Quiz 2 Review

Guowei Zhang

Quiz 2 logistics

- Time: 1pm on Friday April 10
  - Prepare to receive an emailed PDF about 10 minutes before the quiz
- Zoom link: same as recitations
- Handout: released soon

Hazards

- Structural hazards
- Data hazards
- Control hazards
  - Not just branches and jumps!
  - Typically resolved by speculation (eager vs. lazy)

Complex pipelining

- Scoreboard
  - A data structure that detects hazards dynamically
  - Needed because
    - Many execution units
    - Variable execution latency
    - Dynamic instruction scheduling
  - Orthogonal to in-order vs. out-of-order issue

Out-of-order issue

- Strategy: find something else to do

- Difference from in-order issue
  - More hazards to consider (e.g., WAR and control)

- Techniques typically combined with OOO issue
  - Register renaming
    - Critical since it reduces/eliminates WAR and WAW hazards
  - In-order commit
    - Critical since it simplifies speculative execution
    - Speculation requires per-instruction buffering/logging
      - Partial flush is critical
      - Circular buffer management is preferred

OOO design tradeoffs

- Implementations
  - Data-in-ROB
  - Unified-register-file
  - More!

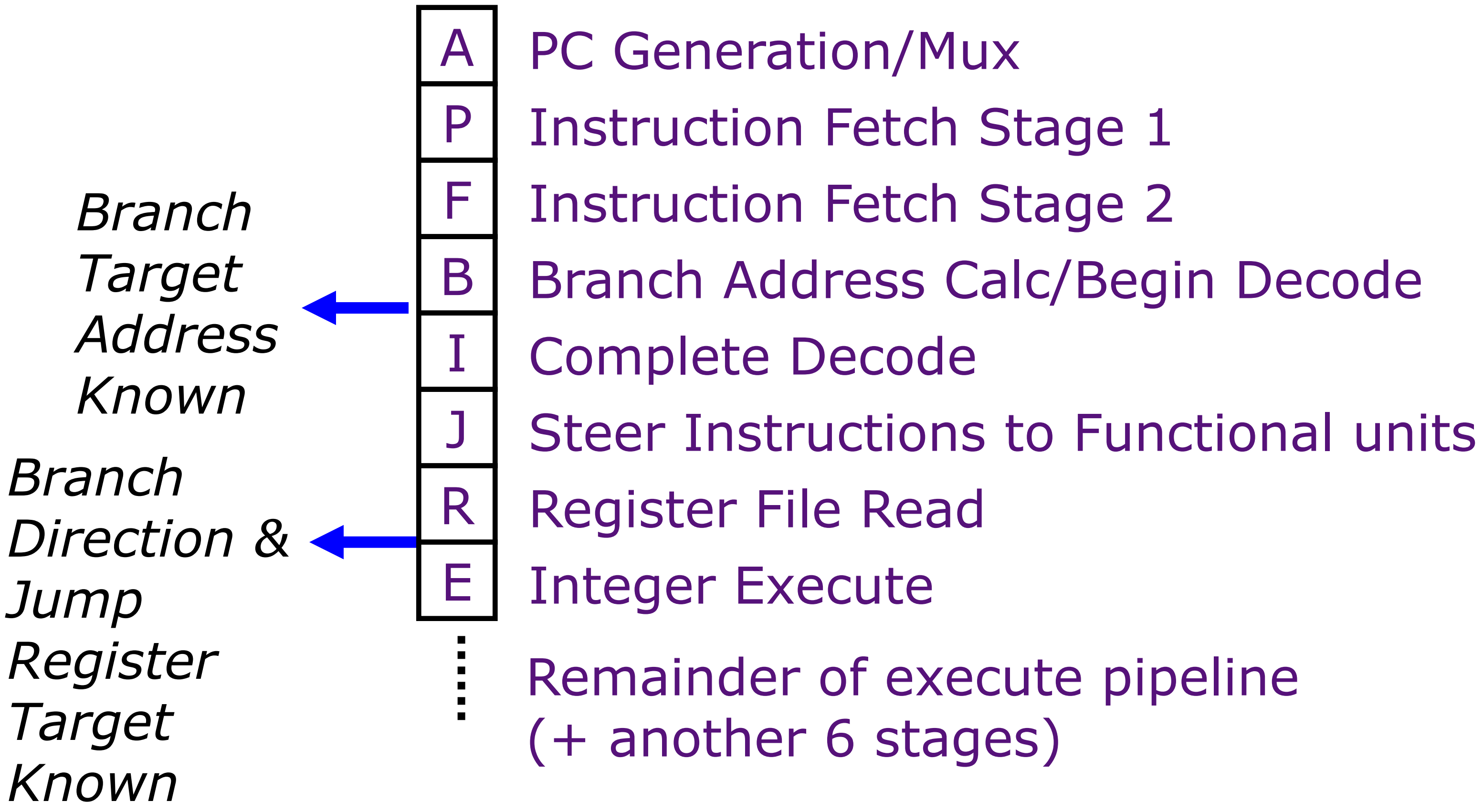
- Tradeoffs
  - Are pointers or values in ROB? Are register reads delayed or immediate?
  - Can speculative values share resources with non-speculative values?
  - Centralized ROB vs. reservation stations
  - ROB vs. issue queue + commit queue

Little’s Law

Throughput (T) = Number in Flight (N) / Latency (L)

Branch prediction

- To reduce the control flow penalty



A	PC Generation/Mux
P	Instruction Fetch Stage 1
F	Instruction Fetch Stage 2
B	Branch Address Calc/Begin Decode
I	Complete Decode
J	Steer Instructions to Functional units
R	Register File Read
E	Integer Execute
⋮	Remainder of execute pipeline (+ another 6 stages)

Branch Target Address Known

Branch Direction & Jump Register Target Known

Branch prediction implementation

- Static vs. dynamic predictor

- Example: two-level branch predictor
  - Access a local/global history in the first level
  - Access a counter in the second level (with or without bits from PC)

- Branch target buffer
- Subroutine return stack
- …

Advanced memory operations

- Write policy
  - Hits: write through vs. write back
  - Misses: write allocate vs. write no allocate

- Speculative loads/stores
  - Cause 1: control dependency
    - Just like other instructions
    - Solution: buffer the stores and commit them in order
  - Cause 2: (memory-location-based) data dependency
    - Simple solution: buffer stores; loads search addresses of all previous stores
    - Problem: addresses of previous stores may be unknown
    - Solution: speculate no data dependency
      - Use a data structure to keep track of this speculation: speculative load buffer

Advanced memory operations

- Prefetching vs. on-demand data movement

Multithreading

- Fine-grain multithreading

- Coarse-grain multithreading

- Simultaneous multithreading
  - Scheduling policies
    - Round-robin
    - ICOUNT

Wish you all the best!