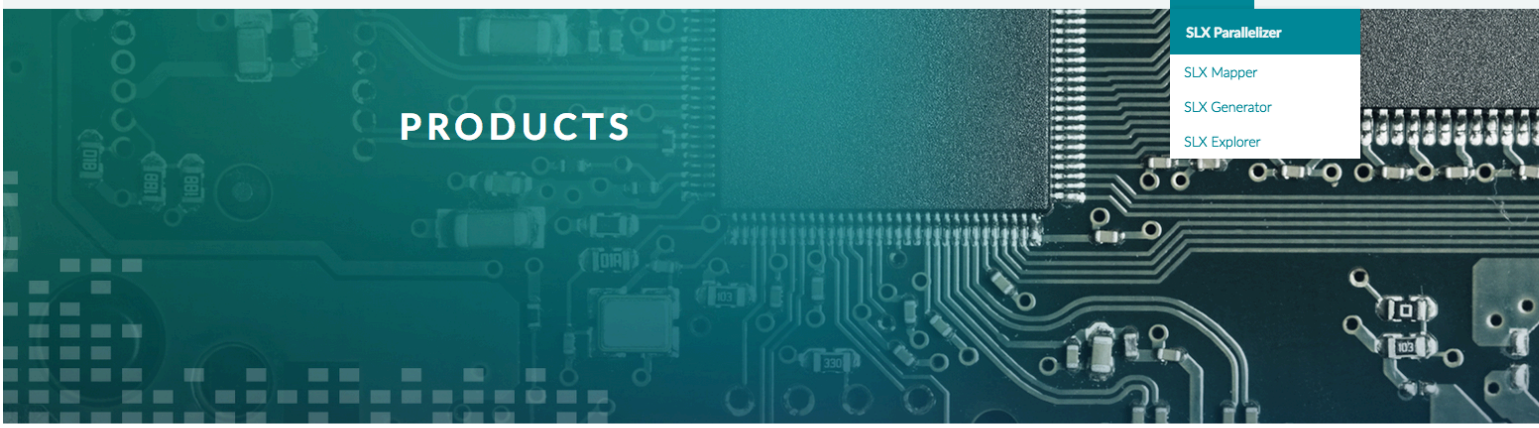
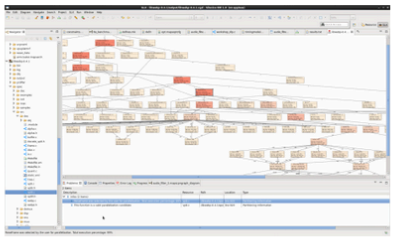


SLX Parallelizer

automatic identification of parallelism

SLX Parallelizer receives sequential C code as input and semi-automatically turns it into a parallel process network implementation. Internally, it uses static and dynamic code analysis techniques to detect typical parallelism patterns that are guaranteed to deliver performance improvements in multicores.



Example of these patterns are **Task-Level Parallelism (TLP), Data-Level Parallelism (DLP), and Pipeline-Level Parallelism (PLP)**. This structured parallelism is commonly present in all compute-intensive applications but is mostly hidden within the sequential implementation.

SLX Parallelizer interacts with the programmer to expose this parallelism. It performs whole-program analysis and uses automatic performance estimation to determine the best parallel representation of the application for a given target multicore. The parallelized version can be **exported as C code with standard parallel APIs**, such as OpenMP, or can be forwarded as a CPN specification ("C for Process Networks") into the SLX Mapper for further optimizations and refinement.