System Generator for DSP Performing Hardware-in-the-Loop With the Spartan™-3E Starter Kit

Introduction

System Generator supports hardware co-simulation, making it possible to incorporate a design running in an FPGA directly into a Simulink simulation. This quick start will guide you through the process of setting up the Spartan™-3E starter kit to enable hardware-in-the-loop verification with JTAG co-simulation via the USB configuration port. This process can be extended to any board with a JTAG connection.

References

- MATLAB help menu → Xilinx System Generator → JTAG hardware co-simulation
- Spartan-3E Starter Kit page (http://www.xilinx.com/s3estarter)
- Spartan-3E Starter Kit user guide (http://www.xilinx.com/bvdocs/userguides/ug230.pdf)
- Design files (http://www.xilinx.com/products/boards/s3estarter/files/s3esk_sysgen_hw_in_loop.zip)

Requirements

- MATLAB R14.1, R14.2, or R14.3 with Signal Processing blockset
- v8.1 ISE™ Foundation with latest service pack
- v8.1 System Generator for DSP
- Spartan-3E Starter Kit (includes power supply and standard USB cable for configuration)

Directory Structure

- /design/user (working directory)
- /design/completed (contains the completed Simulink design file and bitstream for performing hardware-in-the-loop)
- /plugins (contains the completed board support package files for the XUP Spartan-3E Starter Kit. Note that you will generate these files in Step 1 below)

Design Description

Simulate a digital band-pass filter with the following specifications:

- Sampling Frequency (Fs) = 1.5 MHz
- Fstop 1 = 270 kHz
- Fpass 1 = 300 kHz
- Fpass 2 = 450 kHz
- Fstop 2 = 480 kHz
- Attenuation on both sides of the passband = 54 dB
- Pass band ripple = 1

Two different sources are used to simulate the filter:

- The chirp block, which sweeps between the specified frequencies of 6 KHz and 10 KHz without regard for the instantaneous output frequency
- The random source generator, which outputs a random signal of uniform distribution with a range of -1.9 to 1.9. Uniform is a better choice to drive a fixed-point filter because it is bounded.

Steps

- Generate board support package
- Simulate an FIR filter and generate the run-time hardware model
- Connect hardware model to the design and perform a hardware in the loop verification using the USB configuration port

Generate Board Support Package — Step 1

You will quickly generate the board support package for the Spartan-3E starter kit using SBDBuilder, which is a graphical utility to automate the four board support package files. Refer to the MATLAB help menu for detailed information on SBDBuilder and Spartan-3E starter kit user manual for information on the board.

❶ Invoke MATLAB and browse to the **/design/user** directory

❷ Open the **band_pass.mdl** file design.

❸ Double-click on the System Generator token and select **Hardware Co-Simulation → New Compilation Target…** under the **Compilation** field

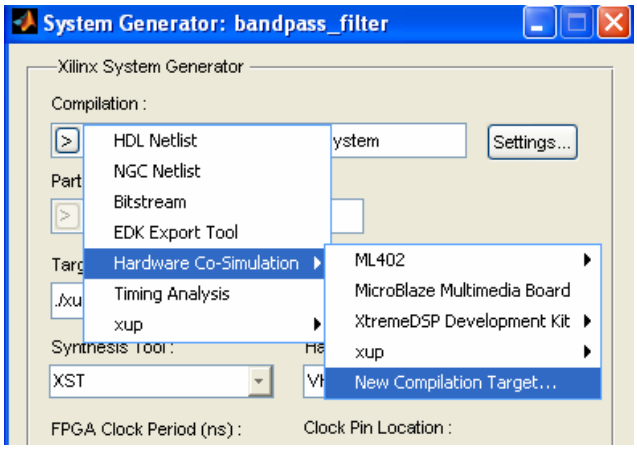


Figure 1. Invoke SBDbuilder to Create New Hardware Compilation Target

Note: The boards listed including the ML402, MicroBlaze Multimedia, and XtremeDSP

❹ Enter a the target board information and clock sections as follows:

- Board Name: s3e_starter

- Frequency (MHz): 50 (This is the system clock frequency on the board)
- Pin Location: c9 (This is the pin location of the system clock)

❺ Connect the USB cable to the board and turn the power on.

❻ Click **Detect** to fill in the JTAG options, which should read as follows:
- Boundary Scan Position: 1 (the xc3s500e is the first device in the JTAG chain, followed by the Platform Flash and then the CoolRunner™-II device)
- IR Lengths: Click the Detect button (should see: 6, 8, 8)

Note: you may verify the position of the devices in the JTAG chain by using the iMPACT programming utility provided with the ISE Foundation software. The iMPACT utility can be invoked as standalone via the start menu.

❼ Under the **Targetable Devices** section, add the xc3s500e-4fg320

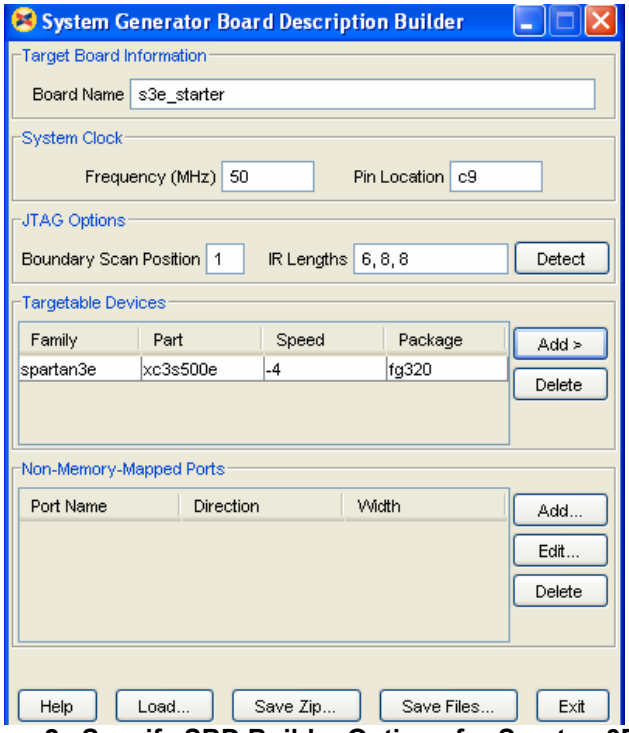


Figure 2. Specify SBD Builder Options for Spartan-3E board

❽ Click the Save Files button and save the files under the System Generator install path as follows:

C:\MATLAB701\toolbox\xilinx\sysgen\plugins\compilation\Hardware Co-Simulation\s3e_starter

The installation of the board support package files should resemble that as follows:

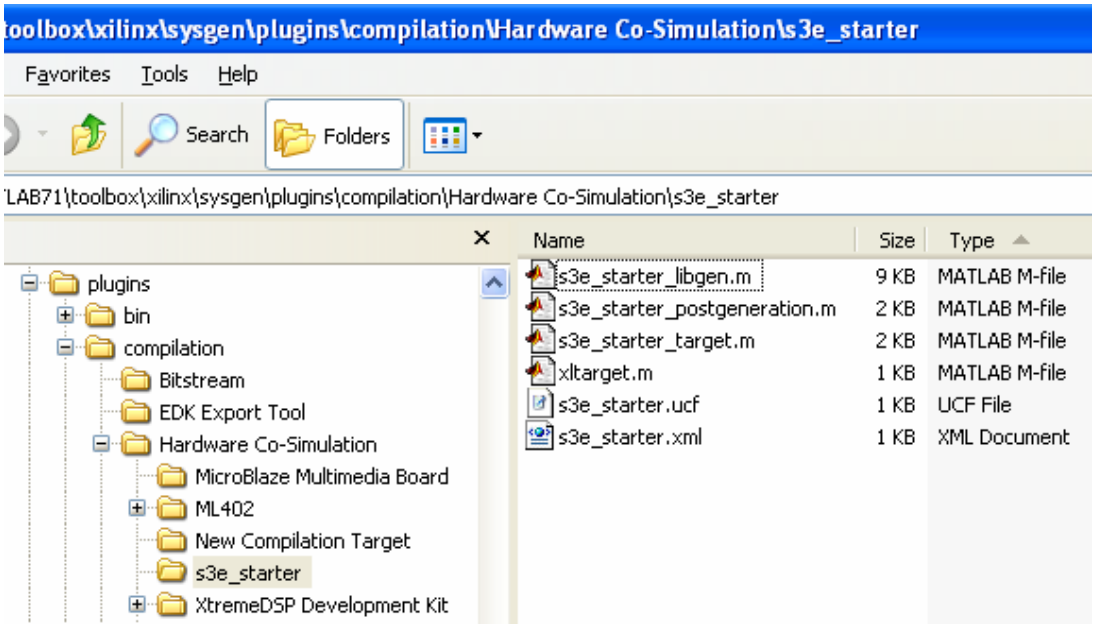


Figure 2. Install the board support package

❾ Exit SBDBuilder

Simulate an FIR Filter and Generate Hardware Model Step 2

Simulate a band-pass FIR filter to verify operation and generate the run-time hardware model.

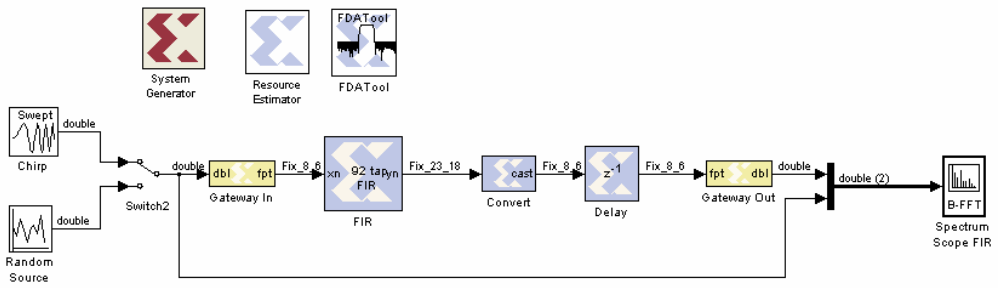
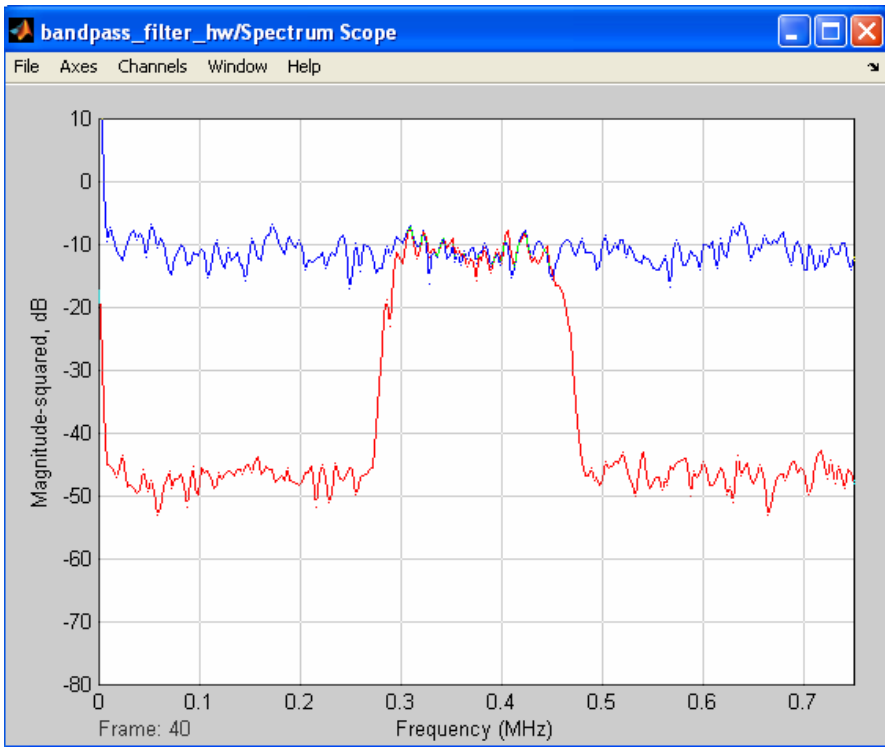


Figure 3. Band-Pass Filter Design

❶ Simulate the **band_pass.mdl** design. You should see the following results



❷ Double click on the System Generator token to verify that the following options are selected:
- Compilation: HDL Co-Simulation → s3e_starter
- Target Directory: ./hwcosim
- Synthesis Tool: XST
- Hardware Description Language: VHDL

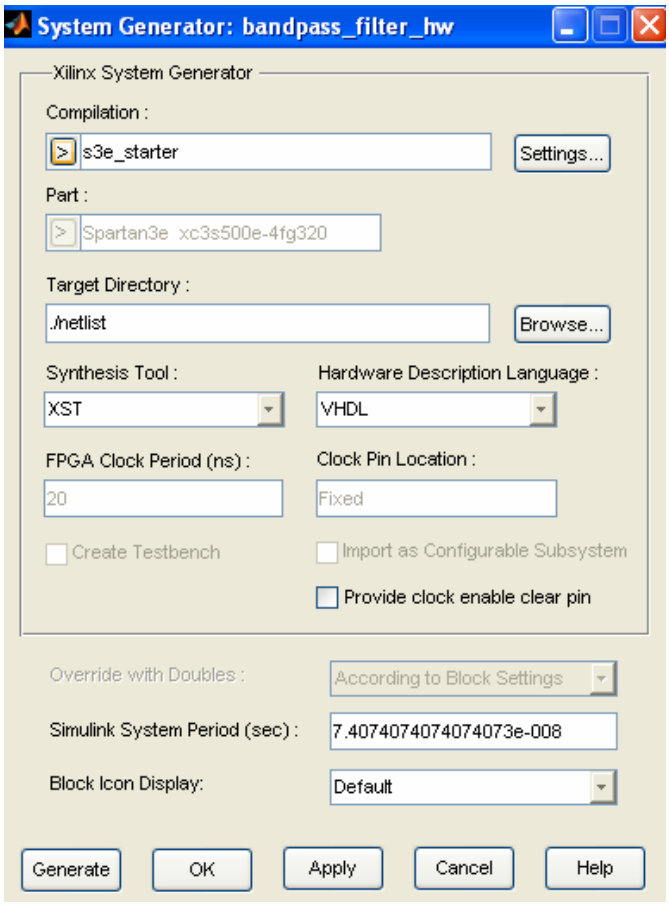


Figure 4. System Generator Token Parameters Box

Note: You may select either VHDL or Verilog

❺ Click the generate button in the System Generator token to generate the hardware model (see illustration below)

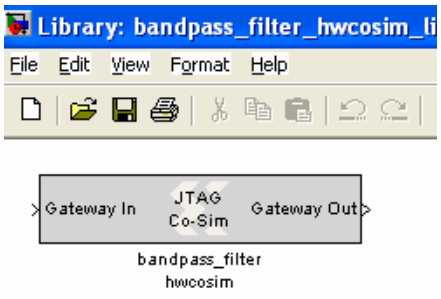


Figure 4. Run-Time Model of Generated Bitstream

Note: System Generator will generate the HDL Code and automatically invoke the ISE Foundation software to generate the bitstream, which can take a few minutes. The run-time block above represents the hardware model of the System Generator design.

Connect HW model and perform JTAG Co-Sim Step 3

Connect the hardware run-time model to the design and perform hardware-in-the-loop verification.

❶ Add the run-time block and connect it according to the figure below, noting that you may simply copy and paste the Spectrum scope and mux.

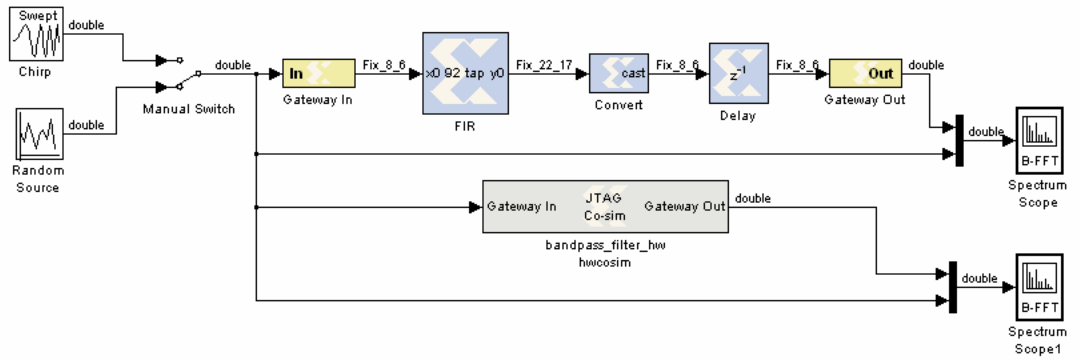


Figure 5. Performing JTAG Co-Simulation

❷ Double-click on the HW co-simulation block and specify Xilinx Platform USB as the download cable.

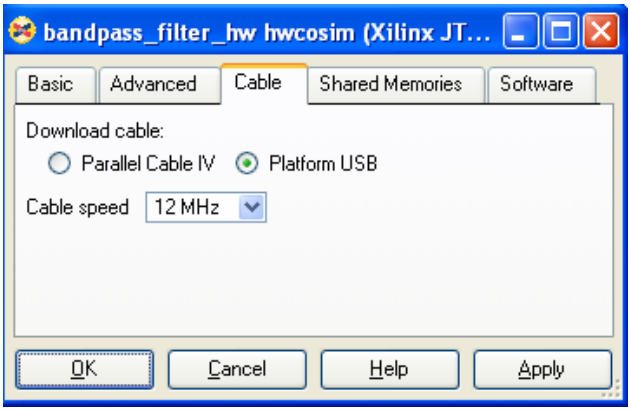
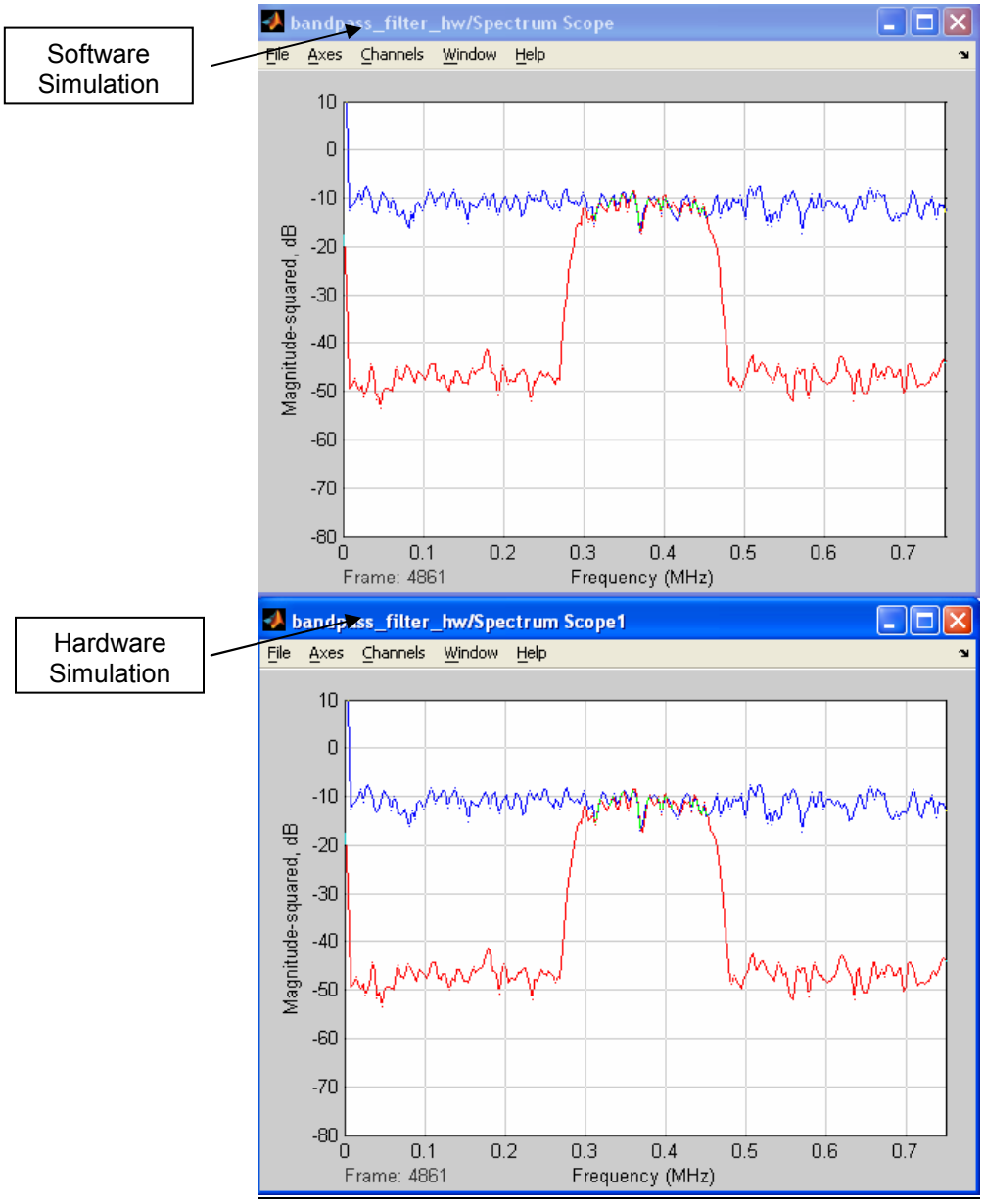


Figure 6. Selecting the Download Cable

❸ Click the **Start Simulation** button to perform the hardware-in-the-loop verification.

System Generator will configure the FPGA and then simulate. You should see results as follows:



Revision History

Rev 1.0 6/8/06 Initial Release
Rev 1.1 12/6/06 Added link to design files