Abstract

During the design of safety-critical real-time systems, developers must be able to verify that a system shows a timely reaction to external events. To achieve this, the *Worst-Case Execution Time* (WCET) of each task in such a system must be determined. The WCET is used in the schedulability analysis in order to verify that all tasks will meet their deadlines and to verify the overall timing of the system. Unfortunately, the execution time of a task depends on the task's input values, the initial system state, the preemptions due to tasks executing on the same core and on the interference due to tasks executing in parallel on other cores. These dependencies render it close to impossible to cover every feasible timing behavior in measurements. It is preferable to create a static analysis which determines the WCET based on a safe mathematical model.

The static WCET analysis tools which are currently available are restricted to a single task running uninterruptedly on a single-core system. There are also extensions of these tools which can capture the effects of multi-tasking, i.e., preemptions by higher-priority tasks, on the WCET for certain well-defined scenarios. These tools are nowadays already used to verify industrial real-time software, e.g., in the automotive and avionics domain. Up to now, there are no mature tools which can handle the case of parallel tasks on a multi-core platform, where the tasks potentially interfere with each other.

This dissertation presents multiple approaches towards a WCET analysis for different types of multi-core systems. They are based upon previous work on the modeling of hardware and program behavior but extend it to the treatment of shared resources like shared caches and shared buses. We present multiple methods of integrating shared bus analysis into the classical WCET analysis framework and show that time-triggered bus arbitration policies can be efficiently analyzed with high precision. In order to get precise WCET estimations for the case of shared caches, we present an efficient analysis of interactions in parallel systems which utilizes timing information to cut down the search space. All of the analyses were implemented in a research C compiler. Extensive evaluations on real-time benchmarks show that they are up to 11.96 times more precise than previous approaches.

Finally, we present two compiler optimizations which are tailored towards the optimization of the WCET of tasks in multi-core systems, namely an evolutionary optimization of shared resource schedules and an instruction scheduling which uses WCET analysis results to optimally place shared resource requests of individual tasks. Experiments show that the two combined optimizations are able to achieve an average WCET reduction of 33%.

During the course of this thesis, a complete WCET analysis framework was developed which can be used for further work like the integration of multi-task and multi-core-aware techniques into a single analyzer.