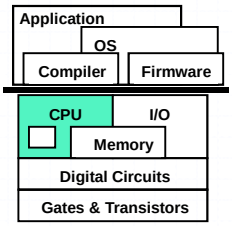
ECE 152
Introduction to Computer Architecture
Pipelining

Duke University

Slides are derived from work by
Amir Roth (U. Penn)
Spring 2011

This Unit: Pipelining



- Basic Pipelining
  - Pipeline control
- Data Hazards
  - Software interlocks and scheduling
  - Hardware interlocks and stalling
  - Bypassing
- Control Hazards
  - Fast and delayed branches
  - Branch prediction
- Multi-cycle operations
- Exceptions

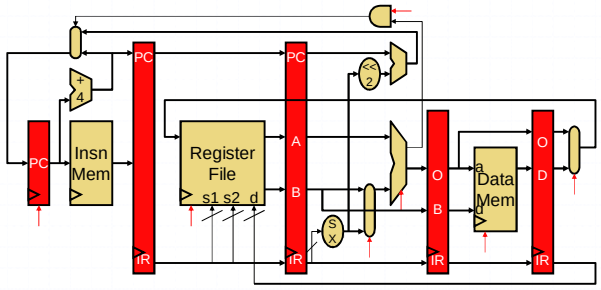
Readings

- P+H
  - Chapter 4: Section 4.5-end of Chapter 4

Pipelining

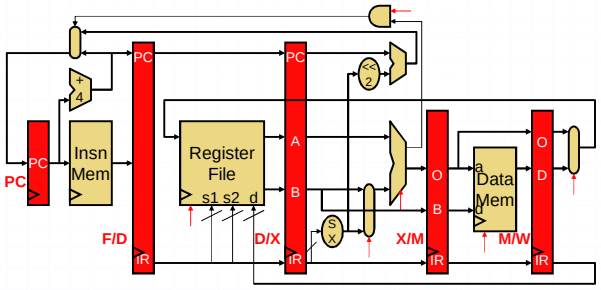
- Important performance technique
  - **Improves insn throughput (rather than insn latency)**
- Laundry / SubWay analogy
- Basic idea: divide instruction's "work" into stages
  - When insn advances from stage 1 to 2
  - Allow next insn to enter stage 1
  - Etc.
- Key idea: each instruction does same amount of work as before
  - + **But insns enter and leave at a much faster rate**

5 Stage Pipelined Datapath



- Temporary values (PC,IR,A,B,O,D) re-latched every stage
  - Why? 5 insns may be in pipeline at once, they share a single PC?
  - Notice, PC not re-latched after ALU stage (why not?)

Pipeline Terminology



- Five stage: **F**etch, **D**ecode, e**X**ecute, **M**emory, **W**riteback
  - Latches (pipeline registers) named by stages they separate
    - **PC**, **F/D**, **D/X**, **X/M**, **M/W**

Aside: Not All Pipelines Have 5 Stages

- H&P textbook uses well-known 5-stage pipe != all pipes have 5 stages
- Some examples
  - OpenRISC 1200: 4 stages
  - Sun UltraSPARC T1/T2 (Niagara/Niagara2): 6/8 stages
  - AMD Athlon: 10 stages
  - Pentium 4: 20 stages
- ICQ: why does Pentium 4 have so many stages?
- ICQ: how can you possibly break "work" to do single insn into that many stages?
- Moral of the story: in ECE 152, we focus on H&P 5-stage pipe, but don't forget that this is just one example