**SLX - MEETING THE AUTOMOTIVE CHALLENGE**

The automotive industry has specialized requirements and standards that offer many challenges to software professionals. SLX makes it easier and quicker to benefit from AUTOSAR interoperability and legacy systems, while taking full advantage of new multicore ECUs.

SLX is a programming technology that helps you meet your most challenging system requirements. State-of-the-art compiler technology and full heterogeneity awareness is comprised into three tasks you need to complete when writing software for a multicore platform:

**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**
- **Faster time to market**

**IMPLEMENT**

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

**FEATURES AND CAPABILITIES**

- **Analyze code**
  Optimal scheduling requires knowledge of the dependencies between runnables and tasks and how often data is accessed. SLX puts analytical results in the human domain.

  - **Inter-Task Data Dependencies**
    Find and display data dependencies across task borders
  
  - **Inter-Runnable Dependencies**
    Display data dependencies of runnables assigned to a single task
  
  - **Intra-Runnable Call Graph**
    Get deeper insights to the implementation of your runnables utilizing the runnable call graph
  
  - **Code Analysis Graph**
    Get a system wide overview of data elements and accessing functions

- **Static and Dynamic Source Code Analysis**
- **Cache-, Memory- and Communication Analysis**
- **Cross-Target Performance Estimation**
- **C/C++, data-flow, task-graphs**

© 2017 Silexica, Inc. All Rights Reserved
Runnable Parallelization
With the analysis performed, the next step in the flow is optimization. SLX optimizes the execution of runnables and task mapping in an AUTOSAR system.

Intra-Runnable Level Parallelization applies the proven technology Silexica developed for sequential code to runnables, as they are also sequential program elements.

Inter-Runnable Level Parallelization speeds the execution of tasks without changing the properties of the original application configuration. SLX distributes a task’s runnables to multiple cores, ensuring data-dependencies are satisfied.

SLX implements a scheduling algorithm to distribute task’s runnables among multiple processors. The results can be used to create Gantt charts and schedules that can be applied to a legacy or AUTOSAR system.

Inter-Task Level Parallelization
SLX supports the decoupling of 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 gain in efficiency.

Power Optimized Parallelization
Combining the tool’s Runnable Level Parallelism and Task Level Parallelism capabilities, SLX reduces the overall execution time of a sensor-actuator system allowing changes to system voltage and frequency. Thus reducing power consumption while still meeting sensor and actuator latency constraints. Alternatively, the increased capacity can be used to augment application features.

Faster Time to Market
AUTOSAR uses an xml-based exchange format to store configuration and design details. As optimization choices and changes are made, Silexica automatically updates the AUTOSAR design and configuration files.
- Generate the desired code
- Configure basic software
- Update RTE - scheduling, communication, implementation

THE SILEXICA SOLUTION
SLX improves your time-to-market, your feature set, and lowers costs and power, all while keeping your application compliant with the AUTOSAR stack.

Clear suggestions to spend less time during multicore migration

www.silexica.com