ELEC 522, Advanced VLSI
Rice University, Fall 2022
1 September 2022

Due: Friday 15 September 2022 at 11:59pm

Project 1: Xilinx Vitis Model Composer / Nvidia CUDA Tutorial Assignment

Please recall the Honor Code policy which is repeated here: Complete this first Project 1 assignment individually. Solutions should be submitted via Canvas by 11:59pm on the day due to receive potentially full credit. You may freely use the 2022 course notes, the course Canvas resources, or course handouts for the assignments. Students may discuss and compare ideas on the homework assignments, but each student must write up (code and block diagrams, etc.) solutions individually without resort to copying. That is, students must not create submissions together; students may discuss problems but may not actually write up their assignments together. You should also not refer to solutions posted on the web (including Xilinx or Nvidia's websites) or from previous years' classes. Clearly state any assumptions that you make in order to solve the problems and show all requested work.

Project 1 Xilinx Files

Complete the three Xilinx Model Composer / SysGen Tutorial lab assignments posted to Canvas under Files/CAD_Tool_Example_Files/Project1_Xilinx_Files. Please generate your own individual answers. You may discuss the labs with others in the class, but please prepare your own block diagrams so that you become familiar with the design tools. Please do not refer to previous year's solutions either on the web or on old files left on lab computers. You may use the starting template files provided in the zip file listed above.

You should use the lab PCs with the Xilinx Vivado 2022.1 design tools. Each PC has a ZedBoard connected for hardware co-simulation. When using the lab PCs, please make an individual local folder in "C:\elec522\xxx" where "xxx" is your username (Netid) for saving your model files when you are using the lab PCs. This would be local to a particular lab machine. On the lab PCs, your networked Rice "storage" is also mounted as the "U:" drive and you can save the final Simulink .slx files there for safe keeping. Do not "build" or "generate" your FPGA files on a network drive as this will be very slow. You may need to screen capture some output in these assignments.

We will use electronic submission via Canvas instead of paper submission. You can upload your answers to the Canvas "Assignments" section. I suggest a Word .doc or PDF format that would include screen capture of your .slx files and some of the Simulink scope outputs. You should answer the questions listed in each of the Labs. You may wish to upload one large file that identifies each lab and your results and answers to each question. You should also upload your .slx files that are created. You should also upload the .bit file that is part of the co-simulation labs along with the .slx file that was modified for hardware co-simulation.

We will be using the new Xilinx Model Composer tutorials which have been updated for the 2022.1 version. These should work well with the Sysgen in Vivado 2022.1 which is on the lab PCs. Recall that the tools support the Xilinx 7 chips. You **should pick the Zynq xc7z020-1clg484 part** which will be needed for ZedBoard projects whenever a chip is mentioned instead of the chip mentioned in the tutorials. The ZedBoard generation target should be pre-loaded on each of the lab PCs which will select the proper chip.

Some notes on the topics of the labs are below, and what to turn in are:

Lab 1: Lab 1: Introduction to Vitis Model Composer HDL Library. Use Xilinx FIR Compiler block, Use over-sampling to create a more efficient design, Design the same filter using discrete blockset parts, Understand how to work with Data Types such as Floating-point and Fixed-point. Include the final block diagram .slx file, a screen shot of the final block diagram, and scope/spectrum analyzer/resource utilization results from each sub-part as appropriate (not all may be done in that lab) that you did in each subpart. Also compare your results with the tutorial results and note any differences or if the same.

Lab 2: Importing Code into a Vitis Model Composer HDL Design, Create a Finite State Machine using the MCode block in Vitis Model Composer, Import an RTL HDL description into Vitis Model Composer, Configure the black box to ensure the design can be successfully simulated, (Note: We will not do the Vitis HLS part mentioned in Project 1. We will do this part in Project 2 or 3. I have removed it from the PDF, ~~Incorporate a design, synthesized from C, C++ or SystemC using Vitis HLS, as a block into your MATLAB design~~) Include the final block diagram .slx file, a screen shot of the final block diagram, and scope/spectrum analyzer/resource utilization results from each sub-part as appropriate (not all may be done in that lab) that you did in each subpart. Also compare your results with the tutorial results and note any differences or if the same.

Lab 3: Timing and Resource Analysis. Identify timing issues in the HDL files generated by Vitis Model Composer and discover the source of the timing violations in your design. Perform resource analysis and access the existing resource analysis results, along with recommendations to optimize. Include the final block diagram .slx file, a screen shot of the final block diagram, and scope/spectrum analyzer/resource utilization results from each sub-part as appropriate (not all may be done in that lab) that you did in each subpart. Also compare your results with the tutorial results and note any differences or if the same.

Note: These are tutorial labs to understand the tools and tool flow. Please complete them. However, you may have slightly different data results than expected due to perhaps different parameters that you might use. That is OK, just explain to describe the potential cause.

Project 1 NVIDIA CUDA Files.

Refer to the documentation on building, compiling, and running a CUDA project in Visual Studio 2022 on Canvas in the Files/CAD_Tool_Documentation/Visual_Studio_CUDA_start.pdf

This tutorial uses the example of a file printing 0 through 9 on the screen. A new test file for this assignment will be posted on Files/CAD_Tool_Examples/Project1_CUDA_Files.

Build a project with this main.cu file. Debug the project and save an upload a screen shot of the results that you received.