AES Embedded Systems Architecture

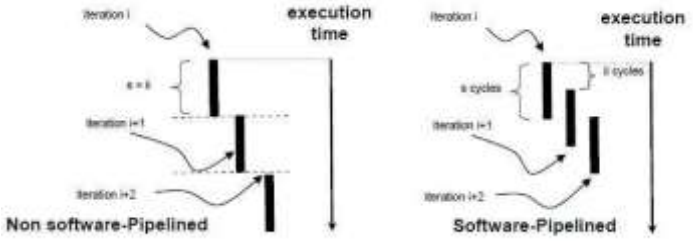
Master Thesis
Compiler Technology

Technische Universität Berlin

Code Optimization for Embedded Processors

Embedded processors have specific requirements in terms of compiler support. On one hand, embedded architectures require more effort to generate efficient code, as hardware limitations push the complexity on the software side. On the other hand, software optimization does not only focus on performance, but it can also target energy/power consumption and code size.

Goals



The objective of this thesis topic is to study and implement compiler techniques that specifically target embedded processors. These will focus on specific a specific problems such as instruction scheduling, power-aware compilation, binary-size optimization, vectorization, et cetera, and will overlap techniques and methodologies from ongoing research work.

Available theses

1. **Energy-aware kernel compilation for GPUs**. GPU energy consumption is not constant and can greatly vary depending on the code executed and the input data. Starting from an architectural analysis on GPU consumption (see Lucas&Juurlink, MASCOTS 2016), the thesis goal is to translate such model into compiler transformations (e.g., registry assignment and instruction scheduling) that reduce the GPU energy consumption while keeping the same runtime performance.
2. **VLIW compilation.** VLIW are architectures that offer high performance at a much lower cost than dynamic out-of-order superscalar processors. By allowing the compiler to directly schedule machine resource usage, the need for expensive instruction issue logic is obviated. The goal of the thesis is to analyze state-of-the-art VLIW compilation techniques such as trace scheduling, speculative execution and software pipelining, and to identify criticality and performance issues for a specific HW (to discuss which one).
3. **Automatic Vectorization.** Compiler-based vectorization is currently addressed by compilers such as LLVM and gcc with two techniques: Loop-Level Vectorization and Superword Level Parallelism (SLP) vectorization. The goal is to investigate cost models and vectorizations techniques for embedded processors.

Desired Skills

- C programming
- Compiler course
- Knowledge of parallel programming model such as OpenCL, CUDA, OpenMP or MPI is a plus

Contact Person

Dr. Biagio Cosenza cosenza@tu-berlin.de (in collaboration with Jan Lucas and Angela Pohl)

References

- Paper on energy modeling for GPU http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7774570
- Auro-Vectorization in LLVM http://llvm.org/docs/Vectorizers.html