

ASIC² Project: Automatic alteration of code designed for loop-unrolling to enable Multithreading

Background: Modern computer systems contain a variety of mechanisms, designed to speed up the execution of certain types of programs. Two examples of such mechanisms are:

- Loop unrolling, a software tweak that takes loops in a source code and unwinds them so that in each modified iteration, several iterations of the original code are performed. The unrolling itself is done either at compilation time or at runtime, and is somewhat dependent on the independence of statements from different iterations.
 - Multithreading: a hardware mechanism that allows multiple threads of code to run "simultaneously". While the processor width is not changed, there can be no actual simultaneity between threads. But when one thread faces a delay in execution, another independent thread can continue its execution, thus increasing the total throughput.
- Each of these may work better with different software, or even with different compilations of the same code. So a SW/HW vendor needs to choose which of the two (if any) to make optimizations for.

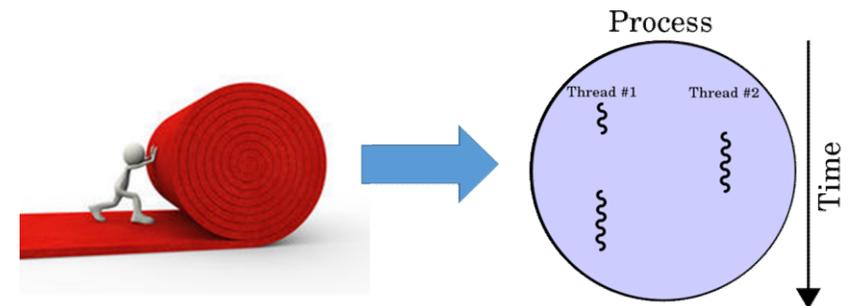
Project Description:

This project is part of a collaboration with CEVA Ltd. The ultimate goal is to examine the option of introducing multithreading techniques (namely SoE-CFMT) to CEVA's SW and HW, currently heavily relying on loop unrolling.

The project requirements are:

- Study the theory and practice of loop-unrolling and multithreading (SoE-CFMT), mostly from a SW perspective.
- Develop a method for evaluating the speedup from using one method or another, and validate using a simple multithreaded code.
- Characterize the code supplied by CEVA in terms of dependency between operations, Loop sizing (number of iterations, commands per iteration), Data structures used, and any other relevant parameters.
- Manually convert a section of code from loop unrolling to multithreading and run it to validate correctness.
- Develop an automatic/semi-automatic tool for code conversion and evaluate results.

The code provided by CEVA for the project, and all converted code, will be in C/C++ language.



Prerequisites:

**234267 or 046267 – Computer Architecture,
236360 or 046266 – Compilation**

Recommended:

**046267 – Computer Architecture,
044160 – EE Lab 1**

Supervisor: TBD – contact nimrodw@tx.technion.ac.il