



SLEC[®] System
Enabling ESL[®]



SLEC[®] System-HLS

SLEC System is the industry's only proven Sequential Logic Equivalence Checker capable of verifying designs with differences in timing, interfaces and levels of abstraction. Enabling ESL with a verifiable path from system-level models to RTL implementations, SLEC System is the cornerstone of advanced hardware design flows.

System Level Design Flows

SLEC System proves that a refined system-level model or RTL implementation is functionally equivalent to the original system-level model. By verifying that both designs produce the same output for all possible inputs, over all time, the quality of verification that SLEC System performs in minutes is equal to years of running simulation.

Because SLEC System does not require testbenches or assertions, engineers spend significantly less time developing their verification environment and more time creating quality design.

SLEC System verifies equivalence between two system-level models or a system-level model and its corresponding RTL implementation. SLEC System supports timed and untimed system-level models written in C, C++ and SystemC.

SLEC System-HLS extends the benefits of SLEC System to ESL design flows by comprehensively verifying the RTL generated from High Level Synthesis (HLS). SLEC System-HLS is integrated with the leading HLS synthesis tools.

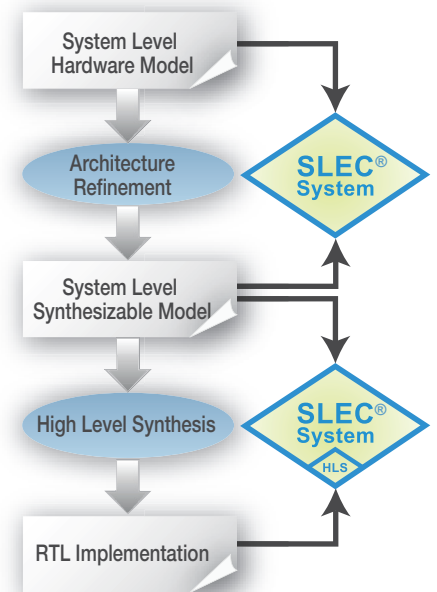
SLEC System and SLEC System-HLS are part of Calypto's proven SLEC product family that also includes SLEC RTL and SLEC Pro. SLEC RTL verifies power and performance optimizations such as clock gating,

retiming, and pipelining. SLEC Pro provides comprehensive, formal verification of PowerPro™ optimizations. The PowerPro product family uses Calypto's sequential analysis technology to evaluate circuit behavior across multiple clock cycles and automatically generate new power optimized RTL. SLEC RTL and SLEC Pro capabilities are included in SLEC System.

SLEC System Verification

Refined system-level models and RTL implementations have vastly different interfaces and internal representations. Unlike traditional combinational equivalence checking tools, SLEC System does not require a one to one mapping of registers, allowing it to verify designs with different timing, interfaces and levels of abstraction. SLEC System leverages the design confidence gained through billions of cycles of system-level verification to ensure that no functional errors are introduced as system-level hardware models are manually refined into system-level synthesizable models or recoded in RTL.

SLEC System supports design blocks over one million gates in size. In ESL design flows, block-level testbenches are not always available. SLEC System allows designers to verify RTL blocks without requiring the development of block-level, RTL testbenches or waiting for the entire design to be assembled.



A Complete ESL Design Flow

Once the system-level model is verified, the system-level hardware model is refined to include architecture details and hardware data-types. SLEC System guarantees the hardware intent remains unchanged throughout the refinement process. After high-level synthesis, SLEC System-HLS comprehensively verifies that the RTL implementation matches the system-level model.

Benefits

- Verification without testbenches or assertions
- Leverages system-level models to verify RTL designs
- Quickly finds bugs that other tools miss
- Isolates bugs with short, concise debug waveforms
- Ensures hardware intent remains unchanged during system-level model refinement
- Integrated with industry leading high-level synthesis tools
- Replaces time consuming simulation regressions with fast results

SLEC System Features

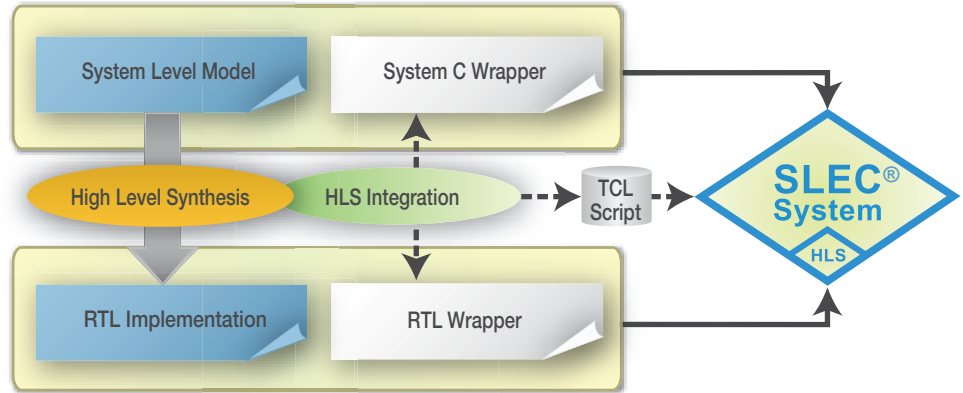
- Supports untimed and timed system-level models
- Verifies designs with different timing, interfaces and levels of abstraction
- Handles designs over one million gates
- TCL interface and batch mode operation
- Generates waveform and testbench counter-examples for debugging
- Integrated with industry standard simulation and debugging tools
- Includes the capabilities of SLEC RTL and SLEC Pro

High Level Synthesis Design Flows

SLEC System-HLS substantially increases the productivity gains of HLS by comprehensively verifying the generated RTL, eliminating time consuming testbench development and RTL simulation. SLEC System-HLS is independent verification of HLS results, providing designers with confidence that all corner cases are verified in their RTL.

SLEC System-HLS is seamlessly integrated with leading HLS tools through the automated generation of the verification environment and the support of HLS specific features and language extensions. The SLEC verification environment consists of a TCL script to setup and control the verification as well as wrappers to align and map design interfaces. Similar to the requirement for formal combination-al equivalence checking in RTL synthesis flows, SLEC System-HLS is an integral part of high-level synthesis design flows.

SLEC System-HLS is an add-on option to SLEC System. SLEC System-HLS solutions include integrations for Cadence® Design Systems' C-to-Silicon Compiler, Forte Design Systems' Cynthesizer™ and Mentor Graphics® Catapult™ C.



SLEC System-HLS is an integral part of High-level Synthesis flows

The integration with HLS tools enables SLEC verification by automating the generation of the TCL script and design wrappers. SLEC System-HLS comprehensively verifies that the generated RTL is functionally equivalent to the original system-level model.

Intuitive User Interface

SLEC verification setup is straight forward, only requiring two designs and a TCL script. The SLEC TCL script identifies the design files, interface mappings, reset conditions and latency/throughput parameters. SLEC includes a rich set of TCL commands that allow users to create advance setups using sequential constraints, hierarchy-based black boxes and intermediate mapping points.

For un-timed system-level models written in C/C++, a simple SystemC wrapper instantiates the design block and a clock signal to align each input and output for comparison. The System C and RTL wrapper are also used to map vastly different interfaces such as C/C++ arrays to bit accurate, RTL memory interfaces.

Advanced Debug

SLEC either proves two designs are equivalent or generates a short, concise counter-example that demonstrates the design difference in a few cycles (typically less than 10 input transactions). Compared to simulation that may require thousands or millions of cycles to reproduce a design

bug, debugging designs with SLEC is fast and efficient. Counter-examples are expressed as waveform traces or simulation testbenches that can be viewed in the user's native simulation or debugging environment. To further assist the user SLEC identifies the signal values and time where the designs first diverge.

System Requirements and Compatibility

- **Languages:**
VHDL 87, 93 & 97 and Verilog 95 & 2001, SystemC 2.1
- **Simulators:**
ModelSim™, VCS®, NCSim™
- **Debuggers:**
Verdi™, SignalScan™
- **Operating Systems:**
Redhat Enterprise Linux 3.0 and 4.0
- **Platforms:**
32-bit and 64-bit x86 compatible hardware
- **Memory:**
2 GB minimum