Ch 21

Improving 5-stage Pipeline Performance

- Lower t_{CLK}: deeper pipelines
  - Overlap more instructions
- Higher CPI_{ideal}: wider pipelines
  - Each pipeline stage processes multiple instructions
- Lower CPI_{stall}: out-of-order execution
  - Execute each instruction as soon as its source operands are available
- Balance conflicting goals
  - Deeper & wider pipelines ⇒ more control hazards
  - Branch prediction

- It all works because of instruction-level parallelism (ILP)

Limits To Single-Processor Performance

- Pipeline depth: getting close to pipelining limits
  - Clocking overheads, CPI degradation
- Branch prediction & memory latency limit the practical benefits of out-of-order execution
- Power grows superlinearly with higher frequency & more OoO logic
- Extreme design complexity

- Limited ILP → Must exploit DLP and TLP
  - Data-Level Parallelism: Vector extensions, GPUs
  - Thread-Level Parallelism: Multiple threads and cores

Amdahl's Law

$$S_{overall} = \frac{1}{((1-f) + (\frac{f}{S}))}$$

f: fraction of a task that is sped up

S: speed up factor