Projects for the Energy Aware Computing Course (coursework 1b)
Submission deadline: 4.30pm, Friday 12th November 2008

Aris Efthymiou (aris.efthymiou AT inf.ed.ac.uk)
thanks Polychronis Xekalakis

The aim of the following projects is to give you a first hand experience of the tools and the methodology employed by researchers in the field of low power architectures. You will be using a version of the Simplescalar simulator, enhanced with additional models for dynamic and static power consumption. The applications that you will evaluate your project on, are going to be the SPEC2000 Benchmark Suite, which are the defacto programs used by the research community. The simulator and benchmark traces are available from the course web page.

The projects are meant to be undertaken by groups of two students. You can select any project from the list below and more than one group can select the same project. If you have an idea of your own for the course project, please contact the lecturer as soon as possible and have a good plan of what you want to do. Obviously, collaboration outside the group for the same project is not permitted except for practical help with the simulator and sharing of relevant reading material. However you are strongly encouraged to exchange information about experimentation methodology, simulator issues, ...

To complete the project you will generally have to read the relevant publications, understand the source code of the simulator, modify the simulator to implement the method described by the project, and evaluate the results. The submission package should contain your modified source code and a short report explaining your implementation, describing your results and comparing them with those published, if any. In addition, shortly after the submission deadline, you will have to give a demonstration of your work to the lecturer. Each student should be able to explain in detail the parts that they implemented themselves.

The project will be marked for completeness of the work, detailed modelling of the corresponding technique, explanation of design decisions, robustness of the evaluation methodology and critical assessment of the method and the simulation results.

Project list

1. **Way Prediction**

   Caches typically used in modern processors are set-associative. In these caches, on each cache-access one has to probe all the ways to find the desired cache line. This leads to unnecessary power

consumption, which in this project we will try to minimise. Towards this end, you are required to design a mechanism that will perform way prediction, so as to access only the predicted way and thus save power. Note that misprediction may cost in power or time or both. You are free to design any predictor (or use existing ones) as long as your mechanism can perform a prediction in one cycle.

Recommended reading material:

- K. Inoue, T. Ishihara; K. Murakami, "Way-predicting set-associative cache for high performance and low energy consumption", In Proc of Int. Symp. on Low Power Electronics and Design, pp. 273-275, 1999
- M. Powell et al. "Reducing set-associative cache energy via way-prediction and selective direct-mapping". In Proc. of the Int. Symp. on Microarchitecture, 2001.
- Z. Zhu and X. Zhang. "Access-mode predictions for low-power cache design". IEEE Micro, 22(2), 2002.

2. **Power Efficiency of Cache Replacement Algorithms**

   Recent research suggests that LRU does not perform very well for L2 (or higher) caches and new replacement algorithms based on Re-reference interval prediction (RRIP) have been proposed. For this project you will first quantify how the cache replacement policies (currently the simulator supports LRU, FIFO and random) affect the energy consumption of the cache (both data L1 and unified L2 cache) and the system. Once you have established which of the current replacement methods is the most energy efficient, you will implement an RRIP-based cache and evaluate it.

   Recommended reading material:

   - Hennessy and Patterson, "Computer Architecture: A Quantitative Approach"
   - A. Jaleel, et al., "High performance Cache Replacement Using Re-reference Interval Prediction (RRIP)", International Symposium on Computer Architecture 2010.

3. **Selective Branch-prediction access**

   Branch predictors are one of the most important components of modern processors since they break the control dependencies and allow the out-of-order execution engine to execute uninterrupted. Accurate predictors can save energy as they keep the number of mispeculated instructions to a minimum. At the same time, a predictor expends energy in every cycle as it tries to predict for each fetched instruction whether it is a branch, its direction and its target address.

   For this project you will examine ways of reducing energy in the branch prediction by eliminating (or reducing) useless accesses, either lookups or updates of the prediction arrays, while keeping the prediction accuracy as high as possible. Some branches are easier to predict than others, so they can be handled by a simpler, more energy-efficient mechanism. Similarly some updates are unnecessary as they will not change the information stored in the prediction arrays. Read the paper below, implement and evaluate SEPAS.

   Reading material: Amirali Baniasadi and Andreas Moshovos, "SEPAS: A Highly Accurate Energy-Efficient Branch Predictor", Proc. of the 2004 Intl' Symposium on Low Power Electronics and Design (ISLPED), Aug. 2004.

4. **Fetch Throttling**

   Fetch Throttling refers to the fact that one might reduce, at run-time, the numbers of instructions fetched in a cycle so as to minimise power consumption. Note that if this is done naively, it will probably degrade the processor's energy efficiency. For this project you will have to implement the fetch throttling mechanism in the processor front-end. Once you have developed it, you will then have to find a heuristic that will turn it on or off. The key to dealing with this problem is determining when fetching more instructions does not give significant benefit in performance. This is most likely when the probability of these instructions being useful is low; in other words the fetched instructions are likely to be squashed due to an earlier branch misprediction

   Recommended reading material:

   - Juan L. Aragon, Jose Gonzalez and Antonio Gonzalez "Power-Aware Control Speculation through Selective Throttling" In Proc. of the 9th IEEE International Symposium on High Performance Computer Architecture (HPCA), Anaheim, CA, USA, February 2003
   - David Brooks, Vivek Tiwari, and Margaret Martonosi. "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," 27th International Symposium on Computer Architecture, June, 2000.