

# ESL Tools Make FPGAs Nearly Invisible to Designers

You can increase your productivity through ESL methodologies.



by Steve Lass  
Sr. Director, Software Marketing  
Xilinx, Inc.  
[steve.lass@xilinx.com](mailto:steve.lass@xilinx.com)

FPGA business dynamics have been rather consistent over the past decade. Price per gate has continued to fall annually by an average of 25%, while device densities have continued to climb by an average of 56%. Concurrent with advances in silicon, design methodologies have also continued to evolve.

In particular, a new paradigm known as electronic system level (ESL) design is promising to usher in a new era in FPGA design. Although the term ESL is broad and its definition subject to different interpretations, here at Xilinx it refers to tools and methodologies that raise design abstraction to levels above the current mainstream register transfer level (RTL) language (Figure 1).

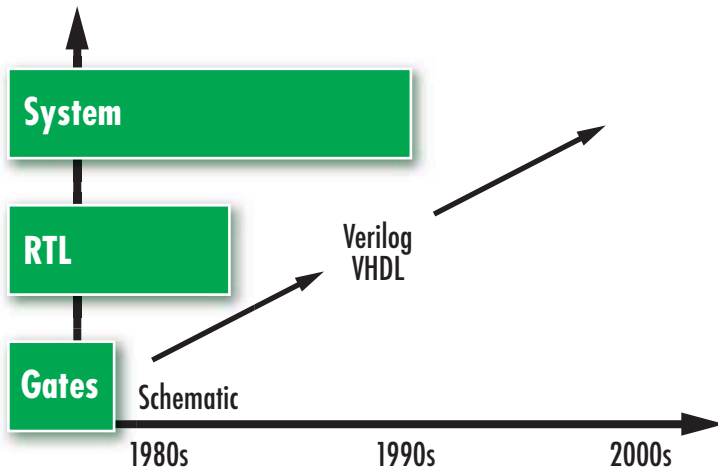


Figure 1 – Design methodologies are evolving to HLL-based system-level design.

Put another way, it is now possible for you to capture designs in high-level languages (HLLs) such as ANSI C and implement them in an optimized manner on FPGA hardware. These advances in software methodologies – when combined with the increasing affordability of FPGA silicon – are helping to make programmable logic the hardware platform of choice for a wide and rapidly expanding set of target applications.

**Defining the Problem**

You might wonder what problem ESL is trying to solve. Why is it needed? To answer this, let’s consider the following three scenarios. First, many of today’s design problems originate as a software algorithm in C. The traditional flow to hardware entails a manual conversion of the C source to equivalent HDL. ESL adds value by providing an automatic conversion from HLL to RTL or gates. For those wishing to extract the most performance, you can optionally hand-edit the intermediate RTL.

Second, for both power and performance reasons, it is clear that traditional processor architectures are no longer sufficient to handle the computational complexity of the current and future generation of end applications. ESL is a logical solution that helps overcome the challenges of processor bottlenecks by making an easy

and automated path to hardware-based acceleration possible.

Third, as the types of appliances that can now be deployed on FPGAs become more sophisticated, traditional simulation methods are often not fast enough. ESL methodologies enable faster system simulations, utilizing very high speed transaction-based models that allow you to verify functionality and perform hardware/software trade-off analysis much earlier in the design cycle.

**Value Proposition for FPGAs**

ESL technologies are a unique fit for programmable hardware. Together, ESL tools and FPGAs enable a desktop-based development environment that allows applications to target hardware using standard software-development flows. You can design, debug, and download applications developed using HLLs to an FPGA board much the same way that you can develop, debug, and download code to a CPU board.

Additionally, with powerful 32-bit embedded processors now a common feature in FPGAs, you can implement a complete system – hardware and software – on a single piece of silicon. Not only does this provide a high level of component integration, but with the help of ESL tools, it is now easy and convenient to dynamically partition design algorithms between the appropriate hardware (FPGA fabric) and software (embedded CPU) resources on the chip (Figure 2).

ESL brings a productivity advantage to both current FPGA hardware designers as well as to a potentially large number of software programmers. Hardware engineers are using HLL-based methods for rapid design prototyping and to better manage the complexity of their designs. Software developers, interested in accelerating their CPU bottlenecks, are using ESL

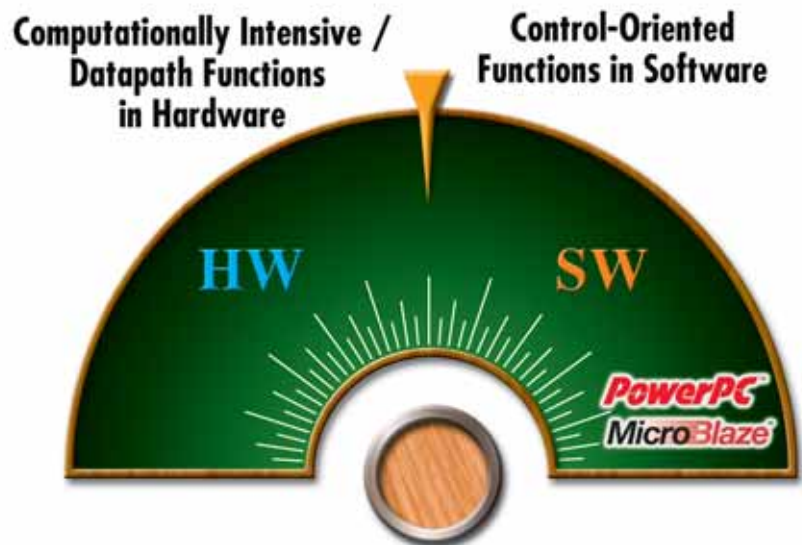


Figure 2 – ESL facilitates dynamic hardware/software partitioning conveniently possible in FPGAs.

## ESL adds value by being able to abstract away the low-level implementation details associated with hardware design. By doing so, the tools simplify hardware design to such an extent so as to render the hardware practically “invisible” to designers.

flows to export computationally critical functions to programmable hardware.

ESL adds value by being able to abstract away the low-level implementation details associated with hardware design. By doing so, the tools simplify hardware design to such an extent so as to render the hardware practically “invisible” to designers. This helps expand the reach of FPGAs from the traditional base


If you remain skeptical about the role of HLLs in hardware design, you are not alone. Your concerns might range from compiler quality of results and tool ease-of-use issues to the lack of language standardization and appropriate training on these emerging technologies. The good news is that ESL suppliers are aware of these challenges and are making substantial and sustained investments to advance

methods are being used with FPGAs to solve practical problems.

You may also be wondering if ESL will signal the end of current RTL-based design methodologies. Although there is little doubt that ESL should be regarded as a disruptive technology, ESL flows today are positioned to complement, rather than compete, with HDL methods. In fact, several ESL tools write out RTL descriptions that are synthesized to gate level using current RTL synthesis. And there will always be scenarios where design blocks may need to be hand-coded in RTL for maximum performance. By plugging into the existing well-developed infrastructure, ESL can help solve those design challenges not addressed by current methodologies.

### Next Steps

If you are looking for a place to begin your orientation on ESL and FPGAs, a good place to start would be the ESL knowledge center at [www.xilinx.com/esl](http://www.xilinx.com/esl). In March 2006, Xilinx launched the ESL Initiative – a collaborative alliance with the industry’s leading tool providers to promote the value, relevance, and benefits of ESL tools for FPGAs (Table 1). The website aims to empower you with knowledge and information about what is available and how to get started, including information on low-cost evaluation platforms that bundle FPGA boards with ESL software and design examples.

This issue of the *Xcell Journal* features a range of articles on ESL and FPGAs written by engineers at Xilinx as well as our ESL partners. These articles are designed to give you better insights into emerging ESL concepts. You may find that some of these technologies do indeed present a better approach to solving your design challenges. We hope that some of you will feel inspired to take the next step and become part of the growing number of early adopters of this unique and revolutionary new methodology. 

Partner	From HLL To FPGA	Xilinx CPU Support	Description (Key Value)
Bluespec	✓		SystemVerilog-Like to FPGA (High QoR)
Binachip	✓	✓	Processor Acceleration (Accelerates Binary Software Code in FPGA)
Celoxica	✓	✓	Handel-C, SystemC to FPGA (High QoR)
Cebatech	✓		ANSI C to FPGA (Time to Market)
Critical Blue	✓		Co-Processor Synthesis (Accelerates Binary Software Code in FPGA)
Codetronix	✓	✓	Multi-Threaded HLL for Hardware/Software Design (Time to Market)
Impulse	✓	✓	Impulse-C to FPGA (Low Cost)
Mimosys	✓	✓	PowerPC APU-Based Acceleration (Ease of Use)
Mitron	✓		Mitron-C to FPGA (Supercomputing Applications)
Mirabilis		✓	High-Speed Virtual Simulation (Perform Hardware/Software Trade-Off Analysis)
Nallatech	✓		Dime-C to FPGA (High-Performance Computing Applications)
Poseidon	✓	✓	MicroBlaze/PowerPC Acceleration (Analysis, Profiling, and C Synthesis)
SystemCrafter	✓		SystemC to FPGA (Rapid Prototyping)
Teja	✓	✓	Distributed Processing Using MicroBlaze/PowerPC (High-Speed Packet Processing)

Table 1 – Xilinx partners provide a wide spectrum of system-level design solutions.

of hardware designers to a new and potentially larger group of software designers, who until now were exclusively targeting their applications to processors. As there are many more software developers in the world than hardware designers, this equates to a large new opportunity for FPGAs.

the state of the art. These efforts are beginning to yield results, as evidenced by an increasing level of interest from end users. For instance, more than two-thirds of the engineers in a Xilinx® survey expressed a keen interest in learning about the capabilities of ESL. Xilinx estimates that hundreds of active design seats exist where ESL