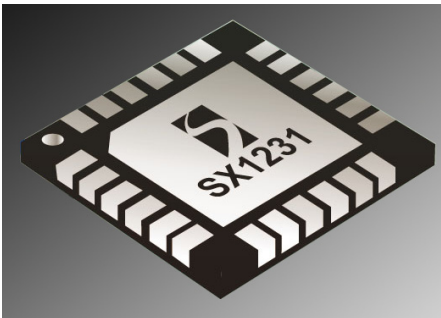


Semtech Speeds Development of Digital Receiver FPGAs and ASICs



The Semtech SX1231 wireless transceiver.

Wireless RF receivers are used in a range of applications, including wireless security systems, industrial monitoring, meter reading, and home automation. In the past, semiconductor suppliers primarily built these receivers using analog designs. Today, suppliers are transitioning to digital and mixed-signal designs to reduce power consumption and ease integration with other components.

Engineers at Semtech are expanding their use of MathWorks tools for Model-Based Design to transition to a digital platform. For years, the engineers modeled and generated HDL code for filters with MATLAB® and Filter Design HDL Coder™. On their most recent project, they used Simulink® and Simulink HDL Coder™ to generate VHDL® for the entire design.

“There is no advantage to writing VHDL by hand,” says Frantz Prianon, IC design engineer at Semtech. “With Simulink and Simulink HDL Coder, we have one model of the system. We simulate it, so we know it works. And we generate code from it, so we can use one model until the end of the project. This is an important capability, because we are sure that what we have implemented matches the design and that the design meets specifications.”

The Challenge

Semtech engineers needed to develop a digital receiver chain for frequency-shift keying (FSK) and minimum-shift keying (MSK) demodulation using a low-IF architecture. They wanted to evaluate multiple design

ideas for performance, power consumption, and layout area during the prestudy phase of the project. Writing VHDL for each design alternative would have been time-consuming, limiting the number of alternatives the team could consider.

Beyond the prototyping stage, Semtech engineers wanted to improve upon their traditional development workflow for production code. “After we modeled our systems to ensure that they met requirements, we used to reimplement them in VHDL and rerun the simulations in a new tool,” says Prianon. “There was always a chance that we would introduce errors, and we could never be sure that the model corresponded exactly to the new VHDL code.”

The Solution

Semtech used MathWorks tools for Model-Based Design to rapidly explore and evaluate design ideas, generate production VHDL code, improve collaboration among engineering teams, and accelerate the development of the digital receiver chain for FSK and MSK demodulation.

In the prestudy phase, Semtech engineers created a floating-point model in Simulink based on the system specifications. They used blocks from Communications System Toolbox™ to model noise in the channel and to implement FSK and MSK demodulation.

Using Signal Processing Toolbox™ and DSP System Toolbox™, one engineer designed and analyzed cascaded integrator-comb (CIC) and finite impulse response (FIR) digital

The Challenge

Accelerate the development of optimized digital receiver chains for wireless RF devices

The Solution

Use MathWorks tools for Model-Based Design to generate production VHDL code for rapid FPGA and ASIC implementation

The Results

- Prototypes created 50% faster
- Verification time reduced from weeks to days
- Optimized, better-performing design delivered

“Writing VHDL is tedious, and the handwritten code still needs to be verified. With Simulink and Simulink HDL Coder, once we have simulated the model we can generate VHDL directly and prototype an FPGA. It saves a lot of time, and the generated code contains some optimizations we hadn’t thought of.” —FRANTZ PRIANON, SEMTECH

filters, while another engineer was working on a sigma-delta analog-to-digital converter (ADC), a phase locked loop (PLL), and other parts of the complete system using Simulink.

Once the separate parts of the digital receiver chain had been simulated, the engineers shared their Simulink models with each other to verify that their component designs would work together before system integration.

The engineers ran simulations to verify the design and used the Error Rate Calculation block from Communications System Toolbox to compute the bit error rate.

With Fixed-Point Toolbox™ and Simulink Fixed Point™, they converted the design from floating point to a fixed-point representation, which they used to conduct bit-true simulations.

Semtech engineers used Simulink HDL Coder to generate VHDL from the Simulink model of the complete receiver chain. To verify the VHDL, they used EDA Simulator Link™ to cosimulate their Simulink design with the Mentor Graphics® Questa® simulator.

Semtech is currently working on the ASIC implementation of the receiver chain.

The Results

Prototypes created 50% faster. “When we wrote the VHDL ourselves, it could easily take two months to create an FPGA prototype,” says Prianon. “With Simulink and Simulink HDL Coder, we eliminate the tedious hand coding of each block and create prototypes in a few weeks.”

Verification time reduced from weeks to days. “On previous projects we would spend at least two weeks writing test benches to verify our VHDL,” recalls Prianon. “Using EDA Simulator Link, we can run cosimulations, test multiple critical points in the model, and verify the VHDL, typically in less than a day.”

Optimized, better-performing design delivered. Model-Based Design enabled Semtech to shorten development time from requirements to tape-out by about 33%. “We used the time we saved to improve the design,” says Prianon. “MathWorks tools enabled us to explore more alternatives and new features, and ultimately deliver a more optimized, better-performing design.”

Industry

- Electronics and semiconductors

Application Areas

- Algorithm development
- System design and simulation
- HDL code generation and verification
- Verification, validation, and test
- FPGA design

Products Used

- MATLAB®
- Simulink®
- Communications System Toolbox™
- DSP System Toolbox™
- EDA Simulator Link™
- Filter Design HDL Coder™
- Fixed-Point Toolbox™
- Signal Processing Toolbox™
- Simulink Fixed Point™
- Simulink HDL Coder™
- Mentor Graphics® Questa®

Learn More About Semtech

www.semtech.com