PERFORMANCE EVALUATION OF PREDICTION CACHES

Abstract

Caches play an important role in hiding latency between processor and memory. Still processor performance is affected by memory latency during cache misses. Prediction caches create a history of cache misses and predict future cache misses. Based on the predicted future cache misses they prefetch the data or instructions. This helps in improving the processor performance by decreasing the processor memory latency gap.

Planned Tasks

1. Learn how to hack into SimpleScalar simulator.
2. Concept of Stream Buffers and Victim Caches.
3. Performance Evaluation of Stream Buffers and Victim Caches.
4. Implementation of Prediction Caches.
5. Performance Evaluation of Prediction Caches.
6. Identify interfaces to change to introduce our proposed features (victim caches etc)

Tasks Completed

1. Hacking the SimpleScalar simulator

We explored various files in the simulator and their functionality. We figured out that we need to change

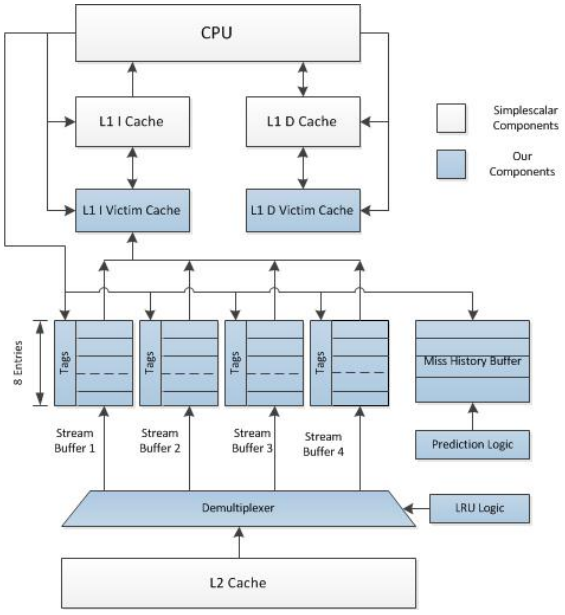
a) cache.h to define the structure of the victim cache and stream buffers. We also need to define block access function which estimates the miss penalty.

b) cache.c to define the functionality of the stream buffers and victim caches.

c) sim-outorder.c to integrate victim cache and stream buffers into the main simulation.

d) Implementation block diagram defined.

2. Rudimentary Block Diagram



Future Tasks

1. Performance Evaluation of Stream Buffers and Victim Caches.
2. Implementation of Prediction Caches.
3. Performance Evaluation of Prediction Caches.
4. Identify interfaces to change to introduce our proposed features (victim caches etc)

References:

1. P. Jouppi, “Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers,” 17th Annual International Symposium on Computer Architecture, pages 364-73, May 1990.
2. S. Palacharla and R.E. Kessler, “Evaluating stream buffers as a secondary cache replacement,” Proc. of the 21st International Symposium on Computer Architecture, pages 24-33, April 1994.
3. V. Phalke and B. Gopinath, “A miss history-based architecture for cache prefetching,” International Workshop IWMM 95 Proceedings, pages 381-98, September 1995.
4. I. Farkas, P. Chow, N. P. Jouppi, and Z. Vranesic, “Memory-system design considerations for dynamically-scheduled processors,” Proc. of the 24th Annual International Symposium on Computer Architecture, pages 133-43, June 1997.