임베디드시스템설계및실습

실습 6

Video Frame Handling on KAU Computer System (VideoInProg01)

Ref: Video IP Cores for Altera DE-Series Boards

Contents

- *Video_In_Subsystem* in *KAU_Computer*
- Exercise 1: Forwarding Video Input to VGA
- Exercise 2: Using Multiple Video Frame Buffers
- Homework 5, Due 5/9

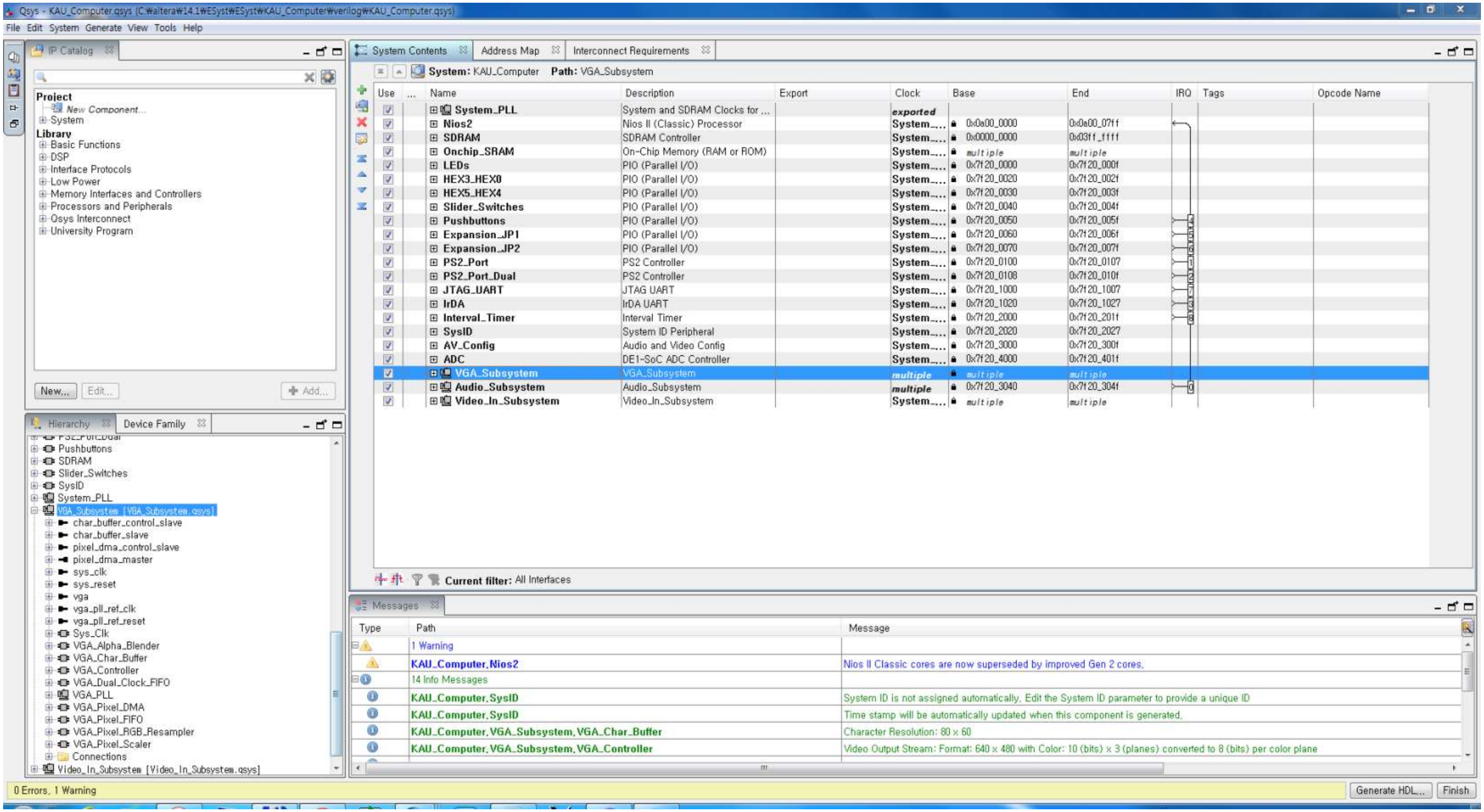
- Schedule
- Term Project – Week 12-15

- Exercise 3: Gray Scale Conversion of Incoming Video
- Exercise 4: VCA Basics – Frame Difference
- Exercise 5: Idea for Hardware/Software Co-Design

Video_In_Subsystem in KAU_Computer

- Video Input Control in KAU_Computer
  - BT656 Video In Decoder
  - Chroma Resampler
  - Color-Space Converter
  - RGB Resampler
  - Clipper
  - Scaler
  - Video DMA Controller

KAU_Computer



System: KAU_Computer **Path:** VGA_Subsystem

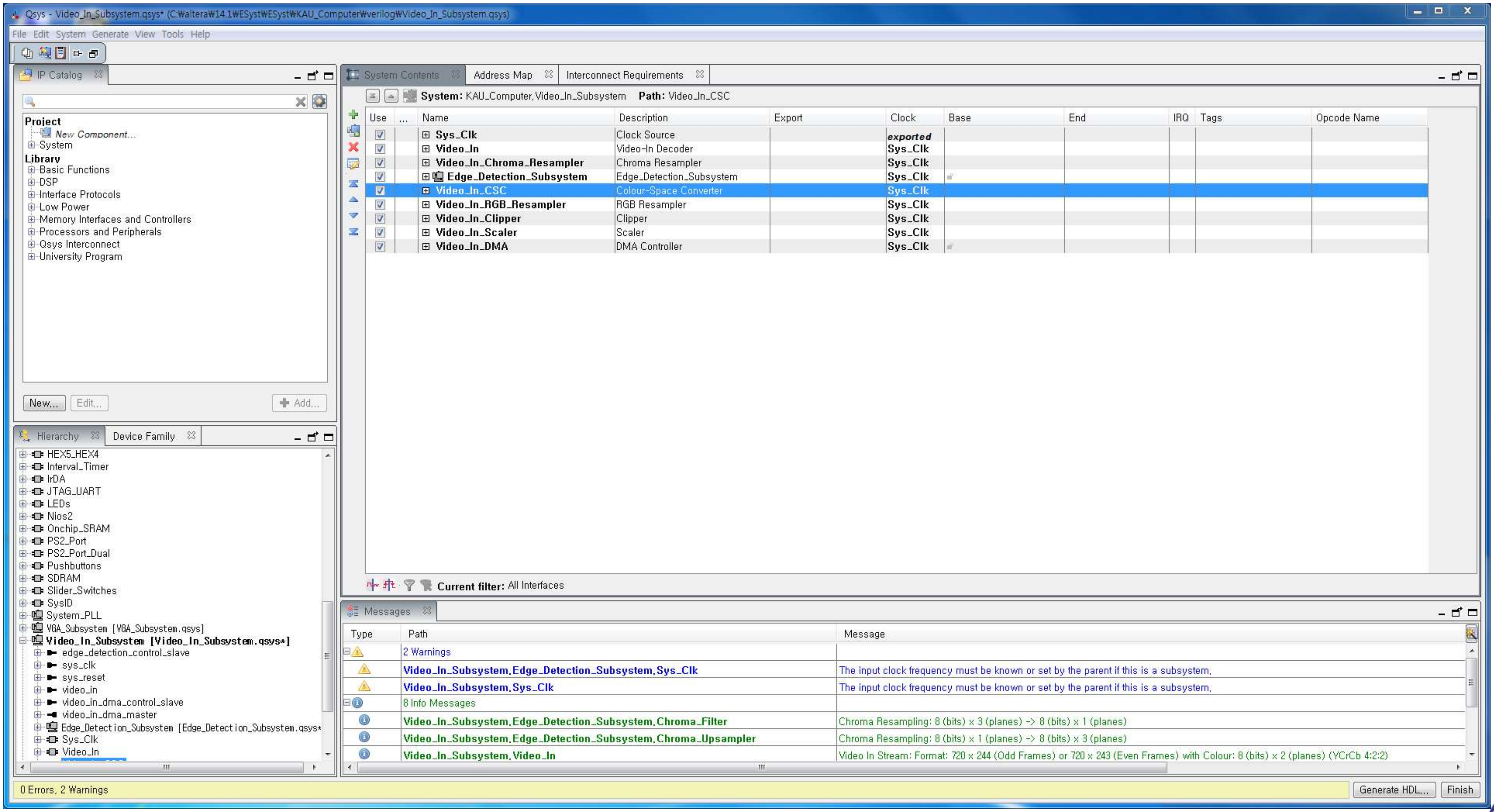
Use	...	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
☑		⊞ System_PLL	System and SDRAM Clocks for ...		*exported*					
☑		⊞ Nios2	Nios II (Classic) Processor		System_...	0x0a00_0000	0x0a00_07ff	←		
☑		⊞ SDRAM	SDRAM Controller		System_...	0x0000_0000	0x03ff_ffff			
☑		⊞ Onchip_SRAM	On-Chip Memory (RAM or ROM)		System_...	*multiple*	*multiple*			
☑		⊞ LEDs	PIO (Parallel I/O)		System_...	0x7f20_0000	0x7f20_000f			
☑		⊞ HEX3_HEX0	PIO (Parallel I/O)		System_...	0x7f20_0020	0x7f20_002f			
☑		⊞ HEX5_HEX4	PIO (Parallel I/O)		System_...	0x7f20_0030	0x7f20_003f			
☑		⊞ Slider_Switches	PIO (Parallel I/O)		System_...	0x7f20_0040	0x7f20_004f			
☑		⊞ Pushbuttons	PIO (Parallel I/O)		System_...	0x7f20_0050	0x7f20_005f	4		
☑		⊞ Expansion_JP1	PIO (Parallel I/O)		System_...	0x7f20_0060	0x7f20_006f	5		
☑		⊞ Expansion_JP2	PIO (Parallel I/O)		System_...	0x7f20_0070	0x7f20_007f	6		
☑		⊞ PS2_Port	PS2 Controller		System_...	0x7f20_0100	0x7f20_0107	1		
☑		⊞ PS2_Port_Dual	PS2 Controller		System_...	0x7f20_0108	0x7f20_010f	2		
☑		⊞ JTAG_UART	JTAG UART		System_...	0x7f20_1000	0x7f20_1007	7		
☑		⊞ IrDA	IrDA UART		System_...	0x7f20_1020	0x7f20_1027	3		
☑		⊞ Interval_Timer	Interval Timer		System_...	0x7f20_2000	0x7f20_201f	8		
☑		⊞ SysID	System ID Peripheral		System_...	0x7f20_2020	0x7f20_2027			
☑		⊞ AV_Config	Audio and Video Config		System_...	0x7f20_3000	0x7f20_300f			
☑		⊞ ADC	DE1-SoC ADC Controller		System_...	0x7f20_4000	0x7f20_401f			
☑		⊞ VGA_Subsystem	VGA_Subsystem		*multiple*	*multiple*	*multiple*			
☑		⊞ Audio_Subsystem	Audio_Subsystem		*multiple*	0x7f20_3040	0x7f20_304f	0		
☑		⊞ Video_In_Subsystem	Video_In_Subsystem		System_...	*multiple*	*multiple*			

Current filter: All Interfaces

Type	Path	Message
⊟⚠	1 Warning	
⚠	KAU_Computer.Nios2	Nios II Classic cores are now superseded by improved Gen 2 cores.
⊟ⓘ	14 Info Messages	
ⓘ	KAU_Computer.SysID	System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID
ⓘ	KAU_Computer.SysID	Time stamp will be automatically updated when this component is generated.
ⓘ	KAU_Computer.VGA_Subsystem.VGA_Char_Buffer	Character Resolution: 80 x 60
ⓘ	KAU_Computer.VGA_Subsystem.VGA_Controller	Video Output Stream: Format: 640 x 480 with Color: 10 (bits) x 3 (planes) converted to 8 (bits) per color plane

0 Errors, 1 Warning

Video_In_Subsystem in KAU_Computer



Qsys - Video_In_Subsystem.qsys* (C:₩altera₩14.1₩ESyst₩ESyst₩KAU_Computer₩verilog₩Video_In_Subsystem.qsys)

Project
- New Component...
- System

Library
- Basic Functions
- DSP
- Interface Protocols
- Low Power
- Memory Interfaces and Controllers
- Processors and Peripherals
- Qsys Interconnect
- University Program

- HEX5_HEX4
- Interval_Timer
- IrDA
- JTAG_UART
- LEDs
- Nios2
- Onchip_SRAM
- PS2_Port
- PS2_Port_Dual
- Pushbuttons
- SDRAM
- Slider_Switches
- SysID
- System_PLL
- VGA_Subsystem [VGA_Subsystem.qsys]
- **Video_In_Subsystem [Video_In_Subsystem.qsys*]**
  - edge_detection_control_slave
  - sys_clk
  - sys_reset
  - video_in
  - video_in_dma_control_slave
  - video_in_dma_master
  - Edge_Detection_Subsystem [Edge_Detection_Subsystem.qsys*
  - Sys_Clk
  - Video_In

System: KAU_Computer.Video_In_Subsystem **Path:** Video_In_CSC

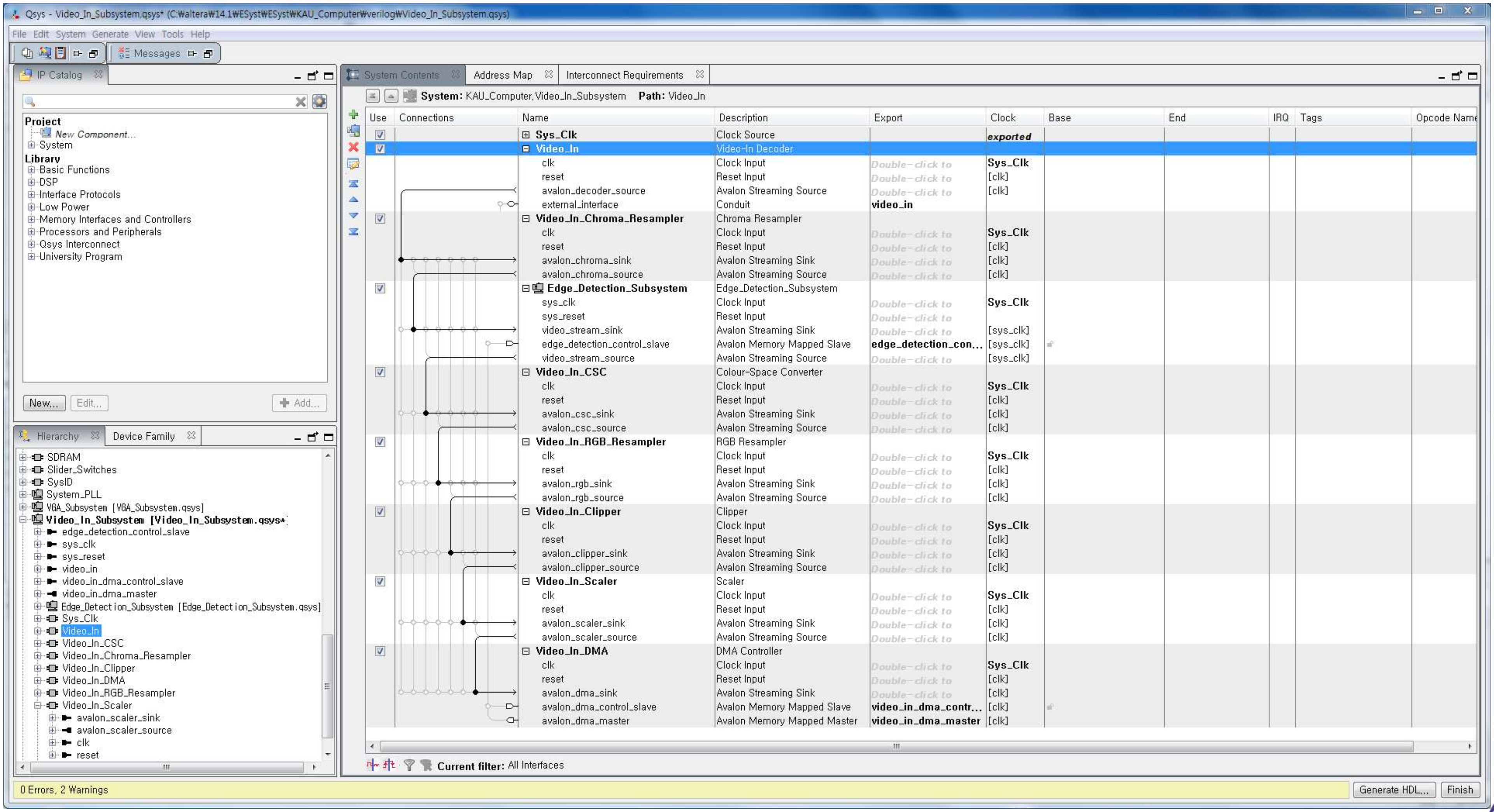
Use	...	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
☑		**Sys_Clk**	Clock Source		***exported***					
☑		**Video_In**	Video-In Decoder		**Sys_Clk**					
☑		**Video_In_Chroma_Resampler**	Chroma Resampler		**Sys_Clk**					
☑		**Edge_Detection_Subsystem**	Edge_Detection_Subsystem		**Sys_Clk**					
☑		**Video_In_CSC**	Colour-Space Converter		**Sys_Clk**					
☑		**Video_In_RGB_Resampler**	RGB Resampler		**Sys_Clk**					
☑		**Video_In_Clipper**	Clipper		**Sys_Clk**					
☑		**Video_In_Scaler**	Scaler		**Sys_Clk**					
☑		**Video_In_DMA**	DMA Controller		**Sys_Clk**					

Current filter: All Interfaces

Type	Path	Message
⚠	2 Warnings	
⚠	**Video_In_Subsystem.Edge_Detection_Subsystem.Sys_Clk**	The input clock frequency must be known or set by the parent if this is a subsystem.
⚠	**Video_In_Subsystem.Sys_Clk**	The input clock frequency must be known or set by the parent if this is a subsystem.
ⓘ	8 Info Messages	
ⓘ	**Video_In_Subsystem.Edge_Detection_Subsystem.Chroma_Filter**	Chroma Resampling: 8 (bits) x 3 (planes) -> 8 (bits) x 1 (planes)
ⓘ	**Video_In_Subsystem.Edge_Detection_Subsystem.Chroma_Upsampler**	Chroma Resampling: 8 (bits) x 1 (planes) -> 8 (bits) x 3 (planes)
ⓘ	**Video_In_Subsystem.Video_In**	Video In Stream: Format: 720 x 244 (Odd Frames) or 720 x 243 (Even Frames) with Colour: 8 (bits) x 2 (planes) (YCrCb 4:2:2)

0 Errors, 2 Warnings

Video_In_Subsystem in KAU_Computer


Qsys - Video_In_Subsystem.qsys* (C:₩altera₩14.1₩ESyst₩ESyst₩KAU_Computer₩verilog₩Video_In_Subsystem.qsys)
File Edit System Generate View Tools Help
Messages
IP Catalog
New... Edit... Add...
Hierarchy
Device Family
System Contents
Address Map
Interconnect Requirements
Generate HDL...
Finish

Project
- *New Component...*
- System

Library
- Basic Functions
- DSP
- Interface Protocols
- Low Power
- Memory Interfaces and Controllers
- Processors and Peripherals
- Qsys Interconnect
- University Program

- SDRAM
- Slider_Switches
- SysID
- System_PLL
- VGA_Subsystem [VGA_Subsystem.qsys]
- **Video_In_Subsystem [Video_In_Subsystem.qsys*]**
  - edge_detection_control_slave
  - sys_clk
  - sys_reset
  - video_in
  - video_in_dma_control_slave
  - video_in_dma_master
  - Edge_Detection_Subsystem [Edge_Detection_Subsystem.qsys]
  - Sys_Clk
  - Video_In
  - Video_In_CSC
  - Video_In_Chroma_Resampler
  - Video_In_Clipper
  - Video_In_DMA
  - Video_In_RGB_Resampler
  - Video_In_Scaler
    - avalon_scaler_sink
    - avalon_scaler_source
    - clk
    - reset

System: KAU_Computer.Video_In_Subsystem **Path:** Video_In

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
☑		⊞ **Sys_Clk**	Clock Source		***exported***					
☑		⊟ **Video_In**	Video-In Decoder							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_decoder_source	Avalon Streaming Source	*Double-click to*	[clk]					
		external_interface	Conduit	**video_in**						
☑		⊟ **Video_In_Chroma_Resampler**	Chroma Resampler							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_chroma_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_chroma_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **Edge_Detection_Subsystem**	Edge_Detection_Subsystem							
		sys_clk	Clock Input	*Double-click to*	**Sys_Clk**					
		sys_reset	Reset Input	*Double-click to*						
		video_stream_sink	Avalon Streaming Sink	*Double-click to*	[sys_clk]					
		edge_detection_control_slave	Avalon Memory Mapped Slave	**edge_detection_con...**	[sys_clk]					
		video_stream_source	Avalon Streaming Source	*Double-click to*	[sys_clk]					
☑		⊟ **Video_In_CSC**	Colour-Space Converter							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_csc_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_csc_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **Video_In_RGB_Resampler**	RGB Resampler							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_rgb_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_rgb_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **Video_In_Clipper**	Clipper							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_clipper_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_clipper_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **Video_In_Scaler**	Scaler							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_scaler_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_scaler_source	Avalon Streaming Source	*Double-click to*	[clk]					
☑		⊟ **Video_In_DMA**	DMA Controller							
		clk	Clock Input	*Double-click to*	**Sys_Clk**					
		reset	Reset Input	*Double-click to*	[clk]					
		avalon_dma_sink	Avalon Streaming Sink	*Double-click to*	[clk]					
		avalon_dma_control_slave	Avalon Memory Mapped Slave	**video_in_dma_contr...**	[clk]					
		avalon_dma_master	Avalon Memory Mapped Master	**video_in_dma_master**	[clk]					

Current filter: All Interfaces

0 Errors, 2 Warnings

Video_In_Subsystem : Overview

- To decode incoming video data and store them to memory
  - Video-In Decoder, Chroma resampler, Color Space Converter, Clipper, Scaler, and Video-In DMA controller
  - The Audio and Video Configuration IP core initializes the ADC with the appropriate settings for use with the UP Video IP cores
- Video-In Decoder IP core
  - The video analog-to-digital converter (ADC) chip on the DE1-SoC board converts video from the composite port and streams it into the FPGA
  - converts the video data from the ADC into Avalon streaming packets and sends them to the video DMA controller
- Video DMA controller
  - writes the video stream to the pixel (or frame) buffer (SRAM or SDRAM)

Video_In_Subsystem : Video Format Conversion

- Video format conversion from Video-In Decoder to Video DMA controller
  - Video-In Decoder : 720 columns by 244 rows with 16 bits per pixel in the 4:2:2 YCrCb color space
  - Video DMA controller : 320 x 240 rows with 16 bits per pixel in the RGB color space for the Pixel DMA in VGA_Subsystem
- Chroma Resampler, Color-Space Converter, RGB Resampler, Clipper and Scaler IP cores are used to perform the conversion

Video_In_Subsystem : Video Format Conversion

- Chroma Resampler
  - converts pixels from the 4:2:2 YCrCb to the 4:4:4 YCrCb formats
  - maintains the frame resolution of 720 x 244
- Color Space Converter
  - converts pixels from 4:4:4 YCrCb to 24-bit RGB color spaces
- RGB Resampler
  - converts the stream between the 24-bit RGB and 16-bit RGB formats
- Clipper
  - trims the stream from 720x244 resolution to a 640 x 240 resolution by dropping the columns and rows around the exterior of the frame
- Scaler
  - reduces the stream to 320 x 240 by dropping every other pixel

Video_In_Subsystem : Video DMA Controller

- Software Programming Model

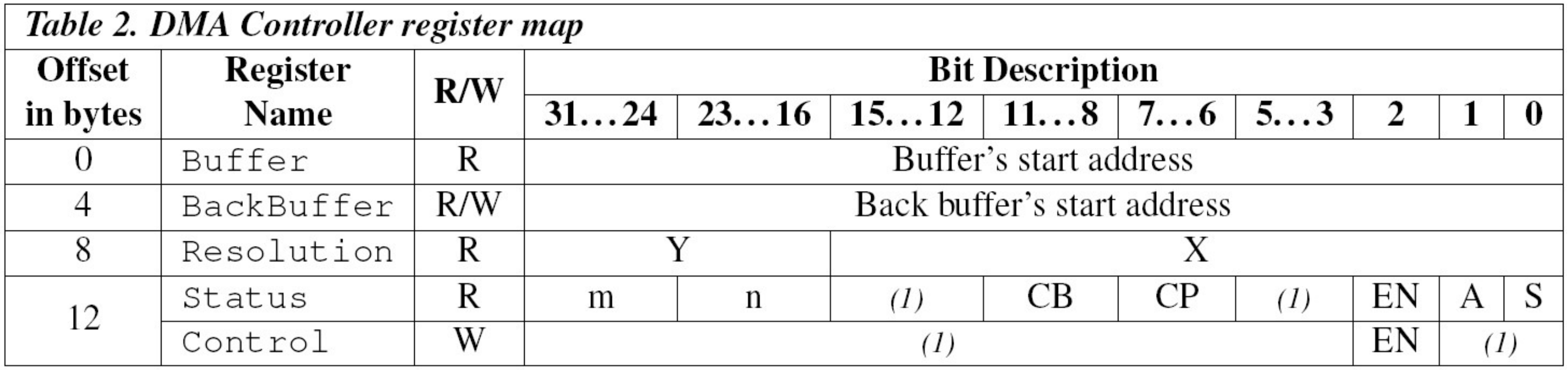


Table 2. DMA Controller register map

Offset in bytes	Register Name	R/W	Bit Description								
			31…24	23…16	15…12	11…8	7…6	5…3	2	1	0
0	Buffer	R	Buffer's start address								
4	BackBuffer	R/W	Back buffer's start address								
8	Resolution	R	Y		X						
12	Status	R	m	n	*(1)*	CB	CP	*(1)*	EN	A	S
	Control	W	*(1)*						EN	*(1)*	

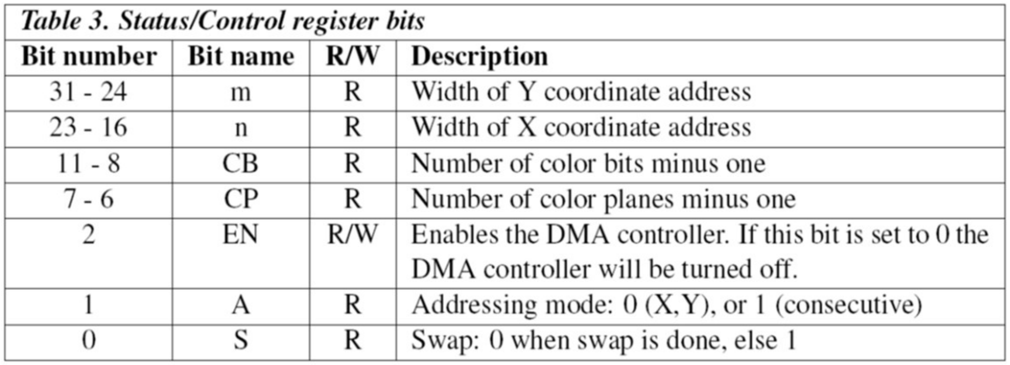
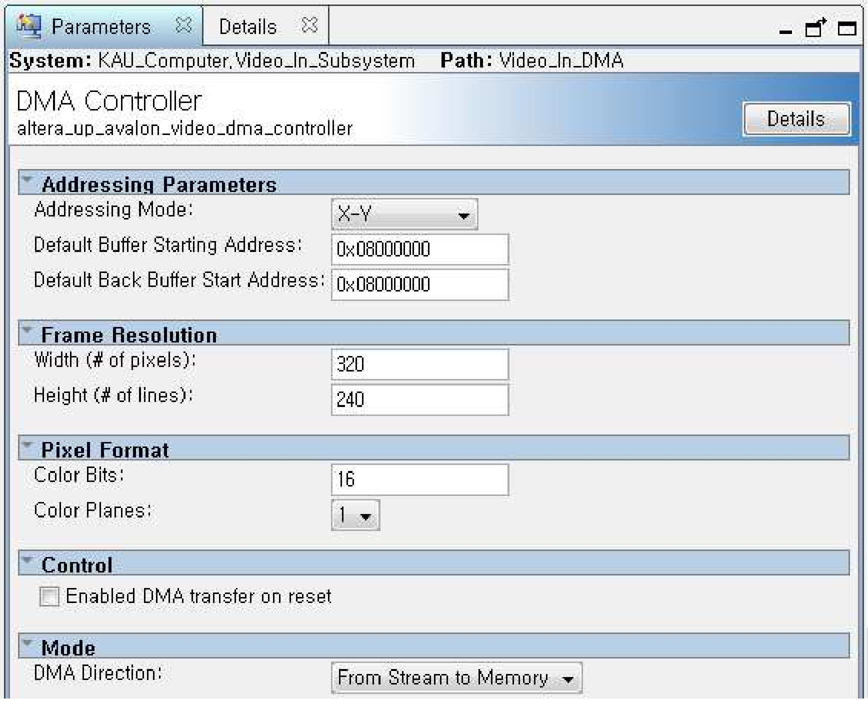


Table 3. Status/Control register bits

Bit number	Bit name	R/W	Description
31 - 24	m	R	Width of Y coordinate address
23 - 16	n	R	Width of X coordinate address
11 - 8	CB	R	Number of color bits minus one
7 - 6	CP	R	Number of color planes minus one
2	EN	R/W	Enables the DMA controller. If this bit is set to 0 the DMA controller will be turned off.
1	A	R	Addressing mode: 0 (X,Y), or 1 (consecutive)
0	S	R	Swap: 0 when swap is done, else 1



System: KAU_Computer.Video_In_Subsystem **Path:** Video_In_DMA

DMA Controller
altera_up_avalon_video_dma_controller

Addressing Parameters

Addressing Mode: X-Y

Default Buffer Starting Address: 0x08000000

Default Back Buffer Start Address: 0x08000000

Frame Resolution

Width (# of pixels): 320

Height (# of lines): 240

Pixel Format

Color Bits: 16

Color Planes: 1

Control

□ Enabled DMA transfer on reset

Mode

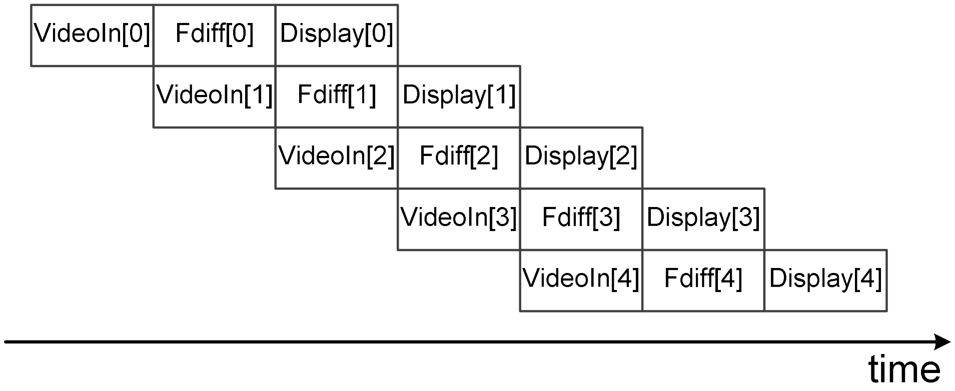
DMA Direction: From Stream to Memory

Exercise 1: Forwarding Video Input to VGA

- This is a simple test designed to store the incoming video data stream to a certain memory location
- Unfortunately, the Altera University Program (UP) does not provide the device driver for the *Video DMA Controller (DMAC)*
  - Read the software programming model carefully
  - Specify the address of a video frame buffer which can be shared with the *Pixel DMA Controller* for VGA display
  - You may need to enable the Video DMAC or swap the buffers
- To verify that you stored the video data correctly, display the stored video data to LCD display via the *VGA_Subsystem* in *KAU_Computer*
  - Check if the video captured by the camera is presented on the LCD display

Exercise 2: Using Multiple Video Frame Buffers

- There exist multiple tasks to be handled simultaneously in a video analytics pipeline
  - For frame differencing, you need to handle Video-In DMAC, Frame Differencing, Pixel DMAC simultaneously

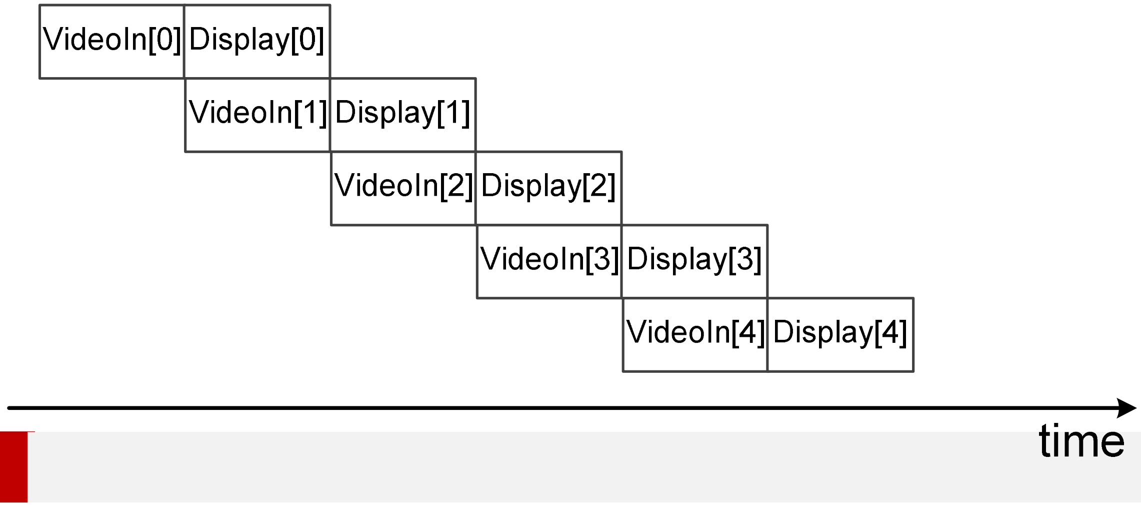


VideoIn[0]	Fdiff[0]	Display[0]				
	VideoIn[1]	Fdiff[1]	Display[1]			
		VideoIn[2]	Fdiff[2]	Display[2]		
			VideoIn[3]	Fdiff[3]	Display[3]	
				VideoIn[4]	Fdiff[4]	Display[4]

  - Frame differencing (Fdiff[n]) is the task calculating pixel difference between the current and the previous frames
  - Then, how many frame buffers do you need for this pipeline? You may assume the frame differencing task can be executed in real-time. Also, data overwriting should be prevented before the data is presented or used for calculation

Exercise 2: Using Multiple Video Frame Buffers

- Define the frame buffers you need for the video analytics pipeline in SRAM or SDRAM memory
- For simplicity, we skip the Frame Difference Task in the pipeline
- Control Video Input and Output Tasks
  - Control the Video DMA Controller and the Pixel DMA such that incoming video frame is first stored to a frame buffer and read for LCD display
  - The Video DMAC and the Pixel DMAC should change their frame buffer pointers to other one to avoid data overwriting or corrupted data display



Homework 5, Due 5/9

- Inspect the IP modules from the Video-In Decoder to the Video DMA Controller in *Video_In_Subsystem* of *KAU_Computer*
  - Read the data sheet for the IP modules
    - Video-In Decoder
    - Chroma resampler, Color Space Converter, Clipper, Scaler
    - Video-In DMA controller
  - Summarizes the functions of each modules briefly (2 page in A4)
- Modify the *Video_In_Subsystem* such that it can handle 8-bit gray-scale image frame
  - Name the system as *KAU_Computer_8BGrayIO*
  - Generate & Compile & program it to display gray-scale incoming video

Schedule

- Week 8 : Video Display
- **Week 9 : Video Capture (5/2)**
- Week 10-11 : Term Project Proposal (Due: 5/23)
- Week 11 : Exam (5/16)
- Week 12-15 : Term Project
- Week 15 : Term Project Final Report, Demo Video Authoring, Presentation

Term Project - Week 12-15

- Design anything with video or audio on DE1-SoC board
  - JukeBox, Digital Camera, Digital Picture Frame, Surveillance Camera (Object Extraction), Rear View Warning System, Around-View Monitoring, Oscilloscope, Motor Control with PWM, etc
- You may use other IP cores in Library in Qsys
- Additional IP cores for surveillance camera will be given upon request
  - But, you should first read the following paper
    - S. Kim, et. Al., “Multi-Object Tracking Coprocessor for Multi-Channel Embedded DVR Systems,” IEEE Trans Consumer Electronics, Vol. 58, No. 4, Nov. 2012.
- Team
  - Equal to or less than 3 students
  - Each student should define his or her role in the team, and report about the role in the final report

Term Project - Week 12-15

- Evaluation
  - Proposal : 30%
  - Demonstration : 30% (Youtube Video Upload)
  - Report : 40% (team 20%, contribution of each student 20%)
  - Bonus : max 30%
    - Successfully used other IP cores which were not used in the class
    - Successfully designed an Hardware IP with Avalon ST or MM interfaces, and included in your embedded system

Exercise 3: Gray Scale Conversion of Incoming Video

- By using the system *KAU_Computer_8BGrayIO,* verify the captured video frames are displayed in gray scale on your LCD monitor
- Is there any difference in the amount of memory used for frame buffers compared with that of 16-bit RGB color system?
- How can you interpret the difference in terms of performance, resources, etc?

Exercise 4: VCA Basics - Frame Difference

- Define Three Frame Buffers, FB[0], FB[1], and FB[2], in SRAM or SDRAM in QVGA size (320x240)
  - Gray scale used for video pixels
- Implement the Frame Difference function in software

$$f_d(x,y) = \begin{cases} 1, & \text{if } |f_b(x,y) - f_c(x,y)| > th \\ 0, & \text{otherwise} \end{cases}$$

  - Disable all the DMAs in the *KAU_Computer*
  - Draw certain patterns for FB[0] and FB[1]
  - Measure the elapsed time for frame differencing of two video frames
    - Frame Difference (FB[0], FB[1]) ➔ FB[2]
    - Check if the difference frame is correct
  - Can this software function be used for the video analytics pipeline handling full video frame rate in Exercise 2?

Exercise 4: VCA Basics - Frame Difference

- Repeat the Frame Difference in software for incoming video frames
  - Enable all the DMAs in the *KAU_Computer*
  - The Video DMA should store incoming video frames to FB[0] or FB[1] alternately
  - You may switch the displayed video channel between the incoming video frame and the difference frame
    - By changing the buffer of the Pixel DMA Controller
  - Measure the elapsed time for frame differencing of two video frames
    - Frame Difference (FB[0], FB[1]) ➔ FB[2]
    - Check if the difference frame is correct (You can do this by switching the display channel to the difference frame buffer)
  - Again, can this software function be used for the video analytics pipeline handling full video frame rate in Exercise 2?

Exercise 5: Idea for Hardware/Software Co-Design

- In Exercise 4, is there any idea to enhance the performance, i.e. frame differencing throughput?
  - You may inspect the DMAs and the tasks in the pipeline for enhancement

끝