

CSC 714 Real Time Computer Systems Project Pipeline and Path analysis for Power aware Embedded architecture

-Kiran Seth

Progress Report

The current objective of the project is to use the existing framework for timing analysis and use it with a different ISA. Currently the timing analyzer works with the microsparc architecture and it is being ported to the ISA used by the SimpleScalar toolset. Other new additions to the timing analyzer are:

- A new pipeline, compatible with the one used in the modified SimpleScalar timing simulator.
- Branch prediction using a static prediction technique (currently Ball Larus heuristic).

Solved issues -

From 11/1 to 11/6 --

- 1) A software patch that uses a PISA assembly file as an input and gives input files for the static cache simulator and the timing analyzer is ready.
- 2) The timing analyzer has been modified to work with the SimpleScalar ISA.

From 11/6 to 11/18 --

- 3) Several bugs in the modified timing analyzer were traced and fixed.
- 4) The timing analyzer was used to generate numbers for various embedded systems and real time benchmarks.
- 5) The numbers were compared with results from Dr. Rotenbergs timing simulator and it is verified that that the timing analyzer provides safe bounds.

Next Steps -

- 1) Currently the data cache simulator is offline. The timing predictions can be made tighter if it can be incorporated into the timing analyzer.

Milestones -

- 1) 11/1 – Changed the instruction set required by timing analyzer from Microsparc to PISA. Also changed the pipeline to resemble the pipeline in Dr. Rotenberg's simulator.
- 2) 11/5 – Added branch prediction to the timing analyzer. All changes complete. Timing analyzer now works with PISA, has a new pipeline and uses Ball-Larus heuristic for branch prediction.
- 3) 11/10 – Traced several bugs in the timing analyzer and fixed them.
- 4) 11/16 – Generated numbers for various real time and embedded systems benchmarks.
- 5) 11/17 – Compared results with the timing simulator and confirmed that the timing analyzer comes up with safe bounds to execution time.