

# ParModelica: Extending the Algorithmic Subset of Modelica with Explicit Parallel Language Constructs for Multi-core Simulation

Mahder Gebremedhin      Afshin Hemmati Moghadam      Peter Fritzson  
Kristian Stavåker  
Department of Computer and Information Science  
Linköping University, SE-581 83 Linköping, Sweden  
{mahder.gbremedin, peter.fritzson, Kristian.stavaker}@liu.se, af-  
she586@student.liu.se

In today's world of high tech manufacturing and computer-aided design simulations of models is at the heart of the whole manufacturing process. Trying to represent and study the variables of real world models using simulation computer programs can turn out to be a very expensive and time consuming task. On the other hand advancements in modern multi-core CPUs and general purpose GPUs promise remarkable computational power.

Properly utilizing this computational power can provide reduced simulation time. To this end modern modeling environments provide different optimization and parallelization options to take advantage of the available computational power. Some of these parallelization approaches are based on automatically extracting parallelism with the help of a compiler [1] [2]. Another approach is to provide the model programmers with the necessary language constructs to express any potential parallelism in their models. This second approach is taken in this work.

The OpenModelica modeling and simulation environment for the Modelica language has been extended with new language constructs for explicitly stating parallelism in algorithms. This slightly extended algorithmic subset of Modelica is called ParModelica. The new extensions allow models written in ParModelica to be translated to optimized OpenCL[3] code which can take advantage of the computational power of available Multi-core CPUs and general purpose GPUs. This enables the Modelica modeler to express parallel algorithms directly at the Modelica language level. The generated code is portable between several multi-core architectures since it is based on the OpenCL programming model.

The implementation has been evaluated on a benchmark suite containing models with matrix multiplication, Eigen value computation, and stationary heat conduction. Good speedups were obtained for large problem sizes on both multi-core CPUs and GPUs. To our knowledge, this is the first high-performing portable explicit parallel programming extension to Modelica.

## References

- [1] Peter Aronsson. *Automatic Parallelization of Equation-Based Simulation Programs*, PhD thesis, Dissertation No. 1022, Linköping University, 2006.
- [2] Håkan Lundvall. *Automatic Parallelization using Pipelining for Equation-Based Simulation Languages*, Licentiate thesis No. 1381, Linköping University, 2008
- [3] Khronos Group, Open Standards for Media Authoring and Acceleration, OpenCL 1.1, accessed Sept 15, 2011. <http://www.khronos.org/opencv/>