30 Oct 2015 COMP 1203
computer systems
10 am WEEK 5

Jed Gibbs
for
Charlton
Rodda

IA32 (Pentium 6 and above) SPARC → MIPS etc
have a hardware variation on branch predictor
and some speculative

Tanenbaum 6th ed chapter 2

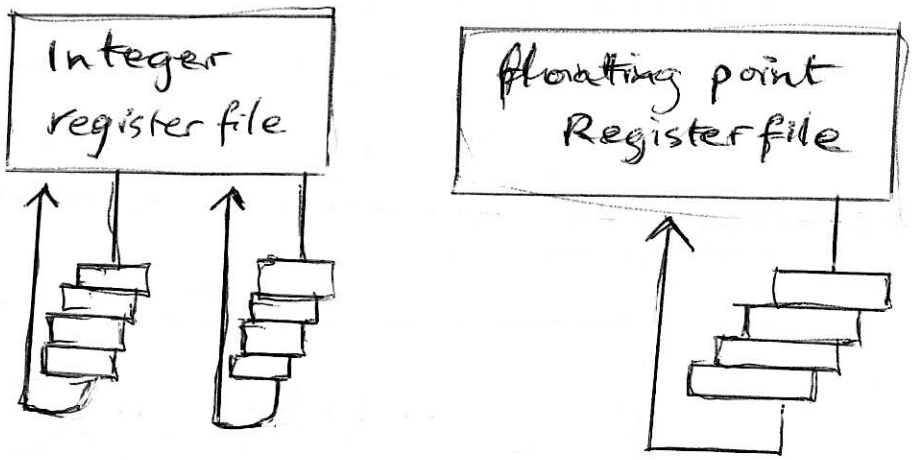
Computer Architecture

Instruction Level Parallelism

SuperScaler - common instructions can be initted independently. Works on CISC and RISC but in practice usually RISC.

Pipelines running parallel - lots of arithmetic units, operands.

Separate Integer from floating point



Instruction Level Parallism

If no dependency between instructions — R1 = 3
can be run parallel. — R2 = 4

BUT if R2 = R1 + 1 then must wait for result of (true)
R1 and so cannot process in parallel = data dependency

Compiler based optimization — the compiler can optimise your code to suit a simpler cpu by changing the order of instructions, assigning registers to values etc.

Procedural Dependency - cannot calculate a value until previous instructions complete.

Resource Conflict — if there are insufficient registers, floating point calculators or comparators this will prevent parallel processing due to bottleneck.

Design Issues:

Instruction Level - software designed so that the order of instructions etc. are optimised to process independently without as dependencies

In Order Issue

Out of Order Completion - if possible without dependencies then the various instructions can complete in their own best times. However beware dependencies which may go wrong if they are actioned before they receive data in the correct order!

30 Oct 2015 Friday
Computer Systems
10am

Jed Gibbs

Antidependency

write-write dependency

To prevent pipeline stalls can employ register renaming to store values temporarily until the order of instructions is ready for the value.

Machine Parallisms

Improving branch predictions, extra pipelines/registers/

The crucial thing is that results are correct whatever techniques are used to speed the process up. See last 2 slides!