Abstract

An instruction-set simulator is a program that simulates a target computer by interpreting the effect of instructions on the computer, one instruction at a time. This study is based on an existing instruction-set simulator, which simulates a general PowerPC (PPC) processor with support for all general instructions in the PPC family. The subject for this thesis work is to enhance the existing general simulator with support for the memory management unit (MMU), with its related instruction-set, and the instruction-set controlling the cache functionality. This implies an extension of the collection of attributes implemented on the simulator with, for example, MMU tables and more specific registers for the target processor.

Introducing MMU functionality into a very fast simulator kernel with the aim not to affect the performance too much has been shown to be a non-trivial task. The MMU introduced complexity in the execution that demanded fast and simple solutions. One of the techniques that was used to increase the performance was to cache results from previous address translations and in that way avoid unnecessary recalculations.

The extension of the simulator, with complex functionality as MMU and interrupts, only decreased the performance approximately two times, when executing with all the MMU functionality turned on. This was possible as a result of the successfully implemented optimisations.