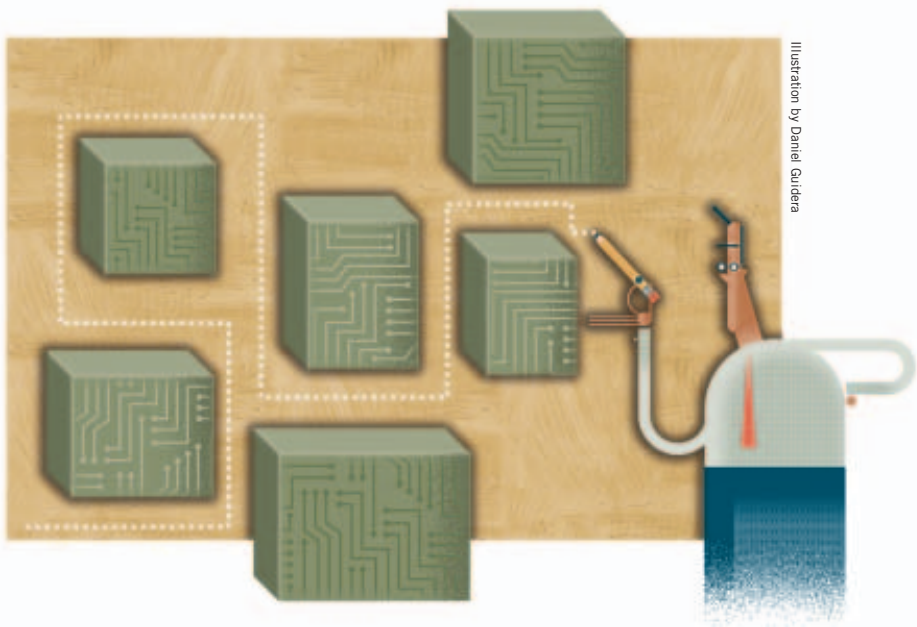


THE DESIGN METHOD IS BECOMING POPULAR, DESPITE A LACK OF A CLEAR DEFINITION OF WHAT CONSTITUTES A PLATFORM.



Platform-based design:

BLOCKS AND BUSES LEAD THE WAY

ENGINEERS DEAL with complex systems by dividing them into subsystems, almost always using functions as the deciding criterion. When you look at a complex ASIC, for example, you see a number of functional blocks connected by buses. Depending on the application market the ASIC is targeting, many of the blocks are common or have functions in common with other products tar-

geting the same market, regardless of the company and design team that implemented it. A chip that targets a graphics application, for example, is likely to contain a graphics-accelerator engine, a graphics memory to buffer display data, a microcontroller, and memory for software that controls the operation of the system. Semiconductor companies and fabless foundries have developed many products with architectures that target popular application areas, such as graphics acceleration and both wired and wireless communications. Vendors market

these products under the category name of ASSPs (application-specific standard products). Products in this area generally offer no way to customize the application within the chip. You must implement your functions using components outside the device.

Companies that have substantial engineering resources often opt to develop their own ASSP by organizing a number of reusable and new functional blocks to develop a proprietary architecture. Some of these blocks usually implement functions using official standards. These

<i>At a glance</i>	56
<i>Exploiting analog platforms in analog/digital co-design</i>	56
<i>For more information</i>	58



blocks do not change from one implementation to another and have a common interface. This observation was an important factor in the development of the virtual-component, or IP (intellectual-property), market. Developing an architecture that uses both proprietary and third-party cores to target an application is a common method in SOC (system-on-chip) design. If you properly develop it, IP reuse saves both development and verification time. This approach helps improve the chance that the product will hit its market window and that it will be profitable.

A collection of properly organized virtual components constitutes a comput-

AT A GLANCE

▶ Platform-based design is more restrictive than ASIC design, but development is faster and cheaper.

▶ Platform-based design can eliminate many of the physical problems you encounter in ultradeep-submicron design.

▶ The VSI Alliance has divided platforms into technology-, architecture-, and application-driven platforms.

▶ A number of vendors support platform-based design methods.

ing platform. The fundamental difference between a platform and an ASSP is flexibility. A platform provides more flexibility to designers by offering ways to customize the product within the device. When engineers use a platform to implement a product, they are doing platform-based design. In the most general sense, any computing system is a platform, but the general definition is not useful, because it does not indicate a methodology. Alberto Sangiovanni-Vincentelli, PhD, who won the Phil Kaufman Award for outstanding work in EDA technology and who holds the Edgar L and Harold H Buttner Chair of electronics engineering and computer science at

EXPLOITING ANALOG PLATFORMS IN ANALOG/DIGITAL CO-DESIGN

By Fernando De Bernardinis and Alberto Sangiovanni-Vincentelli, PhD, University of California–Berkeley

Traditionally, analog-design engineers have designed systems after defining a rigid analog/digital interface and of a set of performance figures that the analog subsystem has to meet. This approach is by no means optimal from the system point of view, and it prunes the potentially large design space that originates from moving the analog boundary and changing performance requirements. It also severely limits the possibilities of a tight analog/digital interaction, because it reduces the amount of control data involved at the analog/digital interface.

Analog platforms allow effective co-design of analog and digital components, thus complementing digital platforms. You can apply analog platforms at different levels of abstraction—from custom circuits, in which the architectural space is a continuum, to programmable fabrics such as Anadigm’s FPAA’s (field-programmable analog arrays), which provide a discrete design-configuration space. The synchronizer and demodulator for the Picoradio receiver at the BWRC (Berkeley Wireless Research Center) provide a concrete example of analog/digital co-design that exploits platforms. The principal aim of this design

is prototyping architectures with minimum power consumption. Because data converters consume most of the energy in digital receivers, the candidate approach should be as much as possible relax ADC requirements by properly processing analog signals.

This design uses an implementation platform comprising an Anadigm FPAA and a Xilinx FPGA. Because the FPAA is dynamically reconfigurable, the overall platform can operate adaptively with varying input conditions. The Picoradio transceiver provides variable-rate communication, ranging from 10 to 160 kbps with on/off keying and low-power-consumption constraints that severely affect noise performances and gain of the RF receiver’s front end.

The basic function the designers need to implement is the advanced delay-synchronization scheme for on/off keying with the corresponding integrator-based demodulator. You can implement the ADC with a simple comparator. You cannot implement this scheme on the FPAA platform as is. You must refine its functions and map them on your selected platform. This modification requires a slightly more complex control

algorithm to synchronize the radio but requires an analog subsystem that you implement on the FPAA. You model the digital control as a set of finite-state machines with a digital interface to the FPAA, and you implement it on the FPGA. The analog baseband section comprises two integrators, two comparators, a delay stage, some filtering stages to lower the input noise, and an input amplifier, exploiting all the FPAA’s resource except for one amplifier.

The design uses Simulink (www.mathworks.com) behavioral models for the platform. The designers use the FPAA synthesis tools to map high-level analog components into switch configurations for the switched-capacitor fabric and BWRC’s Stateflow-to-VHDL flow to map the digital section on the FPGA. The platform’s abstraction level guarantees the system ability to be implemented. Because the FPAA presents noise and offsets in the switched-capacitor fabric, the low-offset programmable-gain input amplifier maximizes the signal level. You can adaptively set the gain of the input amplifier, through proper interaction with the digital controller, to maximize SNR in the FPAA signal path with the input signal,

thus avoiding saturation and clipping. You can thus extend sensitivity from -42 to -51 dBm, which is the achievable limit given the noise in the RF front end. Furthermore, you can adjust filter shapes and integrator gain to different bit rates, so that you can achieve optimal performance over the entire range of operation. Hence, you implement a digitally controlled variable-gain, variable-bandwidth analog baseband. The platform paradigm plays an important role in determining the tight interactions between analog and digital components that allow you to overcome intrinsic analog-performance limits. It thus effectively provides an alternative design approach. As a result, you can obtain satisfying performances for the Picoradio receiver exploiting the flexibility of the FPAA platform.

AUTHORS’ BIOGRAPHIES

Fernando De Bernardinis is a graduate student researcher at the University of California–Berkeley.

Alberto Sangiovanni-Vincentelli, PhD, holds the Edgar L and Harold H Buttner Chair of electronics engineering and computer science at the University of California–Berkeley.

the University of California—Berkeley, has proposed a more useful definition of the term “platform” (Reference 1): “A hardware platform is a family of architectures satisfying a set of constraints imposed to allow reuse of hardware and software.” But even this definition is too imprecise. Without a more restrictive definition, the term “architectures” encompasses any regular hardware or software structure. Thus, a microcontroller with sufficient embedded memory would fall within Sangiovanni-Vincentelli’s definition, as would an FPGA, a structured ASIC, or a general-purpose computer. All satisfy a set of constraints to allow the reuse of hardware and software. Stating that: “A hardware platform is a system providing an architecture consisting of both hard and soft functional blocks satisfying a set of constraints to allow product customization within the component” provides a more precise definition of a platform.

Using a platform to implement a product is more restrictive than using an ASIC, because the predefined functions of the blocks that constitute the platform constrain the design team. For this reason, platforms will never completely replace ASIC design. If a company needs to achieve the higher performance and packaging density possible with a given process node, it will need to develop an ASIC. This approach costs more and takes longer to develop, but the resulting superior capabilities of the chip should produce a more competitive product than alternative solutions.

But new processing nodes at 130 and 90 nm require much greater under-

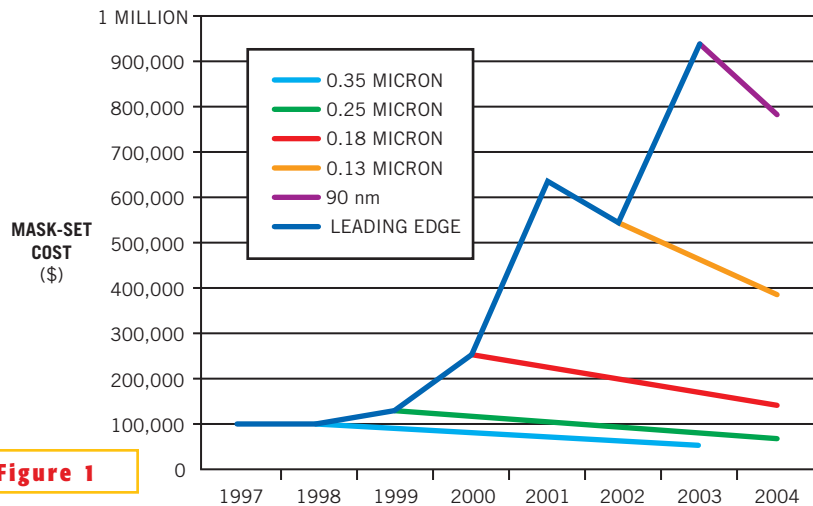


Figure 1

The cost of the mask set is the largest component of a design’s nonrecurring costs.

standing of semiconductor device physics and process technology than the average design team knows. Therefore, alternative methods, such as platform-based design, are becoming popular. Companies are willing to trade optimal performance for lower development costs and shorter time to market.

Only a few years ago, a design team had to develop optimal circuits to achieve the performance that computationally intensive products require, but today’s processes provide sufficient execution speed to even average designs. It is cheaper to adapt the development methodology to the skills of the average designer than to re-educate most designers to understand the physical, mechanical, and optical effects that complicate designing at 130 nm and smaller. Semiconductors companies will find that designs that will

allow them to reach an economically acceptable volume production on 300-mm wafers at 90 nm and smaller will employ some form of platform-based design.

The most significant advantage of platform-based design is the reuse of a verified and manufacturable logic block that is much larger than the single library cell that ASIC design uses. Most of the problems IC manufacturers encounter using 130-nm or smaller processes involve placing and fabricating interconnection traces. Hence, using larger pre-verified blocks lowers the number of unique interconnects. In addition, issues such as power consumption and distribution become more tractable when designers use large blocks that have a verified power profile and well-specified power connectivity. This approach, in turn, decreases the complexity of routing

FOR MORE INFORMATION...

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

Altera
1-408-544-7000
www.altera.com

Chip Express
1-408-988-2445
www.chipexpress.com

Lightspeed Semiconductor
1-408-616-3200
www.lightspeed.com

NEC Electronics
1-408-588-6000
www.necel.com

Telairity Semiconductor Inc
1-408-764-0270
www.telairity.com

Virage Logic
1-510-360-8000
www.viragelogic.com

Anadigm
1-408-879-6677
www.anadigm.com

CoWare
1-888-269-2738
www.coware.com

LSI Logic
1-866-574-5741
www.lsillogic.com

Palmchip
1-408-952-2000
www.palmchip.com

Tensilica Inc
1-408-986-8000
www.tensilica.com

VSI Alliance
www.vsi.org

ARC International
1-408-437-3400
www.arc.com

Fujitsu Technology Solutions Inc
1-877-213-6674
www.us.fujitsu.com

Mentor Graphics
1-800-547-3000
www.mentor.com

Synopsys
1-800-541-7737
www.synopsys.com

Vast Systems Technology
1-408-328-0909
www.vastsystems.com

Xilinx
1-408-559-7778
www.xilinx.com

ARM
1-408-579-2200
www.arm.com

Synplicity
1-408-215-6000
www.synplicity.com

and allows for greater flexibility in die fabrication. Minimizing the impact of physical effects and simplifying the placement-and-routing task opens ultradeep-submicron processes to average designers, thus ensuring sufficient production volume to make the new fabrication processes economically feasible.

To provide the greatest flexibility to ASIC designers, today's cell libraries provide many cells implementing the most basic logic functions. The libraries also provide a number of cells that implement the same logic functions but have different physical characteristics to allow designers to meet product requirements. The percentage of designers that need access to such basic cells will continue to significantly decrease as 90- and 65-nm and lower processes reach production volumes. ASIC design using ultradeep-submicron processes requires the close partnership of designers, manufacturers, and EDA vendors to succeed.

The new-product-development paradigm may change not only how companies structure themselves, but also how the financial revenue model most likely evolves for all three parties involved in ASIC- and custom-product design. To support an average designer, the concept of a cell library must in the next few years evolve together with the approach to synthesis. Substituting "function units" for the traditional logic cells might make "behavioral" synthesis a reality. Synthesis tools will have to hierarchically parse designs. They will first have to partition the designs according to functions and then parse the remaining portion using digital logic. In addition, most designs using platform-based methods do not need the hundreds of millions of transistors available to implement the required functions.

The electronics and EDA industries must take advantage of this general surplus of transistors to implement self-test and redundancy in most—if not all—function blocks. Manufacturers may be able to produce wafers with built-in test circuits that designers can connect to functional units. Even better, vendors can create EDA tools that automatically build scan chains by connecting functional blocks and test structures. The industry must also develop and embrace standards to cover interblock interfaces, design-rule syntax and semantics, libraries, and flexible distribution of hard macros. Adoption of standards will both reduce

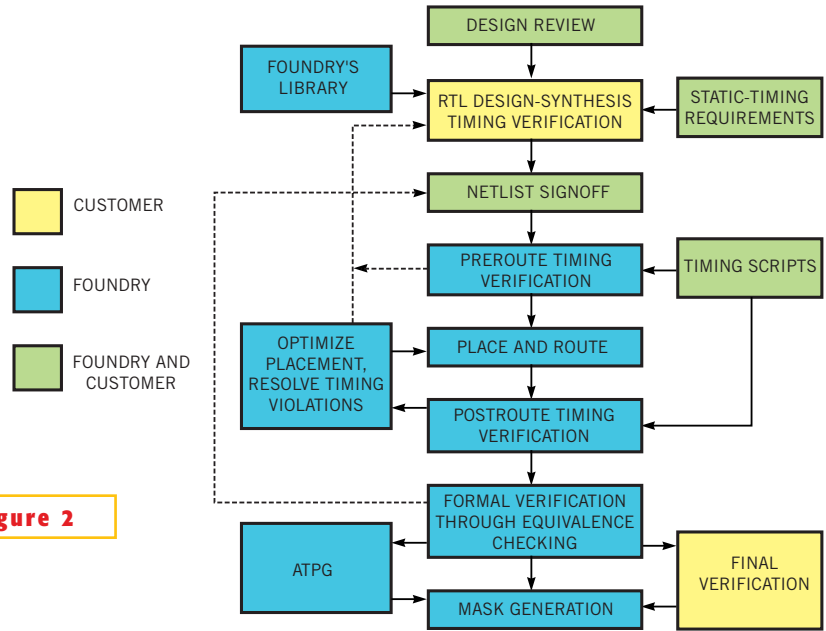


Figure 2

Lightspeed's design flow is a typical example of platform-based design methods.

the overall cost of IC design and generate opportunities for further engineering advances by formalizing what works and thus highlighting potential areas of improvement.

TYPES OF PLATFORMS

Most platform-based-design products target digital design. However, more and more designs now contain analog portions (see sidebar "Exploiting analog platforms in analog/digital co-design" and Reference 2). Semiconductor companies and EDA vendors offer a number of ways to implement and use a platform. The VSI Alliance is developing a document that provides a definition and taxonomy of platform-based design. The work identifies technology, architecture, and application-driven platforms.

Technology-driven platforms fall within Sangiovanni-Vincentelli's definition of a platform but outside my definition. These platforms tend to have cell libraries that offer small functional units, such as adders, multiplexers, and I/O drivers. A few vendors also offer memory blocks with specific characteristics. The existence of standards makes these products feasible, because vendors can prefabricate functional units that implement the most popular logic blocks. The industry recently labeled products in this category as "structured ASICs," recognizing that, although the design-to-man-

ufacturing methods are similar to those of platform-based design, the macrocell library structured ASICs use does not constitute a platform. Structured ASICs offer a way to build a product that is more economical and straightforward than using a traditional ASIC-development flow because they require only one to three custom masks instead of the 20 to 30 necessary to fabricate a standard ASIC product. The mask-set cost approaches \$1 million for a standard, 90-nm ASIC (Figure 1). In comparison, the mask cost for a structured ASIC can be as low as \$100,000. The products vendors develop with this technology require more area, and designers have less flexibility in choosing an implementation than when using a standard ASIC. However, given that the vendors preverify and qualify the logic blocks for the process, the disadvantage is small and financially irrelevant for most designs.

Lightspeed Semiconductor and its customers follow a design flow when using the company's Luminance structured-array product (Figure 2). The flow is typical for designs manufactured using technology-driven platforms. The foundry performs most of the work, and the flow requires team collaboration between it and the customer. In many respects, the flow is similar to the one that gate-level netlist signoff uses at 180-nm and larger processes: The customer needs no know-

ledge of the process characteristics or the requirements it imposes on the place-and-route functions.

Architecture-driven platforms fall within my definition of a platform. These platforms generally center on a CPU core from ARC, ARM, MIPS, or the semiconductor company that markets the platform. They offer a number of blocks implementing graphics and communication protocols, DSP, bus protocols, and substantial amounts of memory. These platforms are not specific to a process node but may be specific to a semiconductor vendor, due to license limitations for some or all of the functional blocks that constitute the platform. The advantages of using this platform derive from reuse.

The vendor often offers IP blocks that it has verified as functional blocks and that therefore come as hard macros. A hard macro is a functional block that a designer has already placed and routed; therefore, the implementers of the block have already solved the issues involving interconnection problems, such as signal integrity. Moreover, designers can count on the fact that other designs have used the blocks, so only a small probability exists that remaining bugs will impact the new design. In addition, the vendor has fabricated the blocks in the target process, so designers face a less challenging place-and-route task.

Reusing verified blocks shortens the development time because you need not develop the blocks, and it simplifies design verification. After you verify that your own circuitry works correctly, you have to verify only the interaction between the circuitry you created and the reused blocks at the transaction level. Architecture-driven platforms offer no significant advantage in manufacturing costs, because each product needs its own mask set. However, because design verification can account for 70% of the design cost, a significant decrease in the cost of this function has a positive impact. Reusing blocks also decreases design time. The result is that the fixed cost that you need to amortize to reach profitability decreases, and you can more quickly get your products to market. Companies choose architecture-driven platforms for financial and actuarial reasons: to diminish development costs, get to market sooner, and diminish the probability of a respin due to design errors.

My definition of “platform” encompasses application-driven platforms. Products in this category are similar to ASSPs. The marketing method is the major difference between the two types: System houses use platforms internally to develop versions of a product family that targets a market segment, whereas semiconductor vendors sell to all those customers ASSPs that target that market segment. The architecture of an application-driven platform is the least flexible of the three platform types, and vendors must, therefore, carefully plan it, considering the entire family of derivative products using the platform that they plan to later release. Engineers must develop the platform to

COMPANIES CHOOSE ARCHITECTURE-DRIVEN PLATFORMS FOR FINANCIAL AND ACTUARIAL REASONS.

provide the best possible operating profile so that, when they add the variable blocks, the product remains competitive with a standard-ASIC product.

Application-driven platforms offer some advantages over standard ASIC products. First, the development cost is lower due to reuse of circuitry. As with architecture-driven platforms, this ability decreases development and verification time. If the family of products covers a market window of years, the follow-on products benefit from the experience users have gained in using the products, and verification costs can significantly decrease as the vendor improves the test suite using customers’ feedback. Because designers must initially consider an entire family of products, they improve system design, in turn improving the quality of each member of the product family and thus providing a subtle competitive edge. Application-driven platforms are generally proprietary and are the result of development work that system houses perform, sometimes in partnership with a foundry. More system houses are working with foundries as system companies move to 130-nm and smaller processes, because logic designers seldom possess this skill. Another reason for team design that includes both the system and the semiconductor company is the choice of the microprocessor in the platform. The architects

of the product family sometimes opt for a third-party rather than in-house-designed microprocessor. In that case, the availability of a license for the chosen microprocessor dictates the choice of semiconductor vendor.

PRODUCTS FOR PLATFORM-BASED DESIGN

Both semiconductor companies and fabless vendors supply technology- and architecture-driven platforms. Suppliers of technology-driven platforms include Altera, Chip Express, Fujitsu, Lightspeed Semiconductor, LSI Logic, NEC, Telairity, Virage Logic, and Xilinx. Due to the similarities between these products and structured ASICs, a future article will cover the vendors of these offerings. ARC, ARM, and Palmchip supply architecture-driven platforms. Altera and Xilinx are also active in this market segment.

ARC International offers the USB Now platform, which includes OTG (On-the-Go) functions for high-speed USB 2.0 peer-to-peer communications between mobile and consumer-electronics devices. The platform comprises the ARC-tangent-A5 synthesizable CPU core, USB 2.0 host/device core with OTG technology, and embedded software stack. David Fritz, vice president of technical marketing at ARC, points out that systems that designers base on multisourced, discrete IP often require more power than platforms designers build as systems.

The generic PrimeXsys platform from ARM defines the CPU, standard peripheral blocks, and the interconnection fabric that enables customers to integrate additional functions to fully characterize the desired final product. Customers have the choice of a number of platforms, depending on the core CPU they prefer, including a dual-CPU architecture.

Palmchip markets the AcurX SOC platform, which allows designers to choose among a number of CPU cores, including those from ARM, MIPS, and ARC. The platform targets low-power, high-performance, and data-intensive applications, such as broadband, networking, wireless, audio, video, network storage, and Internet security.

Altera offers a number of platforms that it built around two processors. Excalibur devices integrate an ARM922T processor with debugging modules, on-chip memory, and peripherals with an Apex 20KE-device-like architecture. For customers requiring more flexibility, the

Nios processor offers greater platform configurability. Designers can integrate memory, processors, peripherals, and other IP into one device. The Nios RISC processor features a 16-bit instruction set, a user-selectable 16- or 32-bit datapath, and a library of standard soft-core peripherals that engineers can configure for a large number of applications. Although the products target the FPGA market, customers can produce ASIC-like devices through Altera's HardCopy program.

Xilinx's platform products are the result of collaboration with both IBM (www.ibm.com) and Wind River Systems (www.windriver.com). You can choose from a number of platforms offering one to four PowerPC cores, high-speed serial I/O, and Wind River Systems' embedded design tools. Xilinx also offers programmable DSP platforms that include a good selection of functional blocks addressing most communication-application needs.

EDA vendors have not developed many tools specifically to support platform-based design. In most cases, engineers use the same tools as for regular design, but using generic tools may sometimes be less efficient than using specifically designed tools. With the purchase of LisaTek, CoWare acquired a presence in the platform-based-design market with an emphasis on application-driven platform development. The Edge processor designer allows designers to address processor-software and -hardware design in the Lisa 2.0 description language. From this description, engineers can develop a model of the hardware in Verilog or VHDL and a model of the software in C or C++. You can use the Rim software designer with Edge. It provides an assembler, a linker, an instruction-set simulator and a graphical debugging environment to support code development for the processor you design. To verify the entire system design, CoWare provides Hub, which allows hardware and software models you develop with the mentioned products to interface with simulation environments from either Synopsys or Mentor Graphics.

Mentor Graphics has developed the Platform Express design-capture and -verification tool, which allows users to progress through the design to determine the suitability of platforms for designs.

Engineers can create the system architecture as block diagrams. Designers begin by browsing the library of available processors, memories, and peripherals from leading semiconductor manufacturers and IP providers. They simply drag and drop the components in the graphical editor and then connect them using standard buses, such as AMBA (Advanced Microcontroller Bus Architecture) and VCI (Virtual Component Interface). The tool instantly creates all the necessary connections between components. The Platform Express GUI includes a memory-map display that shows usage of address space.

Synplicity has entered into a joint development, marketing, and distribution agreement with LSI Logic to provide an optimized physical-synthesis tool expressly tailored to LSI's RapidChip product family, which addresses the technology-driven-platform market. Synplicity is developing a custom physical-synthesis and mapping tool that enables designers to achieve placement-based timing closure of designs that use the RapidChip library.

Vast Systems markets a tool that addresses system-level design; you can use it with application-based platforms. Comet helps system architects quantitatively experiment early in the engineering process with partitioning of hardware and software, the choice of processor and its configuration, and operating-system options.

Tensilica's Xtensa processor is well-suited to application-driven platform-based design. System designers can modify the processor to fit the application by selecting and configuring predefined elements of the architecture and by designing new instructions and hardware-execution units within the processor. The Xtensa Processor Generator automatically generates a software environment, including operating-system support, for each processor configuration. □

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



REFERENCES

1. Sangiovanni-Vincentelli, Alberto, "The context for platform-based design," *IEEE Design & Test of Computers*, November-December 2002, pg 120.
2. Israelsohn, Joshua, "Pour your own programmable analog," *EDN*, June 12, 2003, pg 38.