**ALGorithm parallelization for Multicore Architectures**

**Faster time-to-market for embedded multicore systems with less application development effort**

The ALMA ToolFlow aims to:
- **Hide the complexity** of the underlying hard-ware to the programmer.
- Provide a new approach for compiling annotated Scilab Code to MPSoC architectures.
- Develop a unified SystemC simulation framework for MPSoCs.

**Hide the Complexity by the ALMA Tool Flow**

**ALMA Front-End Tools**
- Scilab Front-End (SAFE)
  - Parses Scilab source code and produces high level intermediate representation (HLIR) expressed in C.
- ALMA profiler (approf)
  - Early performance estimation at the HLIR level.
  - High-Level Optimizer (HLO)
  - Applies platform-dependent optimizations to the HLIR.

**Fine-Grain Parallelism Extraction**
- Floating point to fixed point
  - No hardware support for FP in embedded multicore systems.
  - Provides an automated floating- to fixed-point conversion tool.
- SIMD/ SPAW parallelization
  - Loop parallelization and layout optimization for SIMD instructions.
  - Explore performance vs. accuracy trade-off in fixed-point encodings.

**Coarse-Grain Parallelism Extraction**
- Responsible for global optimization.
- Transformation of ALMA IR CFGD to Hierarchical Task Graph (HTG).
- High-level parallelization transformations to increase schedulable parallelism.
- HTG partitioning to cores.
- Optimal mapping and scheduling of tasks to architecture resources.
- Iterative optimization by using task and communication profiling.

**Parallel Code Generation**
- Generates target-specific C code.
- Maps Scilab variables to memory locations.
- Expresses communication and SIMD instr. by instrumentation for profiling.
- Uses Recore/Kahrisma C compiler.
- Generates executable for the hardware and simulator.

**Multicore Architecture Test Cases**
- Recore Systems’ Multicore DSP Platforms.
- KIT’s KAHRISMA Architecture.

**ALMA Target Architectures**
- Simulate and analyze the architecture description.
- Retargetable.
- Structure defined by ADL.
- Implementation by library of SystemC modules.
- Mixed-accuracy simulation.
- Behavioural or cycle-accurate.
- For individual modules.
- Enables task and communication profiling.

**Application Input Language (Scilab)**
- ALMA dialect of the Scilab language.
- Subset of Scilab language.
- Extended by a preprocessing language.
- Variables declaration.
- Static types specification.
- Maximum size of vector/matrix data type definition.
- Extended by an annotation language for supporting parallelism extraction.

**Application Test Cases**
- Image Processing.
- Object recognition and multi-object tracking.
- Use of Scale Invariant Feature Transform (SIFT).
- Telecommunication.
- IEEE 802.16e PHY Layer in NT x NR MIMO Configuration.
- State-of-the-art WIMAX wireless communication.

**Architecture Description Language (ADL)**
- Enables target independence of the toolchain.
- Used as architecture description for the simulator.
- Enables design-space exploration.
- Compact specification of regular MPSoC structure.
- Structural specification annotated with behavioural information.
- Hierarchical module description for mixed-accuracy simulation support.

**ADL Compiler**
- Compile and analyze the architecture description.
- Extracts high-level information from ADL description (e.g. number of cores, communication bandwidth, available memories).
- Flattens hierarchical description.

**WWW.ALMA-PROJECT.EU**

This work is co-funded by the European Union under the 7th Framework Programme under grant agreement ICT-287733.