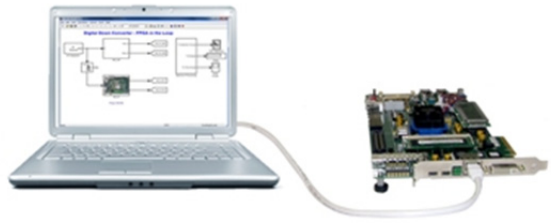
Lab 3: HDL Design Verification using FPGA-in-the-Loop

4 points

This example uses FPGA-in-the-Loop (FIL) simulation to accelerate a video processing simulation with Simulink® by adding an FPGA. The process shown analyzes a simple system that sharpens an RGB video input at 24 frames per second.
At the beginning a predefined model is presented which sharpens the original cat video. Later, the model will be modified. Hence, this example uses the Computer Vision System Toolbox™ in conjunction with Simulink® HDL Coder™ and HDL Verifier™ to show a design workflow for implementing FIL simulation.



FPGA-in-the-Loop for video processing

Software required for this lab:
MATLAB/Simulink R2019a
Toolboxes: Simulink, Fixed-Point Designer, DSP System, Computer Vision System, HDL Verifier, HDL Coder

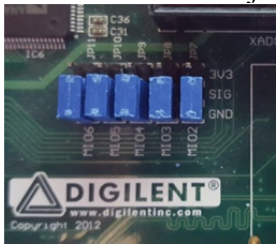
Xilinx Vivado Design Suite 2018.2 WebPACK

Hardware require for this lab:
ZedBoard with Xilinx Zynq SoC
Micro-USB JTAG cable

Step 1: Set Up ZedBoard
Connect the AC power cord to the power plug. Plug the power supply adapter cable into the FPGA development board.

Use the JTAG download cable to connect the FPGA development board with the computer.

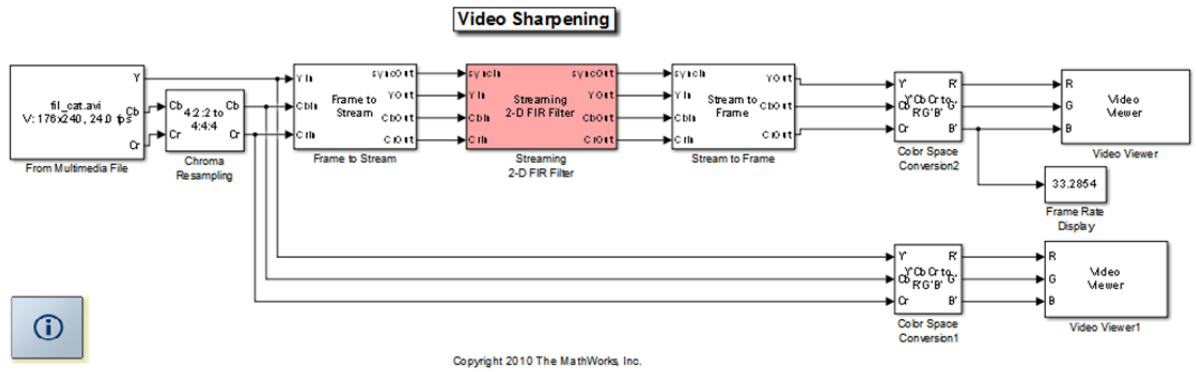
Make sure that the jumpers **JP7-JP11** are in the JTAG position (put to GND)



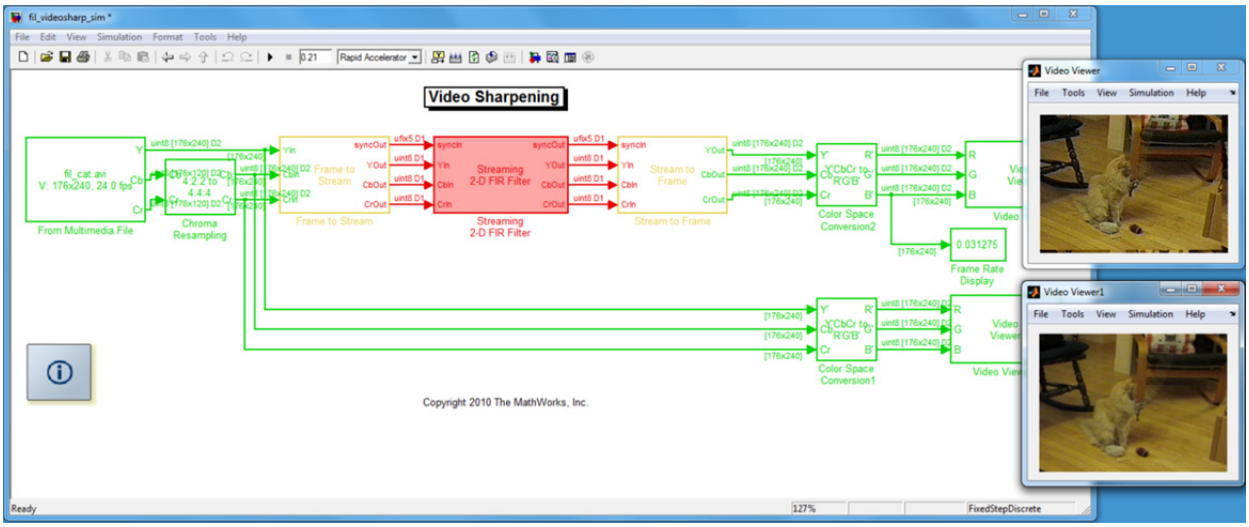
Step 2: Open and Execute the Simulink Model
Start MATLAB. Change folder to your working folder.

```
cd C:\lab_design_test\lab3
```

Open the fil_videosharp_sim.mdl in MATLAB. This is done via the command 'fil_videosharp_sim' which opens the predefined model. Click on Run button to run the simulation for 0.21s.



As you can see in the video viewer, the video sharpening is very slow on the host computer due to large quantity of data process. This shall be improved by using a FPGA-in-the-Loop.



Step 3: Generate HDL Code for FPGA
To shift the FIR filter to the FPGA, make right-click on the block 'Streaming 2-D FIR Filter' and select *HDL Code > Generate HDL for Subsystem*. Check MATLAB Command Window: HDL Code Generation is complete.

Note: If you cannot generate the HDL Code, you can also use the pre-generated HDL files. Copy this files into your current location by the MATLAB command 'copyFILDemoFiles('videosharp');'.

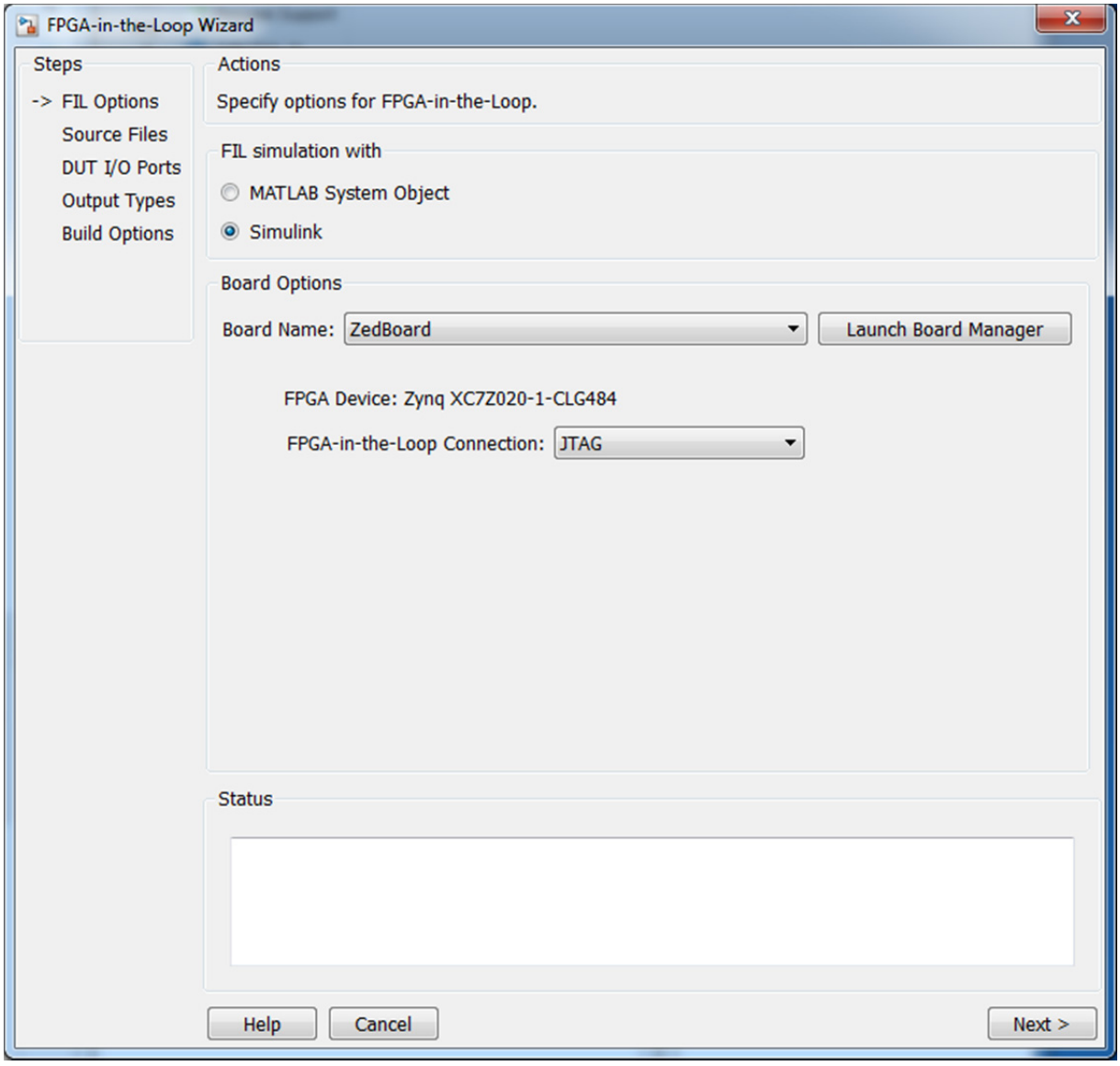
Step 4: Setup the FPGA Design tools
Obviously, the system environment must be setup properly to design the FPGA.
Add Vivado 2018.2 to the system path for the current MATLAB session.

hdlsetuptoolpath('ToolName','Xilinx Vivado','ToolPath',
'C:\Xilinx\Vivado\2018.2\bin\vivado.bat');

Please ensure that the path contains the batch file!

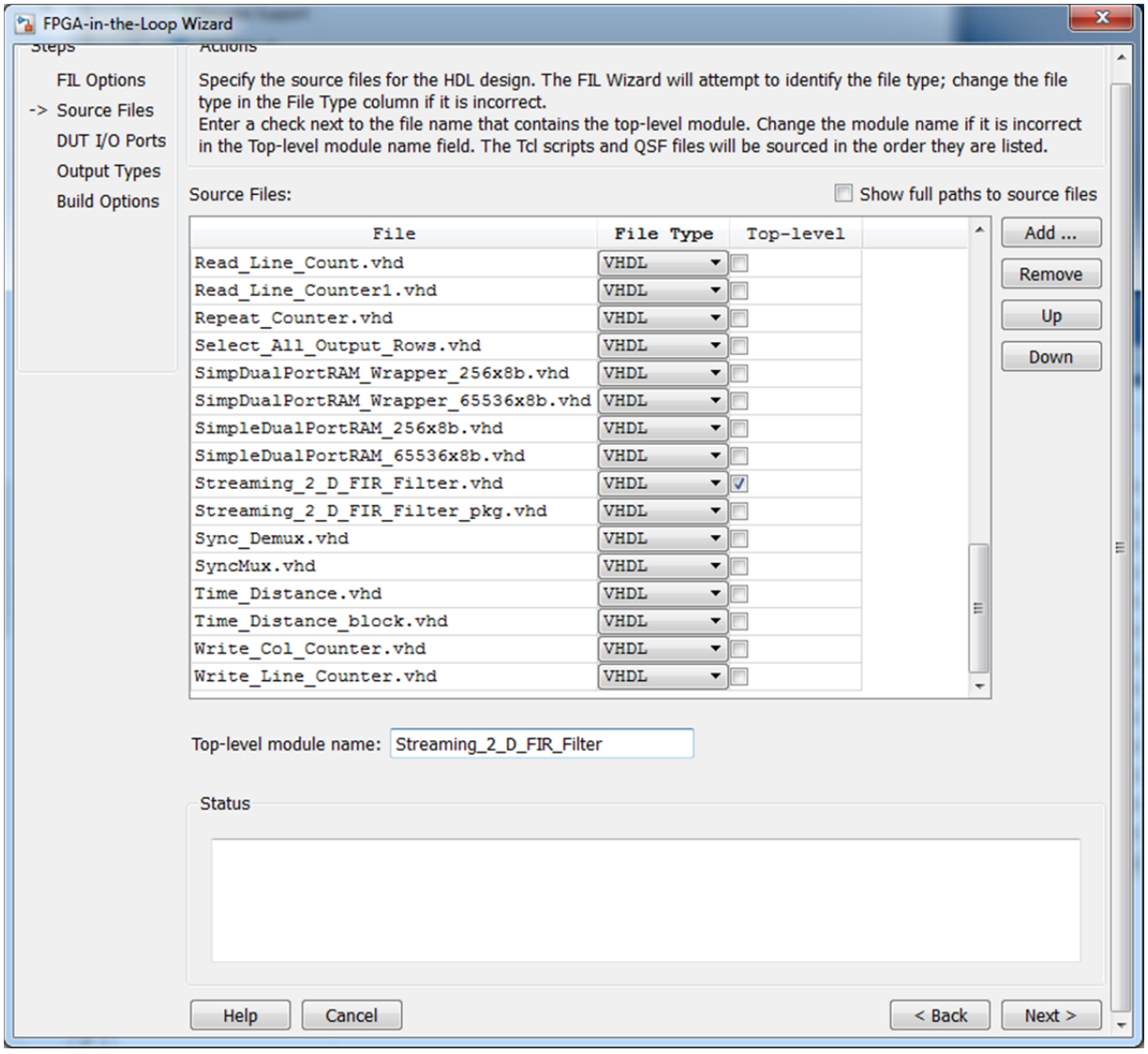
Step 5: Run the FIL Wizard
Type the MATLAB command 'filWizard'. Here you will build the program bitstream for the FPGA.

On the first page, select the FPGA development board 'ZedBoard' which is connected to your host computer via JTAG. If your board is not in the list, click on "Get more boards'. The Support Package Installer will open. Select the HDL Verifier Support Package for Xilinx FPGA Boards and finally install the FPGA board support package. Click cancel to return to FPGA-in-the-Loop Wizard. Select ZedBoard. Click next.

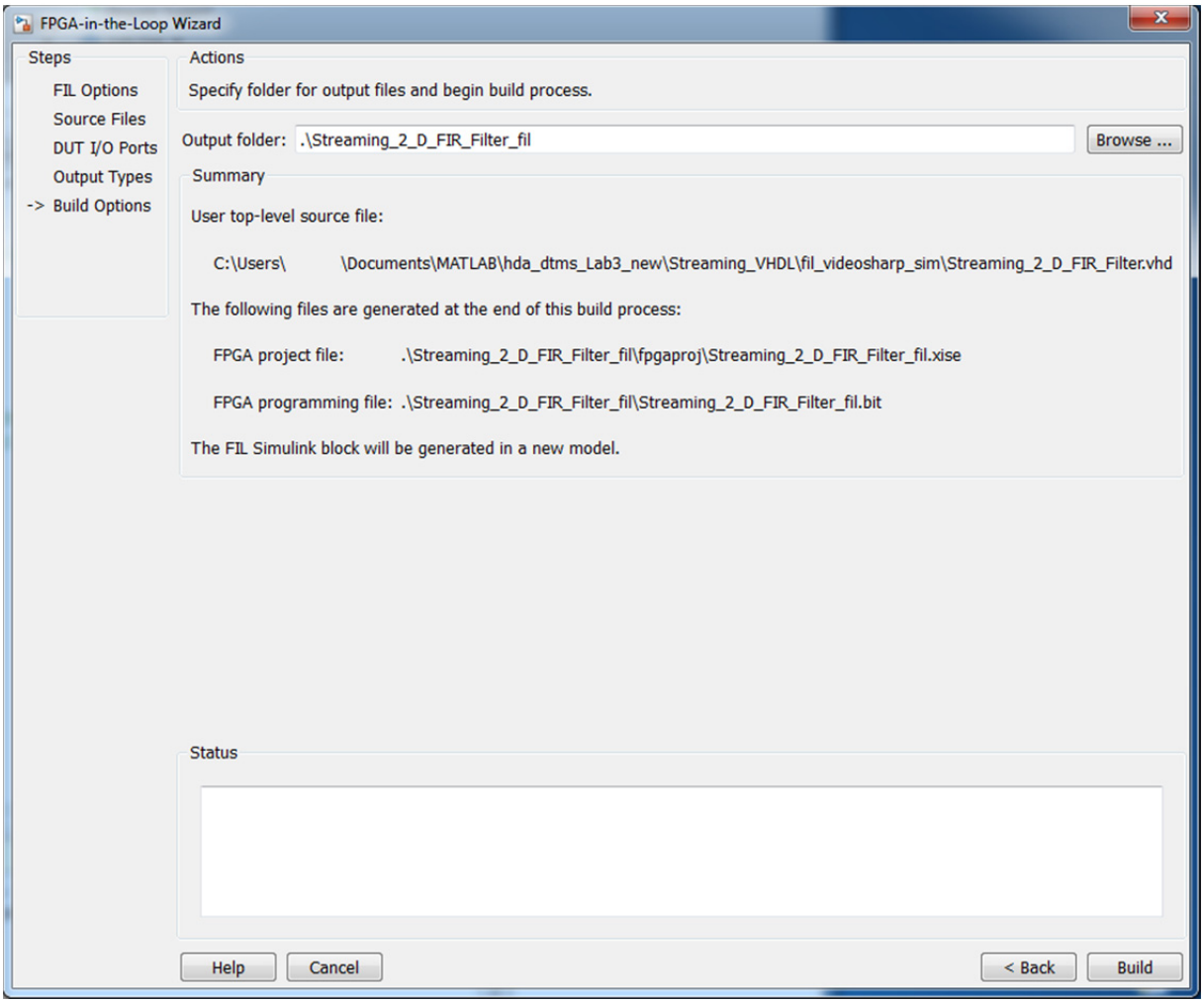


Now include the generated HDL source files from the `Streaming Video Sharpening` subsystem for the HDL design. Click 'Add …' and navigate to

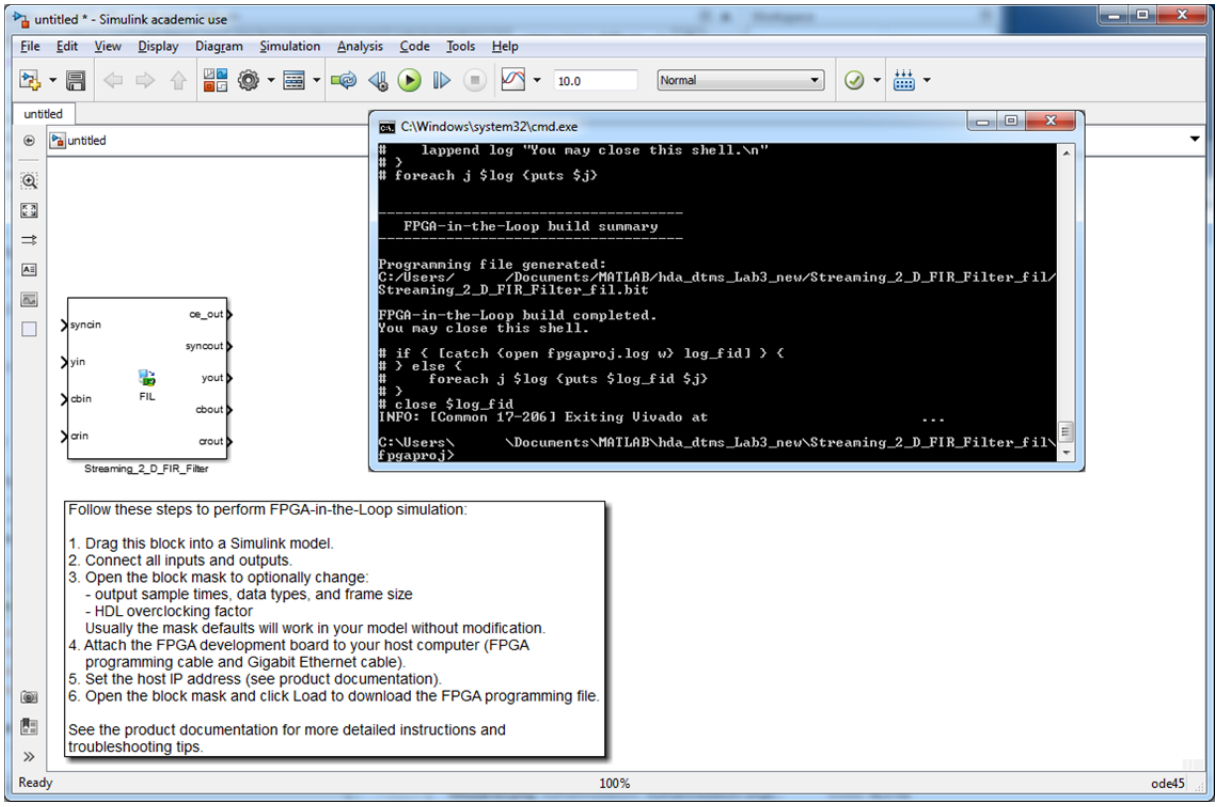
'Streaming_VHDL\fil_videosharp_sim' where all *.vhd files shall be located. Add them all and select the 'Streaming_2_D_FIR_Filter.vhd' file as the Top-Level module. Click Next.



In the next two views 'DUT I/O Ports' and 'Output Types', leave all settings unchanged.



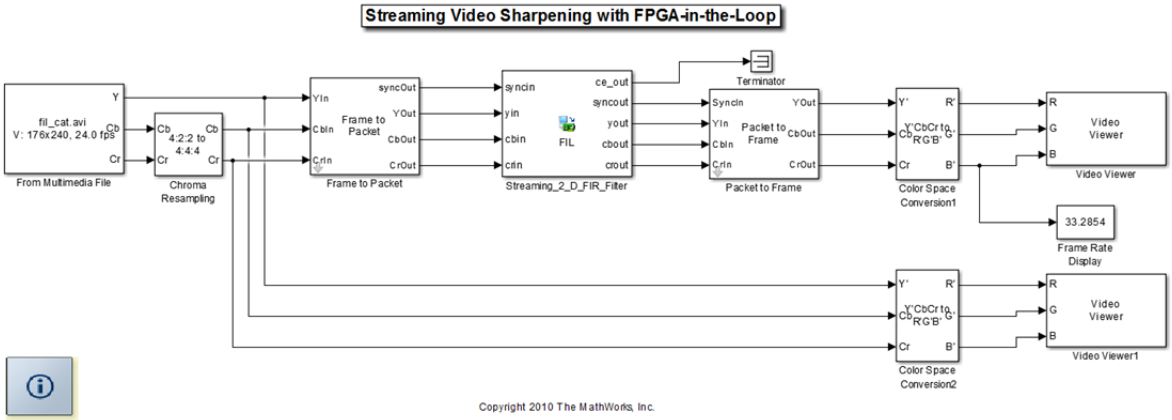
Now, select an output folder for the FPGA bitstream and click 'Build' to build the FIL block and the FPGA programming file. During the build process, the FIL Simulink block, named Streaming_2_D_FIR_Filter, is generated in a new model. Do not close the 'untitled' shell.



After new model generation, the FIL Wizard opens a command shell where the Xilinx Vivado design software performs synthesis, fit, place-and-route, timing analysis, and FPGA programming file generation.
When the FPGA design is done, a message in the command shell lets you know that you can close the shell. Also close the fil_videosharp_sim model shell. But do not close the 'untitled' model which will be applied to the FPGA!

Note: Please be patient, the build process can take up to 15 minutes!

Step 6: Drag VHDL filter block into Simulink model
Type in MATLAB 'fil_videosharp_fpga' to open the origin model for the FPGA version. In this model, replace the subsystem (yellow block) with the previously generated Streaming_2_D_FIR_Filter FIL block. First delete the subsystem, then drag the FIL block



into the Simulink model. Connect all inputs and outputs.
Step 7: Download bitstream to FPGA, run FPGA-in-the-Loop
Connect ZedBoard to host computer via JTAG cable. Switch on power supply.

Double-click the FIL block in the new model and click 'Load' to program the FPGA. Make sure that the File Name is the bitstream which was previously generated in step 5. “FPGA programming file loaded successfully”. Click OK. Close window.

Run the FIL simulation for 10s (click Run in fil_videosharp_fpga). Observe the performance on Video Viewer. Do you see any improvements?

Congratulations, the preparatory work is done. To understand the model a little bit better, the following comprehension questions might help. Please answer to them to get prepared for the practical tasks.

Comprehension questions

a) In the Design Workflow you have done many steps. E.g. you have used the HDL Coder to generate the HDL source files from the Simulink model and you have loaded the generated bitstream to the FPGA. But how does FPGA-in-the-loop verification work? What is the advantage?

b) Describe how basically video sharpening is performed! Which function is transferred to the FPGA?

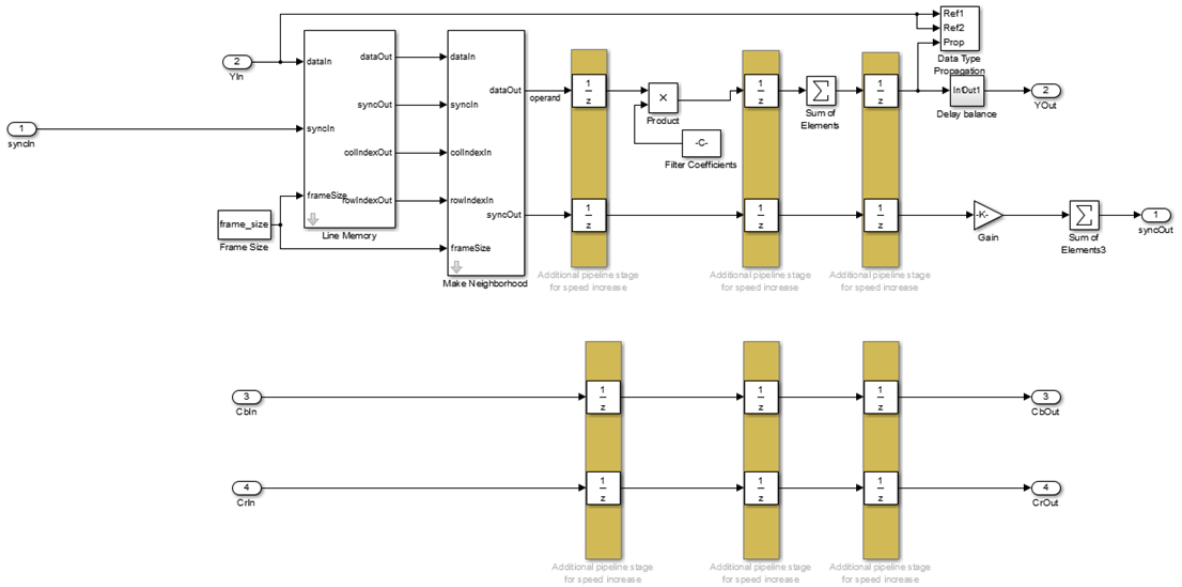
c) Why the Simulink model operates with YCbCr values instead of RGB? Provide the conversion formulas for YCbCr to RGB and vice versa!

d) While reading out the original cat video, the output frame has the format YCbCr 4:2:2. What is the meaning of "4:2:2"? Do internet research on this topic!

e) The chroma resampling block modifies further the chrominance components. Which sampling characteristic exists? How is this done? What impact has it on the frame resolution and why?

f) Which type of FIR Filter is basically used to sharpen the original video? Provide values of the actual filter coefficients (for this check MATLAB code)! Do a plot of filter response (magnitude vs normalized frequency) using MATLAB! What is the filter characteristic?

g) Blocks, which are not part of the Computer Vision System Toolbox and DSP System Toolbox, can be opened to look at the implementation. E.g. the 2-D FIR Filter implementation looks as following:



Describe just basically the structure of the FIR Filter. Why do we need a Memory and Pipeline stages?

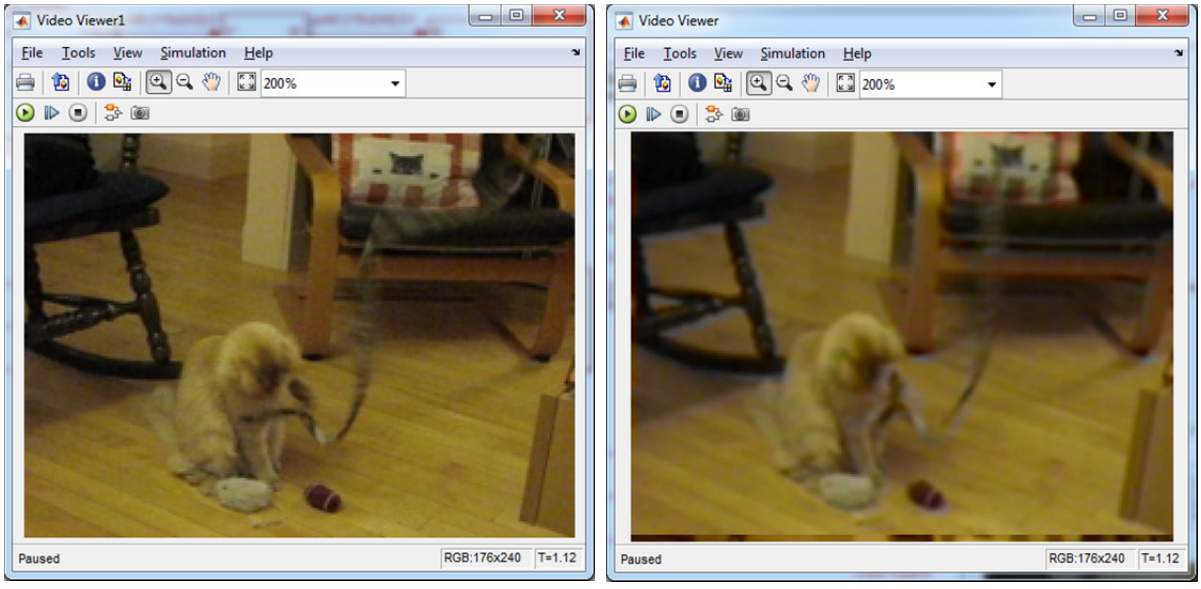
Practical tasks

1. As you have generated the HDL source files and a Vivado project, open it. Typically, its located in '.\Streaming_2_D_FIR_Filter_fil\fpgaproj\Streaming_2_D_FIR_Filter_fil.xpr'.

To understand the relationship between the Simulink FIR implementation and the VHDL sources, compare the filter coefficients in VHDL file with the MATLAB/Simulink model filter coefficients! Do you expect these coefficients in the auto generated VHDL file?

2. After sharpening the video with the FIR Filter, we want to blur the images. For that purpose the filter coefficients of the Streaming 2D FIR Filter needs to be changed. The MATLAB build-in function may help you to find these filter coefficients: Search for 'Sharpen an Image'. Furthermore, find a way to apply the new coefficients to the FIR Filter. For this start with the Simulink model 'fil_videosharp_sim' and double click on the FIR-filter block to define the filter coefficients. Generate VHDL code, execute the FIL wizard again to generate the new FPGA bitstream and load it to the FPGA! Start the new simulation and compare the results.

To get an idea you have an image sample below how the blurred image may have to look.



Original video Blurred video

3. As you got some understanding for the respective components of video processing you have to produce a grayscale output video.

   To have an orientation here you have an image sample how the grayscale image may have to look like.

Prepare before lab: step1-6, questions a) - e)
Note: Tutor will check on your preparation at the beginning of lab! No preparation means you need to leave lab with no points!

Lab report:

1. Perform the workflow for implementing FIL simulation with the predefined sharpening simulation model.
2. Answer all questions of understanding!
3. Do the practical tasks, describe the procedure and make screenshots of the video viewers and your findings! Get the practical tasks 2+3 acknowledged by the tutor.

Each lab group must submit a report (hardcopy) no later than June/28!