Département Informatique Electronique

3ème année du cycle d'ingénieurs **Bus et Interfaces**

Teacher: Julien Dubois

Project: Video chain – Quick prototyping and implantation

Goal:

The project aims to implement a completed video chain and perform processing on the video stream. The processing would be described at one stage using High Level Synthesis prototyping tools.

Background knowledge:

What is an image?

An image is an array of pixels (the word pixel comes from **pi**cture (**x**) **el**ements). For a Black and white image, a pixel is defined by one value representing the brightness. If this value is high, the pixel will be bright (white) and if it is low, the pixel will be dark (black). For a colour image, in most cases the pixel is composed of three values: its level of Red, its level of Green and its level of Blue (this is why we usually call colour images RGB images (for Red Green and Blue)). The combination of these 3 values will give the final colour of the pixel. For example, in a lot a software tools, you can create customs colours by setting the value of Red, Green and Blue. As shown in the figure below, if you select the maximum value for Green and Red (255 in this case) and 0 for Blue, you obtain the colour Yellow.

Introduction to Video

A video is a series of images changing at a particular frequency (typically 50 or 60 times per second). When transmitted from a source to a monitor, a video is sent one image at the time. Each image is sent line by line starting with the top line and each line is sent pixel by pixel starting with the left pixel. For example, the figure 1 below shows how a 3x3 pixel image could be sent to a monitor using 3 data lanes (Red Green Blue).

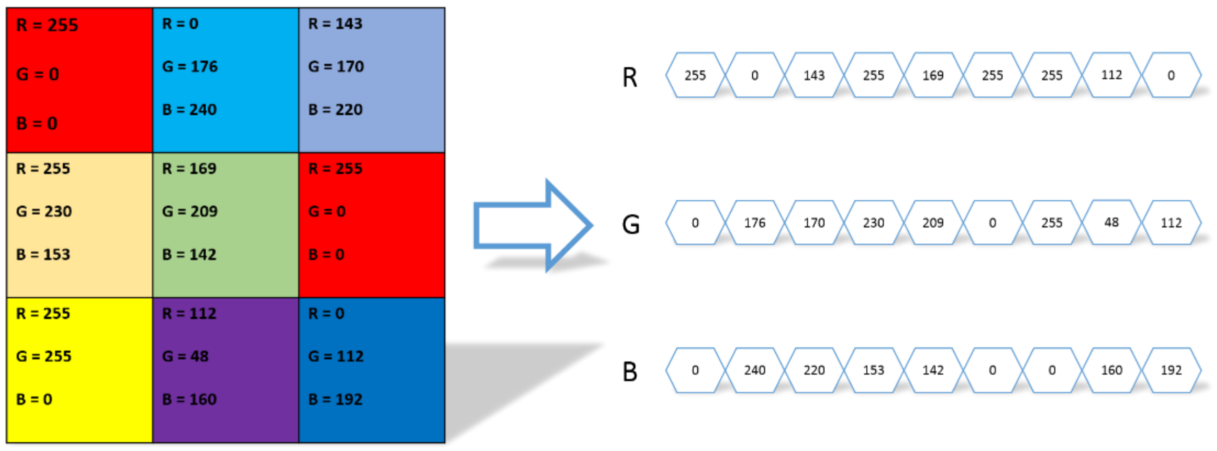


Figure 1: Basic image and corresponding color pixel streams

Along with the pixel values, timing information (carried by the timing signals) is sent to describe the video frame timing. A video frame (one single image of a video) comprises active video and blanking periods.

The timing signals are horizontal and vertical blanking which represent the blanking period (commonly called hblank and vblank) and the horizontal and vertical synchronisation (commonly called hsync and vsync) which happen during the blanking period and are used to indicate when a new line (hsync) or a new frame (vsync) is starting.

Good to know: The time between the start of a blanking period and the start of a synchronization signal is called front porch while the time between the end of a synchronisation signal and the end of a blanking signal is called the back porch.

The figure 2 below shows an example of a video frame along with the synchronization signals.

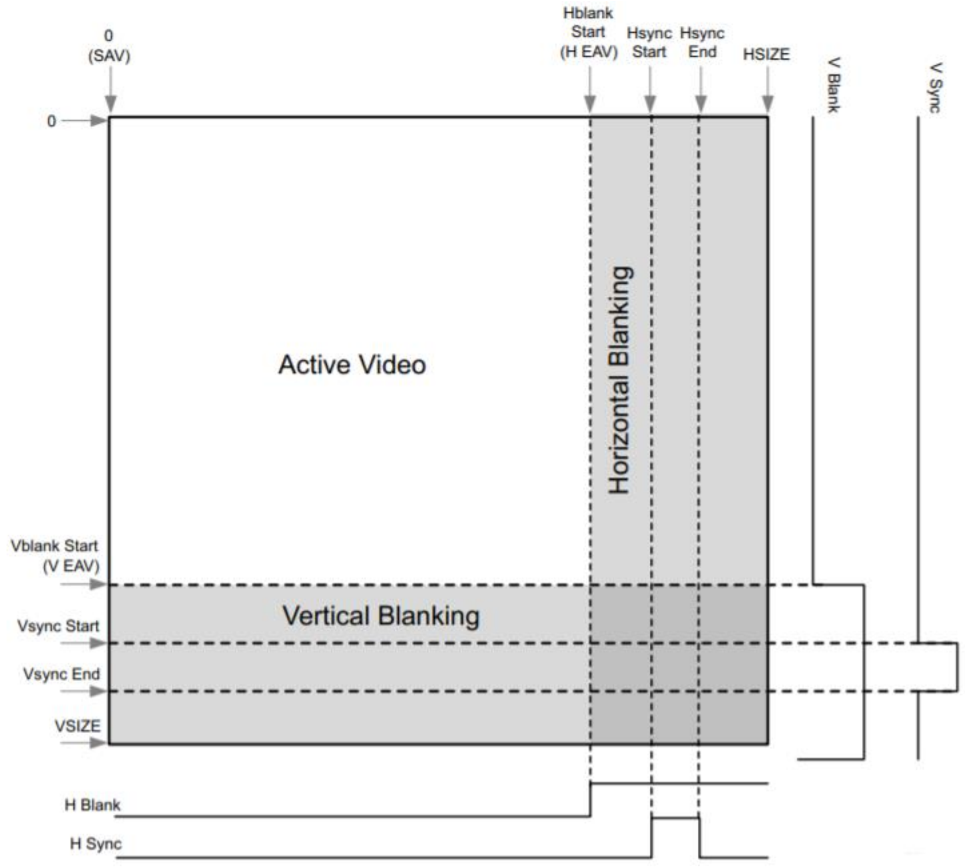


Figure 2: Video synchronization controls

Video systems may utilize different combinations of blanking, synchronization or active signals with various polarities. For example, a VGA interface is only using hsync and vsync.

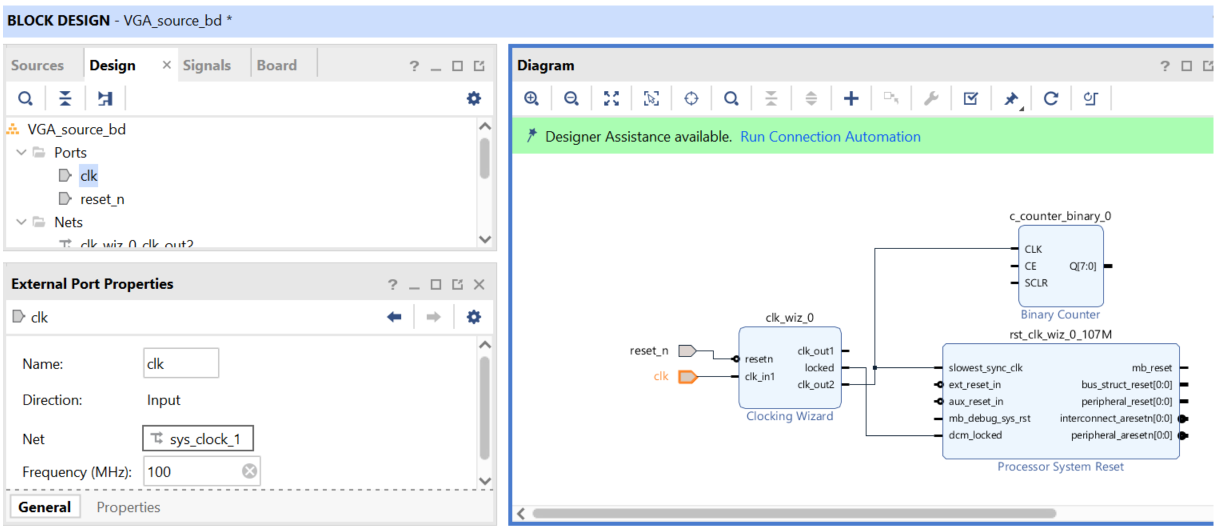
Practical lab:

The main goal is to configure a basic video chain and to apply processing on video stream. The processing would be, first, performed manually and then using High Level Synthesis. First a video stream is setup based on binary counter, generated a grey-level pattern and then display it on VGA interface. Different strategies are presented in this tutorial part to obtain the complete chain. Note that for all the library's components, you can refer to its documentation by clicking on the documentation icon is the configuration window.

Part 1: Configuration of a basic video chain

Create a project selecting Zedboard evalution board. Create Block Design and name it VGA_source_bd. As a first step, we are going to generated video pattern and display via VGA interface. The VGA port is going to be connected to binary counter which will generated a pixel ramp to be displayed.

1) In the Diagram, include a Clock Wizard component. Double click on. In "Output Clocks window" fixes the clk_out1 value to 108 MHz (that could use further on to deal with pixel flow). Select a second clock, clk_out2 to be generated and fix its value to 25.2 MHz. In the same window, go "Enable Optional Input / Outputs for MMCM/PLL and select the reset input and make active low. The second clock is used to generated the ramp evolution.
2) Include a binary counter. Double click on, and fix the output size to 8 bits and add a chip enable and synchronous Clear (respectively named CE and SCLR) to the counter.
3) Check that clk of the binary counter, as well as the clock input of Processor System Reset component (named slowest_sync_clk) is connected the clk_out2 of the clocking wizard component. If not suppress and redraw. Check as well that locked output is connected to dcm_locked input.
4) Rename reset and clock respectively reset_n and clk. Click on the input and use the External Port Properties to do so. The resulting system at this stage is showed in figure 3.

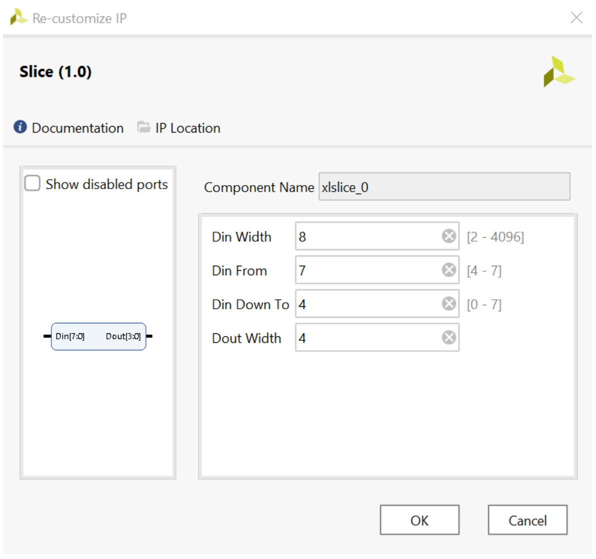


Name: clk
Direction: Input
Net: sys_clock_1
Frequency (MHz): 100

Figure 3: Partial system

5) Connect reset_n to the ext_reset_in imput to enable an external reset to be performed.
6) Add a Video Timing Controller. Double click on it to configure it. Unselect "AXI4-Lite Interface" and "Enable detection" to simplify the component configuration. In the default/constant window, select as a video format a 480p (640x480 active video spatial resolution).

7) Then connect, the clock input of this component to clock_out2. The peripheral_aresetn output is connected to resetn. Click on the vtiming_out interface to show video control signals.
8) Create output ports for hsync_out and vsync_out. Name respectively these ports hsync_out_0 and vsync_out_0. They correspond to horizontal and vertical synchronisation signals which are required to detect the active video. Connect the active_video_out to the CE input of the counter and the horizontal synchronisation of the video. Explain why it makes sense to generate a pattern?
9) Add a slice component. Double click on to configure it. Select only the 4 most significant bits as show figure 4.


Re-customize IP

Slice (1.0)

Documentation   IP Location

Show disabled ports

Component Name xlslice_0

Din Width	8	[2 - 4096]
Din From	7	[4 - 7]
Din Down To	4	[0 - 7]
Dout Width	4	

Figure 4: slice configuration for bit extraction

10) Include three ports on 4 bits respectively named R, G and B. They will correspond to the pixels to be displayed. Connected them to the 4 MSB of the binary counter. The full system for part 1 is generated and presented in figure 5.

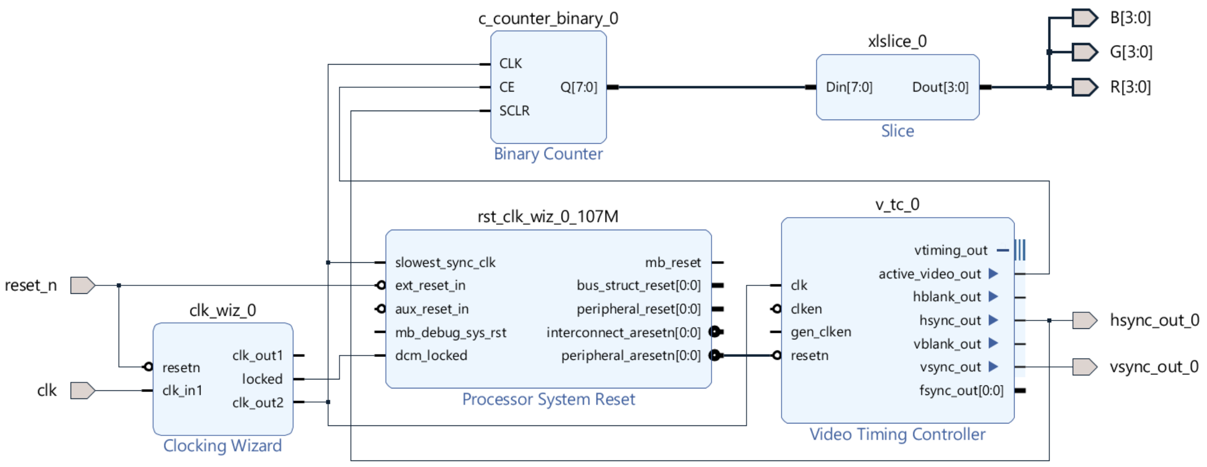
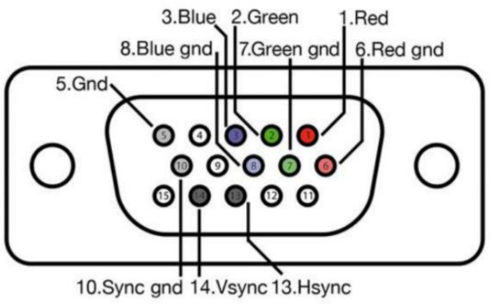


Figure 5: The system's configuration

11) Right click on the block design and generate Create a HDL Wrapper.

12) Include and check the furnished constraint file named Base_line_top.xdc. Check the constraints according the zedboard datasheet and as showed figure 6.



Signal	Zedboard Pin
R	V20 U20 V19 V18
G	AB22 AA22 AB21 AA21
B	Y21 Y20 AB20
Hsync	AA19
Vsync	Y19

Figure 6: VGA physical connectors and pin allocations

13) Now synthetize, perform place and route, generate bit stream and configure the zedboard. The VGA port has to be connected to a screen. Observe the display pattern. According to you, why black bands appears? Propose an easy solution to sort the problem.

14) Modify the bit extraction function as depicted figure 7. The output should be equal to the 4-MSB input when Sel_ActVideo equals one otherwise the output equal zero. The active video signal trigs the Sel_ActVideo. Explain the interest and the potential impact. The IP should be described in VHDL. After the IP validation, include this component into the video chain. Do to so, right click on the vhdl file, and select the “Add Module to Block Design”.

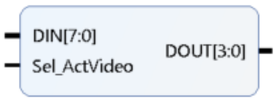


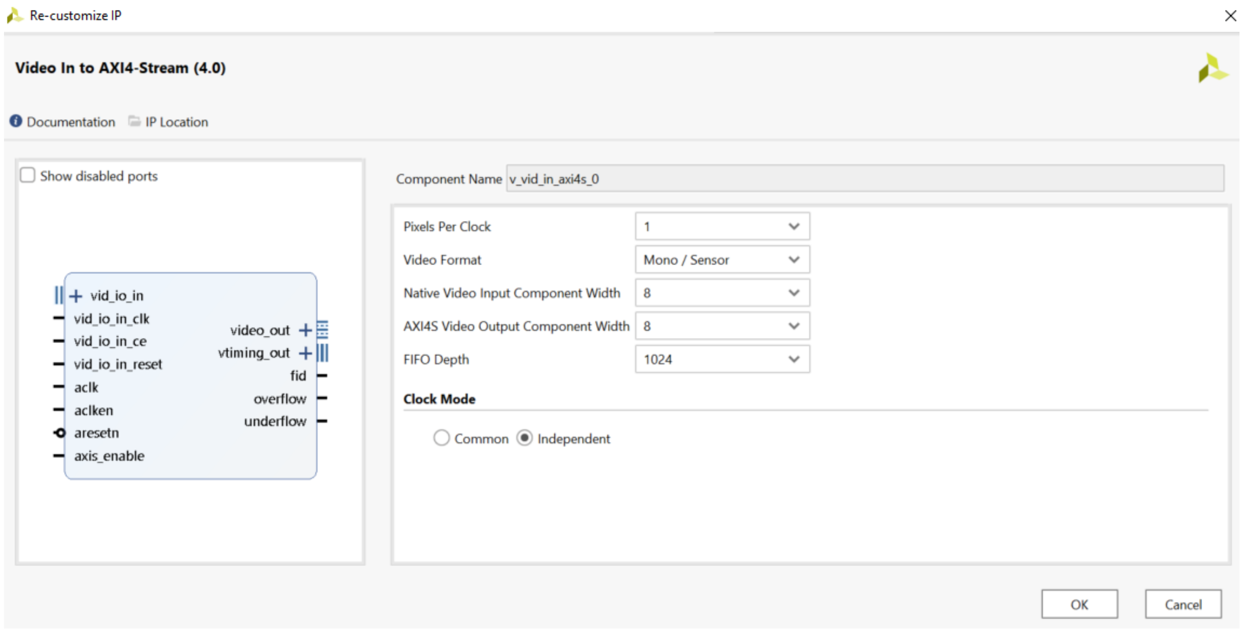
Figure 7: Bit-extraction component sensitive to active video control signal

15) This section aims to introduce hardware debugging as a complement to simulation phase. Create a testbench. Run the simulation. Save screen shots to represent the different phases. Observe the different video signal (blank signal and active video are conformed to what you can expected). Comment. Check using an oscilloscope to check that you obtain the same Hsynch and Vsync. To perform complete validation, use the Xilinx’s Integrated Logic Analyzer (ILA). Do so right click on a wire and select debug. After running connection automation, setup which signals are data and/or trigger. An ILA system is then included. Double click on to configure it. For extra trigger possibilities, select the advanced trigger mode. Warning: this is physical debug, therefore be careful with BRAM memory size and this information is only available after programming the FPGA (please refer to the furnished documentation named ug936). The trigger setup can be done in the Program and Debug section using Hardware Manager after bit-stream generation. Propose a trigger state machine in the debugging window: test first vsync, then hsync, and eventually wait for few cycles before starting trigger.

Part 2: Video chain including AXI4-stream interface

The main idea is to generate a complete video chain including image processing. The processing module should include AXI4-stream interface therefore the previous video chain is going to be modified to support this interface. The AXI-4 stream data interface is a communication standard for internal Xilinx IP.

1. The video and the AXI4-stream frequencies are respectively set to 22.5 and 108 MHz. Firstly, add a “Video In to AXI4-stream” component. The video stream is transformed to respect the AXI4-stream format. Configure the IP to support an 8-bits grey level format. The IP include a FIFO memory that you could set to 1024. Select independent clocks to handle with the two clocks (axi4s and video) as presented in figure 8.


Re-customize IP

Video In to AXI4-Stream (4.0)

Component Name: v_vid_in_axi4s_0

Pixels Per Clock: 1
Video Format: Mono / Sensor
Native Video Input Component Width: 8
AXI4S Video Output Component Width: 8
FIFO Depth: 1024

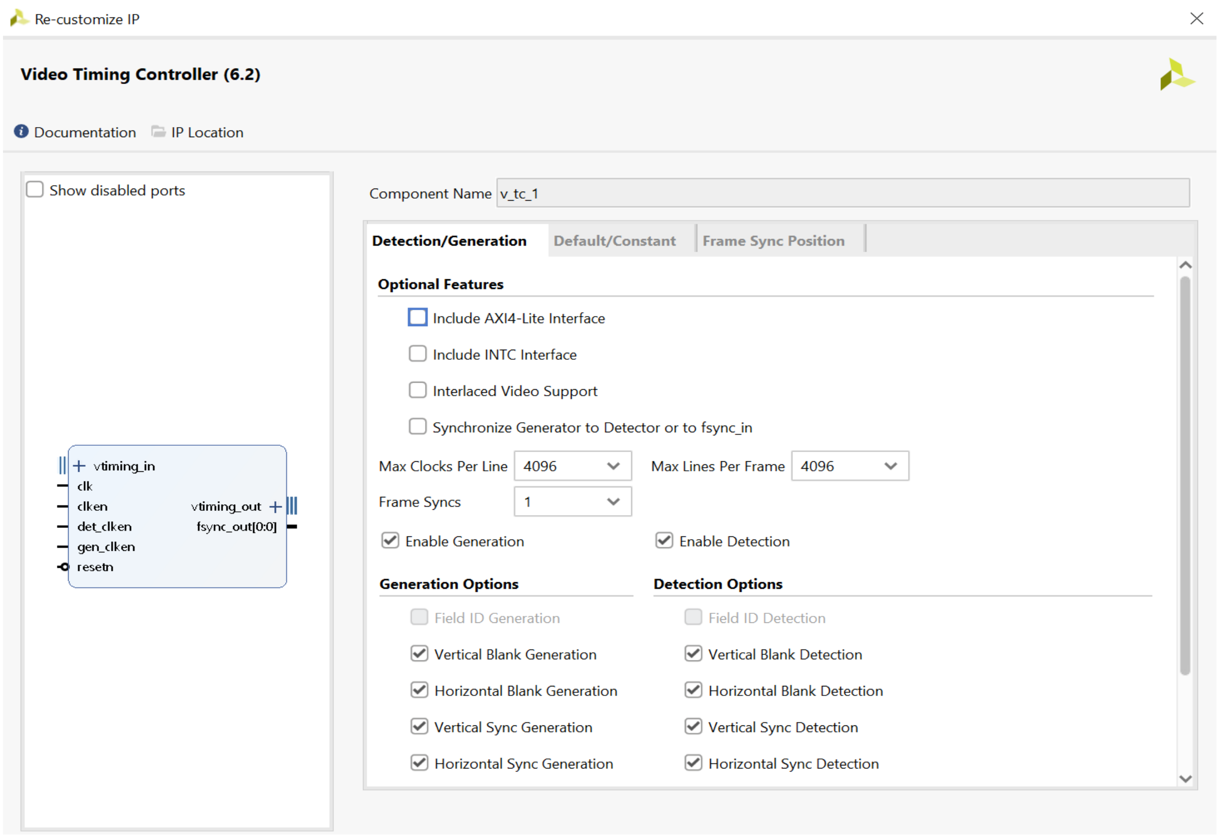
Clock Mode: Common [✓ Independent]

Figure 8: Bit-extraction component sensitive to active video control signal

2. Include a second Video Timing Controller into your design, to detect the input control video signals and to generate the output controls. Double click on the IP, to access to configuration window depicted in figure 9. Deselect the AXI4-Lite. In generation option, check auto generation mode is active. In the default/constant menu select a 480p video format (640x480 pixel spatial resolution for the images).

3. Include a “AXI4-stream to Video Out” component and configure it ad depicted in figure 10. Select a monosensor mode to deal with grey level pixels. Select a pixel per cycle and 8-bits per sample. Use independent clock to manage the axi4 clock separately to the video one. Select a 2048 pixels fifo memory. Explain why the component had to be configure in SLAVE mode?

4. Connect all components and refer to the global system presented in Figure 11 for more details. Propose a testbench, be careful the simulation is then very long. Do it in parallel with other tasks. After how long the system is locked?



Video Timing Controller (6.2)

- [ ] Show disabled ports

Component Name: v_tc_1

Detection/Generation | Default/Constant | Frame Sync Position

Optional Features

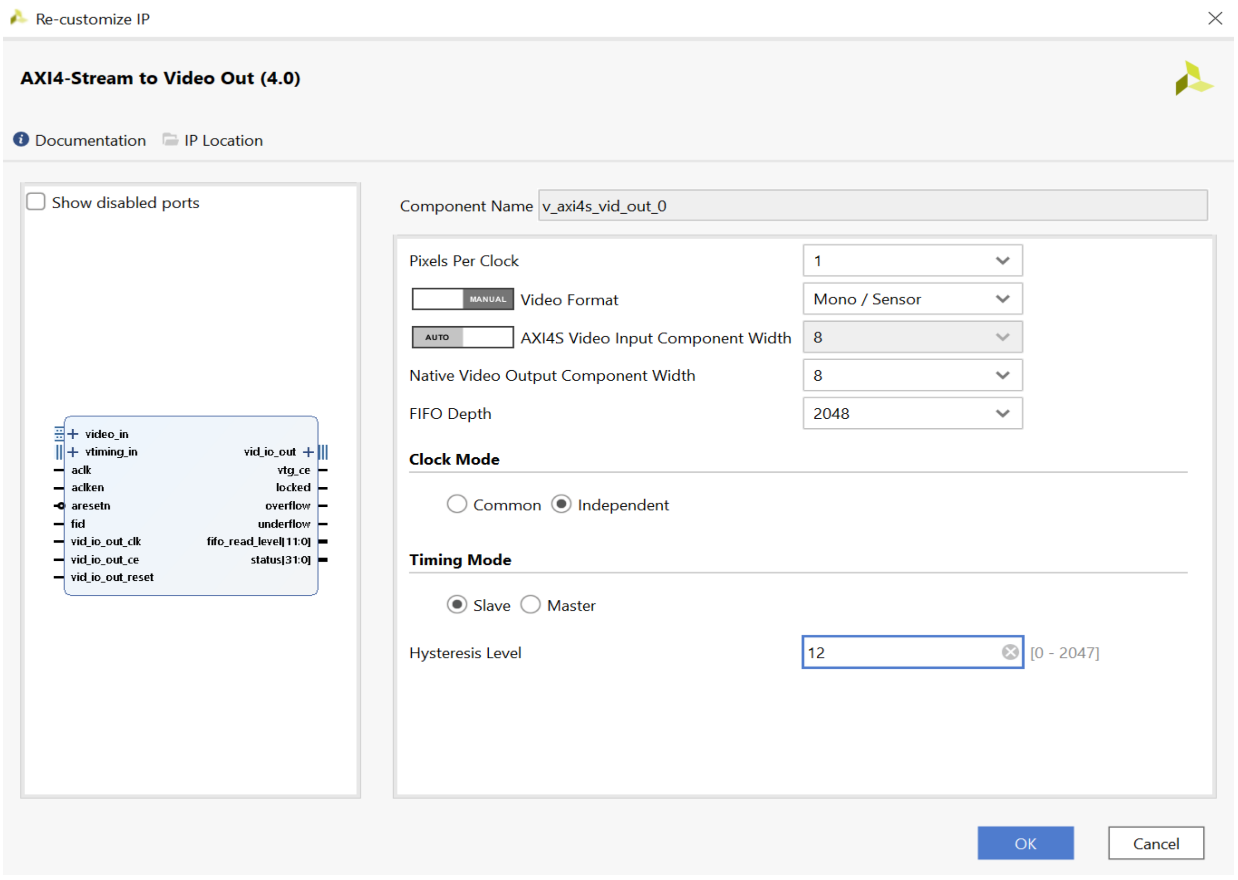
- [ ] Include AXI4-Lite Interface
- [ ] Include INTC Interface
- [ ] Interlaced Video Support
- [ ] Synchronize Generator to Detector or to fsync_in

Max Clocks Per Line: 4096
Max Lines Per Frame: 4096
Frame Syncs: 1

- [x] Enable Generation
- [x] Enable Detection

Generation Options	Detection Options
[ ] Field ID Generation	[ ] Field ID Detection
[x] Vertical Blank Generation	[x] Vertical Blank Detection
[x] Horizontal Blank Generation	[x] Horizontal Blank Detection
[x] Vertical Sync Generation	[x] Vertical Sync Detection
[x] Horizontal Sync Generation	[x] Horizontal Sync Detection

Figure 9. Configuration for the “Video Timing controller”



AXI4-Stream to Video Out (4.0)

- [ ] Show disabled ports

Component Name: v_axi4s_vid_out_0

Pixels Per Clock: 1
Video Format: Mono / Sensor
AXI4S Video Input Component Width: 8
Native Video Output Component Width: 8
FIFO Depth: 2048

Clock Mode

Common [✓ Independent]

Timing Mode

[✓ Slave] Master

Hysteresis Level: 12 [0 - 2047]

Figure 10. Configuration for the “AXI4-Stream to Video Out” component

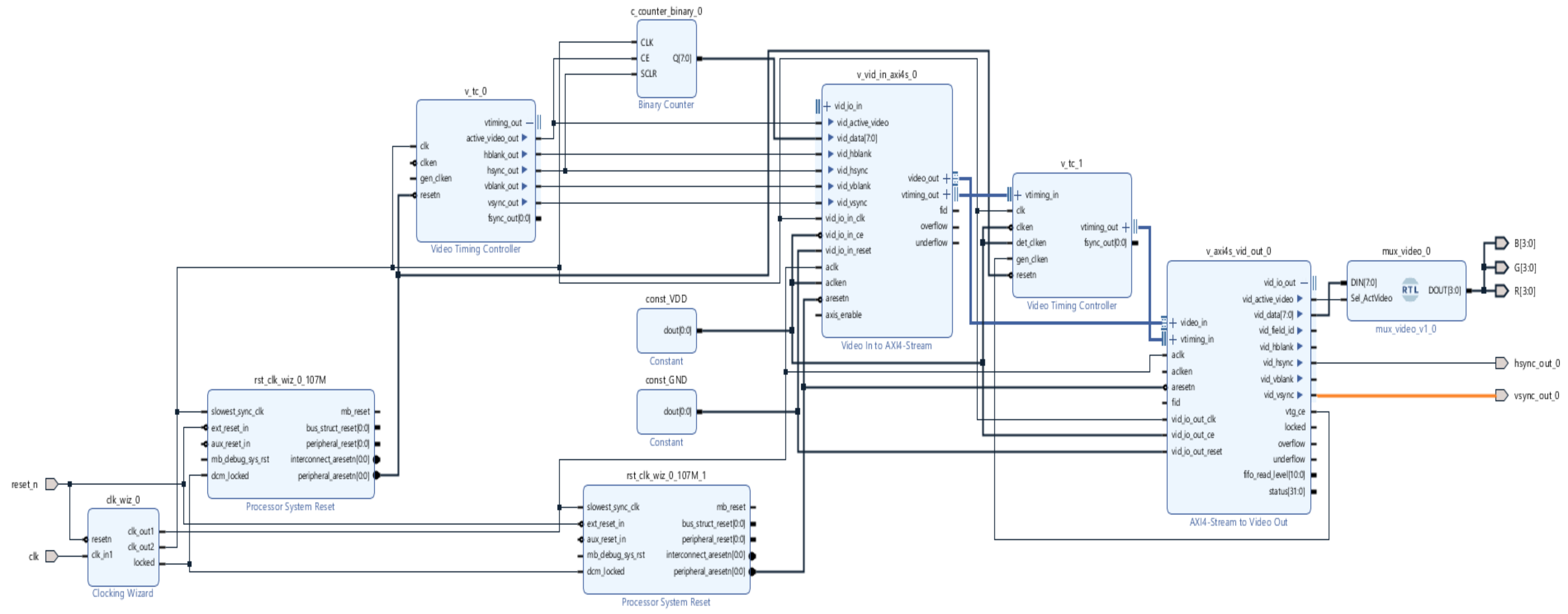


Figure 11. The full video chain

Part 3 - Video Processing and implementation

Processing components are now going to be describe and include into the video chain.

1) To manually generate an IP including an AXI4-stream interface. This IP aims to be include between the “Video In to AXI-stream” and “AXI-stream to Video Out” controllers. The component deals with the control signals to receive the data stream from the first controller, invert the grey-level and the control of the signals to transfer the result data stream to the second controller for the display. Described a VHDL component that includes the ports as depicted in figure 12. The module should include a slave AXI4-stream interface (s00_axis) and a master a master AXI4-stream interface (m00_axis). Please refer to the furnished documentations (ug761 and pg085) if necessary. Moreover, your design should include an input VidOrig_nVideoInv that enable the pixel inversion to be selected (or not). The signal should be connected to the external switch SW1 available on the evaluation board. After the IP validation, include this component into the video chain which has been validated in Part 2. Do to so, right click on the vhdl file, and select the “Add Module to Block Design”. Connected the IP and generate the new architecture and then validate.

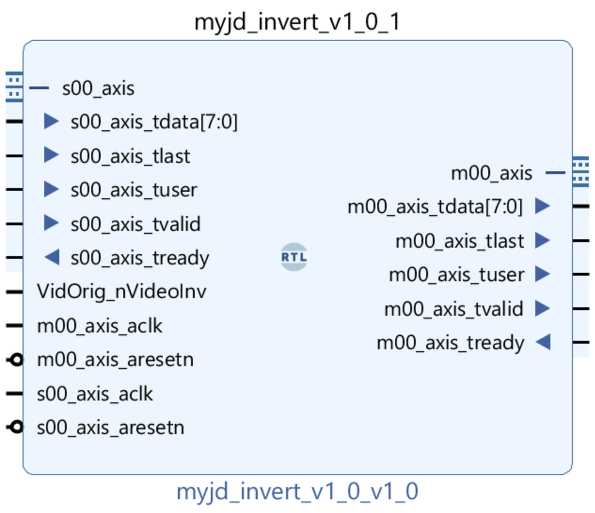


Figure 12. Overview of inversion component

2) Based on the experience that you obtained in the module ITSE51, High Level Synthesis to generate two processing modules. Both components must be described in C/C++ language. Use Vivado HLS to generate VHDL component that should include AXI4-stream interfaces to be included in the final video chain described in part II. The first module should perform grey-level inversion as the component described manually. The second module has to enable include the “binome” number (refer to teacher to obtain this number) in each frame of the video stream.

3) For testing phase, include interconnections and external switches to select on-line the requested configuration (three modes has to be selectable: manually described inversion, HLS inversion, HLS incrustation).