What is pipelining?

- Assembly organization of processing
- Increase utilization of silicon via ILP

Basic pipeline

- Fetch
- Decode (register read)
- Execute
- Memory
- Write-back

Hazards

- Resource conflict
  - Structural
- Data hazard
  - RAW, WAW, (not RAR), WAR
- Control
- Interrupts / Exceptions
- One solution stalling, in practice:
  - Add more hardware!

How to fix hazards?

What about VLIW/EPIC?

What about Vector?

What about n stage pipelines?

Why is pipelining good?

- Impoves efficiency of silicon use
- Improves throughput
- Parallelism
- Improves performance from a faster clock

Why is pipelining bad?

- Overhead
  - More power
  - More area
- Maximum latency for an op increases
- Increases complexity

What is the limit of pipelining?

- An *intrinsic* limit
- Complexity vs. Performance
- Delay of pipeline registers (2 gate delays) versus pipeline stage logic
- Clock distribution
- *Intrinsic* characteristics of applications
  - Data dependencies
  - Control speculation

What about branches?

What to do on a cache miss?

What to do on a TLB miss?