Using Cadence SPW and Virtex-II FPGAs for DSP Design

How to integrate, model, analyze, and implement complex communication standards.

by Michael R. Sturgill
Senior Technical Leader
Cadence Design Systems
sturgill@cadence.com

Today, communication designers face the daunting challenge of rapidly integrating multiple standards within their design to capture time-to-market opportunities. Specifically, these designs must support the seamless integration of WPAN, WLAN, and cellular radio standards into a single application, such as multimedia or digital video broadcast (DVB). In this article we will show how efficiently the Cadence® Signal Processing Worksystem (SPW) integrates, models, analyzes, and implements complex communication standards into a high-end Xilinx Virtex™-II FPGA.

Cadence Signal Processing Worksystem (SPW)

Based on years of proven technology, SPW offers a fully integrated solution for multi-ASIC/FPGA systems or SoCs/SoPCs (system-on-chips/system-on-programmable-chips), from algorithm design to implementation. Figure 1 depicts a simplified SPW flow.

SPW provides a unified design environment that accelerates the development of digital signal processing (DSP) systems, allowing hardware and system engineers to collaborate, share design libraries, and propagate testbenches at every level of design abstraction. In addition, SPW facilitates DSP design by offering the following:

- Integration of C/C++, MATLAB®, SystemC, Verilog-AMS, Verilog, and VHDL blocks into a single design, allowing multiple language flexibility
- Hierarchical design methodology via a graphical block diagram editor, promoting better design, reuse, and documentation
- Architectural convergence that combines datapath and control constructs into a single simulation environment, allowing you to capture and simulate the most advanced electronic systems
- Mix-and-match combinations of a variety of design styles and technologies, enabling your design teams to spend less time writing algorithms and more time optimizing designs
- Tight integration with SPW Hardware Design System (HDS), NC-Sim for RTL (Register Transfer Language) verification, Verilog-AMS (Analog Mixed Signal) for mixed analog and digital simulation, Synplify for logic synthesis, and Xilinx for core generation, place and route – minimizing overall design time.

Variable Interpolation Filter for DVB Applications

Figure 1 illustrates the concept-to-FPGA flow as an interpolation filter designed for use in DVB applications. The filter's output operates in the range of 4 to 48 times the input symbol rate (Rs), while Rs ranges from 1 to 45 megasamples per symbol (MSPS). The multiple interpolation rates available in this model allow the digital-to-analog (D/A) converter output of the DVB transmitter to operate in a relatively narrow frequency span (≈ 45-180 MHz, 4:1 ratio) while operating over a broad symbol rate range (≈ 1-45 MSPS, 45:1

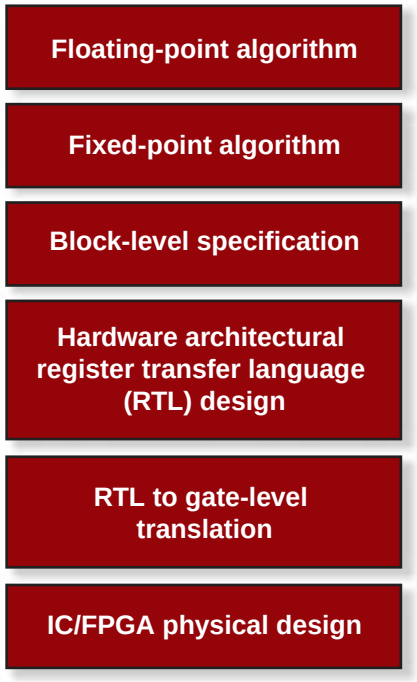

Figure 1 - SPW flow

Modulation Type	Filter Rolloff Factor	Interpolation Factor	Convolutional Encoder Rate
QPSK	35%	4, 8, 16, 32, 48	1/2, 2/3, 3/4, 5/6, 7/8
8PSK	35%	4, 8, 16, 32, 48	2/3, 5/6, 8/9
	25%	4, 8, 16, 32, 48	2/3, 5/6, 8/9
16QAM	35%	4, 8, 16, 32, 48	3/4, 7/8
	25%	4, 8, 16, 32, 48	3/4, 7/8

Table 1 – Interpolation Filter Operating Modes

ratio). This significantly simplifies the design of the analog anti-aliasing filters following the D/A converter.

Operating Modes

The interpolation filter module operates in the modes detailed in Table 1, which also defines the modulation types, rolloff factors, and convolutional encoder rates. The final output frequencies are a function of the Xilinx technology and speed grade chosen, and have been designed to operate at up to 180 MHz. This corresponds to a maximum symbol rate of 45 MSPS.

Theory of Operation

The module operates over a broad range of interpolation factors for each of the modulation types and rolloff factors described above. Figure 2 illustrates the block diagram of this module and the distinct filters used to form the multiple interpolation rates. Combinations of interpolate x4, x3, and x2 make up the specified 4x to 48x range.

The initial interpolate x4 stages for the QPSK (Quadrature Phase-Shift Keying), 8 PSK, and 16 QAM (Quadrature Amplitude Moderation) modes apply the square root raised cosine (SRRC) masks defined in Ref. [1]. These masks also have an x/sinx pre-distortion applied to them. This pre-distortion compensates for the sinx/x droop that occurs when the digital signal is processed through a D/A converter. The output of the D/A converter is then spectrally flat. Filt1, Filt2, Filt3, and Filt4, which are spectrally flat over the passband range of the shaping SRRC filters, are used to interpolate the signal and attenuate the out-of-band images that occur as a result of the interpolation process.

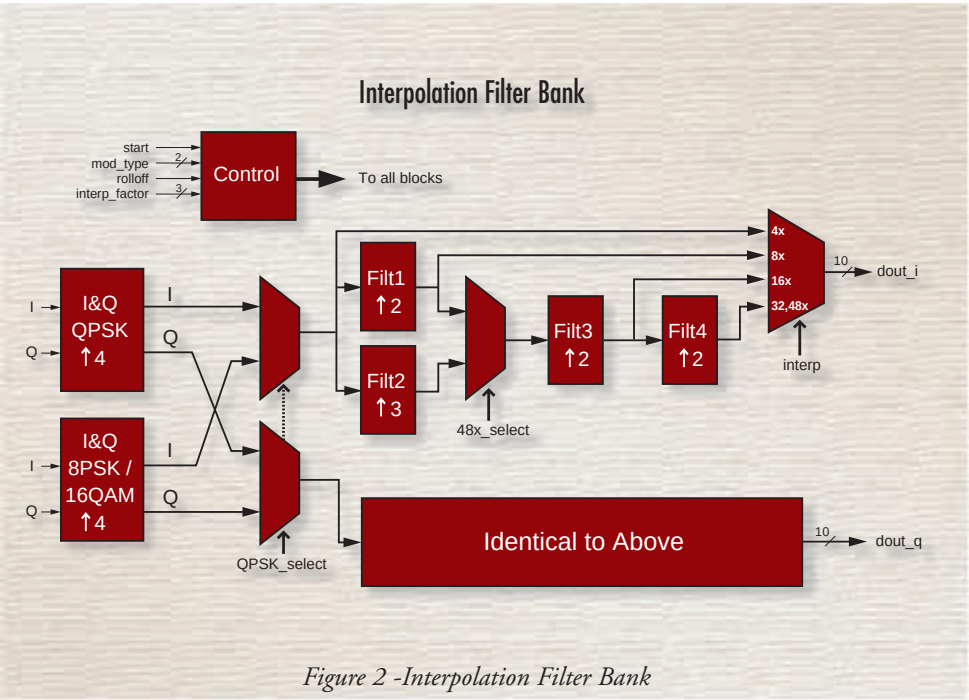

Figure 2 -Interpolation Filter Bank

Fixed Point Architectural (Implementation) Model

There are multiple ways to model systems within SPW. Here, we will focus on the fixed point architectural (implementation) model.

The SPW simulation model, or implementation testbench, depicted in Figure 3 consists of the design under test (DUT), input stimulus data, multirate input controls, parameterizable mode selection and data rate control, and output response capture for visualization and post-processing. All of the SPW building blocks that comprise the DUT are synthesizable. Several blocks in the DUT also have Virtex-II core component instantiations as well.

Implementation Testbench

The simplistic illustration (Figure 3) of this testbench masks the scope of its capabilities. Table 1 lists the operating modes of the design. Careful examination of this table yields roughly 40 different operating modes that must be tested. This testing is achieved by using parameterization at the testbench level.

Figure 3 is color-coded for easy reference to the different parts of the testbench. Together, these blocks make up the testbench. The descriptions are as follows:

- **DUT –** Design under test. Figure 2 shows the modules contained within this block.
- **Simulation Parameters –** These settings allow the DUT to be tested over your required range.
- **Operating Modes –** The top four entries are pointers to the coefficient files used for the QPSK filters. The remaining three enumerated parameters allow you to select the interpolation factor, modulation type, and the filter's rolloff factor.
- **Dynamic Control –** These include the system clock, power on reset signal, and the data input strobe signal.
- **Static Control –** These are constant blocks whose values are computed from the simulation parameters. They provide a convenient way to mimic the control registers in the final design.

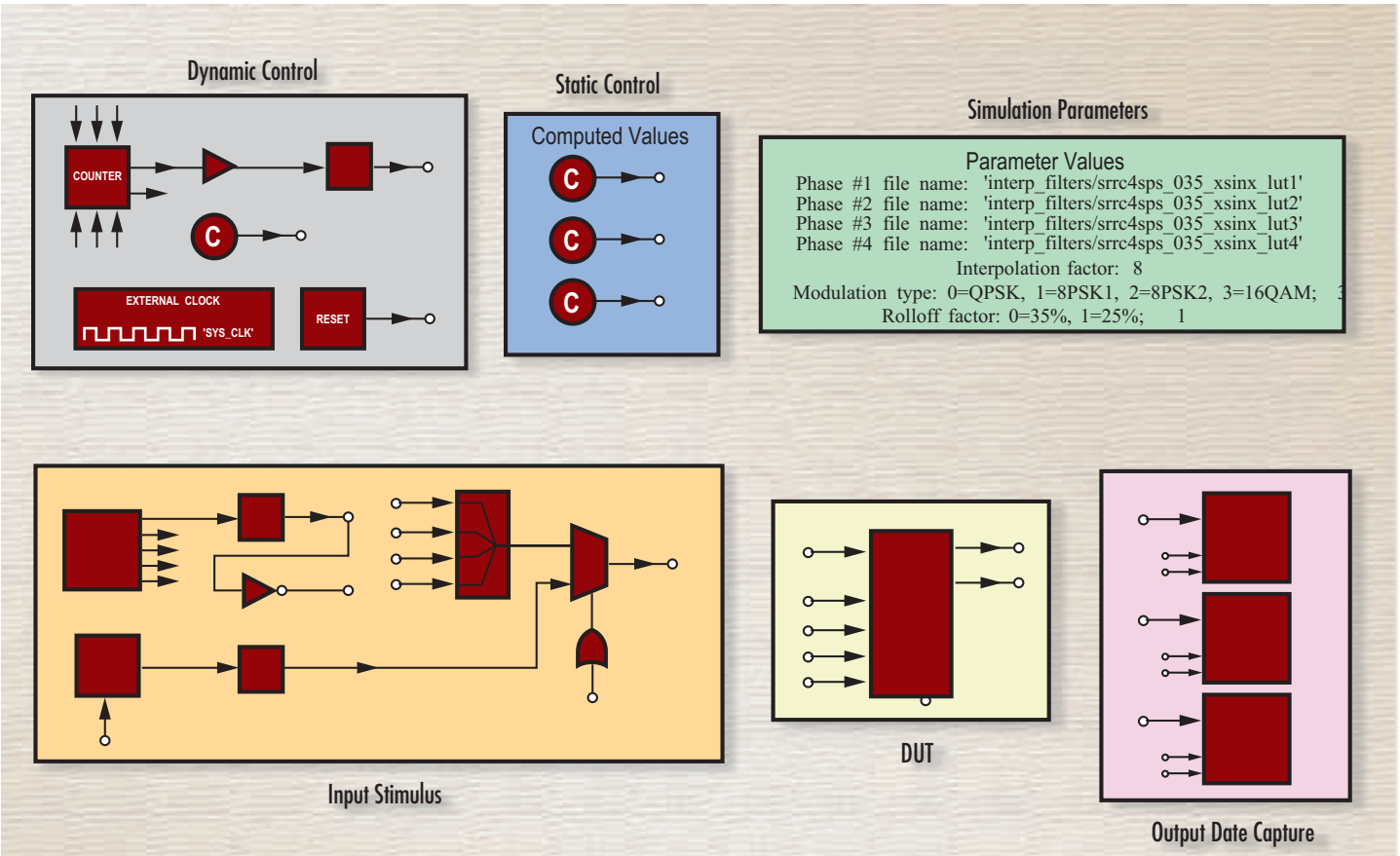

Figure 3 - Testbench for the interpolation filter design

- **Input Stimulus –** There are two possible input selections: The first is a signal source that contains quantized data of a swept sinusoidal waveform. The second input is a random noise generator set to output a uniform white random variable in the range [0, 15]. This allows the filter to operate over all possible transitions of a constellation.
- **Output Data Capture –** The output data capture comprises a set of signal sinks that capture data and write it to a file for post-simulation analysis. You can also use interactive instrumentation to view other aspects of the system, such as eye diagrams, constellations, and so forth.

Implementing the Design

Xilinx Virtex-II devices are an ideal hardware solution for the stringent timing requirements of this design. The Virtex-II devices have dedicated on-board resources specifically meant for high-speed DSP designs, and the design makes efficient use of the hardware multipliers and numerous block RAMs.

SPW has a direct link to the Xilinx CORE Generator™ tool, allowing you to specify the Xilinx core to be imported. Once the core is defined, an SPW library block is created, which is then available for instantiation into all designs. This is the method we used to instantiate the on-board multipliers and the block RAMs in our model.

Using SPW eliminates the need to create Xilinx cores for every block in the design because it allows you to import VHDL and Verilog in combination with Xilinx cores and blocks from SPW's hardware design system. Figure 4 shows the top-level hierarchy for the entire interpolation filter module.

After the design has been captured, simulated, and validated against the behavioral model, it is time to take it to the hardware. The steps involved in this process are as follows:

1. Generate the RTL (VHDL or Verilog) for the design under test.
2. Synthesize the design, targeting the Virtex-II device.
   (Note – If the target synthesizer has the capability of outputting an RTL netlist, this netlist can also be simulated in SPW with the same testbench used for the DUT.)
3. Place-and-route the design using the Xilinx tools. Here again, the netlist created by the Xilinx tools may be simulated in the same SPW testbench.
4. The bit file created is then downloaded to the board, where the design may be run in real time.

The full design runs at data rates of greater than 180 MHz on a Virtex-II XC2V3000-6-FG676 device.

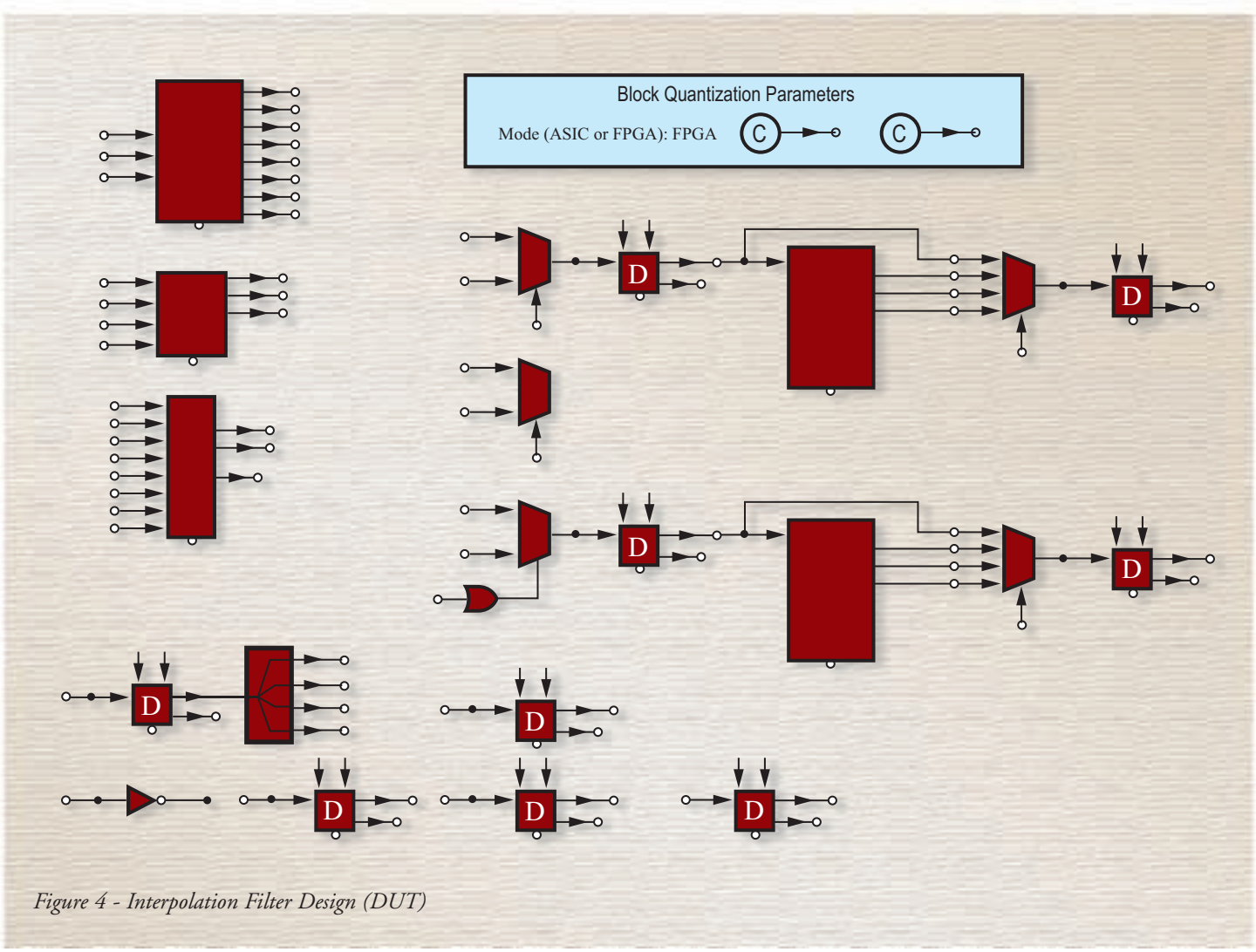

Figure 4 - Interpolation Filter Design (DUT)

Downloading to the Nallatech Development Board

The Nallatech board, which is part of the Xilinx XtremeDSP™ Development Kit, comes with PC software for FPGA configuration and control. The board is populated with a Virtex-II XC2V3000 FPGA for general use, as well as an additional FPGA dedicated to the many clocking schemes available. Connection to the PC is via PCI or USB, and installation is a snap.

We edited the constraint files included with the kit to match the I/O and clocking requirements of the design, and used the included software to download the clock and interpolation filter bit files. The software also includes an interface to the hardware, which enabled us to design a simple bus interface into the interpolation filter to allow configuration changes via the PC software. All modes defined in Table 1 are programmable via this interface. The filter's inputs may also be configured to use either the on-board A/D converters or an internally generated noise source.

Conclusion

Cadence SPW, used in combination with the Xilinx Virtex-II FPGA, creates a powerful and robust solution for meeting the demands of today's DSP designers. By providing a smooth path from system-level design and verification to implementation, SPW offers an effective bridge from system concept to hardware realization. The Virtex-II devices contain the requisite components needed in DSP design, and their speed and density allow entire systems to run in real time.

In this article, we have barely touched on the breadth of capabilities of SPW coupled with Virtex-II devices. We hope, however, that we have given you an indication of the capabilities available. In the past, high-speed designs, such as the one depicted in our model, were feasible only in ASICs, but with Xilinx pushing the speed and density envelope, they are now possible in a reprogrammable device.

For more information on Cadence SPW and NC simulators, visit *www.cadence.com/products.*

References

[1] EN 301 210 v1.1.1 (1999-03) "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for Digital Satellite News Gathering (DSNG) and other contribution applications by satellite," European Standard (Telecommunications series), 1999-03.

[2] M. Sturgill, "Square Root Raised Cosine Transmit FIR Filter For DVB Applications," Cadence Design Systems Publication, 2000.