CS152
Computer Architecture and Engineering
Lecture 14

Pipelining Control Continued
Introduction to Advanced Pipelining

October 19, 2001

John Kubiatowicz (http.cs.berkeley.edu/~kubitron)

lecture slides: http://www-inst.eecs.berkeley.edu/~cs152/

Recap: Summary of Pipelining Basics

- 5 stages:
  - Fetch: Fetch instruction from memory
  - Decode: get register values and decode control information
  - Execute: Execute arithmetic operations/calculate addresses
  - Memory: Do memory ops (load or store)
  - Writeback: Write results back to registers (I.e. COMMIT)
- Pipelines pass control information down the pipe just as data moves down pipe
- Forwarding/Stalls handled by local control
- Balancing length of instructions makes pipelining much smoother
- Increasing length of pipe increases impact of hazards; pipelining helps instruction bandwidth, not latency

Recap: Can pipelining get us into trouble?

- Yes: Pipeline Hazards
  - structural hazards: attempt to use the same resource two different ways at the same time
    - E.g., combined washer/dryer would be a structural hazard or folder busy doing something else (watching TV)
  - data hazards: attempt to use item before it is ready
    - E.g., one sock of pair in dryer and one in washer; can't fold until get sock from washer through dryer
    - instruction depends on result of prior instruction still in the pipeline
  - control hazards: attempt to make a decision before condition is evaulated
    - E.g., washing football uniforms and need to get proper detergent level; need to see after dryer before next load in
    - branch instructions

- Can always resolve hazards by waiting
  - pipeline control must detect the hazard
  - take action (or delay action) to resolve hazards