

SystemCrafter

Bringing Hardware and Software Together

Xilinx's ESL Initiative & SystemCrafter

Xilinx's ESL Initiative - a multi-faceted program aimed at making electronic system level (ESL) design methodologies and tools more accessible to programmable system designers. The initiative expands collaboration across the ESL supply chain to better integrate and optimize ESL tool flows for both hardware designers and software programmers targeting Xilinx FPGAs.

SystemCrafter is one of the early participant of this initiative program and is now *Xilinx ESL Ecosystem Member*.

The ESL Initiative underscores the commitment by Xilinx and SystemCrafter to drive technological innovation and development of practical solutions that deliver on the full potential of this high-level design methodology. The initiative has identified four key areas of focus:

1. Improve ease of use to further simplify and abstract the details associated with FPGA design
2. Optimize support for Xilinx embedded PowerPC™ and MicroBlaze™ processor solutions
3. Improve the quality of results with high level language (HLL) synthesis tools
4. Establish common standards for FPGA ESL tool interoperability

About SystemCrafter Ltd. :

SystemCrafter brings customers the power of fourth-generation electronic design synthesis, and makes it easier, faster and less risky to create advanced IC designs. To fulfill this mission, they have created SystemCrafter SC. SystemCrafter SC is a software tool that automatically synthesizes designs written in the industry-standard SystemC language to electronic hardware.

SystemCrafter is based in Ipswich, England, about 70 miles northeast of London.

For more information contact:

SystemCrafter Ltd., The Old Hall, Parham, Woodbridge,
Suffolk, IP139ND, UK. Tel: +44 (0)1728 724897
URL : www.systemcrafter.com

Distributor for Asia Pacific:

ni logic Pvt. Ltd. (ni2designs), Pune, India.
URL: www.ni2designs.com Email: info@ni2designs.com
Tel: +91-20-25286947/48

Bringing Hardware and Software Together

For many years people have been using C or C++ as a starting point for developing their hardware and systems. This is because these languages are widely known, quick to write, and give an executable specification, which allows very fast simulation. C or C++ versions of standard algorithms are widely available, which allows easy re-use of legacy and publicly available code. For system-level design, they allow hardware and software descriptions to be described in a single framework.

However there have been two drawbacks. Firstly, C and C++ don't support the description of some important hardware concepts, such as timing and concurrency. Secondly, C and C++ have to be translated manually to a hardware description language, such as VHDL or Verilog, for hardware implementation. This step requires specialist resources, is time-consuming, and often introduces errors that are difficult to find.

The first of these problems was solved by the development of SystemC, which is now a widely accepted industry standard that adds hardware concepts to C++.

The second of these problems is solved by the development of tools like SystemCrafter SC, which allows SystemC descriptions to be automatically translated to VHDL/Verilog.

SystemC and ESL

Worldwide industries prepares Transaction Level Modeling (TLM) standard

Electronic System Level (ESL) design refers to evolving design and verification methodologies that begin at a higher level of abstraction than the current mainstream Register Transfer Level (RTL). SystemC has a strong lead as standard language for system-level modeling and ESL designs.

After SystemC became an international IEEE standard there has been a surge in its acceptance and usage. SystemC is now a major and mainstream feature in electronic system level (ESL) design that provides the foundation for a worldwide ecosystem that would otherwise not be possible without a standard language.

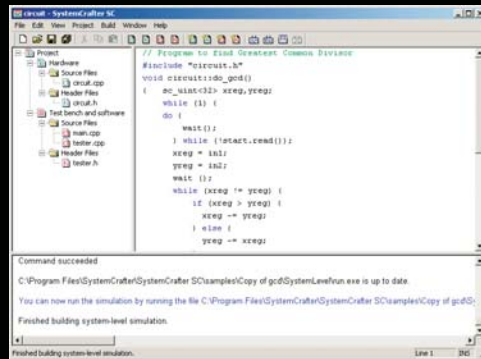
Reports from Open SystemC Initiative (OSCI) confirms worldwide adoption of SystemC is strong and continues to grow, and that SystemC user groups in all geographies are quickly adding in ESL design market.

SystemCrafter SC: Technology

SystemCrafter SC is software tool that synthesizes SystemC automatically to VHDL/Verilog description hardware. Engineers and programmers can design, debug and simulate hardware and systems using their existing C++ development environment. SystemCrafter SC facilitates to simulate hardware and software in same framework. Designers can synthesize SystemC to RTL for implementation using a standard VHDL/Verilog design flow. SystemCrafter SC also writes a structural SystemC description of the synthesized circuit for verification.

Target users of SystemCrafter SC:

SystemCrafter SC is suitable for programmers, scientists, systems engineers and hardware engineers. It enables developing hardware to be viewed as a higher level activity than writing an HDL, and allows the user to focus on the algorithm, rather than the details of the implementation.



"We are delighted that SystemCrafter SC has been recognized by the readers of EE Times, and this award is a great boost for everyone who is working hard to develop the tool further. Our vision is to provide a tool which can bridge the gap between SystemC and HDL, at an affordable price. This not only enables system-level designers to implement their designs quickly and reliably, but also opens up the use of FPGAs to a new market of C and C++ programmers."

Dr Jonathan Saul
SystemCrafter CEO

SystemCrafter SC: Design Flow

SystemCrafter SC design flow automates the process of synthesizing SystemC to RTL VHDL/Verilog. It will also generate a SystemC description of the synthesized circuit, which can be used to verify the synthesized code using your existing test harness.

SystemCrafter SC gives the designer control of the critical steps of scheduling (clock cycle allocation) and allocation (hardware reuse). Thus, the results are always predictable, controllable and match the designer's expectations.

SystemCrafter SC allows you to develop, refine, debug and synthesize hardware and systems within your existing C++ compiler's development environment. You can run fast, executable SystemC specifications to verify your design. You can configure your compiler so that SystemCrafter SC is automatically run when you specify that you want to generate hardware. There is no new GUI to learn.

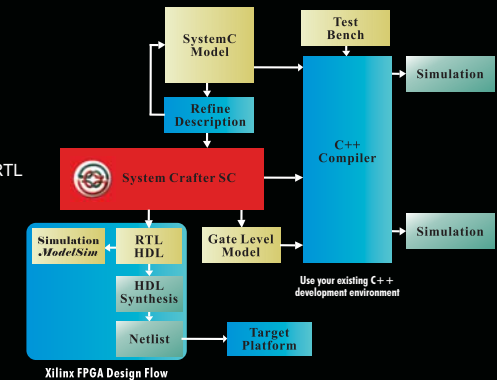
Synthesizing SystemC To Hardware

A typical development process using SystemCrafter is:

- ◆ Develop an initial SystemC description.
- ◆ Write a test bench.
- ◆ Debug, simulate and verify your description.
- ◆ Refine it to describe more efficient hardware.
- ◆ Experiment with trade-offs.
- ◆ Verify the refined description using your test bench.
- ◆ SystemCrafter SC will synthesize your SystemC description to RTL VHDL/Verilog, and a SystemC description of this VHDL/Verilog.
- ◆ Verify the synthesized SystemC model.

Using synthesis tool and Xilinx FPGA flow:

- ◆ Simulate & verify generated RTL HDL using ModelSim.
- ◆ Synthesize the generated VHDL/Verilog to working hardware.
- ◆ Program the netlist to target FPGA.



System-level Design and Co-design

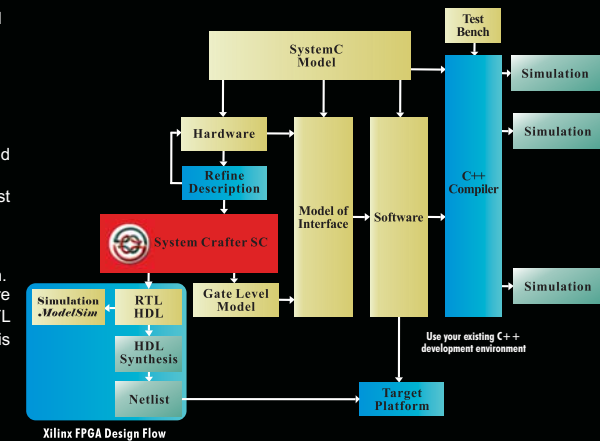
The additional advantage of SystemCrafter SC for co-design is that you can simulate the hardware and software partition in the same framework.

A typical development process is:

- ◆ Develop an initial SystemC description.
- ◆ Write a test bench.
- ◆ Debug, simulate and verify your description.
- ◆ Partition the design into hardware, software and interfaces.
- ◆ Verify the partitioned description using your test bench.
- ◆ Refine it to describe more efficient hardware.
- ◆ Experiment with trade-offs.
- ◆ Verify the refined description using your test bench.
- ◆ SystemCrafter SC will synthesize the hardware parts of your SystemC description to RTL VHDL/Verilog, and a SystemC description of this synthesized circuit.
- ◆ Verify the synthesized SystemC model.

Using synthesis tool and Xilinx FPGA flow:

- ◆ Simulate & verify generated RTL HDL using ModelSim.
- ◆ Synthesize the generated VHDL/Verilog to working hardware.
- ◆ Program the netlist to target FPGA.



FPGA Co-processing & Hardware Acceleration

Using SystemC and SystemCrafter, you can easily develop hardware coprocessing and software acceleration units. Using FPGA hardware with SystemC flows enables powerful new computing applications and domains.

You can use your existing C++ compiler to develop a SystemC description of your application, and experiment with implementing parts of the code in different coprocessors and hardware accelerators. You can use your compiler's facilities to profile candidate architectures, and simulate the whole system. Then SystemCrafter will automatically produce RTL VHDL/Verilog descriptions of the hardware, which you can implement in your Xilinx FPGA.

