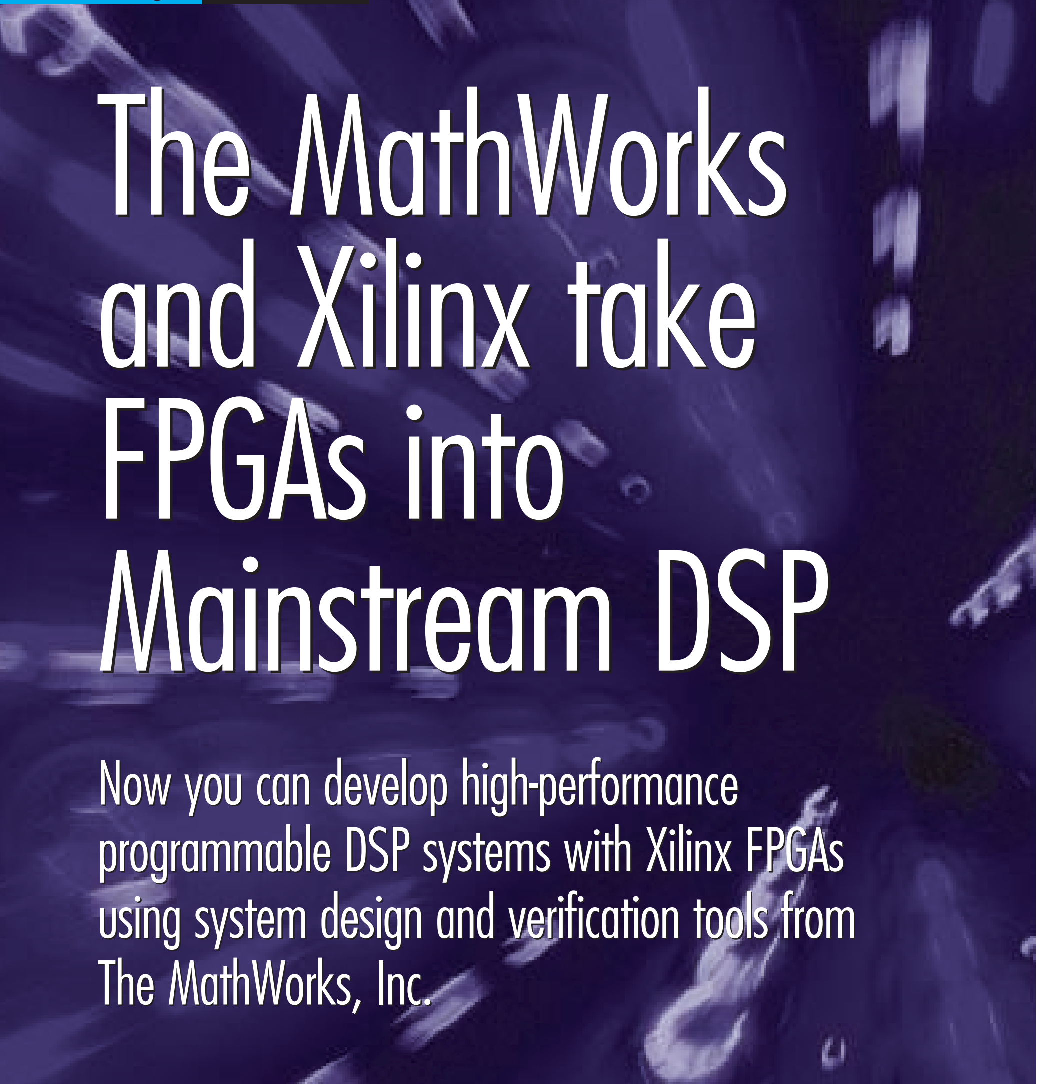

The MathWorks and Xilinx take FPGAs into Mainstream DSP

Now you can develop high-performance programmable DSP systems with Xilinx FPGAs using system design and verification tools from The MathWorks, Inc.

by Per Holmberg
Sr. Product Marketing Manager, Xilinx
per.holmberg@xilinx.com

Anne Mascarin
DSP Market Segment Manager, The MathWorks
amascarin@mathworks.com

The MathWorks and Xilinx have entered a strategic exclusive alliance and joint development agreement for the system-level creation of FPGA-based DSP designs. The first product will be the Xilinx System Generator for Simulink(tm) software. Used with The MathWorks' popular Simulink system-level design tools, and the Xilinx CORE Generator and LogiCORE DSP algorithms, this software is the first to bridge the gap between system-level DSP design and FPGA implementation, allowing you to easily design high-performance DSP applications in Xilinx FPGAs. The overall system flow is illustrated in Figure 1.

System Overview

The Xilinx System Generator automatically generates hardware description language (HDL) code from a system representation model in Simulink. The HDL design can then be synthesized for implementation in Xilinx FPGAs using standard hardware synthesis software. To maximize predictability, density, and performance, the System Generator automatically maps blocks in the system design into optimized LogiCORE algorithms (cores). With only one representation of the design and no manual intervention when translating the system-level design to HDL, a common source of errors is removed.

You can significantly reduce development time by quickly iterating between the system-level model in Simulink and the hardware implementation. This is especially important for DSP applications, because many system-level design trade-offs are based upon the results of the hardware implementation. Consequently, your development time is significantly reduced, and moreover, even if you are new to FPGAs, you can use the familiar tools from The MathWorks, along with the Xilinx FPGA development tools, to create FPGA-based DSP applications. This combination not only gives you the ease-of-use and time-to-market advantages of FPGAs but also the highest possible performance.

The System Generator

The System Generator consists of two components to help you proceed from a system model to actual hardware. The Xilinx Block Set (XBS) allows you to embed bit-true and cycle-true models of FPGA-specific circuitry into a Simulink design, while the System Generator translation software converts the Simulink model into synthesizable VHDL, with Xilinx FPGA hardware as the target.

The XBS provides these elements:

- Parametric blocks for DSP, arithmetic, and logic functions.
- Gateway blocks to communicate with the Simulink environment, where you have access to the extensive set of Simulink DSP libraries.*
- Special tokens to support user-defined black boxes and to invoke the System Generator interface to the Xilinx FPGA software.

The XBS provides the functionality of:

- Simulink S-functions - the Simulink representation of the XBS.
- Synthesizable VHDL - the hardware representation of the XBS, including the use of Xilinx cores where appropriate.

The System Generator software token activates the translation software that converts a Simulink model built from XBS elements into synthesizable VHDL. The VHDL generated may include cores for appropriate functions, as well as corresponding simulation models.

The System Generator graphical user interface (GUI) allows you to customize the Simulink simulation. For example, it can hierarchically override fixed-point values with doubles. This is particularly useful during design and debugging.

Simulink

You can model a VHDL design using any combination of Simulink blocks. By using the XBS black box token, you can then instantiate the design into a generated VHDL model. The black box customization GUI encapsulates the design information necessary for the compiler to create the correct instantiation interfaces. This black box support allows the abstraction of commonly used control signals and ports, and then infers them within the generated VHDL.

Working in Simulink, you create a model of the hardware system as well as one or more test environments in which to simulate the model. When the model is first entered, simulation is typically performed using floating-point data to verify its theoretical performance. You then add the appropriate XBS elements, and then convert data types to the bit-true representations used in the hardware implementation. Then the model is re-simulated to verify its performance with quantized coefficient values and limited data bit widths, which can lead to overflow, saturation, and scaling problems. User-defined black boxes can also be incorporated in the modeling and elaboration process. When the model has been converted to a form realizable in the FPGA and its performance meets specification, you can invoke the netlister and the test bench generator.

The Netlister and Mapper

The netlister extracts a hierarchical representation of the model structure annotated with all the element parameters and signal data types. A mapper then analyzes the elements in the hierarchy and creates a VHDL description of the design. Where possible, the mapper uses the Xilinx CORE Generator to generate optimized LogiCORE implementations for specific design elements.

When an element or its parameter values imply functionality unavailable in the CORE Generator, the mapper instantiates a reference to a parameterized, synthesizable entity in a synthesis library or your own supplied model.The actual hardware entities used have additional inputs and outputs for control signals that are not evident at the level of abstraction used in Simulink. The mapper inserts the necessary control ports and connects them to control logic blocks.

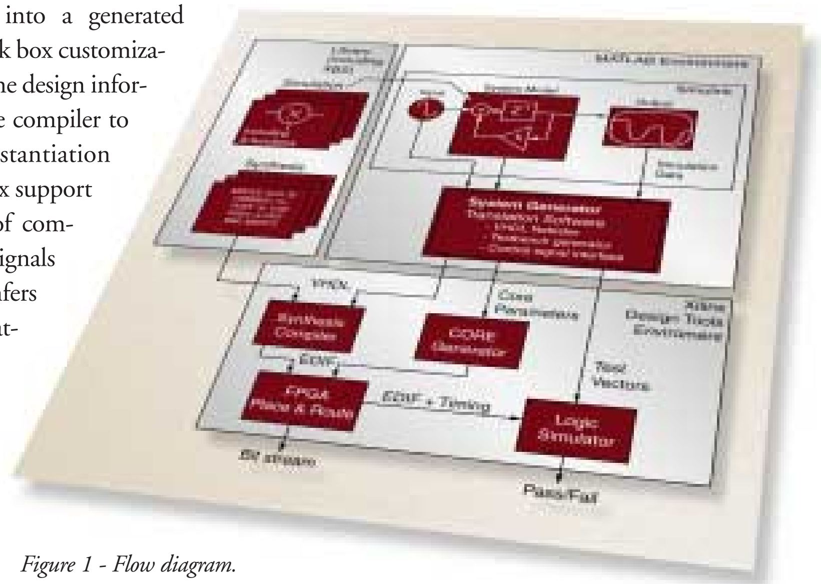

Figure 1 - Flow diagram.

About The MathWorks

The MathWorks develops technical computing software for engineers and scientists in industry and education. An extensive family of products, based on MATLAB and Simulink, provides high-productivity tools for solving challenging mathematical, computational, and simulation problems. For more information see www.mathworks.com

Multi-rate clocking is supported through time step information provided during simulation. Each control logic block is given a default synthesizable behavior which may require alteration to achieve an efficient implementation.

The Testbench Generator

The testbench generator is an interactive tool that runs in the MATLAB environment, in which you capture the input stimuli and system outputs of selected simulation runs for conversion to test vectors. The generator converts the captured simulation data into VHDL code that will exercise the implemented model and test its outputs against the expected results.

Logic Synthesis

The Xilinx Foundation Series, or any synthesis compiler supported through the Xilinx Alliance Series software tools, can be used to synthesize the control logic and elements for which no hardware macros exist, and combine all the pieces into a fully realized netlist. The outputs of this back-end process are a bit stream and an EDIF structural netlist of the hardware, annotated with timing information. This netlist can be simulated with the test vectors produced previously from system simulations to verify the performance of the completed FPGA hardware realization.

Conclusion

With the introduction of the Xilinx System Generator for Simulink, you now have a tool that makes the job of incorporating FPGAs into your DSP designs easier than ever before. Today, Xilinx FPGAs provide you with the world's highest-performance programmable DSP solution, supporting applications equivalent to 160 billion MACs. As FPGA technology continues to advance, it is expected that by the year 2002, you will have access to a ten million gate FPGA offering 0.6 Tera MACS per second.

See www.xilinx.com for more information.