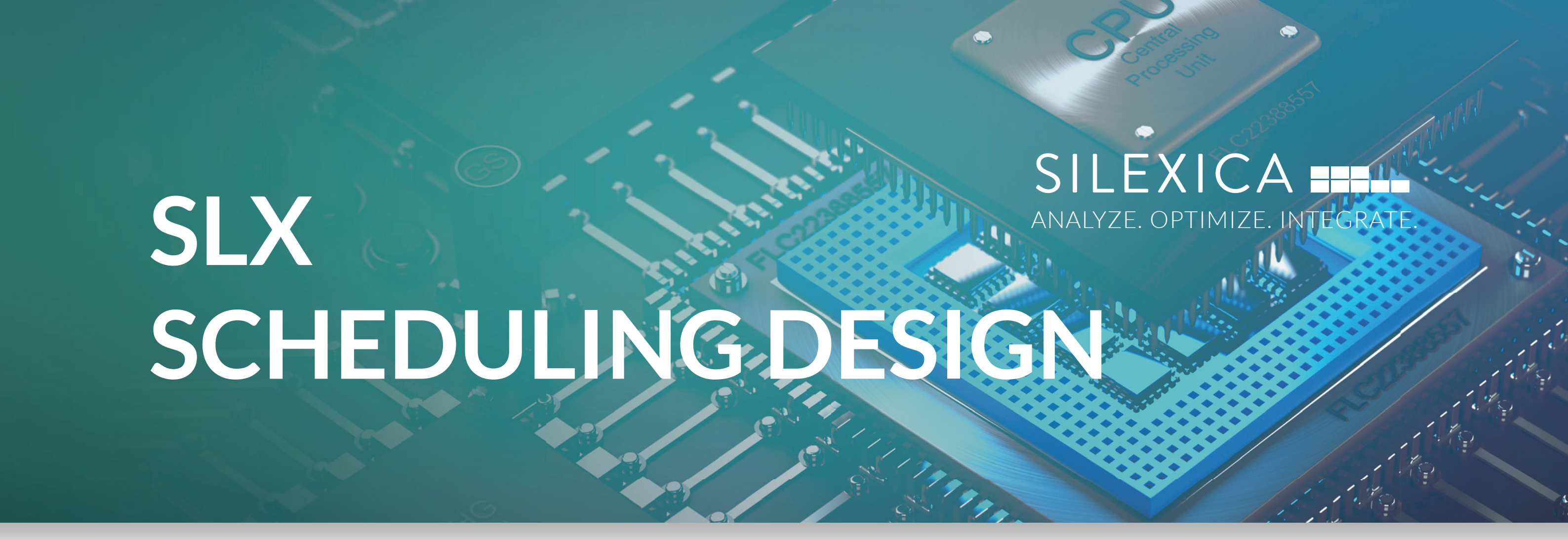


SLX SCHEDULING DESIGN

Scheduling Optimization and Automatic Memory Mapping with SLX

SLX Scheduling Design provides solutions for task-based systems with C/C++, AUTOSAR Classic or AMALTHEA inputs. By giving a full understanding of the dependencies between functions and tasks, efficient multicore schedules can be generated.

By using the power of SLX's Function Level Parallelism and Task Level Parallelism capabilities, power consumption can be reduced while still meeting crucial system latency constraints. SLX for Scheduling Design operates in three phases, Analyze, Optimize and Integrate:

ANALYZE

Analyze your software to fully understand your software architecture including interdependencies of the application's tasks and functions

OPTIMIZE

Optimize system performance through intelligent schedule design which supports runnable distribution and variable to memory mapping

INTEGRATE

Easy integration in AUTOSAR, AMALTHEA and other environments with automatically updated configuration files.

FEATURES AND CAPABILITIES

Analyze Code or Models

Optimal scheduling requires knowledge of the dependencies between functions and tasks and how often data is accessed. SLX puts analytical results in the human domain with deep analysis for:

- **Task Data Dependencies**
  Find and display data dependencies across task borders
- **Function Dependencies**
  Display data dependencies of runnables assigned to a single task
- **Interactive Dependency Graph**
  Add or remove dependencies between functions graphically
- **Function Call Graph**
  Enabling deeper insights to the implementation of your runnables utilizing the function call graph
- **Code Analysis Graph**
  For a system wide overview of data elements and accessing functions

Static and Dynamic Source Code Analysis

Function and Task Dependency Analysis

Automated Runnable Distribution to Processing Elements

Function-Level Parallelization

SLX optimizes the execution of functions and task mapping. Function Level Parallelization utilizes SLX sequential code technology, as they are also sequential programming elements.

1. Function Parallelization

Function Level Parallelization speeds the execution of tasks without changing the properties of the original application. SLX implements a scheduling algorithm to distribute a task's functions among multiple processors, ensuring data-dependencies are satisfied. The results can be visualized as Gantt charts and generated schedules can be applied to a legacy system. SLX also supports Task Level Parallelism by decoupling producer-consumer dependencies, given that the most recent data is not critical for the application behavior. This enables the producer and consumer to run in parallel for an additional efficiency.

2. Memory Mapping

Based on the analyzed variable accesses and the generated multicore schedule, SLX maps variables to the selected platform's memories efficiently. The mapping can be shown visually using mapping tables and charts to help investigate the memory consumption of your application. The variable mapping reduces the amount of costly cross core communication. This is also compatible with the AUTOSAR MemMap feature.

3. Power Optimized Parallelization

Combining Function Level Parallelism and task level parallelism, SLX reduces the execution time allowing changes to system voltage and frequency. This reduces power consumption while still meeting system latency constraints. Alternatively, the increased capacity can be used to augment application features.

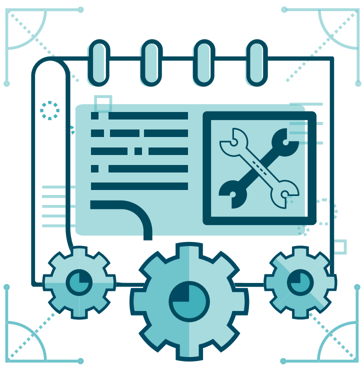
Variable to Memory Mapping

Optimized Task Schedules under Tight Timing Constraints

Performance Estimation and WCET Import

Automatic Code Generation

SLX supports the export of AMALTHEA and AUTOSAR data exchange formats to allow for seamless integration with 3rd party tools. AMALTHEA is a xml-based exchange format for embedded multicore systems which allows for storing both the software and hardware model of the system. The type of information includes Task, Runnable, Variable, Scheduler, OS, Memory and Microcontroller information. AUTOSAR uses an xml-based exchange format to store configuration and design details. As optimization choices and changes are made, SLX automatically updates the AUTOSAR design and configuration files.



The Silexica Solution

SLX gives you absolute code understanding to meet the most challenging multicore system requirements. SLX Scheduling Design can be used on the desktop from a powerful GUI, from command-line or integrated into your agile, continuous workflow. It can support projects in automotive, aerospace & defense, industrial, medical, video, 5G and more. You can find out more at our website here: www.silexica.com or arrange a meeting/demo at: meetus@silexica.com

Actionable Insights for Multicore Migration

Automatic Insertion of Synchronization Primitives

AUTOSAR and AMALTHEA support