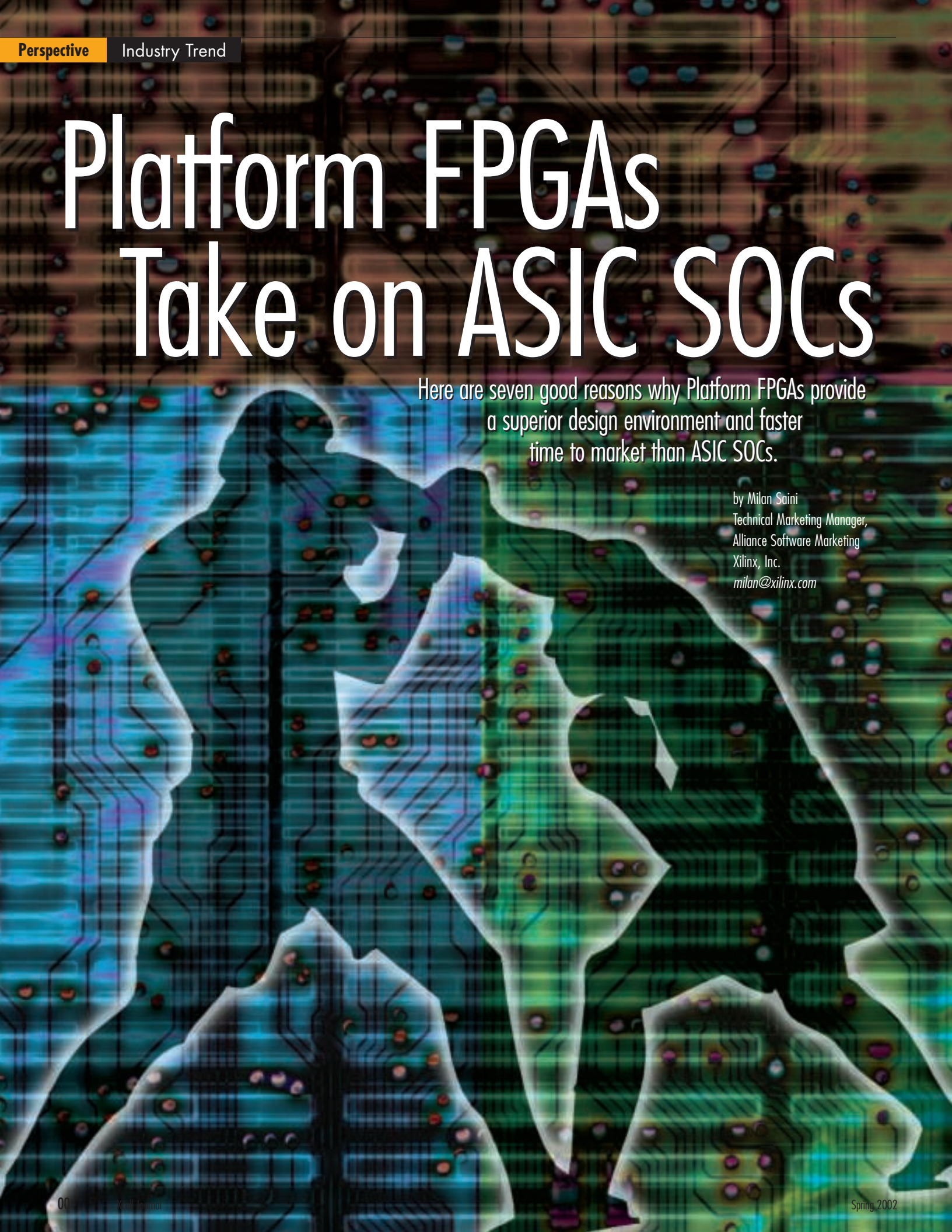
Platform FPGAs Take on ASIC SOCs

Here are seven good reasons why Platform FPGAs provide a superior design environment and faster time to market than ASIC SOCs.

by Milan Saini
Technical Marketing Manager,
Alliance Software Marketing
Xilinx, Inc.
milan@xilinx.com



The system-on-a-chip (SOC) market has experienced steady and consistent growth. Dataquest estimates roughly half of all ASIC design starts are SOC based. That percentage is projected to reach 80% by 2005. There are several reasons for this clear shift in design methodology. Some of the obvious advantages include greater component integration, increased speed (Logic <-> Processor), lower packaging and test costs, and increased overall system reliability. All of these combined can potentially make significant contributions to achieving the often elusive but always important goal of accelerated time to market.

Programmable logic device vendors have entered the SOC solution space with the introduction of a new class of devices known as Platform FPGAs – with the most recent addition being the Virtex-II PRO™. These devices offer the same level of integration as ASIC SOCs, but in contrast, Platform FPGAs facilitate the development of a wide range of applications on the same chip. This article focuses on seven of the key advantages of the Platform FPGA approach.

#1 – Pre-Engineered Platform

Platform FPGAs integrate several fixed and predetermined blocks of hard IP (intellectual property) components (system elements) within the programmable fabric. Notable among these are high-performance RISC CPUs, multi-gigabit and high speed I/Os, block RAM, system clock management, and dedicated DSP processing hardware. This powerful assembly and harmonious blend of components creates a cohesive system design environment. This environment offers unprecedented flexibility and performance, thus enabling the deployment of a wide range of applications.

The critical technological breakthrough is in the ability to tightly interface the various elements into the programmable fabric. Without this tight integration, much of the speed benefits could not be realized. The fact that the choice of system elements is already made greatly simplifies the design and development process. A fixed architecture is particularly beneficial for software tool and IP providers in allowing them to deliver better value, customization, and architecture-optimized solutions.

The assembly of the various hybrid IP blocks in an ASIC adds substantial complexity and hardship to the users' design and development environment, because of a variety of issues relating to tool and IP interoperability, physical layout, timing, and system verification. For Platform FPGAs, on the other hand, it is much easier to tailor and optimize components – such as silicon, software, support, and IP – because FPGAs represent a fixed and pre-engineered target.

Summary: A fixed, pre-engineered but programmable FPGA solution offers a more productive and efficient development environment from both the software and silicon perspective.

#2 – Process Technology

PLD vendors have been able to extract great value and benefits from Moore's Law and shrinking device geometries. While the majority of ASIC design starts are at or higher than 180 nm, FPGAs have raced ahead to bring the cost and performance advantages of 130 nm to its customers. This ability to rapidly migrate to the leading edge of technology is essential to delivering the performance, capacity, and integration that is necessary to challenge and displace the current established platforms of system design.

For instance, FPGAs now make it entirely feasible to build systems of up to 2M (ASIC) gates with CPU(s) running at 400 MHz, serial I/O channels at 3.125 Gbps CLB fabric-switching at 300 MHz, and the entire system clocking over 150 MHz. This surpasses the projected sweet spot of ASIC SOC designs.

Looking at it in pure economic terms, the costs for a typical mask set for a 130 nm ASIC run into the $700K+ range. That raises the bar for entry into the ASIC space, making the Platform FPGA an even more attractive option for a growing percentage of all SOC designs.

Summary: FPGA silicon is best-in-class in process and engineering, thereby delivering best-in-class system performance.

#3 – Software Tools and Methodology

It is well known that EDA designer productivity for ASICs is lagging behind recent silicon advances. The ability to create a productive design and debug environment is absolutely essential to the success of any silicon platform. Therefore, software plays a critical role in not only making the platform easy to use and work with, but also in extracting the most performance and device utilization out of the silicon.

The goal is to insulate the user from having to learn extraneous details about the platform, yet providing empowerment and control when and where it is needed. To this end, FPGA vendors have created customized and user-friendly processor system generator tools that aid in instantiation, initialization, and configuration of all the various system component blocks. In addition, these software tools automate otherwise manual and error-prone tasks, such as the interconnections among the processor, its peripherals, and buses.

Design entry is engineered to tightly couple the HW/SW domains. Such engineer-

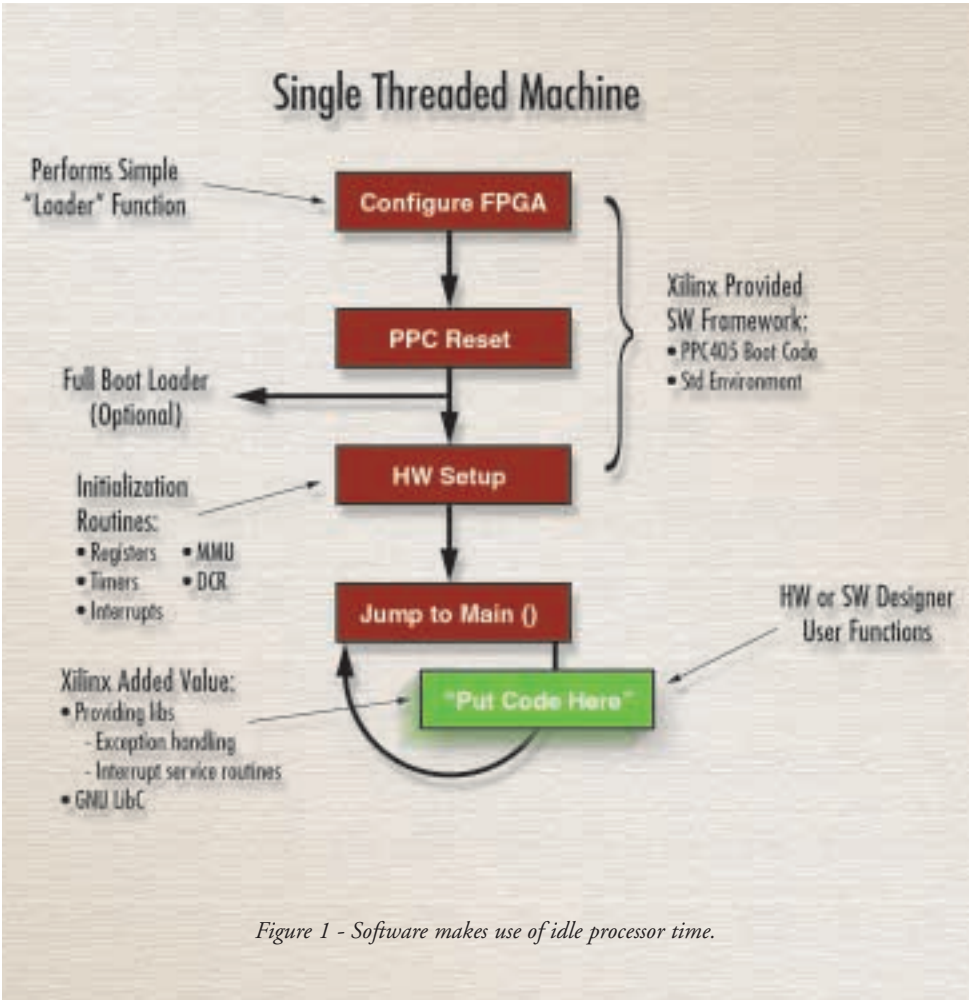


Figure 1 - Software makes use of idle processor time.

ing leads to not only simplified and easy-to-use design flows, but it ensures reliability and robustness from the very start by performing design rule and data consistency checks across the two domains.

Two examples of the advantages of cross-domain HW/SW co-design include:

- Automatic generation of device drivers and header files for SW engineers once a particular block is instantiated by the HW designer
- Tools to automatically populate SW binary code into appropriate FPGA memory bitstreams.

Summary: Software sells silicon.

#4 – Advanced Debug

The importance of finding problems early cannot be overstated. Yet, up-front ASIC verification is extremely designer- and computer-resource intensive. Compared to systems on a board, SOC limits visibility into the internal nodes of the system, making the task of verification and debug more challenging than ever.

The critical part includes the verification of complex interactions between the application software and the custom-designed peripheral hardware. Traditional HDL-centric verification and debug techniques can no longer deal with the rising complexity of system designs.

Consider a typical application, such as MPEG A/V decoding, where a large number of simulation cycles are required to complete a small sequence of frames. Co-verification tools in this case would either take an impractically long time to complete or validate only a mere fraction of the software code, falling far short of what it takes to find problems in the HW/SW interface.

FPGAs overcome this problem in large part by being able to provide access to real or near real targets at a very early stage in the design cycle. Among other things, this means that SW engineers using FPGAs can quickly and easily sort out logic and design flaws by targeting real silicon. The engineers do not have to rely on inefficient ASIC-centric techniques like co-verification or writing stub code. Application software can be debugged at system speeds with full hardware and software register access and control.

Additionally, the FPGA fabric allows for construction of highly customized and value-added cores to enable powerful real-time, on-chip debug capability. Some examples of such instrumentation include:

- Logic and bus analyzer functions
- Bus protocol compliance monitors
- Memory buffers for debug and trace port data
- Cross-domain triggers and breakpoints
- Hardware run control of the CPU
- HW/SW time synchronization logic.

Furthermore, a single cable is able to perform multiple functions, like debugging hardware, debugging software, as well as programming the FPGAs. This greatly simplifies the lab setup making it much easier to exploit the debug advantages.

Summary: Platform FPGAs offer a clearer, more cohesive, and overall more effective debug strategy. Specifically, Platform FPGAs offer up-front silicon access along with unprecedented visibility and control of the processor and its peripherals residing in the programmable fabric.

#5 – Top Tier Partnerships and Vendor Tools Support

In extending the concept of traditional programmable logic to Platform FPGAs, certain critical technologies have had to be developed or acquired. One of most exciting aspects brought forth by FPGA vendors has been to successfully forge

strategic partnerships with vendors holding various system technology components. Driven largely by the successes of the FPGA business models, leading vendors have been eager to partner in fulfilling the vision of building powerful programmable systems platforms.

Areas of cooperation include partnerships in the form of cutting-edge process technology, powerful mainstream processing elements – such as RISC CPU, high-speed I/O – and other components deemed of significance to a system design platform. IBM, Conexant Systems, Inc., Wind River Systems, and other high profile vendors are currently engaged in such strategic partnerships. What this means to you is there is no need for these partners to negotiate licensing, royalty, and integration issues with individual vendors, thereby greatly reducing your engineering, management, and accounting overhead.

The appeal and draw of FPGAs has caught the attention of independent SW tool vendors. Increasing numbers of vendors are able to sustain business models selling to FPGA customers. Several new vendors are setting up shop to write custom tools to help enhance and exploit the unique capabilities of Platform FPGAs.

Summary: The FPGA business model has attracted top-tier silicon and IP vendors to forge strategic partnerships to create powerful system design platforms. Software vendors are able to financially justify investment in research and development leading to a continuous stream of an increasing number of innovative solutions.

#6 – Application Space: Co-Design Flexibility

Programmable HW combined with processors on a single chip softens the HW/SW design boundary. By using hardware-assisted architectural exploration, designers can optimally search for the right HW and SW partitioning, which leads to the increased probability of being able to meet performance and area targets. Sequential computing, exception handling, and control functions, for instance, programmed in HW could be implemented in SW running on the processor to save silicon.

On the other hand, SW structures and algorithms – which can be broken into parallel, non-blocking processes – can achieve significant speed and data throughput improvements by implementing them in HW. In fact, software engineers represent a new and emerging market for Platform FPGAs. Aided by design tools, it is now easier than ever for SW engineers to explore concepts of software acceleration via HW. Both hardware and firmware are reprogrammable and field upgradeable, which enables the development and deployment of several product generations from one base. This translates into a much broader applicability than ASICs and ASSPs.

In addition to being suitable for building complex embedded applications, HW engineers can utilize an otherwise idle processor to run relevant portions of their design algorithms or control logic. As Figure 1 shows, the CPU can serve as a simple microcontroller running a single-threaded application. PLD vendors add value by providing software device drivers and library functions for rapid implementation without requiring the HW engineer to have detailed knowledge of SW practices.

Summary: The Platform FPGAs provide the most flexible co-design platform by enabling dynamic HW-SW design partitioning.

#7 – Risk Management

When compared to Platform FPGAs, ASIC SOCs present a huge design risk. With more variables and issues to worry about, and with limited debug capabilities when mistakes are found, ASIC designers are forced to either resolve problems suboptimally in software, or in the worst case scenario, they are forced to respin the silicon at great cost and loss of time to market.

Reprogrammability comes up big here. Programmable platforms allow early access to silicon. Engineers can validate performance and functionality in real silicon, leading to greater confidence in the reliability of the completed system. The programmability of the platform allows itself to be debugged and upgraded even after the system has been deployed. This helps promote and protect the time-to-market advantage by alleviating a large part of the risk.

Summary: Programmable Platform FPGAs provide better control over the life cycle management of products by minimizing the cost and time penalty for design errors and specification changes.

Conclusion

In an era when SOCs continue to dominate mainstream design, Platform FPGAs are emerging to take a share of the spotlight from ASICs. While ASIC SOCs have and will continue to be strong in certain segments – like low power, small form factor, and high-volume, cost-sensitive consumer electronic gadgets – an increasing range of other infrastructure applications in areas such as networking, telecommunications, industrial electronics, and data storage have compelling reasons to move to a programmable platform.