High-Level Specification an Design of DSP Systems

Michael A. Bohm

In the DSP domain, MATLAB & Simulink are the domain-specific language of choice.

These provide both an efficient system-level verification environment and an
efficient path to implementation.

Leading electronic companies in the commercial, military, and aerospace sectors are now instructing their R&D organizations to investigate solutions to solve the most complex design productivity problem they've experienced in more than a decade. Their electronic systems have become so complex that individual devices can no longer be designed in isolation. The entire system must be modeled in a manner that allows the chip design to be verified early and often against this system-level model. Today's chip design methodologies are based on general-purpose design languages such as C++, SystemC, VHDL, and Verilog and do not provide the necessary efficiency to develop a complex, system-level model in a timely manner.

This demand for greater productivity has given rise to a new set of domain-specific languages (DSLs) that promise a superior solution for system-level design. DSLs are programming languages that sacrifice generality for suitability to a particular problem area. By reducing the conceptual distance between the problem space and the language used to express the problem, programming becomes simpler, easier, and more reliable. The amount of code that must be written is dramatically reduced, increasing productivity and decreasing maintenance costs. Well architected DSLs provide constructs that allow concise representation of large design objects, come complete with visualization tools tuned for the specific design domain, and provide links to the hardware and software implementation processes.

A great example of a DSL now emerging for the logical design of FPGAs and ASICs is

SystemVerilog. The mission of SystemVerilog, as defined by Accellera, is "to dramatically improve productivity in the design of large gate count, IP-based, bus-intensive chips." SystemVerilog is targeted primarily at chip implementation and verification flow, with powerful links to the system-level design flow, and is based on several other DSLs, including SuperLog, Verilog, Vera, OVA, and PSL/Sugar. Improvements in system-level verification capabilities are realized over VHDL and Verilog, while maintaining an efficient path to implementation through synthesis. SystemVerilog has achieved these improvements by focusing on the problem of system-on-chip design, but, unfortunately it does not offer these same advantages to the digital signal processing (DSP) designer.

In the DSP domain, MATLAB® and Simulink are the DSL of choice and provides both an efficient system-level verification environment and an efficient path to implementation. Built-in abstractions liberate the designer from the strict modeling style guides that are required by general purpose languages, allowing large design objects to be represented with a high degree of efficiency. The typical example below can produce large number of gates with vary performance and multiple macro-architectures. "

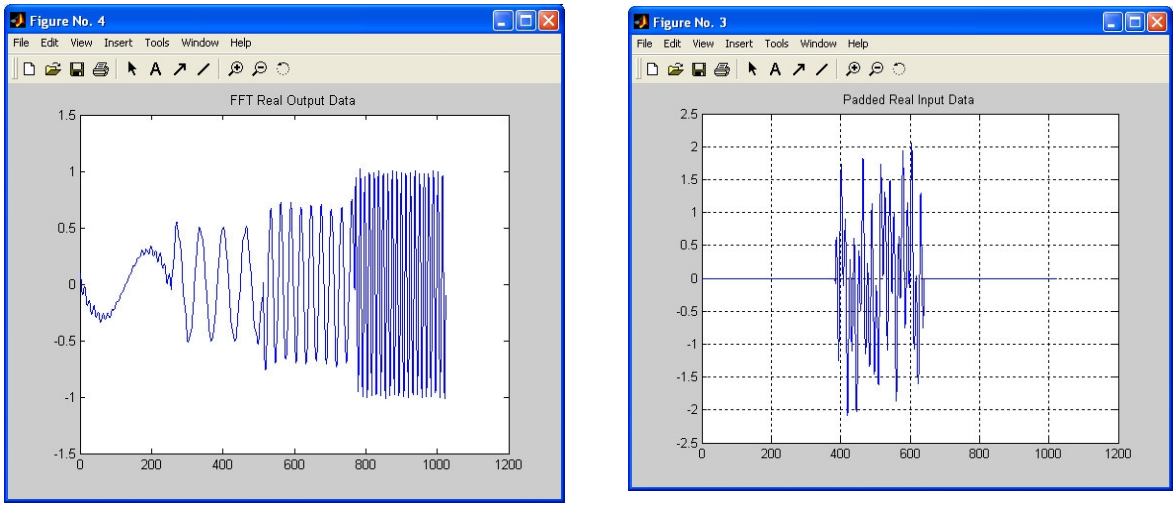
$$\text{“}y=FFT(x.)$$

Table 1 summarizes the results of a case study conducted by The MathWorks, which compares the development efforts to implement the IEEE 802.11b wireless LAN specification in MATLAB vs. C/C++.

Modeling Language	**# of Files**	**Lines of Code**	**Development** (hrs)
C/C++	17	2000	40
MATLAB	1	53	3

The market demand for a more efficient path from MATLAB or Simulink to an ASIC or FPGA has given rise to a new breed of EDA companies, such as AccelChip Inc. and Xilinx that bridge the gap between DSP algorithm development and silicon. AccelDSP and System Generator extends the capabilities of MATLAB as a DSL for chip design by automatically converting a floating-point MATLAB model to a fixed-point, synthesizable VHDL or Verilog model suitable for standard ASIC and FPGA

MathWorks also provides a complete set of advanced graphical tools for data analysis, visualization, algorithm development, and stimulus generation for analysis of these complex algorithms. These are key functions which ensure the system level design is performing correctly.



MATLAB Visualization Graphs

The MATLAB and System Generator DSL currently provides an implementation path when the target hardware platform is a commercially available DSP processor or software running on a standard processor. Both these implementations, however, are performance limited. High performance DSP applications require an ASIC or FPGA as the hardware platform, but targeting these devices will sacrifice all direct paths to implementation currently provided by the MATLAB environment. This “gap” requires the DSP model to be re-implemented in a format suitable for chip design. This can introduce problems into the design process, such as long development cycles, human error, and reduced flexibility to make algorithm changes.

The market demand for a more efficient path from DPS to an ASIC or FPGA has given rise to a new breed of EDA companies, such as AccelDSP Inc and System Generator., that bridge the gap between DSP algorithm development and silicon. Both System Generator and AccelDSP extends the capabilities of MATLAB as a DSL for chip design by automatically converting a floating-point MATLAB model to a fixed-point, synthesizable VHDL or Verilog model suitable for standard ASIC and FPGA.