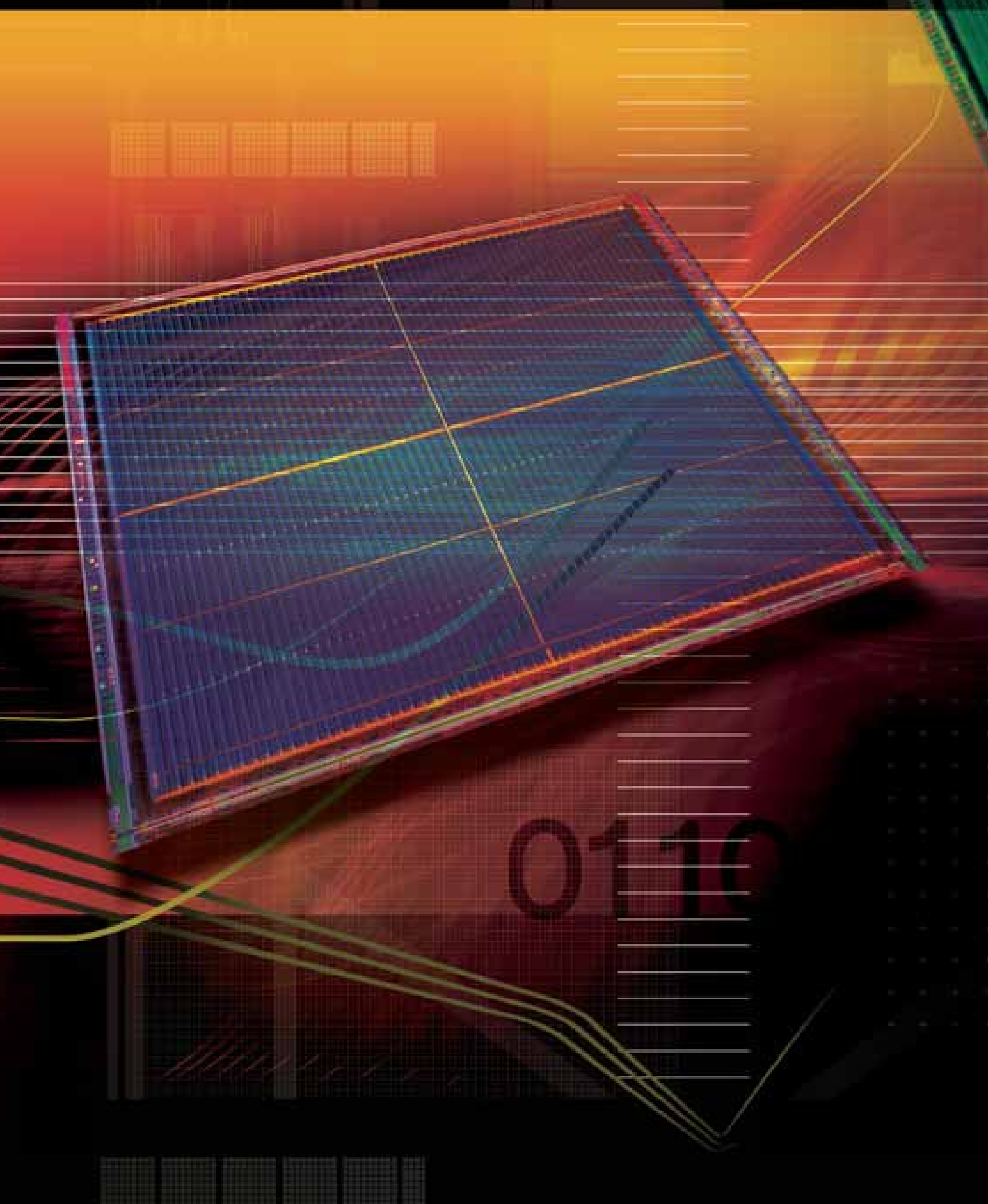
Simulink Brings Model-Based Design to Embedded Signal Processing

The complexity of FPGA-based signal processing systems drives the need for new development approaches.



by Ken Karnofsky
Marketing Director, Signal Processing and Communications
The MathWorks, Inc.
ken.karnofsky@mathworks.com

Many of today's highly integrated embedded hardware and software systems rely on sophisticated signal processing and communications. Dramatic increases in silicon and algorithmic complexity in these systems have triggered a corresponding rise in design and verification costs.

Several studies have noted the impact of the complexity challenge. A Collett International Research study reported by Jack Horgan stated that only 39% of IC designs were bug-free at first silicon in 2002[1]. Embedded Market Forecasters found that more than 50% of embedded projects are behind schedule, and one-third failed to achieve 50% of performance and functional expectations[2]. Figure 1 shows the typical patterns of early defect introduction and late detection that are at the root of these problems.

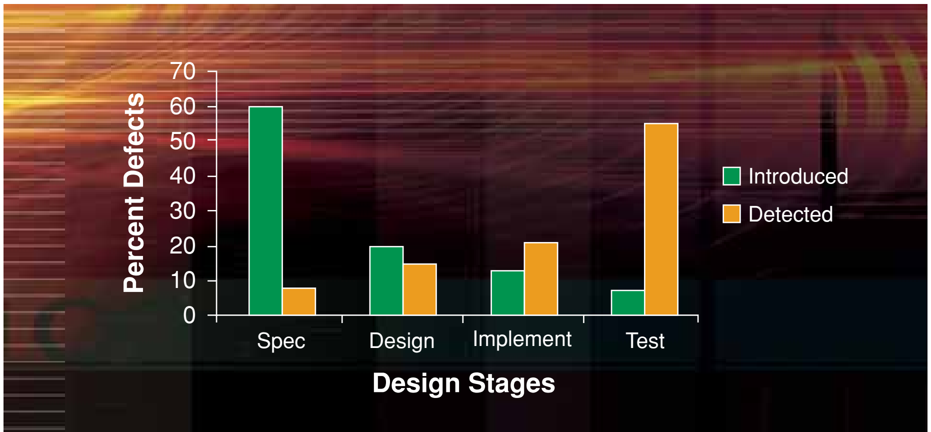


Figure 1 – Typical patterns of design defects using conventional flows, which lead to rising costs and late product deliveries.

Tacking new verification techniques or language extensions onto traditional design tools and flows is not enough to effectively improve the development process. These incremental improvements do not eliminate the aspects of traditional flows – ambiguous text-based specifications, manual implementation, and after-the-fact testing – that produce expensive errors and jeopardize delivery timelines.

In contrast, The MathWorks has demonstrated that Model-Based Design with Simulink® produces dramatic reductions in development time, cost, and risk. These benefits have been documented in the aerospace, automotive, communications, and semiconductor industries – wherever the application requires real-time signal processing, communications, and control logic.

Reinventing Development

The use of FPGAs for high-performance DSP is a natural application for Model-Based Design. Getting the most out of FPGA hardware requires insight into algorithmic and architectural complexities at the same time. To do this, architects need tools that offer direct access to hardware – tools that hardware and software engineers can also use to refine and implement the designs. A design environment like Simulink makes it possible to quickly and accurately simulate system behavior, and provides a direct path to implementation using automatic code generation.

Xilinx® recognized early on that Simulink was a platform that could make FPGA-based DSP design practical. Today, a complete design flow is available from The MathWorks and Xilinx that includes third-party development hardware for real-time prototyping and deployment. Many organizations now enjoy order-of-magnitude returns on their investment in Model-Based Design for Xilinx FPGAs.

Comprehensive, system-level mathematical models form the basis of Model-Based Design. Such modeling was once only in the realm of technology researchers, not mainstream product developers. But facing the limitations of traditional software and hardware description languages for large-scale projects, many design leaders recognize that modeling and simulation is necessary to handle the complexity of today's systems, not only for system design but for hardware and software development as well.

Hardware description languages, even with "system-level" extensions, do not support the rapid modeling and design iteration needed for algorithmically intensive, large-scale embedded hardware systems – that is, virtually all of today's communications and multimedia systems.

Similarly, C-based tools and design flows will not address the software explosion in these systems. In fact, automotive companies and others facing the rapid growth of software-intensive embedded systems have turned to Model-Based Design. Manually developing code in C is no longer an option, because companies cannot hire enough programmers or test engineers to develop and verify the code.

The Elements of Model-Based Design

With Model-Based Design, specification, design, implementation, and verification can be accomplished – and accelerated – using a single Simulink model. Figure 2 depicts these elements, which are described below.

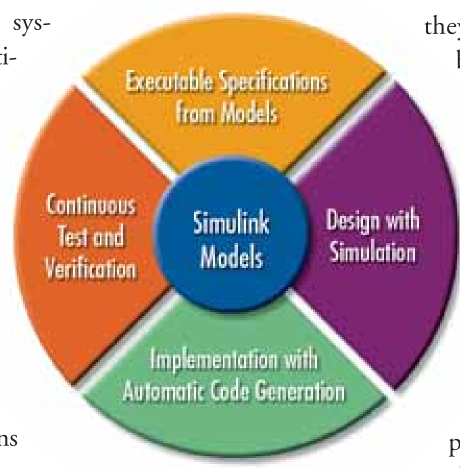


Figure 2 – The elements of Model-Based Design

Executable Specifications from Models

Simulink models serve as executable specifications for system and component behavior, replacing ambiguous text documents. These models can span digital and analog hardware as well as software, and they facilitate clear, unambiguous communication between engineering teams.

Design with Simulation

Simulink is a platform for multi-domain simulation of dynamic systems. The Simulink product family provides an interactive, graphical block diagram environment with a customizable set of block libraries for signal processing, communications, and control. You can create comprehensive system specifications, model channels, and other environmental effects.

These tools also simplify system analysis using quantitative measures such as signal-to-noise ratio and bit-error rate. Simulink is integrated with MATLAB®, providing access to an extensive range of tools for algorithm development and data analysis.

Simulink models are hierarchical; you can partition them easily into subsystems or components. This simplifies comprehension of the design and interaction of

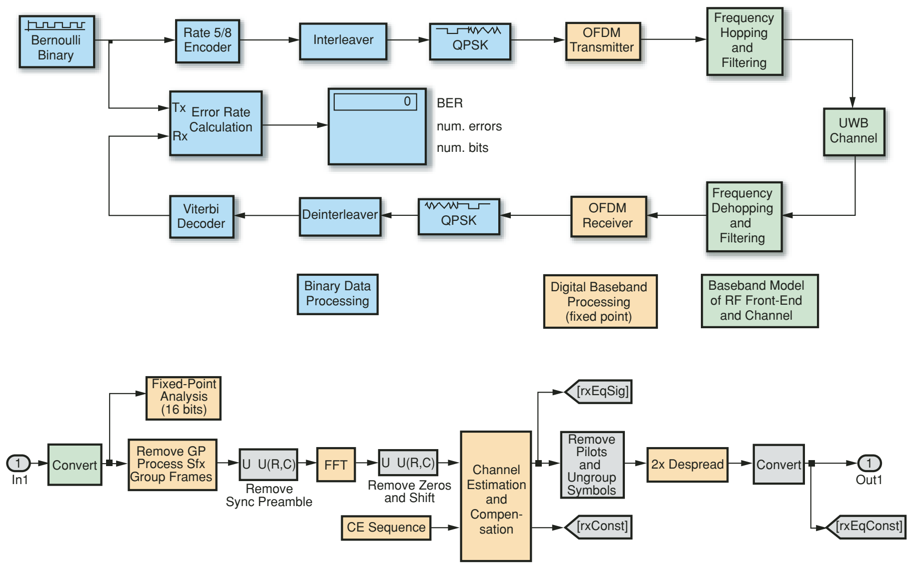


Figure 3 – Simulink model of UWB system (top) with fixed-point OFDM receiver (bottom). Fixed-point computations are shown in yellow.

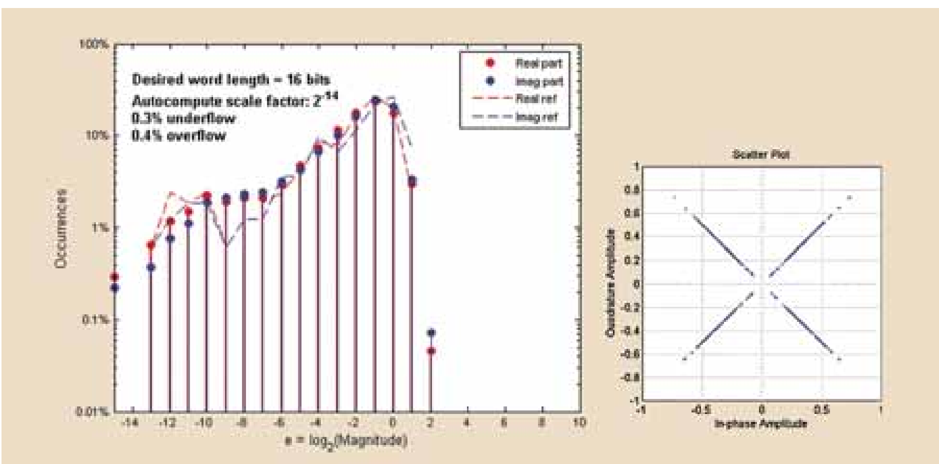


Figure 4 – Optimal fixed-point scaling of OFDM transceiver obtained through iterative simulation and analysis.

subsystems, including software, digital hardware, and RF/analog hardware. You can rapidly simulate and iterate to identify flaws, and refine the model to validate behavior against the requirements.

The Simulink model defines all the information needed to implement the software or hardware, including bit-true fixed-point processing and cycle-accurate timing and synchronization of multi-rate systems. Simulation is used to show that the executable specification defined by the model is complete and works correctly.

You can use model components to create well-defined subsystem interfaces, which simplify reuse in subsequent design efforts, even when those projects employ different target hardware or hardware/software partitioning.

Implementation with Automatic Code Generation

Once you have refined and validated the design, you can automatically generate code from the model, eliminating the need for hand-coding and the errors that manual coding can introduce. You can then use the code for real-time prototyping and deployment in the target system. The strategic partnership between Xilinx and The MathWorks has brought automatic hardware code generation capabilities to Xilinx FPGAs.

Continuous Test and Verification

You can ensure quality throughout the development process by integrating tests into the models at any stage and quantifying test coverage of the model. This continuous verification and simulation helps identify errors early, when they are easier and less expensive to fix, and streamlines the final verification stage.

The system model, or "golden reference," can serve as the test bench for the hardware or software implementation, which you can verify through software or hardware-in-the-loop co-simulation.

Applications of Model-Based Design

Model-based design can accelerate and simplify the development of many technologies. These examples are a small subset of the many applications available on The MathWorks website.

UWB Wireless

The range of ultra wideband (UWB) links is limited by the requirements for low-power, high-speed, and low-cost implementation. Fixed-point word length and scaling have a direct impact on hardware size, cost, and signal-to-noise ratio (SNR) degradation.

Using Simulink, the 10-bit orthogonal frequency division multiplexing (OFDM) transceiver for UWB shown in Figure 3 was designed in a few days. The receiver operates with a 0.5 dB degradation in signal-to-noise ratio, relative to a floating-point reference model.

The optimal word length was determined through simulation over a range of word lengths and channel conditions to evaluate trade-offs between chip size and wireless range. The results are shown in Figure 4. The transceiver operates within a complete end-to-end system model that serves as both an executable specification and a test harness for verifying downstream implementation.

Digital Down Converter for Software-Defined Radio

FPGAs are being used to perform high-data-rate signal processing in many emerging software-defined radio applications. A typical application is the digital down converter (DDC), which is a sequence of multi-rate filters that decimate the RF signal down to the baseband rate. The design challenge is to design an architecture for each filter stage that optimizes the tradeoffs among word length, computational delays, and accuracy of the overall filter response to avoid aliasing and other unwanted numeric effects.

The Simulink model of the cascaded integrator comb (CIC) filter used in the DDC, shown in Figure 5, was automatically generated from the MATLAB filter specification. Models such as this can provide a reference design for developing optimized Xilinx FPGA implementations with Xilinx System Generator.

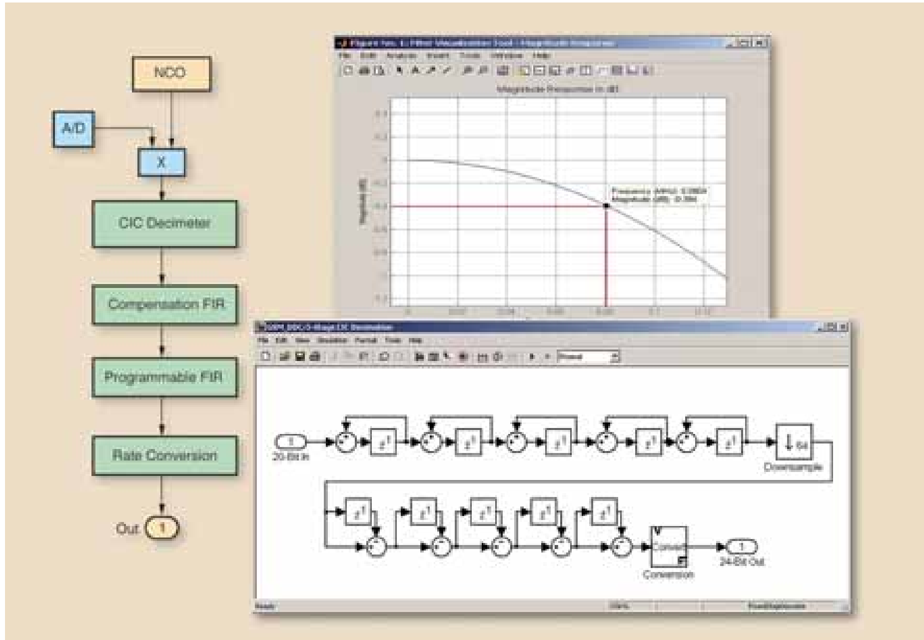


Figure 5 – Frequency specification (top) and Simulink model of a fixed-point CIC filter (bottom) for a digital down converter (left) in a software-defined radio receiver front end.

Reconfigurable Encryption System

Nallatech, a provider of high-performance FPGA systems, used Simulink and Xilinx System Generator for DSP to design a reconfigurable video encryption system in less than two weeks. The system enables their customers to re-verify an entire system when changing components and interfaces – without any knowledge of VHDL. “With this design flow, we efficiently implemented our system and algorithms with a significant improvement on traditional design times, without sacrificing performance,” says Daniel Denning, a research engineer with Nallatech. “Coding in VHDL would have taken us three times as long.”

Simulink 6

In June 2004, The MathWorks introduced Simulink 6, which increases performance, responsiveness, modeling fidelity, and workflow efficiency when modeling large systems. Simulink 6 also extends the scope of Model-Based Design to new domains and applications. These enhancements include:

- Component-based modeling for large-scale systems, including the ability to simulate, test, and implement each design component independently
- Unified data management for model and signal parameters across component models, including a graphical model explorer tool
- Simulink Verification and Validation, which links models to requirements and test cases, and identifies untested portions of models
- Ability to include a subset of the MATLAB language in Simulink models, and automatically generate embeddable C code
- New products for Model-Based Design of signal processing and communications systems, including:
  - Filter Design HDL Coder, for generation of VHDL and Verilog™ code for fixed-point filters
  - Fixed-Point Toolbox, for design and verification of fixed-point algorithms and analysis of fixed-point data in MATLAB
  - RF Toolbox, for design and analysis of networks of RF components
  - RF Blockset, for design and simulation of RF system and component behavior in end-to-end Simulink wireless system models
  - Video and Image Processing Blockset, for design and simulation of embedded video and image processing systems
  - Link for ModelSim™, for co-simulation and verification of VHDL and Verilog using Mentor Graphics® ModelSim

Conclusion

There is growing acceptance of Model-Based Design as the way to handle complexity in embedded hardware and software systems. The MathWorks/Xilinx alliance has enabled the design and implementation of high-performance DSP systems within the Simulink environment, reducing design and schedule risk while capitalizing on the potential of FPGAs for advanced signal processing applications.

For more information, visit *www.mathworks.com/products/dsp_comm/* for technical literature, webinars, and demonstrations.

References

1. Horgan, J. March 29, 2004. “Hardware/Software Co-verification,” EDA Café Weekly.

2. Krasner, J. January 2004. “Model-Based Design and Beyond: Solutions for Today’s Embedded Systems Requirements,” Embedded Market Forecasters, American Technology International, Framingham, Mass.