



designfeature *By Gabe Moretti, Technical Editor*

Design complexity requires system-level design

THE AMBIGUITY OF PRODUCT SPECIFICATIONS WRITTEN IN A NATURAL LANGUAGE OFTEN INTRODUCES DESIGN FLAWS THAT GO UNDETECTED UNTIL LATER, MAKING THE SOLUTION MORE EXPENSIVE.

At a glance28

The evolution of Matlab and Simulink.....28

For more information.....32

ENGINEERS WERE DESIGNING at the system level even before the invention of the transistor. But the focus on system-level design has become greater as design complexity has increased. Product specifications generally comprise hundreds of pages of written documents that are too often ambiguous and difficult to interpret. Ken

Karnofsky, director of marketing, signal processing, and communications at The MathWorks, observes, "Design flaws are introduced at the specification stage and typically not detected until the validation phase." These flaws make development much more expensive than it has to be, because developers have to redo work under the pressure of much tighter schedules (**Figure 1**).

In the last four years, market-researcher Dataquest has begun to report on a segment of the EDA market that Gary Smith, its chief analyst for EDA

worldwide, has labeled the Electronic System Level. Products in this market aim to enable engineers to describe an electronic design at an abstraction level higher than RTL (register-transfer level), the traditional starting point for hardware design since the introduction of logic synthesis. Although many EDA companies have introduced products for this market, revenues have grown significantly less than expected, and a leading technical trend remains undeveloped. Designs' continued growth in complexity and the introduction of the 65-nm

semiconductor-fabrication process make solving the system-design problem even more imperative.

TRADITIONAL APPROACH

Most EDA companies have approached the complexity problem by assuming that design engineers would naturally become systems designers if only familiar tools were available. Therefore, the industry is witnessing three major and distinct efforts to extend the language-based design used at the RTL to the system level by providing constructs that can describe concept at a higher level of abstraction. Each effort uses an existing language as its starting point—namely, Verilog, VHDL, or C.

The Accellera organization, which promotes worldwide standards, has extended the Verilog language using contribu-

AT A GLANCE

▶ Increasing design complexity requires better system-design tools.

▶ Hardware designers need retraining to become proficient system designers.

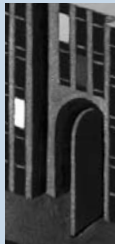
▶ Languages designed to develop and prove algorithms are best for system design.

tions from its member companies to create SystemVerilog—a proper superset of Verilog that adds many behavioral constructs. Because Verilog is the most widely used language for RTL design, the consortium decided that extending it would minimize the need for retraining engineers and result in considerable cost savings for the industry.

The IEEE has begun an effort within the Computer Society DASC (Design Automation Standards Committee) to extend VHDL for the same purpose. The

VHDL base language already provides most of Verilog's additional behavioral constructs, simplifying, according to the committee, the transition to a new version of the language. However, this work is proceeding slowly and runs the risk of missing the market window.

THE EVOLUTION OF MATLAB AND SIMULINK



In 1984, Cleve Moler, Jack Little, and Steve Bangert founded The MathWorks. The company's mission was to make Matlab into a product. Computer scientist Moler had, in the late 1970s, developed MatLab in Fortran. Little and Bangert

translated the program into C and ported it to the PC shortly before the company's founding. Matlab is a high-level technical-computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation.

Matlab allows users to solve technical-computing problems faster than with traditional programming methods and in a range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. It provides a number of features for documenting and sharing work, and users can integrate Matlab code with other languages and applications.

Simulink, another MathWorks product, is a platform for multidomain simulation and model-based design of dynamic systems. It

provides an interactive graphical environment and a customizable set of block libraries that let users accurately design, simulate, implement, and test the same systems they solve using Matlab.

In 2004, Mentor and The MathWorks developed a method of linking Simulink with ModelSim, thus allowing engineers to concurrently simulate both algorithms and hardware implementations. The product, Link for ModelSim—which received the 2004 *EDN* Innovation award for EDA—narrows the gap between algorithm and hardware implementation by adding Verilog or VHDL to the simultaneous direct mixed-language-simulation capabilities of Matlab and Simulink.

Vojin Zivojnovic, co-founder of Axy Design Automation, which ARM recently purchased, explains, "Matlab is a software tool that almost transparently converts the mathematical theory of signal processing and control systems into real-life execution. Its deep entrenchment in well-established and barely changing mathematical theory allowed the tool to steadily evolve from early statistical libraries of the '60s and '70s. From its primary use in academia, Matlab spread into the industry over two decades, forming one of the largest user communities in the technical-software market."



Another significant effort based on C or, more precisely, C++ produced SystemC, which the OSCI (Open SystemC Initiative) consortium promotes and a number of companies have adopted. Proponents of this approach argue that because most engineers have learned to use C at the university level to complete class assignments, a C-based tool would not require significant retraining.

Although the technical details of the three approaches differ (Reference 1), each has the same premise: Engineers doing designing at the RTL will also design at the system level. However, even a superficial study of the history of product development, whether in EDA or other industries, questions that assumption. In the electronic industry, design engineers are not involved in the initial product specification, developing functional requirements, and setting project guidelines and market-positioning parameters. Rather, these tasks fall to marketing, finance, and systems-engineering professionals, who are generally unfamiliar with hardware-description languages. These individuals have used documents written in a natural language and pictures to describe a system because nothing else has been available. For them, using SystemVerilog or VHDL, two languages with heavy hardware-implementation semantics, or C++ with its structured-programming constructs, is as difficult as using a new language written for system-level design.

Cadence, Mentor, and Synopsys—the three leading EDA companies—are supporting both SystemC and SystemVerilog, although with varied emphasis. Cadence is more active in SystemC promotion, and Synopsys champions SystemVerilog. Mentor has taken a neutral approach, providing equal support for both. Companies that support SystemC have been more vocal in defending their strategy at conferences and in the press. Small companies that best exemplify this approach to system design include Celoxica, Summit Design, and CoWare.

Celoxica focuses support on the implementation of designs with FPGA devices, believing that, given the proper tools, any soft-

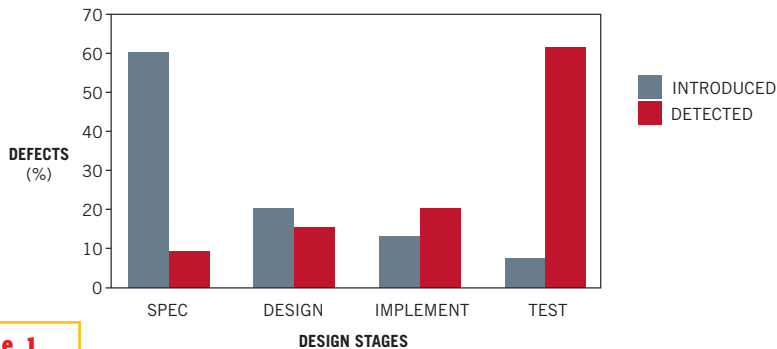


Figure 1

Traditional methods identify design problems late in the design cycle.

ware designer can code a correct algorithm and generate working hardware. Summit Design believes that only by using C and SystemC can designers produce a complete system design. Emil Girzyc, president and chief executive officer of Summit, explains, “Algorithm design is only one part of the design problem facing SOC (system-on-chip) teams. After using Matlab, the design team must determine the optimal implementation of the algorithm. The performance, power, and architectural trade-offs are best performed in a procedural modeling language—not Matlab.” In addition, he maintains that design teams must also design the control-dominated portions of the application and integrate IP (intellectual-property) models. “This is where C and SystemC come in and where they will replace Verilog and VHDL as the languages of choice,” he says.

Almost 15 months ago, CoWare got

new life when Cadence invested in the company and gave it the rights to its SPW product, which provides an environment for system-level development of communication systems. According to Johannes Stahl, director of marketing for SPW, “CoWare addresses the mainstream market with a direct replacement for Simulink, used for C-based verification across a very wide range of Matlab users.” Critics argue, however, that this approach places too much emphasis on the silicon. When designers must use fabrication processes that employ optical methods to produce features with sizes below the wavelength of visible light, developing unique working patterns on silicon becomes costly, and error probability increases significantly; sometimes, the cost of fixing these errors results in design teams’ having to scrap entire projects. Using the most functionally powerful known building blocks to implement the

IC may alleviate the problem. Therefore, many companies are turning to platforms, which are application-specific collections of IP cores. With this approach, the proprietary portion of the design, whether hardware or software, is more manageable and thus less prone to errors.

NOVEL SOLUTION

According to Tom Feist, vice president of marketing and international sales at AccelChip, the demand for greater productivity has given rise to a new set of DSLs (domain-specific languages) that promise a better approach to system-level design: “DSLs are programming languages that sacrifice generality for suitability to a par-

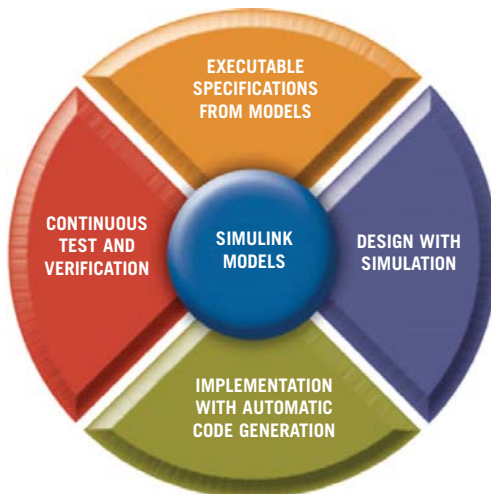


Figure 2

A model-based design method allows engineers to continuously refine the specification while maintaining a verifiable link to the implementation.

ticular problem area. 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.”

In the last two years, the DSL that has generated the most interest in the EDA community is Matlab from The MathWorks (see sidebar “The evolution of Matlab and Simulink”). Matlab is not a DSL in the strict use of the term, however, because engineers and scientists use it to solve problems in many application areas. Within EDA, designers have used Matlab to develop algorithms that are appropriate to both the wired- and wireless-communication fields. Companies such as Accelchip, for example, allow designers to develop and test their algorithms in Matlab and provide tools to implement those algorithms in hardware.

The industry has for some years debated how to arrive at an executable specification and created languages such as UML (Unified Modeling Language) to aid in solving the problem. The MathWorks and some EDA companies that have developed support for Matlab describe a process called model-based design. It replaces a written specification with a system model comprising algorithmic blocks that describe the desired behavior without implying a specific implementation method (Figure 2). Using Matlab and Simulink, a design team can try various architectures, ensure that

You can reach Technical Editor Gabe Moretti at 1-941-497-9880, fax 1-941-497-9887, e-mail gmoretti@edn.com.



each version can be tested consistently, make certain that RTL code can be generated from the design, and thus maintain consistency by avoiding manual-translation steps. This approach is the best option for system-level design because it allows a design team to verify that the requirements are realistic, giving an implementation a better chance of being

on time and on budget. Vinod Kathail, vice president of engineering and chief technology officer at Synfora, points out, “Matlab provides some significant benefits in terms of providing a familiar and easy-to-use environment for rapidly creating and exploring algorithm alternatives.”

In addition to Mentor, both Synopsys and Cadence provide a method for their customers to use Matlab during hardware development. Synopsys, in fact, has since 2000 allowed its System Studio customers to use Matlab. But only recently, with Matlab’s growing popularity, has the company heightened its public commitment to the language. Both Altera and Xilinx offer their customers methods that allow the use of Matlab and Simulink during FPGA development and refer customers to The MathWorks if they want to purchase the tools. □

REFERENCE

1. Moretti, Gabe, “The search for the perfect language,” *EDN*, Feb 5, 2004, pg 40.

TALK TO US

Post comments via TalkBack at the online version of this article at www.edn.com.

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, contact any of the following manufacturers or organizations directly, and please let them know you read about their products in *EDN*.

Accelchip
www.accelchip.com

Catalytic
www.catalyticinc.com

Mentor
www.mentor.com

Summit Design
www.summit-design.com

Accellera
www.accelera.org

Celoxica
www.celoxica.com

Synfora
www.synfora.com

Tensilica
www.tensilica.com

Altera
www.altera.com

CoWare
www.coware.com

Synopsys
www.synopsys.com

The MathWorks
www.mathworks.com

ARM
www.arm.com

CriticalBlue
www.criticalblue.com

Synplicity
www.synplicity.com

Xilinx
www.xilinx.com

Cadence
www.cadence.com