SILEXICA

SLX C/C++

Master Heterogeneous Computing with SLX C/C++

Optimizing C/C++ applications on complex multicore SoCs requires a complete understanding of the software architecture and its behaviour on the hardware. Developers are challenged with the task of optimizing sequential and parallel code for multicore SoCs that comprise a variety of compute engines. For the most efficient utilization of CPUs, DSPs and FPGAs, a full understanding of the code structure and interdependencies between applications, threads and variables is required for streamlined software development, guided refactoring and software design optimization.

SLX C/C++ enables the development and optimization of concurrent C/C++ applications for heterogeneous multicore platforms with unparalleled actionable insights. This allows for software professionals to achieve an optimized software architecture and improved performance for a specific code on a particular multicore SoC. SLX blends hardware understanding with code analysis to give unrivalled insights into the application's execution while providing a feedback loop to the original source code to provide full traceability. SLX C/C++ can be used on the desktop from a powerful GUI, from command-line or integrated into your continuous workflow.

FEATURES AND CAPABILITIES

Joint Static and Dynamic Code Analysis for Multiple Applications

State-of-the-art code analysis methods provide only a limited view of the software architecture. Static code analysis can find simple bugs or disregarded coding guidelines. Dynamic analysis provides a single profiling run without an understanding of the variance of its behaviour or any root-cause connection back to the source code.

The SLX C/C++ analysis allows for understanding of the static software architecture down to the dynamic behavior of concurrency, synchronization and data flow. This provides a live overview from the source code to prevent architecture erosion and provide actionable insights to optimize the software implementation.

Static, Dynamic and Semantic Source Code Analysis

Actionable Insights for Improved Software Performance

Eclipse GUI or CI Workflow Integration

Software Architecture Analysis with SLX C/C++

SLX gives unprecedented insights into execution behaviour. It weighs the costs and benefits of software optimization to further exploit the target's resources. Key features include:

- **Provide deep application insights to multi-binary and multi-threaded applications.** Through the combination of static, dynamic, and semantic analysis, SLX visualizes thread genealogy, communication, synchronization and data dependencies, providing a live architectural overview from the source code which can be checked for consistency with the envisioned architecture. SLX is the only development tool available today that provides this level of actionable insights.
- **Identify communication and memory bottlenecks by performing analysis at the function, thread and application level.** SLX shared memory (POSIX shared memory variables) analysis makes you aware of how the application communicates among threads and with other applications. SLX shows all accesses to variables including sub-objects of arrays and structs even when accessed through pointers. This provides an up to date architectural overview based on the actual source code to assist with functional debugging, code refactoring and generation of documentation.
- **Identify missing inter-thread and inter-process shared memory protection (semaphores, mutexes) that may lead to data corruption by performing protection analysis.** SLX understands protection mechanisms and can point directly to the problematic source lines so the code can be fixed. This enables not only the detection of data races in between threads, but also in between separate processes and applications.
- **Optimize software distribution for the hardware compute blocks on a heterogeneous multicore system by performing fast "what-if" analysis to visualize code execution.** Optimizations are driven by performance, power and memory requirements for a specific code base given a combination of CPUs, DSPs, FPGAs described in a standardized hardware description format (SHIM2 by the Multicore Association)
- **Save development time by automatically identifying optimization opportunities in the code.** SLX provides guidance to assist code refactoring for improved performance and parallelism detection. Different levels of parallelism are supported including task, pipeline, and data level parallelism.

Multiple Application Analysis

Data Protection and Dependency Analysis

Automatic Software Distribution

Parallelism Detection

The Silexica Solution

SLX C/C++ meets the growing requirement to understand the interaction between multiple applications, processes and threads and gives actionable insights to improve architecture and performance. The tool gives you absolute code understanding to meet the most challenging multicore system requirements. SLX C/C++ can be used on the desktop from a powerful GUI, from command-line or integrated into your continuous workflow.

SLX can support projects in automotive, aerospace & defense, industrial, medical, video, 5G and more. You can find out more at our website **www.silexica.com** or arrange a meeting/ demonstration with us at: **meetus@silexica.com**