



CS 152 Computer Architecture and Engineering

Lecture 4 - Pipelining

Krste Asanovic

Electrical Engineering and Computer Sciences
University of California at Berkeley

<http://www.eecs.berkeley.edu/~krste>

<http://inst.eecs.berkeley.edu/~cs152>

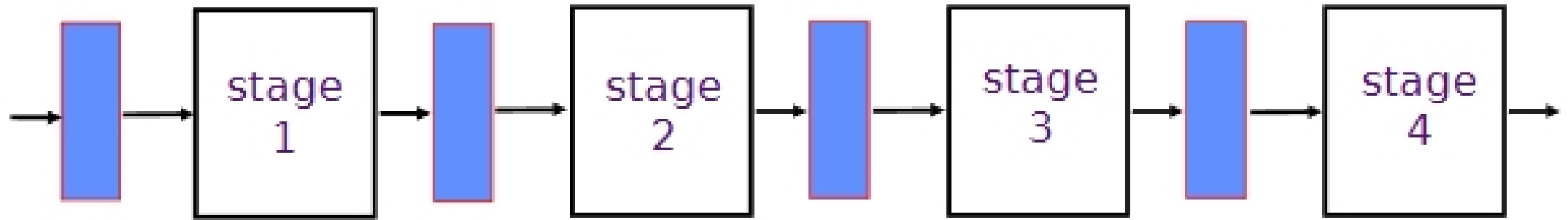


Last time in Lecture 3

- Microcoding became less attractive as gap between RAM and ROM speeds reduced
- Complex instruction sets difficult to pipeline, so difficult to increase performance as gate count grew
- Iron-law explains architecture design space
 - Trade instructions/program, cycles/instruction, and time/cycle
- Load-Store RISC ISAs designed for efficient pipelined implementations
 - Very similar to vertical microcode
 - Inspired by earlier Cray machines (more on these later)



An Ideal Pipeline



- All objects go through the same stages
- No sharing of resources between any two stages
- Propagation delay through all pipeline stages is equal
- The scheduling of an object entering the pipeline is not affected by the objects in other stages

These conditions generally hold for industrial assembly lines, but instructions depend on each other!