CSC 714 Real Time Computer Systems Project

Pipeline and Path analysis for Power-aware embedded architecture

-Kiran Seth

1. Problem Overview

In Real-Time systems, it is critical to have guaranteed temporal and logical performance. Correct operation requires task deadlines to be met while maintaining correctness of the operation. As the deadlines of Real-Time system tighten, performance increasing techniques like caches, pipelining, branch prediction and out of order execution are added to embedded processors. These complex pipelines increase the performance of embedded processors, but it is difficult to guarantee the performance of a complex pipeline. The Worst Case Execution Time (WCET) for tasks is required for schedulability analysis and also for creating the schedules. It is very difficult to accurately measure WCET of tasks on a complex pipeline. But for simple pipelines, it is possible to calculate the WCET accurately. Due to the WCET requirements, simple pipelines are preferred in hard Real Time Systems. But simple pipelines cannot show the same performance as a complex pipeline.

Another issue in Real Time systems is the conservation of power. Because of naï ve WCET bounds, significant power is wasted in current Real Time systems. Therefore the required processing speed or clock frequency is higher than required because a deadline cannot be missed. This inflated frequency leads to higher power consumption.

A simple solution to both the problems is to use a dual frequency approach. The embedded processor will usually work at a low frequency with a complex pipeline. It is expected that the actual execution time required is smaller than the WCET. The tasks are now given intermediate deadlines. If the task misses any intermediate deadline, the frequency is increased. At this time, we can also switch the pipeline from a complex to simple pipeline. This means that since we can guarantee the performance of a simple pipeline, we are assured that no deadline will be missed. The low and high frequencies are chosen depending on the WCET of the task and the actual execution time (which can be found by using simulations of the program).

One of the main requirements in this approach is accurate WCET analysis of a program. The goal of this project is to modify existing tools to provide accurate WCET predictions for the SimpleScalar ISA and then to add parametric timing analysis paradigms to the tools.

2. Key Outcomes and Tasks

It important to perform accurate WCET of tasks for simple pipelines. The program is first analyzed using a static instruction cache simulator and a static data cache simulator to calculate the caching potential of all the instructions. The instructions are categorized into always hit, always miss, first hit and first miss. To get an accurate WCET, we need to perform path analysis for the given program and look at the interaction between paths. The path analysis along with the caching categorizations are used to predict the WCET of the program. Overlapping of two operations is also taken into account (like a high latency floating point operation and a cache miss may overlap) thus making the WCET more accurate and less naï ve.

Currently tools are available to perform static instruction cache simulation (work by Dr. Mueller), static data cache simulation (work by Dr. White) and the timing analysis of programs using path analysis and caching categorizations (work by Dr. Healy). The tools take inputs files generated by a research compiler (vpcc/vpo). The timing analyzer also takes input files from the static instruction cache simulator and the static data cache simulator. But we are going to use a modified version of the SimpleScalar simulator (provided by Dr. Rotenberg) to calculate the actual execution times of programs using simulation. This means that all the WCET analysis must also be done for the SimpleScalar ISA (PISA). The static instruction cache simulator is ISA independent. But the timing analyzer currently works with the microSparc ISA. Also, there is no compiler that generates PISA binaries and generates input files for the various tools.

My goal is to reverse engineer the PISA assembly to produce inputs for all the tools (like the vpcc/vpo compiler). After creating a software patch for reverse engineering the assembly, I will work on the Timing Analyzer to change its ISA to PISA and also the internal pipeline it uses for simulation so that it looks and works just like the pipeline in the other simulator. Thus we will be able to analyze programs compiled using the PISA gcc and determine an accurate and tight WCET for the programs.

The main objective is to derive a tool that provides power-aware static timing analysis. The challenge is to devise a concept to perform path analysis in the presence of parametric expressions. We will use the parametric paradigms for representing the execution characteristics in response to frequency scaling. The timing analyzer will also be modified to add power-aware path analysis.

3. Project Homepage

www4.ncsu.edu/~krseth/index.htm

4. References

[1] C. A. Healy, R. D. Arnold, F. Mueller, D. Whalley, and M. G. Harmon. Bounding pipeline and instruction cache performance. *IEEE Transactions on Computer*s, 48(1):53–70, January 1999.

[2] E. Vivancos, C. Healy, F. Mueller and D. Whalley. Parametric Timing Analysis. ACM SIGPLAN Workshop on Languages, Compilers and Tools for Embedded Systems, ACM SIGPLAN Notices, Aug 2001, pages 88-93

[3] F. Mueller. Timing Analysis for Instruction Caches. Real-Time Systems Journal, Vol. 18, No.2/3, May 2000, pages 209-239

[4] R. White, F. Mueller, C. Healy, D. Whalley and M. Harmon. Timing Analysis for Data Caches and Set-Associative Caches. Real-Time Technology and Applications Symposium, Jun 1997, pages 192-202

[5] D. Burger, T. Austin and S. Bennett. Evaluating Future Microprocessors: The Simplescalar Toolset. Technical report CS-TR-96-1308, Computer Sciences Department, University of Wisconsin-Madison, July 1996.

[6] E. Rotenberg. Using variable-Mhz microprocessors to efficiently handle uncertainty in real-time systems. *34th International Symposium on Microarchitectur*e, December 2001.

[7] Thomas Lundqvist. "A WCET Analysis Method for Pipelined Microprocessors with Cache Memories," PhD thesis, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, June 2002.

[8] Jakob Engblom. Effects of Branch Predictors on Execution Time. Dept. of Information Technology Technical Report 2002-013, April 2002.