COST Action: IC1202
Title: Time-predictable memory hierarchy for single-path code
Applicant: Bekim Chilku (TU WIEN)
Host: Martin Schoeberl (DTU)
Reference: ECOST-STSM-IC1202-111015-068518
Dates: from 11-10-2015 to 08-11-2015

Report

Purpose of the STSM

Single-path conversion is a strategy to eliminate the complexity of the WCET analysis of the code. The approach reduces the complexity of timing analysis by converting all input-depended alternatives of the code into pieces of sequential code and with that it's eliminating all control-flow induced variations in execution time. The new transformed code has a single execution trace that forces the execution to become constant. To obtain information about the timing of the code it is sufficient to run the code only once. The major drawback of single-path code is its potentially long execution time.

The predictable properties that are gained from the transformed code, in combination with cache prefetcher, have been considered as a potential solution that would improve the execution time properties of the single-path code. By having the knowledge on the execution stream of the instructions, the prefetcher can be guided to prefetch in advance the exact instructions from the main memory without polluting the cache with useless data.

The aim of the Short-term Scientific Mission (STSM) was to build and integrate the prefetcher in Patmos processor and run it on the FPGA. The reason for choosing Patmos is that, on one side, the compiler for this processor also supports conversion of the code into a single-path. On the other side, DTU has the comprehensive and detailed knowledge of Patmos for effective implementation and integration of the prefetcher into the processor.

Description of the work carried out during the STSM

The work carried out during the visit to DTU followed generally the main points of the Working Plan presented with the STSM application.

- A presentation of the prefetcher and its functionality was given in the front of the research group in DTU. During this presentation it was shown how the prefetcher gather information on instruction stream and how it will react in order to bring the right instruction into the cache without interfering the execution.

- A detailed analysis of Chisel code for instruction cache was done. The goal was to get the knowledge on how the processor stall is implemented on

cache miss and how the processor was informed that the whole cache line have arrived into the cache.

- The instruction cache was redesigned in order to separate the part of the cache that serves the fetch stage and the prefetcher form the part that brings the instruction from the main memory. The goal was to make the cache-memory communication to bring fetched and prefetched instructions from the main memory in aggressive form (fully utilized) without being dependent on the fetch-cache communication. Also, the cache had to consider not replicating the requests from the fetch stage if the requested was already generated from the prefetcher.

- The Patmos had only direct-mapped cache implementation. A 2-way set associative cache with fifo replacement was also build.

- A prefetcher for direct-mapped and 2-way associative cache was implemented and integrated into Patmos. The signal analysis was also done in order to verify the functionality of the cache and the prefetcher for both variants.

- Patmos processor configuration file was adapted in order to make it easy for any user to choose deferent configuration of the memory hierarchy (direct-mapped or 2-way associative cache with/without prefetcher) when processor is build.

The whole work is now available at github repository of Patmos processor[1]

Description of the main results obtained

During the STSM time period we managed only small examples of searching algorithms to convert them into a single-path and run it with Patmos. The results showed improvement of the execution time of 10-15%. However these are just a preliminary numbers. The compiler for Patmos is still under development and needs additional effort to adapt the code in order to convert them into a single-path code. At the moment the work is continued at TU Wien in order to convert the set of Debie, Malardalen and MiBench benchmarks to do evaluation of the Patmos with prefetcher.

Also, a joint paper between TU Wien and DTU has been started and the result from this STSM work will be presented.

Future collaboration with host institution

Even if the single-path concept has the goal of achieving constant execution time, the research so far has been focused only on the instruction cache. During the STSM period, possible solutions on the data cache have also been discussed in order to make the data access also stable. This has been considered as a new

[1] https://github.com/t-crest/patmos/tree/icache_with_prefetcher

possibility for continuing the collaboration between TU Wien and DTU on the single-path code in this direction.

Acknowledgment

I would like to express my very grad appreciation to Martin Schoeberl for his valuable and constructive suggestions during my stay at DTU. I would also like to thanks Wolfgang Puffitsch for helping me with the integration of the prefetcher into the Patmos processor, and TACLE Action for enabling me to do this STSM research at DTU.