Using Vivado HLS IP with SysGen Demo Script

Introduction

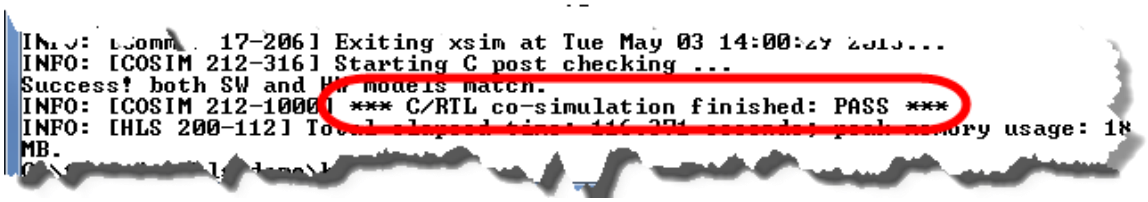
This demonstration script provides high-level instructions on how the RTL created by HLS can be packaged as IP and how the design can be used inside the System Generator for DSP.

Preparation:

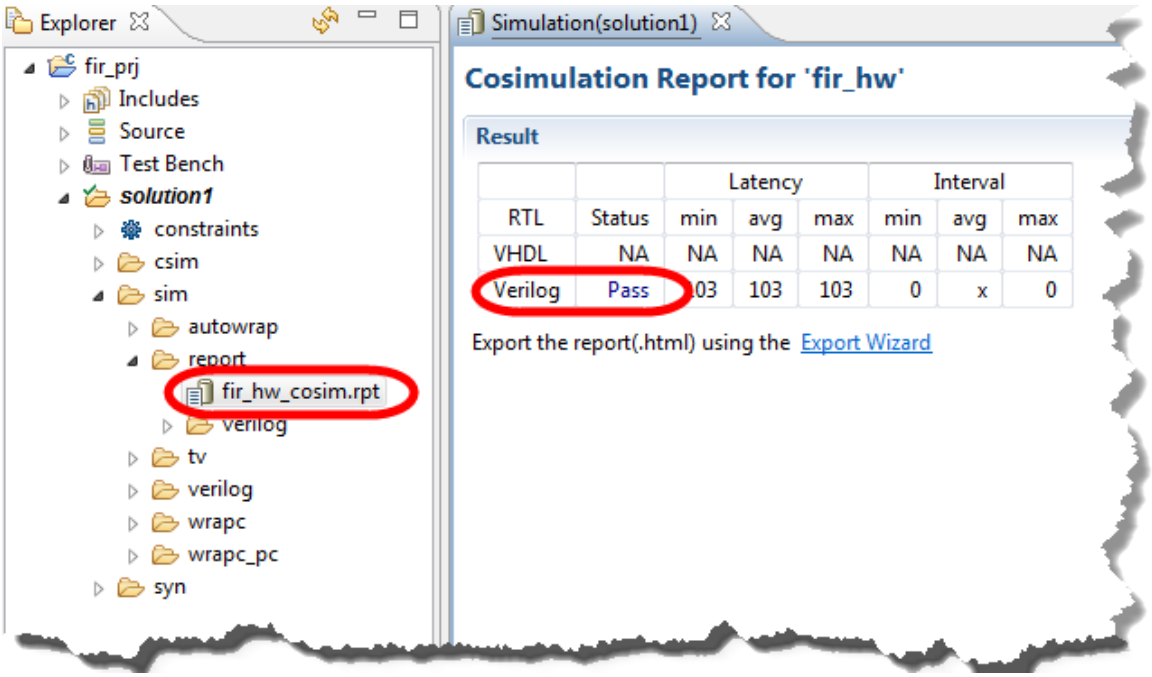
- Required files: Necessary files are located at *C:\training\HLx\demos\hlx_sysgen*
- Required hardware: None
- Supporting materials: None

Using Vivado HLS IP with SysGen

Action with Description	Point of Emphasis and Key Takeaway
• Launch the Vivado® HLS tool command line prompt.	You will run the Tcl script file provided to create a Vivado HLS tool project.
• Change the directory to where the script file is located: `cd C:\training\hls\demos\ hlx_sysgen`	The design files and Tcl file are located in this directory. The sample design is a FIR filter that uses streaming interfaces modeled with the high-level synthesis hls::stream class. The design is fully pipelined at the function level. The optimization directives are embedded into the C code as pragmas.

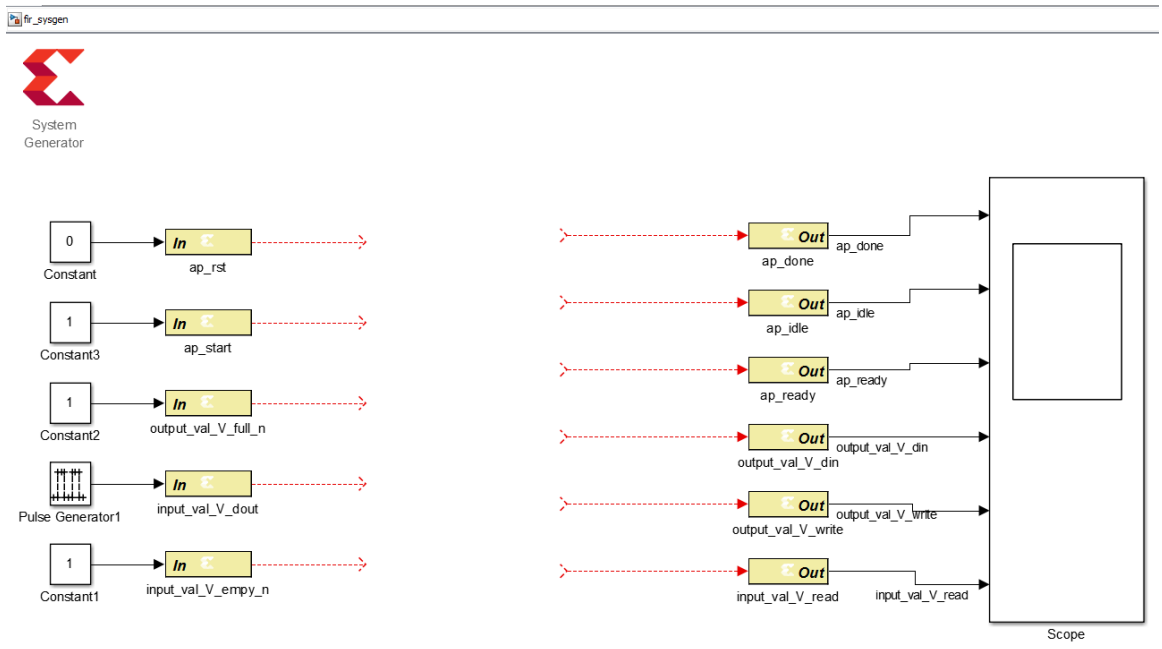
Action with Description	Point of Emphasis and Key Takeaway
• Enter the following command to run the script file: `vivado_hls -f run_hls.tcl`	Before running the script, open and review the `run_hls.tcl` file. This tcl script performs the following: • Creates a new HLS project • Adds the design files • Specifies the top-level function for synthesis • Creates a new solution • Specifies the Xilinx device and clock period • Simulates the design • Synthesizes the design • Co-simulates the design The `export_design` command is commented out. The exporting will be done by opening the HLS GUI.
• You will notice that the co-simulation finished with the results displayed. 	
• Launch the Vivado HLS tool and open the project created from the directory *C:\training\hls\demos\hlx_sysgen\fir_prj*.	Review the Vivado HLS tool project created by the Tcl script in the previous step.

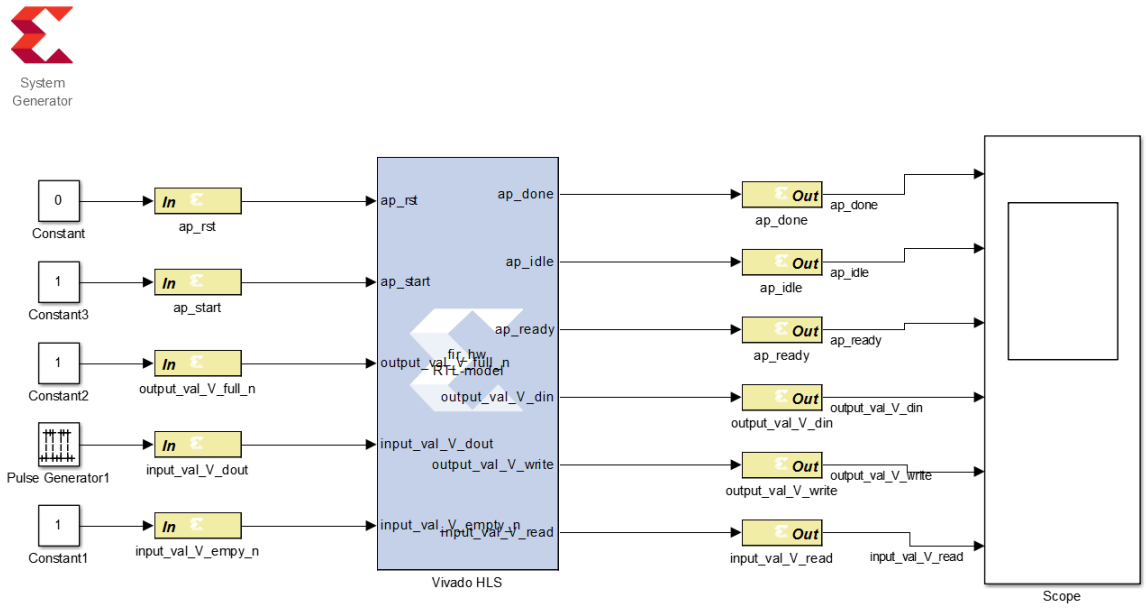
Action with Description	Point of Emphasis and Key Takeaway
• Expand **fir_prj** > **solution1** > **sim** > **report** and double-click the **fir_hw_cosim.rpt** file.	Since cosimulation has already been performed, the Cosimulation report will be accessible from the report folder.

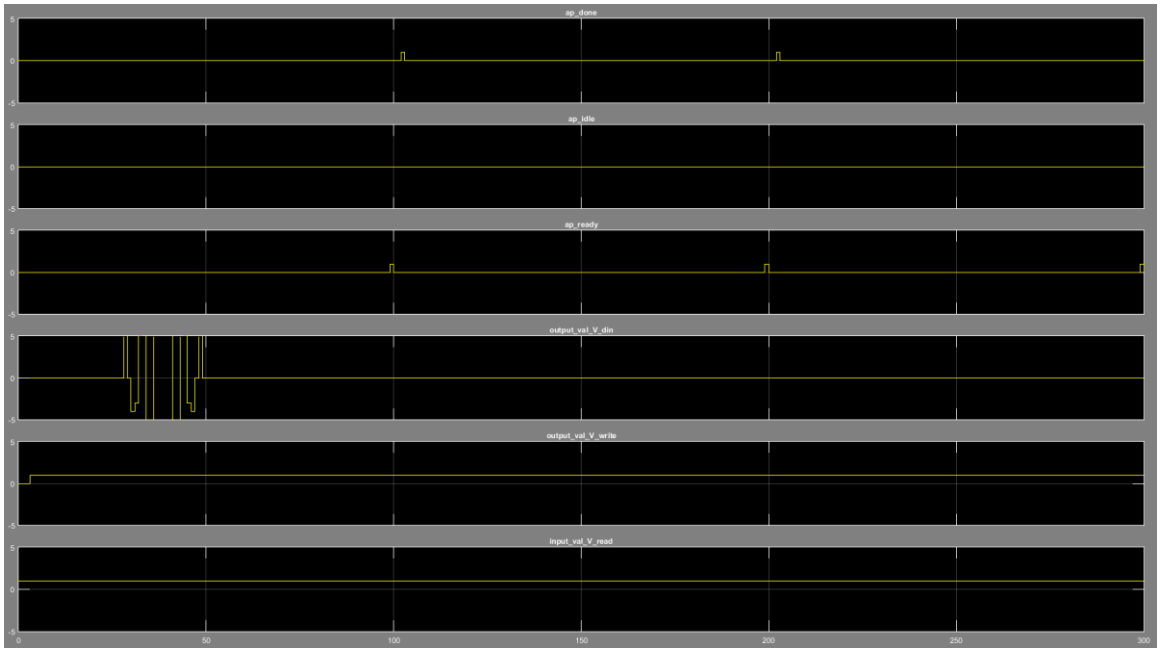


Action with Description	Point of Emphasis and Key Takeaway
• Select **Solution** > **Export RTL**. • Select **System Generator for DSP** from the Format Selection. • Select **Verilog** from the Options section. • Click **OK**.	The final step in the Vivado HLS tool flow is to export the RTL design as a block of intellectual property (IP) that can be used by other tools in the Xilinx design flow. The RTL design can be packaged into the following output formats: • IP catalog-formatted IP for use with the Vivado Design Suite • System Generator for DSP IP for use with Vivado System Generator for DSP • Synthesized checkpoint (.dcp) In this demo, you will export the design as a System Generator for DSP IP. You will see the progress on the bottom right of the tool. In addition to the packaged output formats, the RTL files are available as standalone files (not part of a packaged format) in the verilog and vhdl folders located within the implementation folder: `[project_name]/[solution_name]/impl`. If the **Evaluate** option were selected, RTL synthesis is executed and the final timing and resources reported but not included in the IP package.

Action with Description	Point of Emphasis and Key Takeaway
• Select **Start** > **All Programs** > **Xilinx Design Tools** > **Vivado 2016.1** > **System Generator** > **System Generator 2016.1** to launch the System Generator.	Launch the System Generator to add the Vivado HLS IP. System Generator is a DSP design tool from Xilinx that enables the use of the MathWorks® model-based Simulink® design environment for FPGA design. Designs are captured in the DSP-friendly Simulink modeling environment using a Xilinx-specific blockset. The System Generator design can then be imported into a Vivado IDE project via the IP catalog.
• Click the **Open** toolbar icon and navigate to the project directory *C:\training\hls\demos\hlx_sysgen* . • Select the **fir_sysgen.slx** file.	Browse to the project location to open the *fir_sysgen.slx* model. After few seconds, the design opens with incomplete connections as shown below.



Action with Description	Point of Emphasis and Key Takeaway
• Right-click the **fir_sysgen** window and select **Xilinx BlockAdd**. • Type **Vivado** in the Add field and double-click **Vivado HLS**.	Add the Vivado HLS block to System Generator from the Xilinx Blockset library. The Xilinx Blockset library contains building blocks for constructing DSP and other digital systems in FPGAs using Simulink. The blocks are grouped into libraries according to their function, and some blocks with broad applicability (e.g., the Gateway I/O blocks) are linked into multiple libraries.
• Double-click the **Vivado HLS** block and in the Solution field and browse to the *C:\training\hls\demos\hlx_sysgen\fir_prj\solution1* directory. • Click **Choose** and click **OK** to load the IP block.	Add the path to the created Vivado HLS IP.
• Connect the design I/O ports to the ports in the FIR IP block as shown below. 	

Action with Description	Point of Emphasis and Key Takeaway
• Ensure that the simulation stop time is 300. • Click the **Run** icon ( ) in the toolbar to start the simulation.	Now run the simulation for 300. The Scope output below shows the output.
• Double-click the **Scope** block to view the simulation waveforms. 	
• Save the model and close the MATLAB tool.	

Summary

In this demo, you used a Tcl script to create a Vivado HLS tool project, added the design files, set the top function, run C simulation using a provided self-checking testbench to verify the results, run C-based synthesis to generate RTL, and run RTL co-simulation to verify the RTL results. You then exported the RTL as an IP block to use with System Generator designs.

References:

- Supporting materials
  - *Vivado Design Suite Tutorial: High-Level Synthesis* (UG871)
  - *Vivado Design Suite User Guide: High-Level Synthesis* (UG902)