ECE 511: Computer Architecture October 25, 2004

Lecture 16

Lecturer: Matt Frank *Scribe: Agnes Lo*

ILP HISTORY AND PROSPECTS

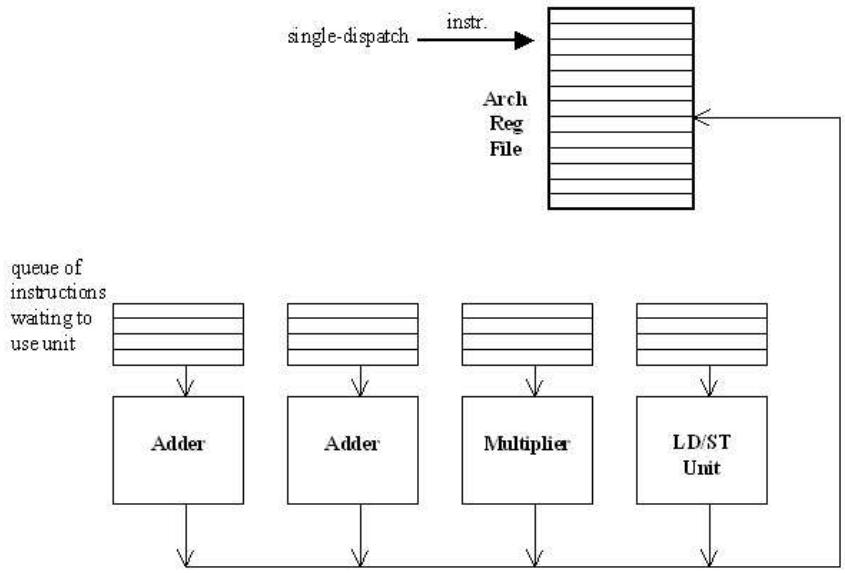
Checkpoint Repair (doing renaming and having a reorder buffer):
(Checkpoint repair is really neat!)
Two directions:
1. Historical perspective: Why is checkpoint repair important?
   - Tomasulo did renaming, but he didn't have a ROB, so he couldn't speculate across branches
2. The new: The way that we do it now is the state of the art in industry.
   - What are people doing today to take advantage of checkpoint repair?

A Little History
CDC 6600 - c. 1965
(Control Data Corporation, a processor manufacturer in Minneapolis)
- Jim Thornton
- had out-of-order execution, scoreboarding, but no renaming
Cray machines:
- defined own instruction set (similar to the one in class)
- arithmetic, LD/ST, BR are separate
- didn't want CISC-like instructions; wanted to keep things simple for scoreboarding, so they could get out-of-order execution working
- the first RISC machine
- also didn't have to deal with a huge legacy code base

IBM 360 line
- line of computers that all had the same instruction set
- hugely original idea
- a program written in assembly would run on all machines!
- the difference between them was how much hardware support there was to make them fast

IBM 360/90 Tomasulo et al c. 1967
- out-of-order issue and renaming
- no checkpoint repair (no branch speculation)
- the floating point part of the instruction set was more RISC-like than the rest of the instruction set
- 4 floating point registers
- take code that you cared about and unroll it (do branch prediction yourself)

Distributed Execution Unit



- instruction comes in
- puts itself in the correct queue
- when the Destination Register is ready, it is sent to the architectural register file
- the WB bus is snooped by the other units to check for certain physical tags
  (similar to wakeup/select, but spread out between 4 units)
- note that the same physical register value can be in a few different places
  (this can be nice since all the operands are easily accessible)
- the 4 queues act as the scoreboard and also the physical register file
- unlike our machine, this is single-dispatch, multiple-issue
- this only gave benefit for FP-intensive work

IBM 360/85
- first machine to have a data cache
- most people bought this instead of the 360/90 if they wanted high performance at a lower cost

1970's
- not much going on
- people didn't really understand the issues
- no one knew how to get around the issue of branch speculation

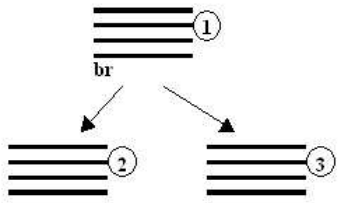
c. 1970
- Tjaden and Flynn
- Mike Flynn did performance evaluations on a bunch of programs and came to the conclusion that there was no ILP in the programs
- "Flynn bottleneck" - "There's nothing to look for, so don't look for it."

1972
- Riseman and Foster tried to measure why the Flynn bottleneck was happening
- conclusion: (conventional wisdom) Doomed!

1975
- Richard Keller pointed out that there is a difference between scheduling and renaming
- renaming gets rid of anti- and output-dependencies

c. 1979
- Josh Fisher (student at NYU) is a microcoder who asked himself, "How do I write high performance microcode?"
- he discovered that he was speculating across branches

Speculating across branches
- write extra code to fix mess-ups



If you know that the branch usually goes to 3, it's fine to move 3 up to 1 if the destination is not used (it can be overwritten later).

What he was doing was profiling, finding easy-to-predict branches
2 techniques:
1. moving an instruction up
2. moving an instruction down to both paths
If things go badly, code fixes it up.
This is trace scheduling.

Fisher invented the concept of VLIW processors and started building them.

Josh Fisher - Multiflow Computer Inc. - built VLIW mini-supercomputers
Bob Rau - Cyndrome Inc. - also built VLIW mini-supercomputers
Both are cheaper than Cray machines.

mid-80's
c. 1985
- Wen-Mei Hwu, Yale Patt, Michael Shebanow
- described checkpoint repair in a processor for the first time
- architecture: HPS
- first machine that had some sort of reorder buffer (renaming)
- used a renaming mechanism later used by Alpha 21264
- checkpoints every branch speculation; has multiple RATs, RAT silo
- at the time, people thought this to be impractical

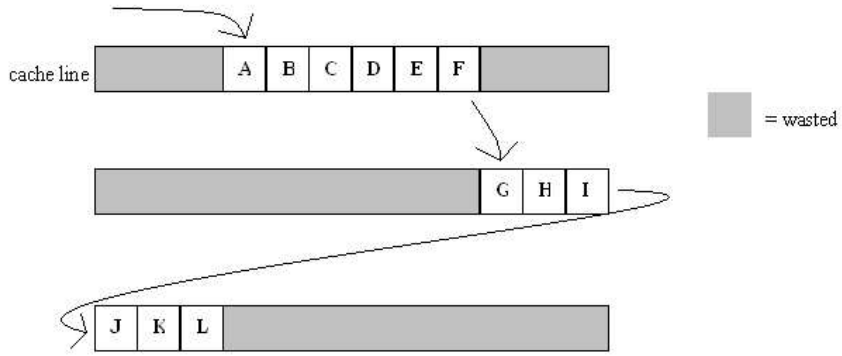
1986
- Jim Smith and Andrew Pleszkun
- proposed what today is the ROB and examined different locations for register files
- i.e. history file, future file

1987
- Guri Sohi
- register update unit (RUU)

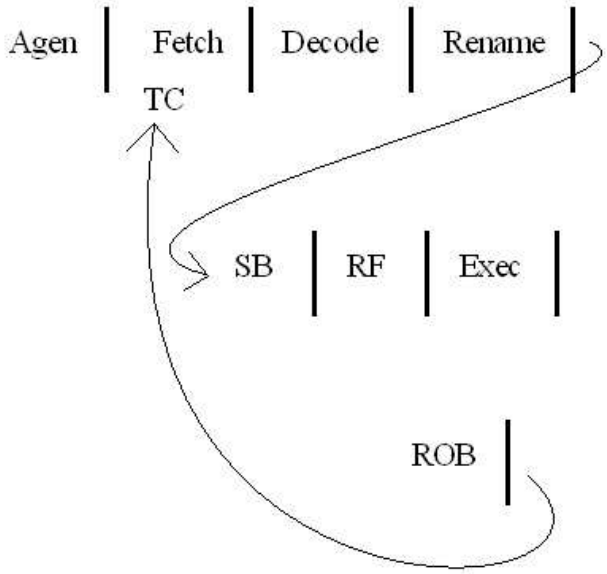
1994
- MIPS R10000: used Smith and Pleszkun's method
- Alpha 21264: used Hwu and Patt's method
- they understood that checkpoint repair only works as well as your branch predictor...

Early 90's
- Yale Patt and Jim Smith
- initial descriptions of high performance branch predictors
- at the time, FP boundary conditions were left to software (trap) since there wasn't enough silicon area to handle it in hardware
- Smith/Pleszkun/Sohi looked at this to do FP exceptions out of order

TRACE CACHING
- original proposal: fix problem of basic blocks not fitting evenly into a cache line
- generate address of basic block; everything done simultaneously in basic block



- only 1 fetch per cycle (limits throughput)
- basic blocks got split between cache lines



- since stream out of ROB is predictable, put it in an alternative set of cache lines
- first time we execute, we go through the normal path
- next time, use trace cache to get high throughput

This leads to higher throughput through fetch stage.

A	B	C	D	E	F	G	H	I	J	K	L

The Pentium 4 implements a trace cache since the bandwidth through decode is important. All Pentium-style processors translate a CISC instruction into a sequence of RISC instructions (stored in trace cache) in decode. This is good because it saves power, area, and time since you don't need as much decode bandwidth (and decode is no longer in critical path).