Avalon Video Input Module

Version 1.0, December 2004 Application Note 373

Introduction

The Avalon™ video input module provides a flexible video capture solution, which may be implemented in Altera® Cyclone™ or Stratix® devices, and has the following features:

- Component video interface to VGA camera module
- Color-bar test pattern generator
- Clipping of input image
- Horizontal (Y) scaling of input image
- Vertical (X) scaling of input image
- Avalon direct memory access (DMA) master to write image(s) to frame buffer memory
- Avalon register slave for control and status

Functional Description

Figure 1 shows the Avalon video input module block diagram.

Figure 1. Block Diagram

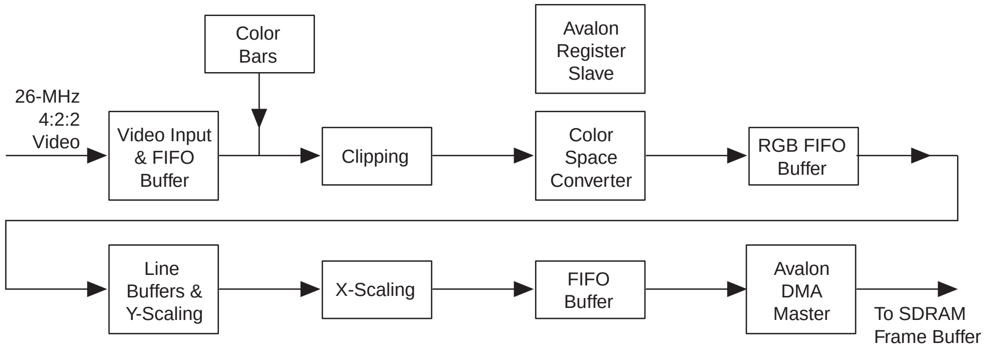


Figure 2 shows the signals.

Figure 2. Signals

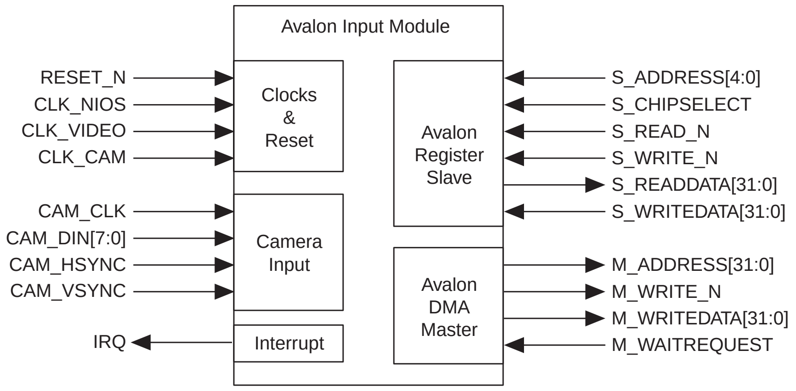


Table 1 shows the signals.

Table 1. Signals (Part 1 of 2)

Signal	Direction	Description
Clocks & Reset		
`RESET_N`	Input	Active low asynchronous reset.
`CLK_NIOS`	Input	Nios® II and Avalon clock.
`CLK_VIDEO`	Input	Video clock.
`CLK_CAM`	Input	Pixel clock.
Camera Input		
`CAM_CLK`	Input	Clock input from camera.
`CAM_DIN[7:0]`	Input	Data input from camera.
`CAM_HSYNC`	Input	Horizontal synchronization signal from camera.
`CAM_VSYNC`	Input	Vertical synchronization signal from camera
Interrupts		
`IRQ`	Output	Interrupt request.
Avalon Register Slave		
`S_ADDRESS[4:0]`	Input	Register address.
`S_CHIPSELECT`	Input	Device select module input.
`S_READ_N`	Input	Avalon read enable.

Table 1. Signals (Part 2 of 2)

Signal	Direction	Description
`S_WRITE_N`	Input	Avalon Write enable.
`S_READDATA[31:0]`	Output	Avalon read data.
`S_WRITEDATA[31:0]`	Input	Avalon write data.
Avalon DMA Master		
`M_ADDRESS[31:0]`	Output	Avalon address to frame buffer.
`M_WRITE_N`	Output	Avalon write enable.
`M_WRITEDATA[31:0]`	Output	Avalon write data to frame buffer.
`M_WAITREQUESTS`	Input	Avalon wait request.

Clocks

The Avalon video input module requires the following clocks:

- Camera clock
- Pixel clock
- Video clock
- Nios II and Avalon clock

Camera Clock (CAM_CLK)

A global clock pin is used as the clock input from the camera module. data from the camera is clocked into a FIFO buffer on the rising edge of this clock. The FIFO buffer allows for variations in phase and frequency between the camera clock and the pixel rate clock.

Pixel Clock (CLK_CAM)

The pixel clock should be the same frequency as the camera clock (or very close to it) and clocks the read side of the camera input FIFO buffer, the clipping block, color-space converter (CSC) and RGB input FIFO buffer write port. The FIFO buffer and separate clock allow for slight differences in frequency between the camera clock and the pixel clock.

Video Clock (CLK_VIDEO)

The horizontal and vertical scaling blocks are driven by the video clock, which can normally be connected to the Nios II clock (see "Clocking Requirements" on page 8 for video clocking requirements).

Nios II & Avalon Clock (CLK_NIOS)

The Avalon DMA master and register slave must be driven from the same clock that the Nios II processor and Avalon bus fabric use.

Component Video Input

The input to this block is progressive scan (not interlaced) 4:2:2 component video (Cb, Y, Cr, Y, …) at 26 MHz based on the output from a VGA camera (part number DA3530-30XF1) from Dialog Semiconductor. Figure 3 shows the timing requirements of this interface.

Figure 3. Component Video Interface Timing

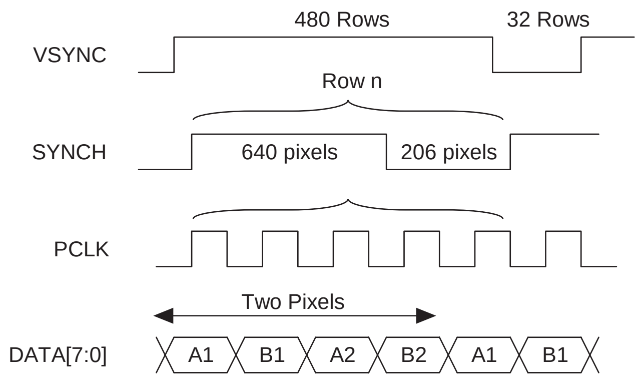


Table 2 shows the timing requirements of this interface.

Table 2. Component Video Interface Data

Data Pins	YCbCr			
	A1	B1	A2	B2
`data[7]`	`Cb[7]`	`Y1[7]`	`Cr[7]`	`Y2[7]`
`data[6]`	`Cb[6]`	`Y1[6]`	`Cr[6]`	`Y2[6]`
`data[5]`	`Cb[5]`	`Y1[5]`	`Cr[5]`	`Y2[5]`
`data[4]`	`Cb[4]`	`Y1[4]`	`Cr[4]`	`Y2[4]`
`data[3]`	`Cb[3]`	`Y1[3]`	`Cr[3]`	`Y2[3]`
`data[2]`	`Cb[2]`	`Y1[2]`	`Cr[2]`	`Y2[2]`
`data[1]`	`Cb[1]`	`Y1[1]`	`Cr[1]`	`Y2[1]`
`data[0]`	`Cb[0]`	`Y1[0]`	`Cr[0]`	`Y2[0]`

The input Cb and Cr samples are replicated (rather than interpolated) to fill the output samples giving parallel Y, Cb, Cr samples at an effective clock rate of 13 MHz.

Color-Bar Generator

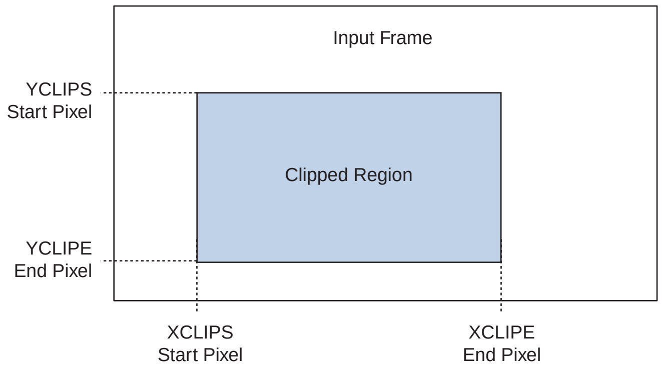
The color-bar generator gives a very simple pattern of vertical color bars to aid in debugging a system using the video input module. The color-bar generator decodes bits [10:7] of the pixel address of each pixel in a line and sets the pixel colors according to the following Verilog HDL code fragment:

```
case (pixel[10:7])
4'h0:color = white;
4'h1:color = yellow;
4'h2:color = cyan;
4'h3:color = green;
4'h4:color = magenta;
4'h5:color = red;
4'h6:color = blue;
4'h7:color = black;
4'h8:color = white;
4'h9:color = yellow;
default:color = cyan;
endcase
```

Clipping

Clipping allows a portion of the input frame to be selected for processing by further blocks in the video input module. In this way, only the portion of interest is processed and written to frame buffer memory, conserving memory bandwidth. The region to be clipped is specified by four registers defining the start and end line (Y clipping, `YCLIPS` and `YCLIPE` registers) and start and end pixel (X clipping, `XCLIPS` and `XCLIPE` registers). Figure 4 shows the video input clipping.

Figure 4. Video Input Clipping



In addition to the scale factor registers, the horizontal and vertical length registers must be loaded with the following correct values:

```
XLEN = XCLIPE – XCLIPS
YLEN = YCLIPE - YCLIPS
```

Color-Space Converter

The CSC converts the Y, Cb, Cr format video into red, green and blue components. The R, G and B values are each 6 bits so that one pixel (18 bits) can be written to each word of an M4K RAM block in 256 × 18 mode.

The following equations are used for color-space conversion and for gamma corrected RGB:

```
R’ = 1.164(Y - 16) + 1.596(Cr - 128)
G’ = 1.164(Y - 16) – 0.813(Cr - 128) – 0.392(Cb - 128)
B’ = 1.164(Y - 16) + 2.017(Cb - 128)
```

No dithering is applied during the color-space conversion.

RGB Input FIFO Buffer

The RGB input FIFO buffer transfers data from the component video clock domain to the system clock domain.

The FIFO buffer depth required depends upon the processing rate of the Y scale block and the maximum scaling factor, see “Clocking Requirements” on page 8.

Line Buffer & Vertical Scaling

Vertical (Y) scaling is achieved by interpolation between lines. Two line buffers are used to store two input lines, each using 3 M4K RAM blocks in 256 × 18 mode. Interpolation uses 3 multipliers per line buffer (1 each for R,G, and B samples).

The scale factor is specified in the `YSCALE` register as the reciprocal of the required scaling. The `YSCALE` register allows a 4-bit integer and a 12-bit fractional part to be specified. For example when scaling up by 2, the scale factor is 0.5 (i.e. the reciprocal of the required scaling) or `0000.100000000000` binary.

As each output line is generated, the `YSCALE` value updates the notional position of the current output line with respect to the input lines. Two adjacent input lines are combined in proportion to their distances from the output line. Figure 5 shows an example of the generation of output lines when the scaling is greater than 1 (scaling up or adding lines) with `YSCALE` = 0.4 or a scaling of 2.5.

Figure 5. Generation of Output Lines With Scale Factor > 1

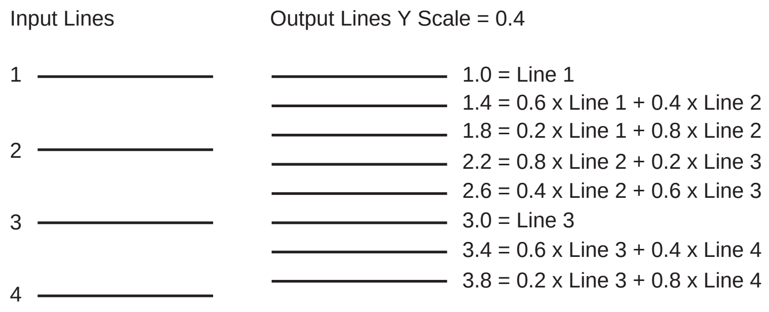
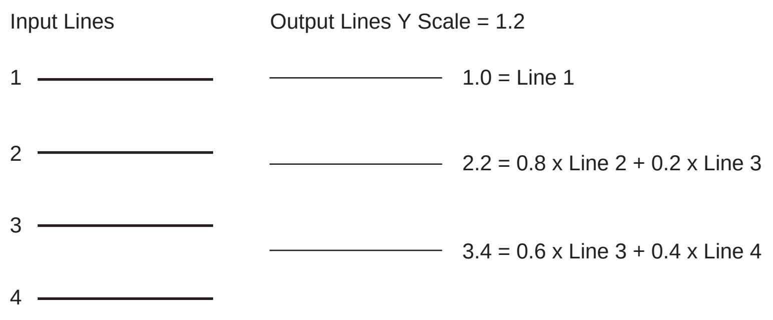

Input Lines
Output Lines Y Scale = 0.4
1
2
3
4
1.0 = Line 1
1.4 = 0.6 x Line 1 + 0.4 x Line 2
1.8 = 0.2 x Line 1 + 0.8 x Line 2
2.2 = 0.8 x Line 2 + 0.2 x Line 3
2.6 = 0.4 x Line 2 + 0.6 x Line 3
3.0 = Line 3
3.4 = 0.6 x Line 3 + 0.4 x Line 4
3.8 = 0.2 x Line 3 + 0.8 x Line 4

Figure 6 shows an example of the generation of output lines when the scaling is less than 1 (scaling down or removing lines) with `YSCALE` = 1.2 or a scaling of 0.833.

Figure 6. Generation of Output Lines With Scale Factor < 1


Input Lines
Output Lines Y Scale = 1.2
1
2
3
4
1.0 = Line 1
2.2 = 0.8 x Line 2 + 0.2 x Line 3
3.4 = 0.6 x Line 3 + 0.4 x Line 4

The number of lines written to the frame buffer in each frame depends upon the number of lines in the clipped region and the scale factor and can be found from the following formula using integer arithmetic:

```
Number of lines = ((YCLIPE – YCLIPS)× 4096)/YSCALE + 1
```

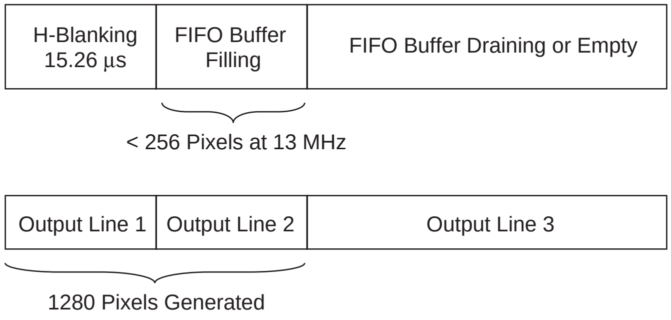
For example, setting `YCLIPE` = 100, `YCLIPS` = 50, `YSCALE` = `0x0666` (scaling of 2.5) results in 126 output lines being generated for each frame.

Clocking Requirements

The RGB input FIFO buffer must be deep enough to hold the new input samples that arrive while the Y scale block is interpolating two extra output lines. Interpolation begins once a complete line has been received, that is, during the horizontal blanking period. For a worst case scale factor of three, the RGB input FIFO buffer must buffer samples while two output lines are being interpolated. During interpolation of the third output line the RGB input FIFO buffer begins draining as samples in the line buffer are replaced by those for the next input line.

If the RGB input FIFO buffer has 256 entries (one M4K), it fills in 19.69 µs at a pixel rate of 13 MHz. This time plus the horizontal blanking (206 pixels or 15.26 µs) period must be sufficient to generate two interpolated output lines, so the Y scaling block must be able to process 1,280 pixels in less than 34.95 µs. The Y scaling clock must be greater than 37 MHz. Figure 7 shows the clock requirements.

Figure 7. Clock Requirements



Horizontal Scaling

The horizontal (X) scaling is achieved by interpolation between pixels in a similar manner to the Y scaling. No extra storage is required by the X scaling block.

The number of pixels written to the frame buffer in each line depends upon the number of pixels in each line of the clipped region. The horizontal scale factor can be found from the following formula using integer arithmetic:

```
Number of  lines = ((XCLIPE – XCLIPS)× 4096)/XSCALE + 1
```

For example, setting `XCLIPE` = 100, `XCLIPS` = 50, `XSCALE` = `0x0666` (scaling of 2.5) results in 126 output pixels being generated for each line.

RGB Output FIFO Buffer

One M4K RAM block in 256 × 18 mode is employed to gather pixels from the X scaling block for Avalon burst transfers to the frame buffer.

Avalon DMA Master

The Avalon DMA master is a write only master that writes to the frame buffer(s) in system memory (usually some form of SDRAM). The master waits for sufficient data to be available from the RGB output FIFO buffer to perform a burst transfer.

To conserve memory bandwidth, pixels are written to memory as 16-bit data, with R, G, and B represented by 5, 6, and 5 pixels, respectively. The least significant bit of the R and G samples is discarded with no dithering applied.

The DMA master assumes the use of a linear frame buffer in which all lines are contiguous. The start address of a frame buffer must be word aligned in memory.

Master Interrupt

The DMA master may be configured to generate an interrupt request at the end of each video frame written to memory.

Avalon Register Slave

The Avalon register slave interface gives access to control and status registers to configure the operation of the video input module.

Table 3 shows the registers. All registers are 32-bit aligned to a word boundary. Unused bits should be written as zero.

Table 3. Registers

Address (h)	Access	Mnemonic	Name
`00`	W	`CR`	Control register.
`00`	R	`SR`	Status register.
`10`	RW	`CAMXLEN`	Camera line length.
`20`	RW	`XCLIPS`	Horizontal clipping start pixel.
`24`	RW	`XCLIPE`	Horizontal clipping end pixel.
`28`	RW	`YCLIPS`	Vertical clipping start line.
`2C`	RW	`YCLIPE`	Vertical clipping end line.
`30`	RW	`XSCALE`	Horizontal (X) scale factor.
`34`	RW	`XLEN`	Clipped line length.
`38`	RW	`YSCALE`	Vertical (Y) scale factor.
`3C`	RW	`YLEN`	Clipped height.
`40`	W	`MCONTROL`	Master control register.
`40`	R	`MSTAT`	Master status register.
`44`	RW	`MINTEN`	Master interrupt enable register.
`48`	RW	`MICR`	Master interrupt clear register.
`50`	RW	`FBSTART`	Frame buffer start address.

Control Register (CR)

Table 4 shows the control register format.

Table 4. Control Register Format

Bit	Mnemonic	Description
0	`CB`	0 = input to clipping block comes from component video interface. 1 = input to clipping block comes from color-bar generator.
31:1	0	–

Status Register (SR)

Table 5 shows the status register format.

Table 5. Status Register Format

Bit	Mnemonic	Description
0	`CB`	Returns the current state of CB bit in control register.
31:1	0	–

Camera Line Length (CAMXLEN)

Table 6 shows the camera line length register format.

Table 6. Camera Line Length Register Format

Bit	Mnemonic	Description
9:0	`CAMXLEN`	Specifies the line length in pixels of the component video source.
31:10	0	–

Horizontal Clipping Start Line (XCLIPS)

Table 7 shows the horizontal clipping start line register format.

Table 7. Horizontal Clipping Start Line Register Format

Bit	Mnemonic	Description
9:0	`XCLIPS`	Start pixel of region to be selected from input source.
31:10	0	–

Horizontal Clipping End Line (XCLIPE)

Table 8 shows the horizontal clipping end line register format.

Table 8. Horizontal Clipping End Line Register Format

Bit	Mnemonic	Description
9:0	`XCLIPE`	End pixel of region to be selected from input source.
31:10	0	–

Vertical Clipping Start Line (YCLIPS)

Table 9 shows the vertical clipping start line register format.

Table 9. Vertical Clipping Start Line Register Format

Bit	Mnemonic	Description
8:0	`YCLIPS`	Start line of region to be selected from input source.
31:9	0	–

Vertical Clipping End Line (YCLIPE)

Table 10 shows the vertical clipping end line register format.

Table 10. Vertical Clipping End Line Register Format

Bit	Mnemonic	Description
8:0	`YCLIPE`	End line of region to be selected from input source.
31:9	0	–

Horizontal Scale Factor (XSCALE)

Table 11 shows the horizontal scale factor register format.

Table 11. Horizontal Scale Factor Register Format

Bit	Mnemonic	Description
11:0	`FRAC`	Fractional part of scale factor.
15:12	`INT`	Integer part of scale factor.
31:16	0	–

Clipped Pixel Count (XLEN)

Table 12 shows the clipped pixel count register format.

Table 12. Clipped Pixel Count Register Format

Bit	Mnemonic	Description
9:0	`XLEN`	Line length in pixels after clipping and scaling.
31:10	0	–

Vertical Scale Factor (YSCALE)

Table 13 shows the vertical scale factor register format.

Table 13. Vertical Scale Factor Register Format

Bit	Mnemonic	Description
11:0	`FRAC`	Fractional part of scale factor.
15:12	`INT`	Integer part of scale factor.
31:16	0	–

Clipped Line Count (YLEN)

Table 14 shows the clipped line count register format.

Table 14. Clipped Line Count Register Format

Bit	Mnemonic	Description
8:0	`YLEN`	Number of lines per frame after clipping and scaling.
31:9	0	–

Master Control Register (MCONTROL)

Table 15 shows the master control register format.

Table 15. Master Control Register Format

Bit	Mnemonic	Description
0	`EN`	1 = DMA master is enabled.
31:1	0	–

Master Status Register (MSTAT)

Table 16 shows the master status register format.

Table 16. Master Status Register Format

Bit	Mnemonic	Description
2:0	0	–
3	`FB`	Frame buffer done bit is set when a whole frame has been written to memory.
31:4	0	–

Master Interrupt Enable Register (MINTEN)

Table 17 shows the master interrupt enable register format.

Table 17. Master Interrupt Enable Register Format

Bit	Mnemonic	Description
2:0	0	–
3	`FB`	Frame buffer interrupt enable.
31:4	0	–

Master Interrupt Clear Register (MICR)

Table 18 shows the master interrupt clear register format.

Table 18. Master Interrupt Clear Register Format

Bit	Mnemonic	Description
2:0	0	–
3	`FB`	Write 1 to clear frame buffer interrupt.
31:4	0	–

Frame Buffer Start Address (FBSTART)

Table 19 shows the frame buffer start address register format.

Table 19. Frame Buffer Start Address Register Format

Bit	Mnemonic	Description
31:0	`FBSTART`	Frame buffer start address.

Resource Usage

The Avalon video input module consumes approximately 2,300 logic cells when implemented in a Cyclone device. A Stratix or Cyclone II device implementation uses less logic cells if you employ hardware multipliers in the horizontal and vertical scaling blocks.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com

I.S. EN ISO 9001