

# Faster and More Flexible Embedded Systems

Programmable Platform FPGA devices and intelligent tools combine to create higher performance processing solutions.

by Jay Gould  
Product Marketing Manager,  
Xilinx Embedded Solutions Marketing  
Xilinx, Inc.  
[jay.gould@xilinx.com](mailto:jay.gould@xilinx.com)

Copyright © 2005 Xilinx, Inc. All rights reserved.

Are there any real-time computing requirements that don't mandate that your new systems be faster and more flexible than your previous designs? Ever-changing industry standards dictate that tomorrow's embedded system designs accommodate a new level of customization, while high performance demands challenge traditional processing designs.

You don't want to be restricted by an overly customized design, and you can't just keep toggling the system clock at ever-increasing speeds to improve performance. There has to be a better way to create faster and more flexible embedded processing systems.

Platform FPGAs are programmable SOCs that support a multitude of sophisticated designs, and include on-board memory, DSP capability, embedded processing, and hardware accelerated co-processing. The re-programmability and field upgradeability of these new devices mean that you can fix bugs, enhance features, optimize performance, and add emerging industry standard support throughout product life cycles and even after deployment in the field.

With these powerful capabilities immersed in a programmable SoC device, all you need are the appropriate tools to unleash and harness this embedded performance.

## Intelligent Tools

In recent industry surveys, design engineers made it clear that they often value intelligent tools more than the actual devices and operating systems they use to build their own end products. If this trend is accurate, choosing the appropriate tool suite before beginning your next embedded system design will be crucial to your product schedule and overall success.

Today's development environments need to provide "platform-aware" tools that understand all system options and support multiple types of processor cores, as well as the creation and customization of co-processing and IP. Design wizards and automatic module



generation will reduce errors and streamline the development process, while integrating hardware and software debuggers together will enable you to find and fix bugs faster. If you choose wisely, intelligent tools will accelerate development and optimize performance.

### Standard Flows and Innovation

Xilinx® created the Xilinx Platform Studio (XPS) tool suite with the development of Platform FPGAs in mind, supporting existing traditional flows for both hardware (HDL/netlists for FPGAs) and embedded

hard- and MicroBlaze™ soft-processor core designs, the tools are smart enough to remove MicroBlaze options if you have chosen PowerPC, and vice versa.

Importing, creating, and customizing IP is streamlined through a separate design wizard, and supports an IP repository to facilitate IP reuse elsewhere on this design or in the future on a different design.

XPS additionally innovates and accelerates the development process with a variety of automatic generators that replace tedious and error-prone manual design steps. Being aware of the Platform FPGA

ware debug capabilities through a single probe. Other traditional methods require multiple probes and switching hardware connections between the different steps.

In fact, XPS uniquely integrates the hardware and software debuggers together so that they can cross-trigger each other. This new visibility into the system allows embedded design teams to find and fix bugs faster, regardless of whether the flaws originate in hardware or software.

### Acceleration Through Co-Processing

Let's say that you now have a flexible processor-based platform that satisfies most of your system requirements. How fast can you clock the core to meet your performance requirements?

You probably have realized that clocking your processor faster won't take care of all of your performance challenges. Besides the physical limitations of discrete processors and heat dissipation, accelerated clocking can't ensure that your core can service and complete all the real-time event responses and applications with which you have burdened it. More and more "multi-processor" solutions are emerging to partition and offload lower priority tasks from a main control processor so that the main unit can ensure real-time responses.

Programmable platforms introduce some additional ways to approach this problem, with off-the-shelf devices that you can customize yourself for your own unique applications. Supporting both hard and soft processor cores, one solution offered by Platform FPGAs is to focus high-priority tasks on an immersed hard processor while offloading lower priority tasks to a soft-processor core instantiation. You have the option to add one or more MicroBlaze soft processors to a Platform FPGA device already running an embedded PowerPC engine. Example devices supporting this are the Virtex™-II Pro FPGA or new Virtex-4 FX family devices with built-in PowerPCs. The PowerPC cores in these devices can be complemented with MicroBlaze IP cores inserted as macros and built out of FPGA hardware resources in the silicon.

Another alternate and promising approach is to implement the concept of

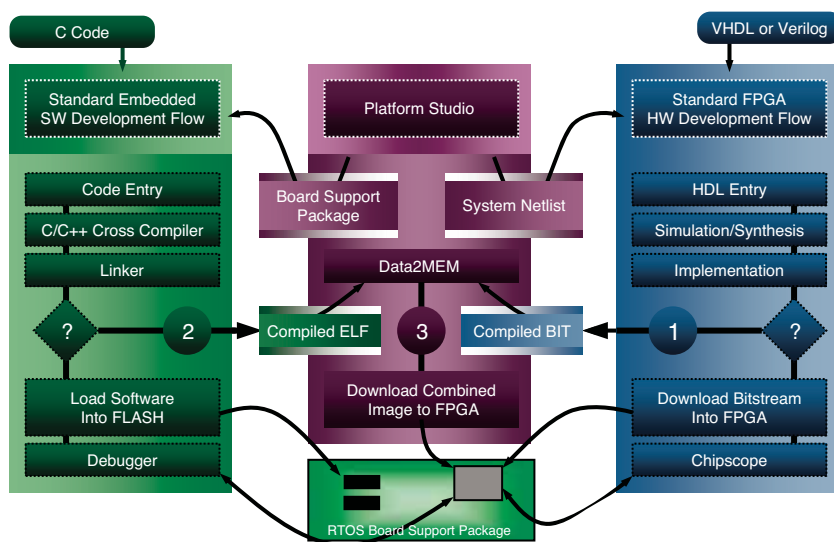


Figure 1 – XPS design flow

software design (C/ELF code for processing core engines) (Figure 1). In addition to providing a unified development tool suite for supporting the complete spectrum of programmable processing solutions, XPS won the IEC's (International Engineering Consortium) DesignVision Innovation Award for introducing new capabilities that accelerate embedded development. With platform-aware tools like XPS, you can quickly create a real-time hardware/software system through an abstract flow of design wizards.

The design wizards guide you through the process of creating a basic system and can reduce errors by masking-off design options not supported by your initial selections and assumptions. For example, although XPS supports both PowerPC™

silicon properties and options, XPS can automatically generate software drivers for selected peripherals, generate sample test code for board options, and even create BSPs (board support packages) for some of the more widely used RTOS/eOS (real-time operating systems/embedded operating systems) such as Wind River Systems's VxWorks or embedded Linux.

XPS also provides a unique utility (Data2Mem) that merges C code into the FPGA bitstream, enabling software development and debug to proceed in real time without time-consuming re-runs of FPGA place and route tools.

Xilinx even provides new efficiencies with a unified JTAG connection methodology that combines FPGA download, FPGA debug, C code download, and soft-

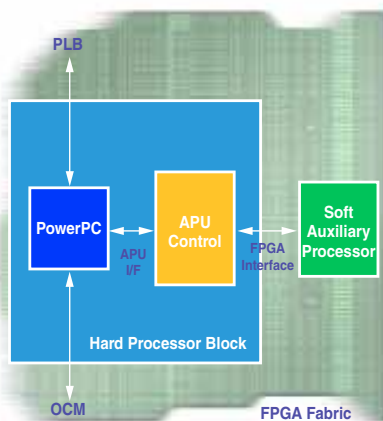


Figure 2 – Virtex-4 FX APU

“co-processing” and use the intelligent tools to build a direct connect from the embedded PowerPC cores to high-performance FPGA fabric, where hardware accelerator functions can operate as extensions to the PowerPC. As shown in Figure 2, you can improve the overall system performance by offloading computationally demanding applications from the main CPU.

By its very nature, FPGA hardware fabric is parallel in structure and can be used to accelerate system functions orders of magnitude faster than clocking methods can provide. In this example, the PowerPC core is complemented by an APU (Auxiliary Processor Unit), which inter-

faces to a parallel soft processor that can handle applications such as data processing, floating-point mathematics, and video processing. This direct connection provides a high-bandwidth, low-latency solution with parallel advantages over other multi-core processor and arbitrated busing solutions.

### Performance Analysis

Do you need to find out where your performance is lost in your design? Embedded software debugging and analysis is always a bit of a challenge because code execution is often “invisible” to you. On paper, your design looks like it meets specifications, but when running in real-time hardware with asynchronous interrupts and real-world situations, you find that often you don’t meet your own performance requirements. Now is the time when intelligent tools can provide you with a unique view inside the operating device rather than leave you guessing outside of a black box.

Version 7.1 of Xilinx Platform Studio introduces a series of performance analysis tools and views that provide great insight as to how your software is actually executing and where performance is leaking away from you (Figure 3). By knowing which software functions take up the most execution time and which functions call other functions – as well as the number of times called – you can

get an illuminating view of exactly how your embedded design is running. Functions that take a long time to execute, or functions that are called a large number of times by other routines, may be excellent candidates to accelerate by moving them to parallel hardware as co-processing extensions.

Figure 3 also shows that if the tools track and display your software execution clearly, you can quickly and easily identify areas that could be more efficient. This can save a lot of what-if experiment scenarios that are time-consuming and often result in relatively small performance improvements. In-lining some C code or an entire function may provide tiny localized speed-ups, but moving time-consuming routines into high-performance FPGA hardware can often result in an order-of-magnitude improvement. With intelligent views of the code execution by specific function names, you can see exactly which software routines to adjust, providing a much higher return on improving system performance.

### Conclusion

Intelligent platform-aware tools can help you identify the inefficiencies in your embedded software code and allow you to optimize performance. Knowing which specific software functions you need to streamline allows you to evolve your hardware/software partitioning and accelerate more modules in programmable FPGA fabric.

The high-performance nature of parallel FPGA hardware resources and the advent of easy-to-use, programmable co-processing technologies like the Virtex-4 FX APU enable you to create faster and more flexible embedded processing systems.

Xilinx offers clear advantages for embedded processing over traditional discrete or competitive FPGA solutions. Our tools, combined with our programmable embedded Platform FPGAs, offer a significant performance improvement for real-time developers.

To learn more about the Platform Studio tool suite, please visit [www.xilinx.com/edk](http://www.xilinx.com/edk). A good starting point to learn about all of our embedded processing solutions is [www.xilinx.com/processor](http://www.xilinx.com/processor).

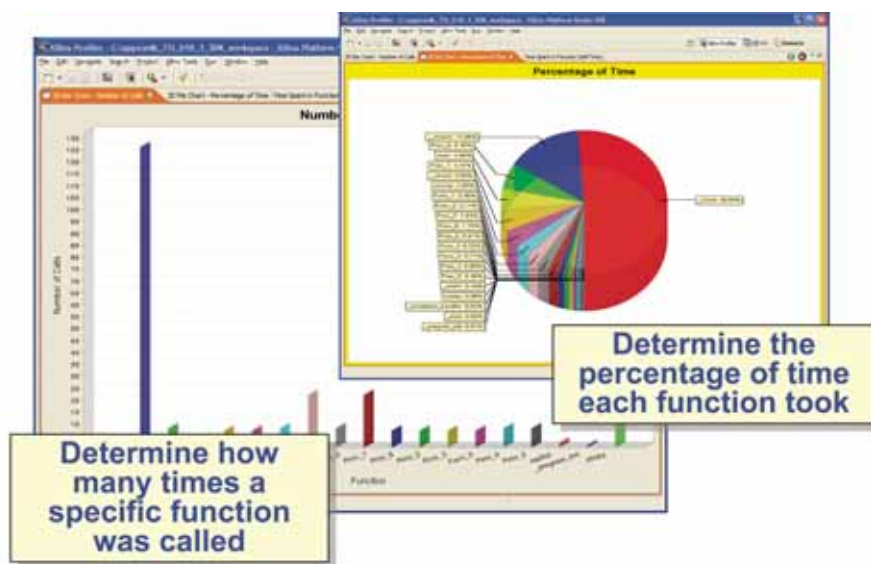


Figure 3 – XPS performance analysis views