

# Increase Performance and Lower Cost Through System Design

Poseidon's breakthrough ESL tools shorten the system-level design cycle by using Xilinx embedded processors.

By Farzad Zarrinfar  
VP of Worldwide Sales & Marketing  
Poseidon Systems  
[Farzad.Zarrinfar@poseidon-systems.com](mailto:Farzad.Zarrinfar@poseidon-systems.com)

Bill Salefski  
VP of Engineering  
Poseidon Systems  
[bills@poseidon-systems.com](mailto:bills@poseidon-systems.com)

Stephen Simon  
Director of Sales & Business Development  
Poseidon Systems  
[ssimon@poseidon-systems.com](mailto:ssimon@poseidon-systems.com)

The introduction of embedded processors for FPGAs (like the immersed PowerPC™ and Xilinx® MicroBlaze™ soft-processor core) has greatly expanded the benefits of platform FPGAs. These advantages include lower FPGA power consumption, lower device cost, and more scalable performance, particularly when balancing your application's execution between what will run in the fabric of the FPGA versus the embedded processor(s).

Designing efficient processor-based system architectures and optimizing overall system performance for a specific application is challenging. For example, effectively partitioning the hardware and software early in the design is difficult, yet critical to

the development of the architecture. Also, architectures must be verified early in the design cycle to maximize the benefits of processor-based designs. You cannot wait until RTL development to discover that your architecture does not support the system requirements. This "redesign loop" can easily delay the overall design cycle.

Newer and better alternatives exist to partition, analyze design bottlenecks in, implement, and verify system architectures for Xilinx FPGAs with embedded hard or soft processors. Electronic system-level (ESL) tools and predefined cores can help you make critical architectural decisions, simplify your design tasks, and reduce your overall design cycle. In this article, we'll show how Poseidon's new ESL tools can help you realize these benefits.

## Triton Tool Suite

Poseidon's Triton Tool suite is a system design and acceleration environment that enables you to quickly develop, analyze, and optimize system architectures. With these ESL tools, the abstraction level of the design is above RTL – you can quickly address system issues without having to solve the detailed issues surrounding RTL implementation. Thus, you can quickly

perform architectural "what-if" analysis and accelerate time to market. The tool suite was created specifically for processor-based systems that require efficient robust architectures with the need to optimize performance, power, and cost.

Triton comprises two main tools:

- Triton Tuner – a system and software analysis tool
- Triton Builder – a hardware/software partitioning tool

Triton Tuner is a simulation and analysis environment based on SystemC. Simulation is performed at the transaction level from models of both the processor and surrounding buses and peripherals. Using Tuner, you co-simulate the hardware with the application software.

During simulation, the tool collects both hardware and software performance data. The environment then provides tools to visualize and analyze the data, reducing the effort required to identify inefficiencies in the design. Figure 1 shows a bus activity graph – a visualization tool to help identify bus congestion and bottlenecks in the system design.

Triton Tuner also links hardware and software events, providing a key tool in

determining causal relationships between hardware and software systems. This feature greatly simplifies the task of evaluating and optimizing system performance. Tuner also provides tools to optimize the memory structure. This can greatly reduce the need for costly caches and other high-speed memories.

Triton Builder is an automated partitioning tool, which simplifies the task of accelerating the performance of processor-based algorithms. With Triton Builder, you can easily offload compute-intensive loops and functions from software to hardware. The tool automatically creates a DMA (direct memory access)-based hardware accelerator to perform the offloaded task. All hardware and source code modifications are also created to greatly simplify the repartitioning process.

With the builder tool, you can control a number of key parameters in which to optimize the solution to your system requirements. Together, these tools provide a system design and optimization environment that unobtrusively plugs into a Xilinx EDK design flow, enabling you to quickly make dramatic improvements in the performance and power consumption of your application.

Triton Builder generates the RTL for the complete accelerator, the driver required to invoke the accelerator, and inserts the driver into the proper place in the original application. The RTL can be generated in either Verilog or VHDL. This integral generation process ensures that necessary hardware and software components are exactly matched for trouble-free design. The tool also generates an RTL test bench and SystemC model for verifying the new hardware.

### Design Architecture of FPGA-Based Accelerator

The builder tools create a design architecture ideally matched to FPGA design. Using Triton Builder, you can quickly create a peripheral using the C code that runs on the processor. The tool moves the computationally intensive portions of the code into a



Figure 1 – Bus activity graph

hardware accelerator. This acceleration hardware is connected directly onto the processor bus and implemented on the FPGA fabric. The block diagram of a typical accelerated architecture is shown in Figure 2.

If while executing the application code the processor hits a section of code that has been moved to hardware, control is passed to the accelerator through the inserted driver, which then performs the accelerated function. The accelerator runs independently from the processor, freeing it to perform other tasks. When the accelerated task is completed, the results are passed back to the application program. You can implement multiple independent accelerators within the same system.

To create a complete accelerator peripheral, you need more than just a C synthesis tool. Poseidon Builder includes not just a C-to-RTL synthesizer that creates the compute core, but it also generates the communication and control hardware. The Poseidon tool creates a complete accelerator peripheral, the block diagram of which is shown in Figure 3. In addition to the compute core, the accelerator includes a multi-channel DMA controller, bus interface, local memories, and other logic to create a plug-and-play peripheral.

### Interfacing to Xilinx Tools

Triton Tools are extremely flexible, allowing you to use either Triton Tuner or Triton Builder independently or together as an integrated suite. These tools were developed to enhance the productivity in developing processor-based designs and vastly increase their capability.

The Triton tools link seamlessly to EDK tools. A typical system design flow is usually an iterative process where you would analyze the system performance, determine inefficiencies, modify the system, and check the resultant performance. When the resultant architecture performs to the desired level, the system is then transferred back into the Xilinx

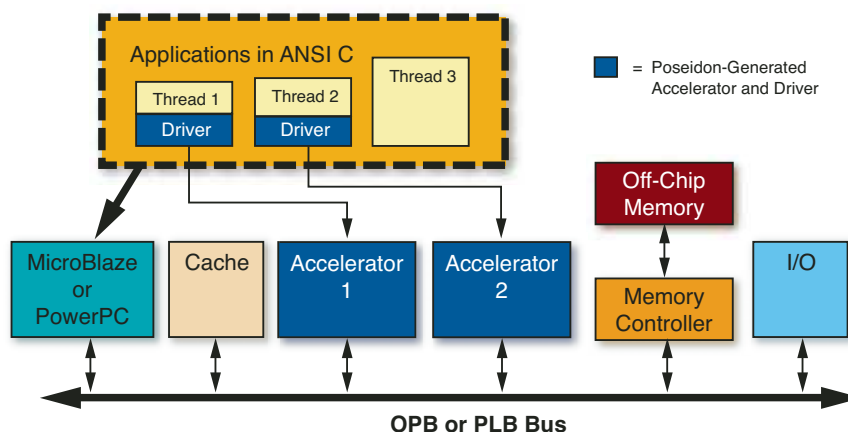


Figure 2 - Accelerated system architecture

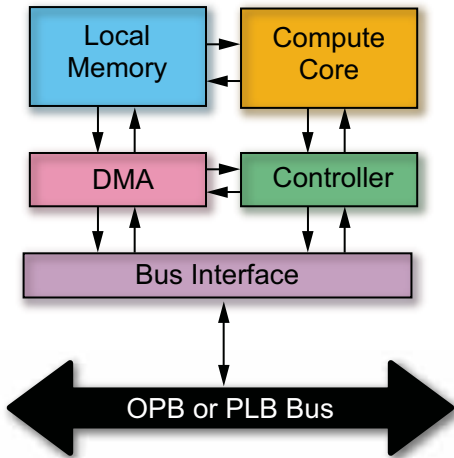


Figure 3 - Accelerator block diagram

tool chain. The Triton tools accelerate the process of identifying problem areas and establish an integrated flow that allows you to move between the tools to develop the optimal architecture.

A typical flow (shown in Figure 4) comprises these steps:

- The designer develops the target architecture using selected Xilinx tools
- The architecture description is read from the microprocessor hardware specification file (.mhs) from EDK, and the ANSI C application source code is read into the Triton tools
- Triton Tuner profiles the ANSI C code, reveals bottlenecks in the code or architecture, and eliminates inefficiencies
- Triton Builder partitions compute-intensive algorithms in ANSI C into hardware and generates a hardware accelerator
- Triton Tuner verifies that the new system performs to the desired level
- RTL, test bench, modified C code, driver, and architecture are exported back into the Xilinx environment

### Conclusion

Poseidon's Triton Tool suite enables you to rapidly and predictably analyze, optimize, and accelerate processor-based architectures. With Triton Tuner's SystemC

simulation environment, you can develop robust efficient processor architectures. With Triton Builder, you can add sophisticated hardware acceleration to your processor-based systems.

The Triton tool suite enables design architects to achieve higher system throughput, reduced power consumption,

and cost, as well as shorter design cycles. For more information, visit our website at [www.poseidon-systems.com](http://www.poseidon-systems.com) or contact the sales department at (925) 292-1670. To be qualified for a free tool evaluation of the Poseidon Builder and Tuner and a free white paper, e-mail [farzad.zarrinfar@poseidon-systems.com](mailto:farzad.zarrinfar@poseidon-systems.com).

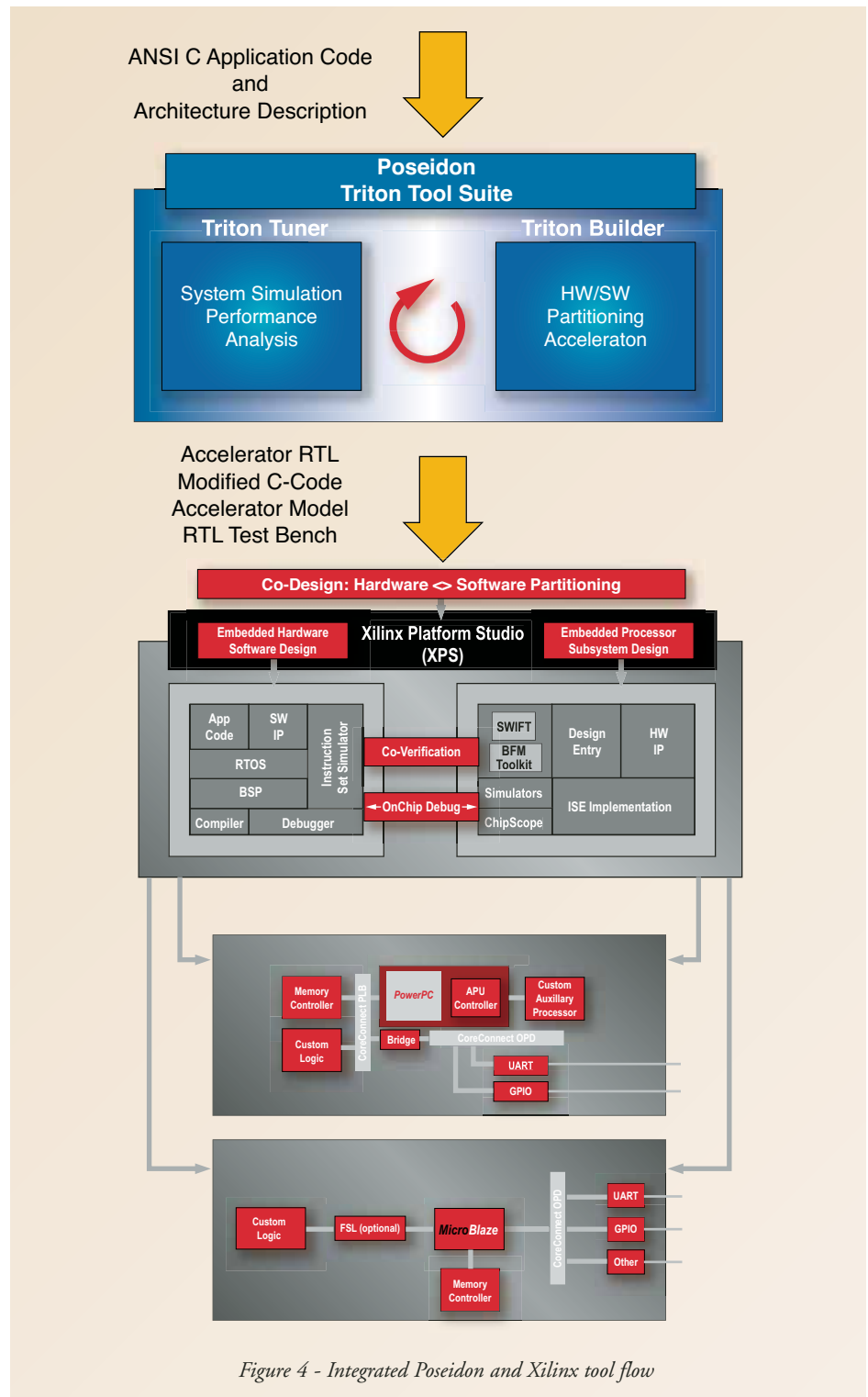


Figure 4 - Integrated Poseidon and Xilinx tool flow