SILEXICA

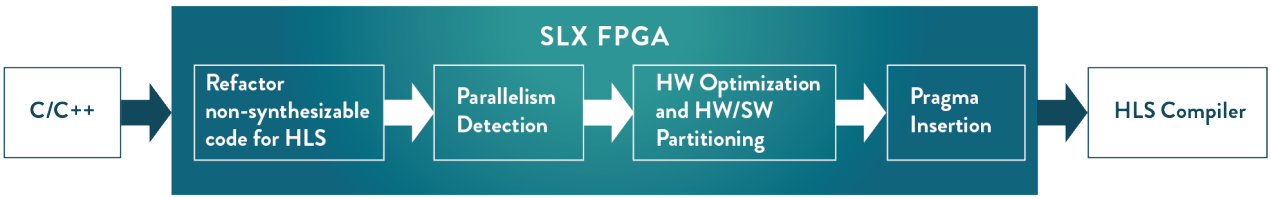
SLX FPGA

Accelerating the journey from C/C++ to Hardware

SLX FPGA helps to convert your C/C++ code into an FPGA bitstream more easily, faster, and with higher performance. Leveraging standard HLS (High Level Synthesis) tools from FPGA vendors, SLX FPGA tackles the challenges associated with the HLS design flow including non-synthesizable C/C++ code, non-hardware aware C/C++ code, detecting application parallelism, where to insert pragmas, and how to determine optimal SW/HW partitioning. Using SLX FPGA enables you to get to market faster by leveraging the benefits of HLS for FPGA design entry. These benefits include improved productivity through designing at a higher level of abstraction, orders of magnitude faster simulation than traditional RTL simulation, and higher QoR through high-level optimizations and design space exploration.

SLX FPGA addresses the challenges of using a HLS design flow by performing static and dynamic code analysis and providing deep insights in to the user's C/C++ code. Through this code analysis, SLX FPGA identifies non-synthesizable C/C++ code, detects non-hardware aware data types, and pinpoints parallelism within the SW that can be implemented in HW for acceleration. SLX FPGA provides guided and automatic code refactoring for HLS synthesizability helping to solve the biggest barrier and most time-consuming aspects of using HLS design flows. SLX FPGA then uses the detected parallelism to automatically generate and insert HLS pragmas which optimize the design for performance and area utilization.

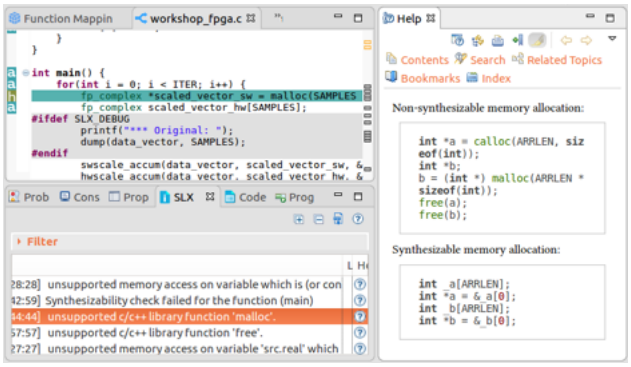
SLX FPGA provides a step by step flow to optimize C/C++ code for High Level Synthesis, significantly reducing development time and ensuring an optimized hardware implementation of accelerators coded in C/C++.



Refactor non-synthesizable code for HLS

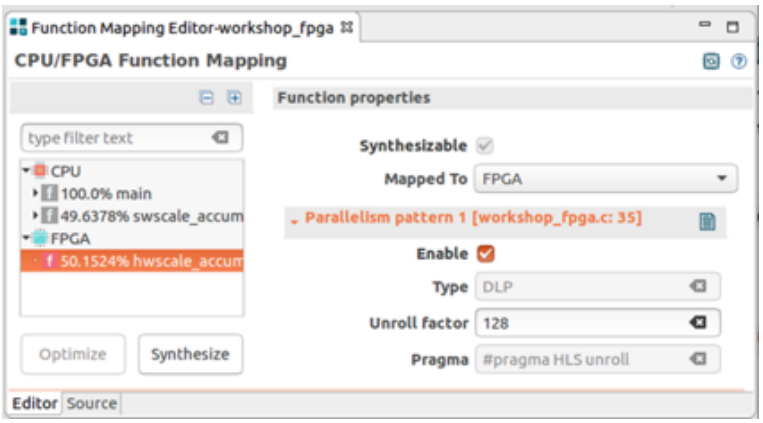
C/C++ coding guidelines for HLS compilers are extensive and can be over 1000+ pages of documentation that needs to be comprehended when writing or refactoring C/C++ code for HLS synthesis. SLX FPGA eliminates the need to be an expert in coding for HLS by:

- **Identifying C/C++ code which is non-synthesizable**
- **Performing automatic code refactoring for common non-synthesizable libraries**
- **Providing guided code refactoring by supplying code examples to help re-write the code to make it synthesizable**



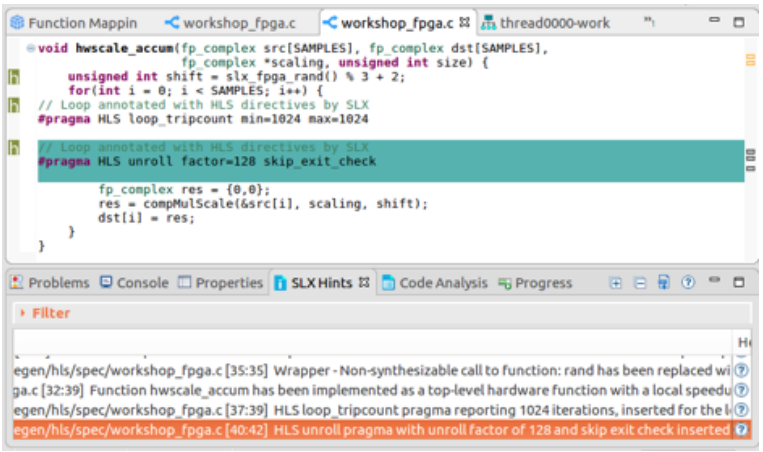
Parallelism Detection

C/C++ code is typically executed sequentially on standard processors but implementing functions in dedicated hardware allows operations to be executed in parallel, accelerating the execution of the code in hardware. SLX FPGA analyzes application C/C++ code and identifies functions that can be accelerated by executing in parallel when implemented.



HW Optimization and HW/ SW Partitioning

After identifying functions that can be implemented to execute in parallel, SLX FPGA performs analysis of the functions to determine the theoretical maximum for the achievable speedup. Using Silexica's proprietary algorithms, SLX FPGA then determines the ideal implementation of the parallel function based on user supplied constraints, ensuring an optimized implementation.



Pragma Insertion

Once the optimized hardware implementation is determined, SLX FPGA inserts HLS Pragmas to direct the HLS complier on how to implement the function in hardware.

SLX FPGA is fully integrated with Xilinx Vivado HLS and the SDSoC Development Environment to create a complete path from C/C++ to FPGA synthesis. It can be used on the desktop from a powerful GUI, from command-line or integrated into your agile, continuous workflow. You can find out more at our website here: www.silexica.com or arrange a meeting/demonstration with us at: meetus@silexica.com

Convert non-synthesizable C/C++ code with Auto and Guided Code Refactoring

Detect Parallelism in C/C++ code to implement in hardware

HW optimization and SW/HW partitioning

Automatic insertion of HLS pragmas