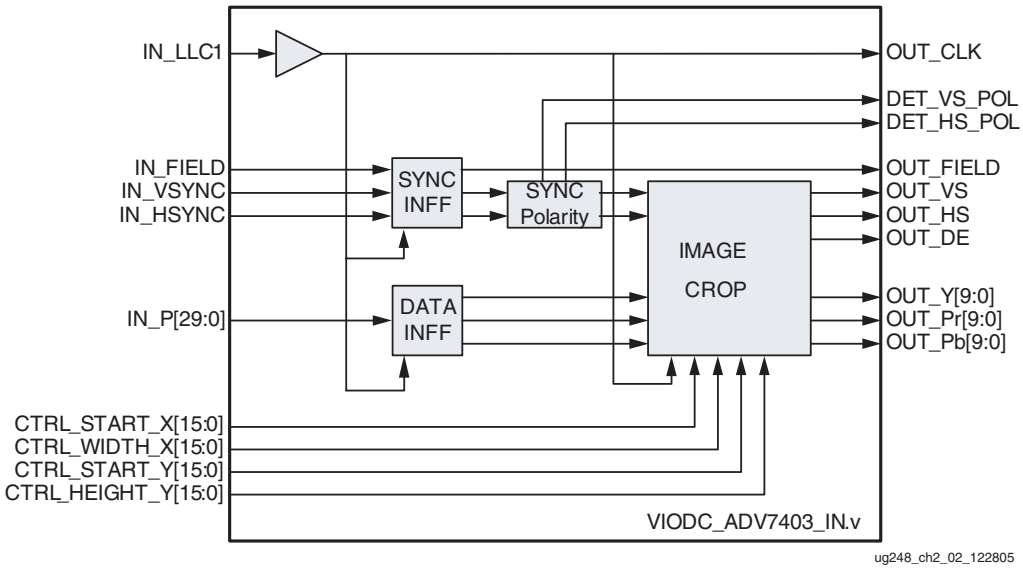

Figure 2-2: **Input Module `VIODC_AD9887_IN.v`**

Initialization Tables

To run various input modes, the ADV7403 must be configured as well as the control inputs to the `VIODC_AD9887_IN.v` module. All I2C communications to the ADV7403 is at address 0x40/41.

Composite Input

Table 2-1: **Composite Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x0096	Start X coordinate of data
CTRL_WIDTH_X[15:0]	0x05A0	Width, in pixels
CTRL_START_Y[15:0]	0x001C	Start Y coordinate of data
CTRL_HEIGHT_Y[15:0]	0x00F0	Height, in lines

Table 2-2: **Composite ADV7403 Register Settings**

Register Name	Register Address	Register Value	Description
Input Control	0x00	0x0F	CVBS on Ain11
Output Control	0x03	0x11	30-bit output on P29-P0
CP TLLC Control 3	0x89	0x18	Swap CR_CB_WB
Hsync control	0x36	0x7F	Hsync position control
Shaping Filter Control	0x17	0x41	Select SH1

Table 2-2: **Composite ADV7403 Register Settings *(Continued)***

Register Name	Register Address	Register Value	Description
ADI control 2	0x1D	0x47	Enable Crystal
ADC Control	0x3A	0x17	Power down ADC 1-3, set latch clock
BIAS Control	0x3B	0x80	External Bias enable
Manual Window Control	0x3D	0xA2	MWE enable, kill at 2.5%
Reserved	0x3E	0x6A	BLM optimization
Reserved	0x3F	0xA0	RGB optimization
CP Misc Control 1	0x86	0x0B	Enable STDI line count mode
AFE Control 1	0xF3	0x01	Enable anti alias filter on ADC0
VS Mode Control	0xF9	0x03	Set max V lock range
	0x0E	0x80	ADI recommended Sequence
	0x52	0x46	ADI recommended Sequence
	0x54	0x00	ADI recommended Sequence
	0x7F	0xFF	ADI recommended Sequence
	0x81	0x30	ADI recommended Sequence
	0x90	0xC9	ADI recommended Sequence
	0x91	0x40	ADI recommended Sequence
	0x92	0x3C	ADI recommended Sequence
	0x93	0xCA	ADI recommended Sequence
	0x94	0xD5	ADI recommended Sequence
	0xB1	0xFF	ADI recommended Sequence
	0xB6	0x08	ADI recommended Sequence
	0xC0	0x9A	ADI recommended Sequence
	0xCF	0x50	ADI recommended Sequence
	0xD0	0x4E	ADI recommended Sequence
	0xD1	0xB9	ADI recommended Sequence
	0xD6	0xDD	ADI recommended Sequence
	0xD7	0xE2	ADI recommended Sequence
	0xE5	0x51	ADI recommended Sequence
	0x0E	0x00	ADI recommended Sequence

S-Video Input

Table 2-3: **S- Video Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x0096	Start X coordinate of data.
CTRL_WIDTH_X[15:0]	0x05A0	Width, in pixels
CTRL_START_Y[15:0]	0x001C	Start Y coordinate of data
CTRL_HEIGHT_Y[15:0]	0x00F0	Height, in lines

Table 2-4: **S-Video ADV7403 Register Settings**

Register Name	Register Address	Register Value	Description
Output Control	0x03	0x11	30-bit output on P29-P0
CP TLLC Control 3	0x89	0x18	Swap CR_CB_WB
Hsync control	0x36	0x7F	Hsync position control
ADI control 2	0x1D	0x47	Enable Crystal
ADC Control	0x3A	0x13	Power down ADC 2 & 3, set latch clock
BIAS Control	0x3B	0x80	External Bias enable
Manual Window Control	0x3D	0xA2	MWE enable, kill at 2.5%
Reserved	0x3E	0x6A	BLM optimization
Reserved	0x3F	0xA0	RGB optimization
Configure 1	0x69	0x02	YC on Ain10, Ain12
CP Misc Control 1	0x86	0x0B	Enable STDI line count mode
AFE Control 1	0xF3	0x03	Enable anti alias filter on ADC0&1
VS Mode Control	0xF9	0x03	Set max V lock range
	0x0E	0x80	ADI recommended Sequence
	0x52	0x46	ADI recommended Sequence
	0x54	0x00	ADI recommended Sequence
	0x7F	0xFF	ADI recommended Sequence
	0x81	0x30	ADI recommended Sequence
	0x90	0xC9	ADI recommended Sequence
	0x91	0x40	ADI recommended Sequence
	0x92	0x3C	ADI recommended Sequence
	0x93	0xCA	ADI recommended Sequence
	0x94	0xD5	ADI recommended Sequence
	0xB1	0xFF	ADI recommended Sequence

Table 2-4: **S-Video ADV7403 Register Settings *(Continued)***

Register Name	Register Address	Register Value	Description
	0xB6	0x08	ADI recommended Sequence
	0xC0	0x9A	ADI recommended Sequence
	0xCF	0x50	ADI recommended Sequence
	0xD0	0x4E	ADI recommended Sequence
	0xD1	0xB9	ADI recommended Sequence
	0xD6	0xDD	ADI recommended Sequence
	0xD7	0xE2	ADI recommended Sequence
	0xE5	0x51	ADI recommended Sequence
	0x0E	0x00	ADI recommended Sequence

Component 525i Input

Table 2-5: **Component 525i Cropping Controls**

Port Name	Port Value	Description
CTRL_START_X[15:0]	0x0072	Start X coordinate of data.
CTRL_WIDTH_X[15:0]	0x059D	Width, in pixels
CTRL_START_Y[15:0]	0x0010	Start Y coordinate of data
CTRL_HEIGHT_Y[15:0]	0x00F0	Height, in lines

Table 2-6: **Component 525i ADV7403 Register Settings**

Register Name	Register Address	Register Value	Description
Primary Mode	0x05	0x00	Standard definition
Video Standard	0x06	0x0A	525i
ADI control 2	0x1D	0x47	Enable Crystal
ADC Control	0x3A	0x11	Power down ADC 3, set latch clock
BIAS Control	0x3B	0x80	External Bias enable
TLLC Control	0x3C	0x52	Setup PLL Qpump
CP output selection	0x6B	0xC0	Set to 30 bit output, field
CP AV Control	0x7B	0x1C	No AV codes
CP Detection Control 1	0x85	0x19	Turn off SSPD and force SOY, int VS
CP Misc Control 1	0x86	0x0B	Enable STDI line count mode

Table 2-6: **Component 525i ADV7403 Register Settings**

Free Run Line Length 1	0x8F	0x77	FR_LL to 1820 and enable 27Mhz LLC
VBI info	0x90	0x1C	FR_LL to 1820
AFE Control 1	0xF3	0x07	Enable anti alias filter on ADC0-2
	0x0E	0x80	ADI recommended Sequence
	0x52	0x46	ADI recommended Sequence
	0x54	0x00	ADI recommended Sequence
	0x0E	0x00	ADI recommended Sequence

Component 525p Input

The ADV7403 is configured to generate a DE signal, so no cropping is required.

Table 2-7: **Component 525p ADV7403 Register Settings**

Register Name	Register Address	Register Value	Description
Primary Mode	0x05	0x01	Component Video
Video Standard	0x06	0x06	525p
ADI control 2	0x1D	0x47	Enable Crystal
ADC Control	0x3A	0x11	Power down ADC 3, set latch clock
BIAS Control	0x3B	0x80	External Bias enable
TLLC Control	0x3C	0x5B	Setup PLL Qpump
CP output selection	0x6B	0x80	Set to 30 bit output, DE
CP Detection Control 1	0x85	0x19	Turn off SSPD and force SOY, int VS
CP Misc Control 1	0x86	0x0B	Enable STDI line count mode
	0x0E	0x80	ADI recommended Sequence
	0x52	0x46	ADI recommended Sequence
	0x54	0x00	ADI recommended Sequence
	0x0E	0x00	ADI recommended Sequence

Component 720p Input

The ADV7403 is configured to generate a DE signal, so no cropping is required.

Table 2-8: **Component 720p ADV7403 Register Settings**

Register Name	Register Address	Register Value	Description
Primary Mode	0x05	0x01	Component Video
Video Standard	0x06	0x0A	720p
ADI control 2	0x1D	0x47	Enable Crystal
ADC Control	0x3A	0x21	Power down ADC 3, set latch clock
BIAS Control	0x3B	0x80	External Bias enable
TLLC Control	0x3C	0x5D	Setup PLL Qpump
CP output selection	0x6B	0x80	Set to 30 bit output, DE
CP Detection Control 1	0x85	0x19	Turn off SSPD and force SOY, int VS
CP Misc Control 1	0x86	0x0B	Enable STDI line count mode
	0x0E	0x80	ADI recommended Sequence
	0x52	0x46	ADI recommended Sequence
	0x54	0x00	ADI recommended Sequence
	0x0E	0x00	ADI recommended Sequence

Component 1080i Input

The ADV7403 is configured to generate a DE signal, so no cropping is required.

Table 2-9: **Component 1080i ADV7403 Register Settings**

Register Name	Register Address	Register Value	Description
Primary Mode	0x05	0x01	Component Video
Video Standard	0x06	0x0A	720p
ADI control 2	0x1D	0x47	Enable Crystal
ADC Control	0x3A	0x21	Power down ADC 3, set latch clock
BIAS Control	0x3B	0x80	External Bias enable
TLLC Control	0x3C	0x5D	Setup PLL Qpump
CP output selection	0x6B	0x80	Set to 30 bit output, DE
CP Detection Control 1	0x85	0x19	Turn off SSPD and force SOY, int VS
CP Misc Control 1	0x86	0x0B	Enable STDI line count mode

Table 2-9: **Component 1080i ADV7403 Register Settings *(Continued)***

Register Name	Register Address	Register Value	Description
	0x0E	0x80	ADI recommended Sequence
	0x52	0x46	ADI recommended Sequence
	0x54	0x00	ADI recommended Sequence
	0x0E	0x00	ADI recommended Sequence

Output Interface

The output module `VIODC_ADV7321_OUT.v` (Figure 2-3) takes a pixel stream with sync signals and drives the signals necessary to interface to the ADV7321 video encoder. To drive the clocks to this part, DDR output flip flops are used. This ensures a static timing relationship between the clocks and the data.

The only control input of this module is MODE_S_P. Although the ADV7321 video encoder can support both standard definition (SD) and high definition (HD) simultaneously, this module is designed to only support one at a time. The MODE_S_P input selects which mode. A logic one selects SD mode; a logic zero selects HD mode. In HD mode, the input data passes through directly to the output flip flops, but in SD mode, the Pr and Pb. data must be interleaved onto the OUT_C bus. A MUX_TOGGLE signal is generated to do this and is reset by IN_HS.

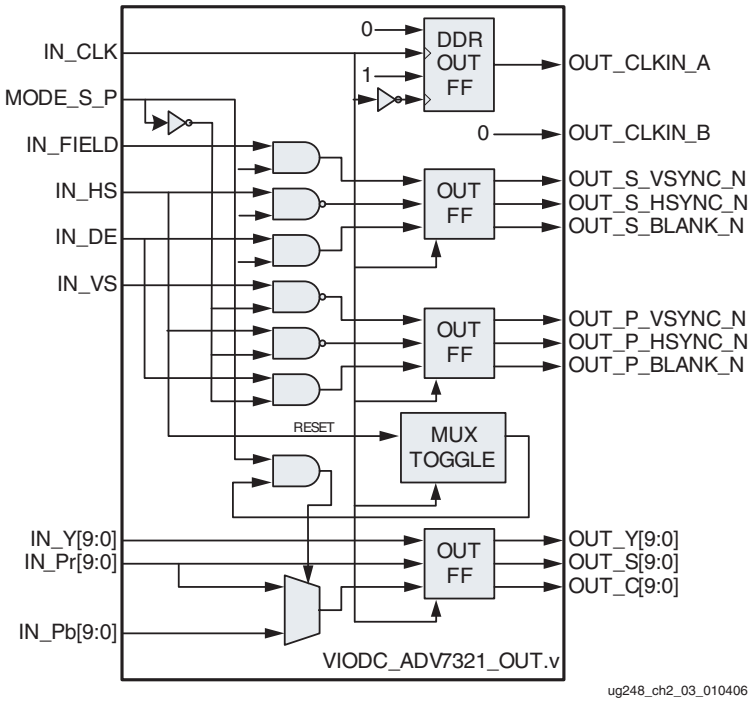

Figure 2-3: **Output Module `VIODC_ADV7321_OUT.v`**

Initialization Tables

To run various input modes, the ADV7321 must be configured as well as selecting the mode for the `VIODC_ADV7321_OUT.v` module. All I2C communications to the ADV7321 is at address 0x54/55.

SD Output

Table 2-10: **SD ADV7321 Register Settings**

Register Name	Register Address	Register Value	Description
Power Mode	0x00	0x7E	DAC B-F, no PLL
Mode Select	0x01	0x80	SD only, Y/C swap
SD Mode R0	0x40	0x08	NTSC, Y Notch NTSD, C 1.3MHz
SD Mode R1	0x42	0x41	Output Configuration
SD Mode R6	0x48	0x18	20-bit
SD Timing R0	0x4A	0x02	Slave Mode 1

Component 525p Output

Table 2-11: **Component 525p ADV7321 Register Settings**

Register Name	Register Address	Register Value	Description
Power Mode	0x00	0x1E	DAC D-F, no PLL
Mode Select	0x01	0x10	PS only
HD Mode R2	0x11	0x01	Pix data valid
HD Mode R4	0x13	0x04	Mode 4:4:4

Component 720p Output

Table 2-12: **Component 720p ADV7321 Register Settings**

Register Name	Register Address	Register Value	Description
Power Mode	0x00	0x1E	DAC D-F, no PLL
Mode Select	0x01	0x20	HDTV
HD Mode R1	0x10	0x38	720p@30Hz
HD Mode R2	0x11	0x01	Pix data valid
HD Mode R4	0x13	0x24	Mode 4:4:4, SSAF

Component 1080i Output

Table 2-13: **Component 1080i ADV7321 Register Settings**

Register Name	Register Address	Register Value	Description
Power Mode	0x00	0x1E	DAC D-F, no PLL
Mode Select	0x01	0x20	HDTV
HD Mode R1	0x10	0x68	1080i@30Hz
HD Mode R2	0x11	0x01	Pix data valid
HD Mode R4	0x13	0x24	Mode 4:4:4, SSAF

Running the Demo

DIP Switch Settings

The DIP switches on the VIODC are used to control the demonstration. Only the right four switches are used. These switches select the operating mode. The four corresponding LEDs reflect the mode selected by these switches. The Table 2-14 indicates the SD/HD loop-through modes.

Table 2-14: **DIP Switch Settings**

Mode ID	VIODC DIP Switch Settings	Mode Name	Description
5	XXXX:0101	SD Composite	Standard definition composite input, composite, S-Video and component output
6	XXXX:0110	SD S-Video	Standard definition S-Video input, composite, S-Video and component output
7	XXXX:0111	SD Component	Standard definition component input, composite, S-Video and component output
8	XXXX:1000	Component 525p	Component 525p input and output
9	XXXX:1001	Component 720p	Component 720p input and output
10	XXXX:1010	Component 1080i	Component 1080i input and output

Chapter 3

Control and Initialization Interface

Description

The various ICs on VIODC require initialization via the I2C bus for operation in various modes. The FPGA modules that interface to these ICs also have various controls. All mode settings and I2C initialization is handled by a PicoBlaze processor. This PicoBlaze processor accesses the switches to detect the selected mode and settings and configure the related ICs and modules. The PicoBlaze processor also operates the status LEDs to indicated various status information depending on the operating mode.

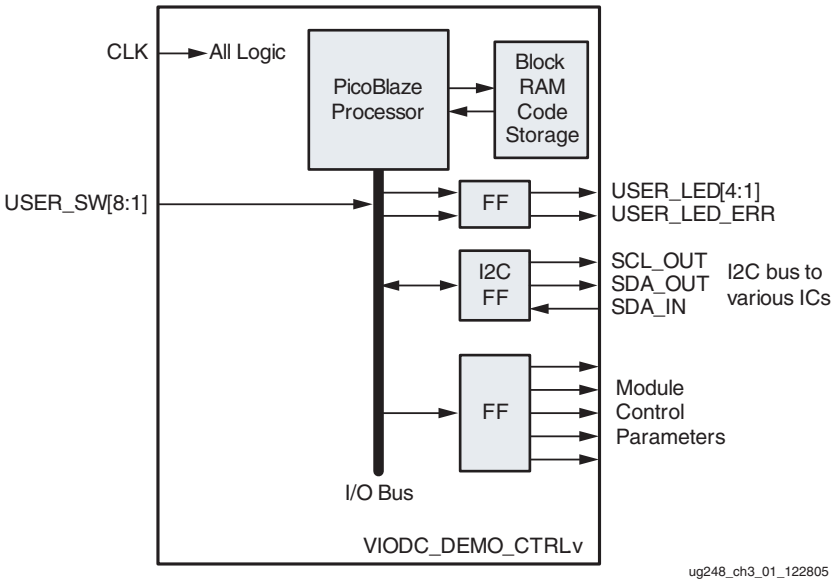

Figure 3-1: **Module Block Diagram**

The VIODC_DEMO_CTRL module is made up of the PicoBlaze processor and logic to interface it to the modules and ICs which it controls. The PicoBlaze processor is a very simple 8-bit microcontroller. It utilizes a single block RAM to store the processor code.

The I/O bus on the PicoBlaze connects the processor to various inputs and outputs. For user control and status for the various demo modes, the I/O bus is connected to the USER_SW input and USER_LED outputs. I2C FFs are connected to the I/O bus to allow the processor to interface to an I2C bus for the various I2C devices. This module also has numerous other registered outputs for control of the various Verilog modules in this design.

The PicoBlaze processor application is coded in assembly for the Mediatronix pBlaze IDE version 3.6. This application monitors the USER_SW inputs. When ever a new mode is

selected, the processor configures the external devices via the I2C busses and configures the interfacing modules via the control parameters.

I2C Operation

The PicoBlaze processor implements the I2C protocol in software by discretely controlling the SCL and SDA lines (bit-banging). Software delay loops control the timing of this interface. This protocol is encapsulated into several I2C subroutines which make the interface easy to use.

The primary function used for the register initialization is I2C_REG_WRITE. This function requires the device address, register address, and register data to write. It returns a bit indicating success or error.

Chapter 4

How to Make the Demo

Files to Make the Demo

This project is implemented using ISE 8.2i. The project settings are stored in the project file, `VIODC_DEMO.ise`. All of the listed source files (Table 4-1) are required to synthesize and implement the project.

All of the files associated with this demo are provided on the VSK CDROM at: `Examples\HDL_Demos\viodc_IO_verilog`

Table 4-1: **Files Needed to Make the Demo**

Filename	Description
`VIODC_DEMO.ise`	Project file
`VIODC_DEMO.v`	Top level of the design
`VIODC_DEMO.ucf`	Constraints file for design
`VIODC_DEMO_CTRL.v`	Control module for design
`kcpsm3.v`	PicoBlaze processor core
`PB_VIODC_DEMO.v`	Assembled PicoBlaze code
`VIODC_AD9887_IN.v`	Module for interfacing to AD9887A
`IMAGE_CROP.v`	Image cropping module
`VIODC_DVI_VGA_OUT.v`	Module for interfacing to ADV7123 and TFP410
`VIODC_ADV7403_IN.v`	Module for interfacing to ADV7403
`VIODC_ADV7321_OUT.v`	Module for interfacing to ADC7321

Files to Assemble PicoBlaze Code

This demonstration uses the PicoBlaze3 8-bit soft-core microcontroller. The PicoBlaze processor is responsible for reading the mode selection switches, which are described in the previous chapters, and for programming the various peripheral chips on the VIODC. More information about the PicoBlaze microcontroller can be found at the PicoBlaze web page and at the PicoBlaze User Resources page.

Table 4-2 contains the files needed to assemble PicoBlaze code. The PicoBlaze processor assembly code file is `VIODC_DEMO.psm`. The source file requires the `helpers.inc` include file and uses the `PB_CODE_template.v` file to generate the Verilog file

PB_VIODC_DEMO.v. The PB_VIODC_DEMO.v file must then be copied into the Xilinx ISE project area.

The files are assembled using the Mediatronix pBlazeIDE version 3.6. Documentation and a free-of-charge download of the tool can be found at the Mediatronix website.

The pBlazeIDE tool supports multiple versions of the PicoBlaze microcontroller. The user will need to specify which version is being targeted by selecting the "Settings" menu and then selecting "PicoBlaze 3".

Table 4-2: **Files Needed to Assemble PicoBlaze Code**

Filename	Description
pBlazeIDE.exe	Assembler/IDE
VIODC_DEMO.psm	PicoBlaze Assembly file
PB_CODE_template.v	Verilog template file for assembled code
helpers.inc	Include file of useful equates