

# Harris Accelerates Verification of Signal Processing FPGAs

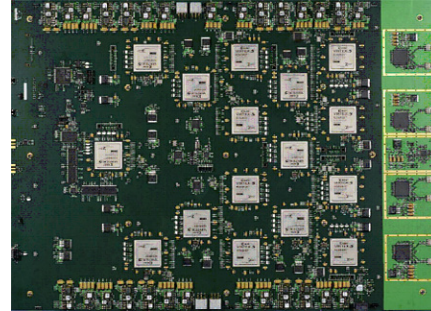
Engineers at Harris Corporation are accustomed to delivering sophisticated FPGA-based signal processing systems within tight schedules. To meet their customers' often stringent requirements and their own quality standards, the engineers thoroughly verify the HDL design of each system before it is synthesized.

In the past, HDL verification required several manual steps. Harris engineers have automated the process by using EDA Simulator Link® to provide a bidirectional link between the MATLAB® system model and the HDL design simulated in Cadence® Incisive®. The new process eliminates ambiguity between the algorithm specification and HDL verification, reduces duplication of effort, and improves communication between system and HDL engineers.

“Cosimulation with MATLAB and EDA Simulator Link not only made it easier to simulate at the subsystem level, it also enabled us to verify the overall system more completely,” says Jason Plew, senior engineer at Harris. “We greatly reduced the time needed to develop subsystem test benches, which enabled us to verify and debug our designs much earlier.”

## THE CHALLENGE

Harris engineers needed to design and verify a signal processing system that spanned multiple Xilinx® Virtex®-5 FPGAs. In this labor-intensive process, once a fixed-point model of the subsystem had been simulated, the team exported large text files containing stimulus data and expected



Harris FPGA-based system.

results from those simulations and imported them into Cadence Incisive. An experienced engineer then wrote test benches in HDL to read in the data, apply the stimulus, and verify the results. If the results did not match, the team had to delve into the massive results files to find out where and why the simulation failed. For each change in the characteristics of the test, they had to re-export the simulation data, which required 8 hours of tedious manual work.

“We spent a considerable amount of time churning through manual steps for each subsystem, and as a result, we only had time to complete 30% of the desired system-level test cases before our deadline,” Plew recalls.

Harris needed a way to drive the input signals in the logic simulator directly, eliminating the need to write the HDL test benches and the overhead of managing huge text files of test data.

## THE SOLUTION

Harris engineers modeled and simulated their signal processing systems in MATLAB. The MATLAB model became

### THE CHALLENGE

Streamline a time-consuming manual process for testing signal processing FPGA implementation

### THE SOLUTION

Use EDA Simulator Link to verify the HDL design from within MATLAB

### THE RESULTS

- Functional verification time cut by more than 85%
- 100% of planned test cases completed
- Design implemented defect-free

a reusable test bench that they could use interactively to cosimulate with their Cadence Incisive simulator using EDA Simulator Link.

With MATLAB and Filter Design Toolbox™, the engineers developed an ideal floating-point model of the baseband signal processing data path, which they used to verify that the design met the functional requirements.

After converting the model to fixed point, they optimized their algorithms for implementation in hardware. Additional simulations were performed in MATLAB to ensure that the fixed-point implementation performed within specifications.

Using the fixed-point model as an executable specification, Harris engineers then implemented the system in HDL.

The team used EDA Simulator Link to enable the MATLAB code to act as the test bench for the implementation. Via the cosimulation interface, MATLAB was used to supply stimulus to and analyze the results from the HDL simulation.

To accelerate the execution of multiple test cases, Harris engineers developed a MATLAB control script that managed multiple parallel cosimulations across a Linux computing grid.

The team used this approach to rapidly verify individual subsystems as well as the overall signal processing chain before synthesizing and successfully demonstrating the device. Harris engineers have begun applying the same approach on other projects, including one that requires a million-point fast Fourier transform.

“EDA Simulator Link enabled us to greatly reduce functional verification development time by providing a direct cosimulation interface between our MATLAB model and our logic simulator. As a result, we verified our design earlier, identified problems faster, completed more tests, and compressed our entire development cycle.”

Jason Plew, Harris Corporation

## THE RESULTS

■ **Functional verification time cut by more than 85%.** “It used to take an experienced engineer about 8 hours to prepare a subsystem test bench,” says Plew. “With EDA Simulator Link, we can create one in an hour, and because MATLAB generates the stimulus and performs the analysis, we have all the test results ready in MATLAB for postprocessing.”

■ **100% of planned test cases completed.** “Our previous process was so complex that we often performed little verification at the subsystem level, and our schedule kept us from running more than 30% of the desired test cases at the system level,” says Plew. “With MathWorks tools, we generated and simulated 100% of those test cases using an automated process that enables us to find most defects at the subsystem level.”

■ **Design implemented defect-free.** “We eliminated weeks of lab debug time because we had so thoroughly verified the design by using EDA Simulator Link to cosimulate our HDL code in MATLAB and Cadence Incisive,” Plew notes. “In fact, the data path elements of the FPGA performed as designed right from the start.”

Learn more about Harris:  
[www.harris.com](http://www.harris.com)

### INDUSTRY

- Aerospace and defense
- Electronics and semiconductors
- Communications

### APPLICATION AREAS

- Digital signal processing
- FPGA design
- Verification, validation, and test

### PRODUCTS USED

- MATLAB®
- EDA Simulator Link™
- Filter Design Toolbox™
- Cadence® Incisive®
- Xilinx® Virtex®-5

[www.mathworks.com](http://www.mathworks.com)



© 2010 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

91778v00 04/10