PRODUCT BRIEF

SLX FOR FPGA

SLX- HARDWARE/SOFTWARE PARTIONING FOR FPGA-ENABLED SOCS

SLX combines a deep understanding of legacy source code, how it is executed and the implications of a system that enables the partitioning of applications into software to run on several processors, and hardware to be mapped to an FPGA. It gives comprehensive processor modeling, and characterization of the timing and resource capabilities of the FPGA. SLX searches for parallelism patterns from static and dynamic information. The design space is explored to fully optimize hardware/software partitioning. Pragmas are automatically inserted based on the optimization decisions for the parallelization of the software and for guiding the High Level Synthesis (HLS) process for the hardware implementation. SLX uniquely connects its analysis and advice to your source code, simplifying your task of understanding the opportunities for, or blockers of, parallelism and hardware acceleration.

ANALYZE

Analyze your software to fully understand your code and automatically identify further parallelization opportunities.

Absolute code understanding

OPTIMIZE

Optimize the distribution of your application on your target platform, driven by performance, power and memory constraints.

Meet challenging requirements

INTEGRATE

Integrate easy-to-use recipes and automatically generate code, instantly improving your software.

Faster time to market

FEATURES AND CAPABILITIES

ANALYZE

Analyze code

SLX builds a complete application model from sophisticated compiler technology utilizing both static as well as dynamic analysis to identify and reason about all data and control dependencies. The model includes the application call graph, read and write accesses to local, heap and global variables, and a complete understanding of memory accesses within the system architecture.

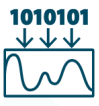
A pattern-based framework identifies potential opportunities to transform your application to expose more parallelism. SLX identifies:

- Data-Level, Task Level and Pipeline-Level parallelism
- Offloading opportunities for custom cores and hardware accelerators
- Parallelism potential blocked by dependencies or irregular control patterns

Based on the timing and resource information from SLX's platform models, the designer's analysis is focused only on the parts of the application that can provide an effective gain. This eases the designer task, and increases productivity.

Static and Dynamic Source Code Analysis

Cache-, Memory- and Communication Analysis



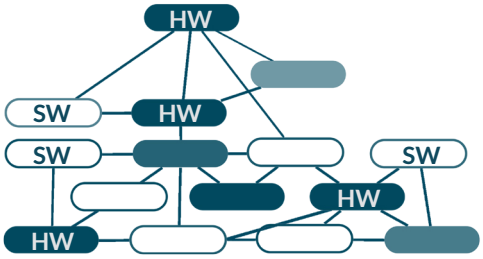
Cross-Target Performance Estimation

Call-graph and Variable Access

OPTIMIZE

Find the Optimal Hardware/Software partitioning

Based on the platform models, SLX is able to determine the timing and resource consumption for the implementation of a detected parallelism pattern, running in a processor as software, or in the FPGA as a hardware accelerator. An optimization algorithm utilizes this information to calculate the final partitioning of the application, and is able to:



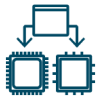
- Provide graphical representation of the partitioning, directly linked with the source code
- Report global and local performance improvement for each detected parallelism pattern
- Report the final resource consumption on the FPGA, for the calculated hardware/software partitioning
- Provide recommendations on the most beneficial partitioning for the target FPGA-enabled SoC

SLX allows the user to override the calculated partitioning, and manually test alternative implementations. This workbench approach guarantees to keep the designer in the loop, and exploit human ingenuity while increasing productivity.

DLP, TLP, PLP Extraction

Identification of Blocking Dependencies and Control

Automatic and user-annotated timing resource consumption

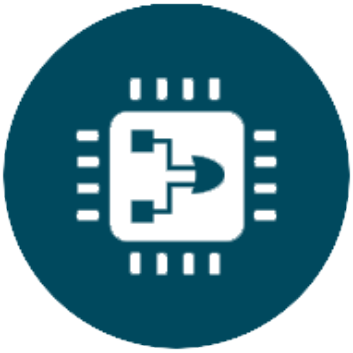


Target specific

INTEGRATE

Guide and Rewrite

SLX helps users migrate their existing application by automatically rewriting the code once a partitioning has been established. For the software portions of the application that exhibit parallelism, OpenMP 4.5 Pragmas are inserted to guarantee a multithreaded integration in the target platform. Xilinx pragmas are inserted for the hardware parts of the application, tuned to guarantee an efficient integration for the resulting accelerators.



SLX also provides code refactoring hints for the parts of the application that could not be parallelized or implemented in hardware, pin-pointing the reasons that impeded these activities. SLX has an end-to-end integration with Xilinx Vivado HLS and SDSoC tools, that creates a way between legacy code and a partitioned application running on the target FPGA-enabled SoC.

THE SILEXICA SOLUTION

SLX gives you absolute code understanding to meet the most challenging multicore system requirements.

Clear suggestions to spend less time during multicore migration

Mapping dependent code generation for FPGA multicores

Automatic insertion of pragmas (OpenMP, HLS, etc.)