548 - Pipelining

Lecture 3
Pipelining?

- Where: Breaking computation down into smaller pieces that can execute concurrently with other pieces of other instructions
- Where: front-end of processor, execution, retirement; the memory system
What is a basic processor pipeline?

- fetch, decode, reg, exec, mem, writeback
Pros?

• Utilize hardware more efficiently
• Increases throughput / Increases instruction level parallelism
• Design compartmentalization
Cons?

- Increases overall complexity
  - Hazards: forwarding, stall (WaW, ...)
- Increases size
- Forwarding networks, hazard detection,
- Increases latency
  - Branch misspeculation penalty
-
Complexities?

• Can be difficult to split the logic (design complexity)
• Rollback (miss-speculation)
• Speculative scheduling
“Limits”?  

- Fixed cost: latch overhead (area, power)  
- Loops : “Lose Loops Sink Chips”  
  - Load-use, branch resolution, high-latency ops  
- Balancing the whole-system design