

Interfacing Simulink to the Analog World

The P160 Analog Module adds analog I/O to Memec/Xilinx development boards.

by Luc Langlois
DSP Specialist
Memec Insight
luc_langlois@ins.memec.com

If your product performs digital signal processing, it probably must interface to real-world analog signals. To avoid surprises, it's best to introduce analog I/O early in the design process; the ideal point is in the modeling phase with Xilinx® System Generator for DSP, under Simulink® from The MathWorks.

Consider the development of a digital QPSK demodulator in a software-defined radio. The FPGA performs several signal processing tasks, including carrier recovery DPLL (digital phase-locked loop), down conversion to baseband, down-sampling, pulse-shaping, and symbol timing recovery. You may wish to compare simulated demodulator performance against the real thing by injecting a heterodyned analog signal with noise through an ADC (analog-to-digital converter) to the FPGA running at full clock speed.

We need a framework to receive digitally sampled analog input signals from an ADC as stimuli into the Simulink model, either as real-time streaming data or as captured data for repeatable playback. We also need our framework to deliver processed data from the Simulink model to a DAC (digital-to-analog converter) to produce an analog output signal.

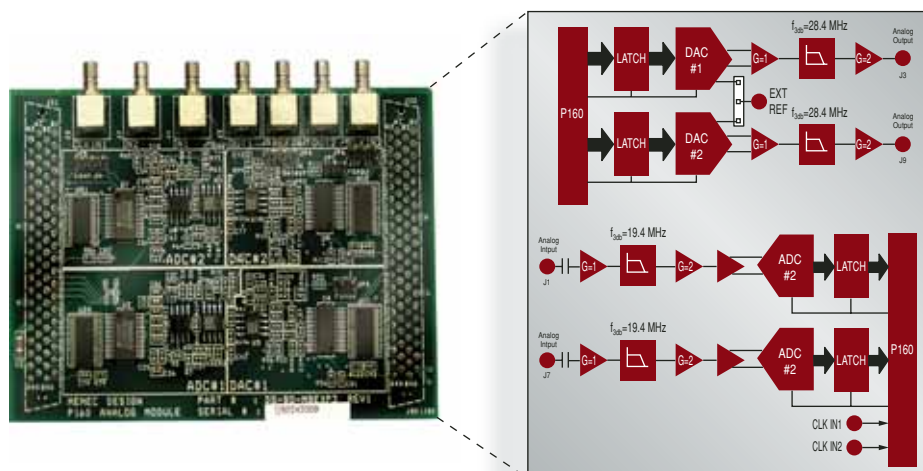


Figure 1 – Memec P160 Analog Module

The Memec™ P160 Analog Module is a daughtercard to interface external analog signals to Memec's wide assortment of Xilinx FPGA development boards. In this article, we'll present design techniques using the Memec P160 Analog Module in Simulink DSP models, making creative use of several new blocks from Xilinx System Generator version 6.3. If you are an FPGA designer, these techniques offer practical starting points, providing you with a head start on your DSP development using external analog signals.

Memec Simulink Library

The Memec P160 Analog Module is shown in Figure 1. It provides two channels of analog I/O through 12-bit data converters from Texas Instruments™:

- Two 165 megasamples per second DAC902 DACs, driving single-ended analog outputs.
- Two 53 megasamples per second ADS807 ADCs. The digital data out of the ADCs is latched into external buffers and then passed to the FPGA through the P160 interface.

The Memec P160 Analog Module DAC and ADC blocks are delivered as a Simulink library (shown in Figure 2). It offers the following features:

- Drag-and-drop P160 analog components from the Simulink library browser

- Supports HDL co-simulation from Simulink
- Supports common compilation types in Xilinx System Generator for DSP 6.3
 - Hardware co-simulation type: generates board-specific I/O ports to FPGA pins connected to the P160 Analog Module
 - HDL netlist type: generates top-level FPGA I/O pins connected to the P160 Analog Module
- Automatically detects target part/package from the System Generator token
- Supports all Memec FPGA development boards with P160 expansion connector
- Installer for automatic Simulink library creation

Interfacing to External Analog Signals

Let's describe three design techniques using various features of the P160 Analog Module to interface to analog signals during development of a DSP design in Simulink.

Memec P160 Analog DAC in HDL Co-Simulation

Our first design technique uses HDL co-simulation, a Xilinx System Generator feature that lets you incorporate your HDL code into Simulink through a black box. Simulink doesn't interpret HDL directly;

rather, it invokes the Mentor Graphics® ModelSim™ HDL simulator, with which it exchanges I/O data during simulation.

When the design is compiled to hardware, the HDL code is included for synthesis. This technique is invaluable if your engineering staff wishes to preserve any existing investment in proven, re-usable HDL code.

System Generator blocks don't expose the system clock directly. Nevertheless, as digital designers, we sometimes prefer to see digital waveforms referenced to the clock, especially for logic that drives signals onto FPGA pins to off-chip. For this reason, the Memec P160 analog DAC block is built using a black box.

Figure 3 shows a model in which we drive the DAC block with a sinusoid signal stored in a ROM look-up table. The ModelSim waveform window opens automatically during simulation, displaying all inputs and outputs of the black boxes and all clock and clock enable signals supplied by System Generator. The signal display can be customized with an auxiliary Tcl script.



Figure 2 – Memec Xilinx DSP library in the Simulink library browser

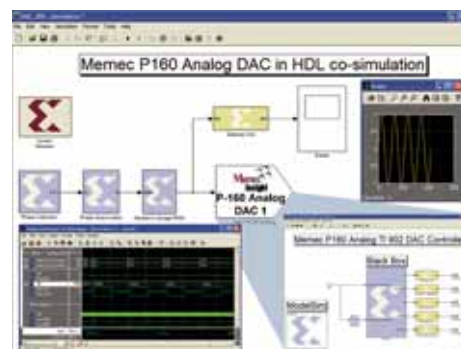


Figure 3 – Memec P160 Analog DAC in HDL co-simulation

Hardware Co-Simulation with the ADC

You can synchronize a System Generator hardware co-simulation block with its associated FPGA hardware in one of two clock modes. In single-step mode, the FPGA is clocked from Simulink; in free-running mode, the FPGA runs off an internal clock, and is sampled asynchronously when Simulink awakens the hardware co-simulation block. Let's examine a design that demonstrates switching between free-running and single-step modes.

The Memec P160 analog ADC controller from the Simulink library is built with gateway-ins defined as board-specific I/O ports. When used in a hardware co-simulation block, the Xilinx implementation tools bring these gateways to FPGA input pins, which connect to the P160 analog daughtercard, according to location constraints for the target device on the board. Consequently, sampled data from the P160 analog ADC can reach the FPGA.

As Figure 4 illustrates, we first compile the model on the left for hardware co-simulation and then connect it, as shown on the right. We use the Xilinx pause simulation block to switch clock modes of the co-simulation block.

Hardware co-simulation starts in free-running mode as we sample an analog input signal through the P160 ADC at a sampling rate derived from the system clock on the Memec development board. Sampled data fills a FIFO, while the Simulink model polls the FIFO's full flag asynchronously. When the full flag goes high, simulation pauses; the clock mode is switched to single-step and captured data samples are read out from the FIFO to Simulink. The captured data can be saved as stimuli for subsequent simulation.

ChipScope Validation of DAC-to-ADC Loopback

Our third design technique is useful to validate P160 analog DAC and ADC

functionality in a loopback configuration. As shown in Figure 5, we generate a waveform that drives the P160 analog DAC to produce a continuous analog output signal. The waveform, stored in FPGA block RAM, is defined as an expression from The MathWorks MATLAB®; either a periodic function such as a sinusoid or arbitrary data from a MATLAB array.

When compiling the model to a bit-stream, the DAC and ADC output ports are mapped to FPGA I/O that connects to the P160 Analog Module, according to the selected target FPGA.

In operation, a loopback cable connects the analog output signal from the P160 Analog Module DAC back into the ADC input. The Xilinx ChipScope™ tool – available as a block in System Generator – captures both the generated waveform and the sampled loopback version.

The design runs in the FPGA at the system clock rate – 100 MHz on most Memec development boards. The data-sampling rate is set in the model to a convenient sub-multiple of the system clock rate, as the ADC can operate at a maximum 53 megasamples per second.

Conclusion

The Memec P160 Analog Module is ideal for interfacing FPGAs to external analog signals. We have shown example techniques to quickly get you started to capture, process, and produce analog signals in your DSP applications under Simulink.

As your needs evolve to meet customer demand for ever-increasing analog I/O performance, expect Memec's next generation of its analog module to deliver faster sampling rates and higher resolution within a consistent support framework in System Generator for DSP and Simulink.

The P160 Analog Module specs, Memec Xilinx DSP Simulink library, and reference designs are available to all current P160 analog customers. You can obtain the free download through the Memec Reference Design Center at <http://legacy.memec.com/solutions/reference/xilinx/>.

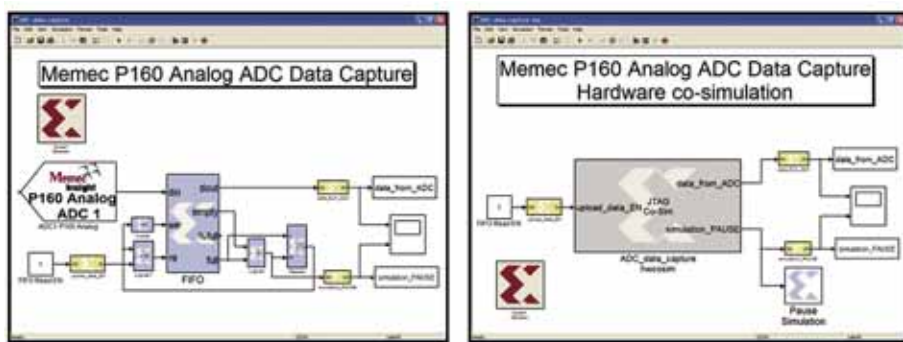


Figure 4 – Simulink model for data capture through the P160 ADC

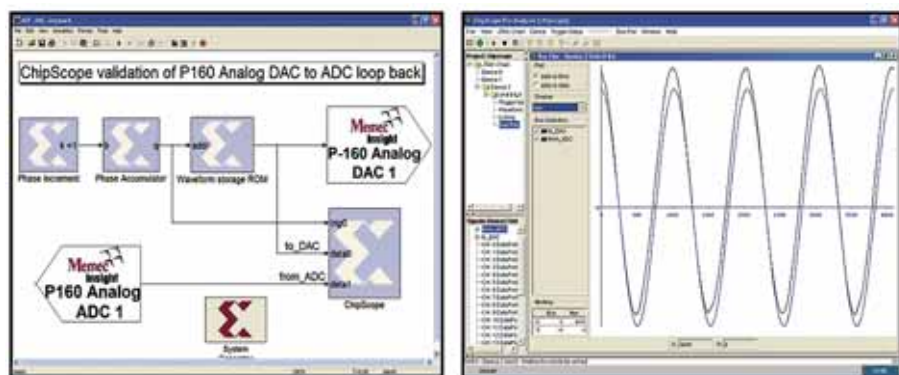


Figure 5 – ChipScope validation of P160 analog DAC-to-ADC loopback