Time-predictable Stack Caching

Embedded systems are computing systems for controlling and interacting with physical environments. Embedded systems with special timing constraints where the system needs to meet deadlines are referred to as real-time systems. In hard real-time systems, missing a deadline causes the system to fail completely. Thus, in systems with hard deadlines the worst-case execution time (WCET) of the real-time software running on them needs to be bounded.

Modern architectures use features such as pipelining and caches for improving the average performance. These features, however, make the WCET analysis more complicated and less imprecise. Time-predictable computer architectures provide solutions to this problem. As accesses to the data in caches are one source of timing unpredictability, devising methods for improving the time-predictability of caches are important. Stack data, with statically analyzable addresses, provides an opportunity to predict and tighten the WCET of accesses to data in caches.

In this thesis, we introduce the time-predictable stack cache design and implementation within a time-predictable processor. We introduce several optimizations to our design for tightening the WCET while keeping the time-predictability of the design intact. Moreover, we provide a solution for reducing the cost of context switching in a system using the stack cache. In design of these caches, we use custom hardware and compiler support for delivering time-predictable stack data accesses. Furthermore, for systems where compiler support or hardware changes are not practical, we propose and explore two different alternatives based on only software and only hardware support.