Lecture 18: Pipelining

- Today's topics:
 - Hazards and instruction scheduling
 - Branch prediction
 - Out-of-order execution
- Reminder:
 - Assignment 7 will be posted later today

- Example: a unified instruction and data cache → stage 4 (MEM) and stage 1 (IF) can never coincide
- The later instruction and all its successors are delayed until a cycle is found when the resource is free → these are pipeline bubbles
- Structural hazards are easy to eliminate increase the number of resources (for example, implement a separate instruction and data cache)

Data Hazards



Bypassing



• Some data hazard stalls can be eliminated: bypassing

Example



Example



Example



Control Hazards

- Simple techniques to handle control hazard stalls:
 - for every branch, introduce a stall cycle (note: every 6th instruction is a branch!)
 - assume the branch is not taken and start fetching the next instruction – if the branch is taken, need hardware to cancel the effect of the wrong-path instruction
 - fetch the next instruction (branch delay slot) and execute it anyway – if the instruction turns out to be on the correct path, useful work was done – if the instruction turns out to be on the wrong path, hopefully program state is not lost

Branch Delay Slots



Pipeline without Branch Predictor



Pipeline with Branch Predictor



Bimodal Predictor



- For each branch, maintain a 2-bit saturating counter: if the branch is taken: counter = min(3,counter+1) if the branch is not taken: counter = max(0,counter-1) ... sound familiar?
- If (counter >= 2), predict taken, else predict not taken
- The counter attempts to capture the common case for each branch

- Perfect pipelining with no hazards → an instruction completes every cycle (total cycles ~ num instructions)
 → speedup = increase in clock speed = num pipeline stages
- With hazards and stalls, some cycles (= stall time) go by during which no instruction completes, and then the stalled instruction completes
- Total cycles = number of instructions + stall cycles

Multicycle Instructions



© 2003 Elsevier Science (USA). All rights reserved.

- Multiple parallel pipelines each pipeline can have a different number of stages
- Instructions can now complete out of order must make sure that writes to a register happen in the correct order

An Out-of-Order Processor Implementation



Title

• Bullet