

GEORGIA INSTITUTE OF TECHNOLOGY

College of Computing

CS/EE6760 — Parallel Computer Architecture I

Fall 1998

CS/EE6760
Homework 2

Issued: October 15, 1998
Due: October 22, 1998

Purpose: This homework covers instruction-level parallel machines.

Reading: H&P Chapters 3 and 4
[Fisher91] (*high-level overview, optional but worth reading*)
[Smith88] (*problem 2*)
[Song97b] and [Gwennap96] (*problems 2, 3*)

Problems:

1. ILP, ILP, ILP.
2. Precise Exceptions.
3. Caches in the Real World.

Appendix: State table for Tomasulo's algorithm (also on-line).
Short perspective on state of industry.

Problem 1: ILP in Software, Hardware

The problem is composed of selected parts of Problem 4.14 in the book. For all parts, use the SAXPY code given in Problem 4.14 and the pipeline latencies given in Figure 4.2.

A: Do part **a.**, vanilla DLX.

B: Do part **b.**, loop unrolling for the vanilla DLX.

F: Do part **f**, Tomasulo-style version of the DLX, except assume the floating-point latencies tabulated in Figure 4.2 and the general architecture shown in Figure 4.8 (plus an integer unit). A worksheet of the state registers used in Tomasulo's algorithm for this hardware is attached and also on-line (in case you want to modify it). Note that integer instructions need no reservation stations because they all complete on the cycle issued. Also, there is only one load station because loads are assumed to always complete in one cycle.

G: Do part **g**, unrolling/scheduling for a superscalar DLX.

Problem 2: Precise Exceptions

Early out-of-order execution machines (CDC 6600, IBM 360/91) did not maintain precise exceptions. Precise exceptions make exception handling (e.g. for virtual memory) much cleaner and are effectively required by the IEEE floating point spec.

A: The [Smith88] paper catalogs several ways to achieve precise exceptions, including reorder buffers, history buffers and a “future” file. What makes a future file better than a reorder buffer or a history buffer? (size? speed? design complexity?)

B:

Easy: What technique does the UltraSparc-3 ([Song97b]) use?

Harder: What technique does the 21264 ([Gwennap96]) use?

Optional: What did the Multiflow Trace ([Colwell88]) do?

Problem 3: Memory Systems Revisited

Since you're reading about the UltraSparc-3 and the 21264 anyway and spent a lot of time in Project 1 contemplating on-chip caches, what is the cache structure of those two processors to the extent that you can tell from these papers? I.e. organization, sizes, access time, virtual vs. physical, write policy, replacement policy, any oddball features.