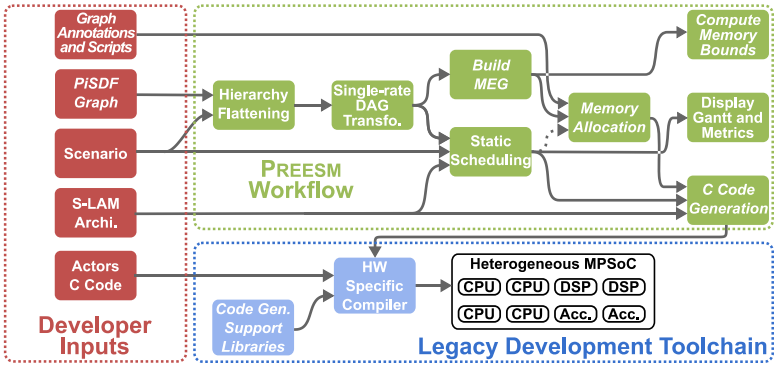
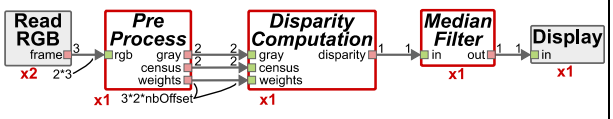
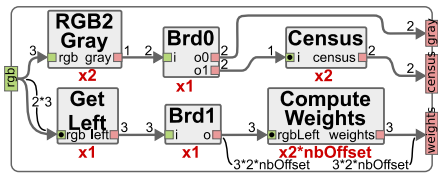
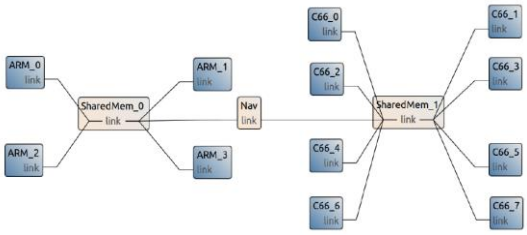
Users	• Software developers/embedded system engineers with little to no knowledge of the hardware
Key Features	• Design parallel application intuitively with a graphical editor of dataflow graph, independently from any implementation consideration. • Simulate and generate correct-by-construction parallel code for a wide range of embedded systems, including customizable heterogeneous multi- and many-core systems. • Optimize automatically the latency, the core loads, and the memory footprint of applications.
Benefits for the User	• PREESM automates inter-PE (processing element) communications. Communication libraries are distributed with PREESM that automatically synchronizes cores. • PREESM offers performance predictability at the early stages of heterogeneous digital system design. • PREESM fosters legacy code reuse. Existing functions can be easily called from PREESM generated code
Inputs	• Block-based hierarchical description of the application: PiSDF Dataflow graph • Imperative code describing the internal behaviour of dataflow "blocks": C code • High-level description of the targeted architecture: S-LAM (based on IP-XACT standard) • Deployment scenario: Specify constraint for a given pair of application and architecture
Outputs	• Optimized parallel (Multi-threaded) code for heterogeneous embedded platforms. • Simulation of application deployment: Gantt diagram, core load & memory metrics
Block Design (Preesm Overview)	
Example of Usage (PiSDF graph)	 Stereo Matching application graph  Hierarchical sub-graph of the *Pre-Process actor*
Example of Usage (S-LAM graph)	
Role in the Toolchain	Crossing layers from high-level models to embedded SW implementation