Lecture 9: Datapaths and Pipelining

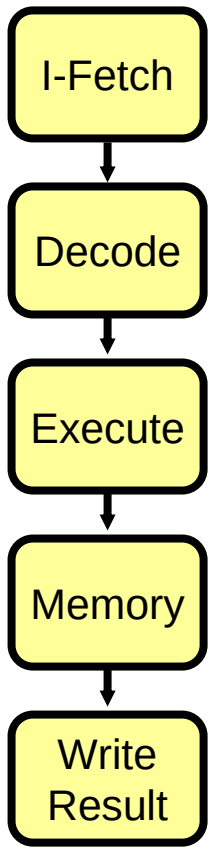
- Logistics
  - Homework #2 due
  - Homework #3 will be posted later today – due next Thur.
- Last Time
  - RISC vs. CISC + discussion
  - Carry-lookahead addition, shift, multiplication
- Today
  - Datapath organization
  - Introduction to pipelining

Observations on Multiplication

- Can speed up algorithm by doing 2 bits at a time, instead of just one
  - How????
  - See Booth encoding strategy in chapter 4
- Multiplication algorithm is still slow
  - Each step is doing a full carry-propagate add
  - Can use carry save adders instead
  - Build circuit to combine the partial products
- Division algorithm is very similar to multiplication
  - See section 4.7

Instruction Execution

- 5 basic steps
  - fetch instruction (F)
  - decode instruction/read registers (R)
  - execute (X)
  - access memory (M)
  - store result (W)



Change to paper lecture media…
[see PDF file for Part 2 of lecture]

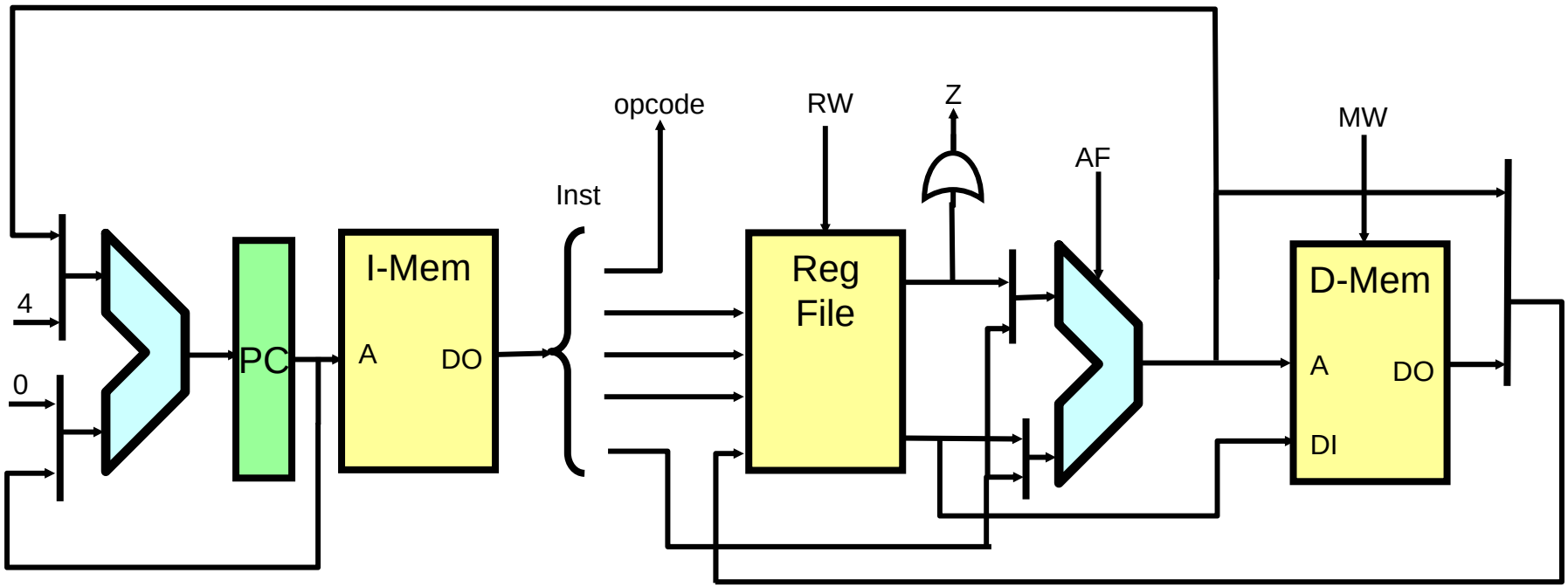
(Datapath Microarchitecture)

Pipelining

Optimizing the Processor

No shared resources

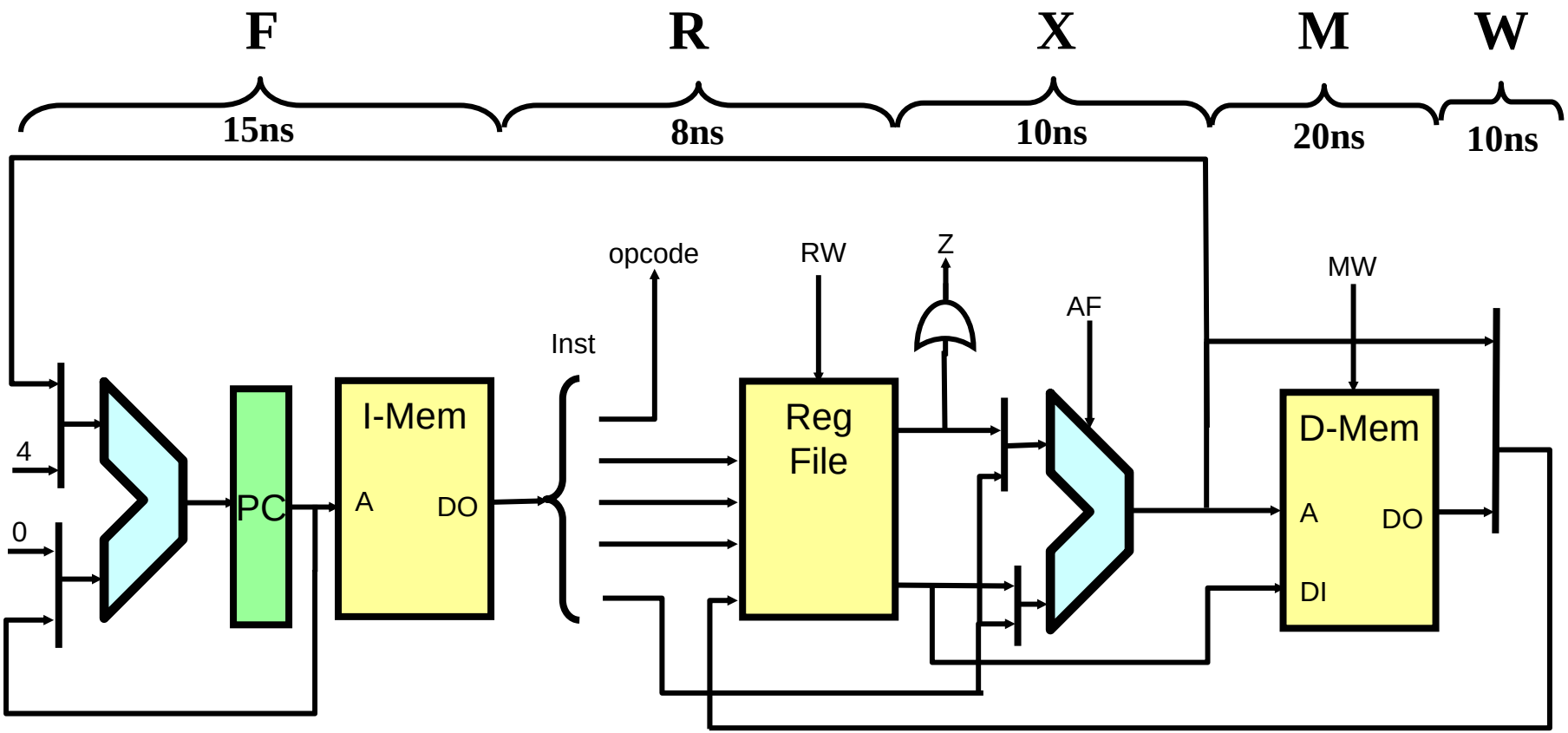
No hidden state



Interestingly – we can get rid of most of the datapath flip-flops
"Single-cycle execution"

Datapath Timing

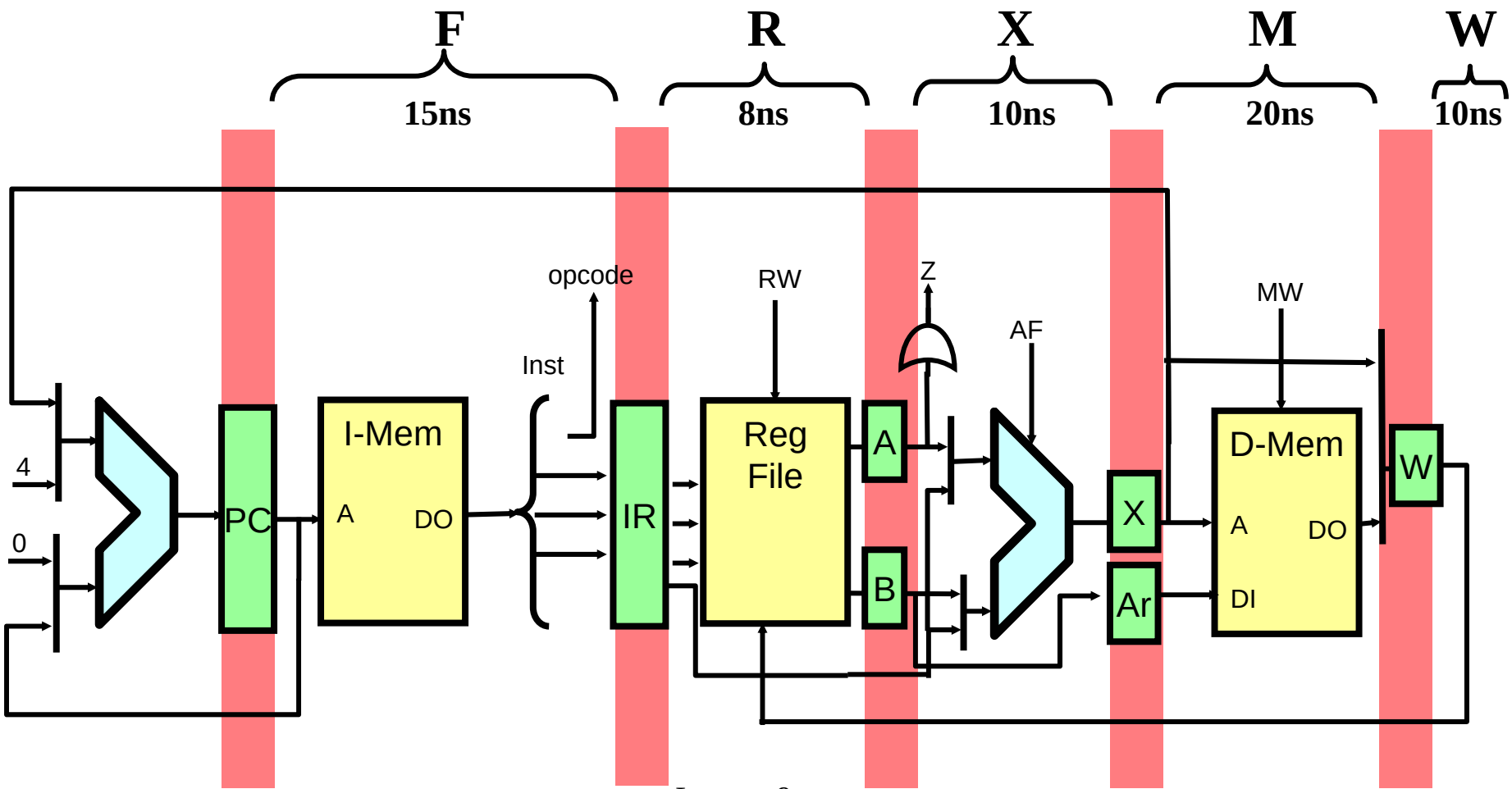
Instruction execution time = ? Clock frequency = ?



Can Re-insert Datapath Flip-Flops

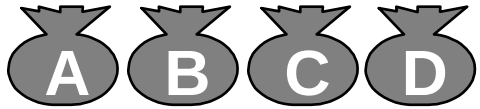
"Multicycle Execution"

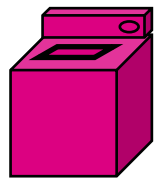
Instruction execution time = ? Clock frequency = ?

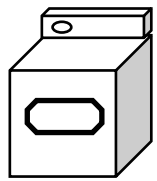


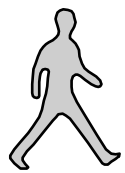
Pipelining is Natural!

- Laundry Example
- Ann, Brian, Cathy, Dave each have one load of clothes to wash, dry, and fold
- Washer takes 30 minutes
- Dryer takes 30 minutes
- "Folder" takes 30 minutes
- "Stasher" takes 30 minutes to put clothes into drawers

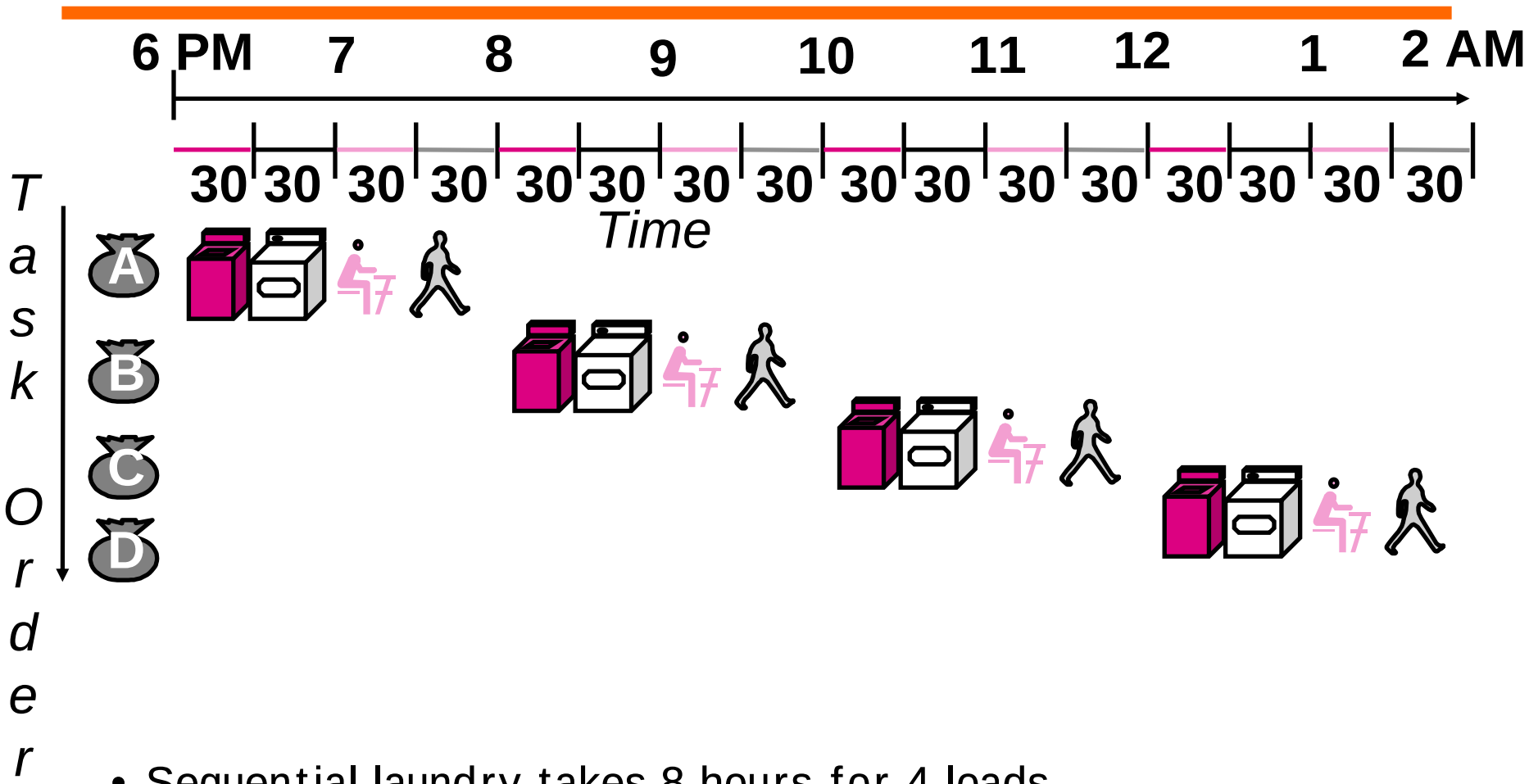






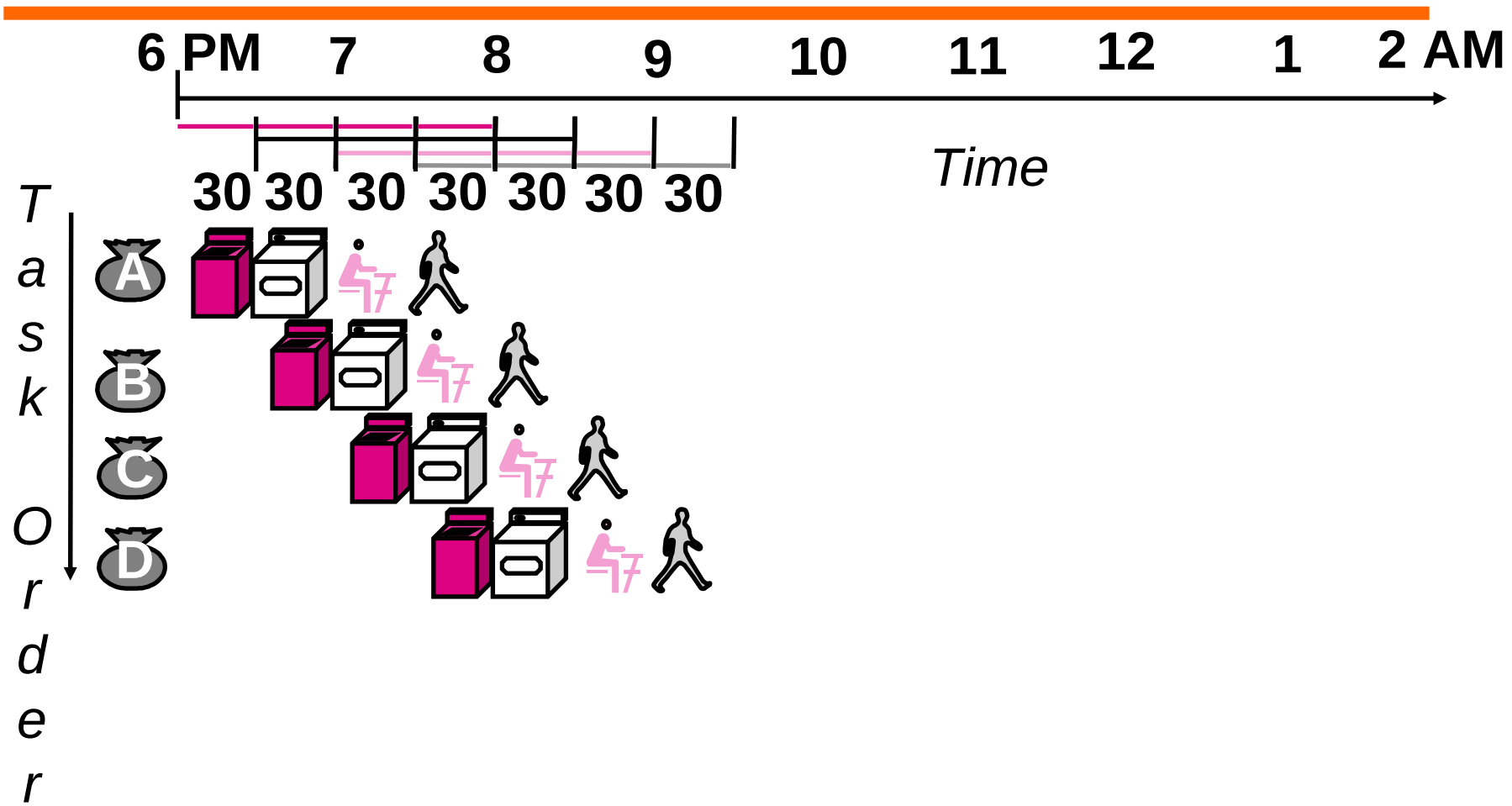


Sequential Laundry



- Sequential laundry takes 8 hours for 4 loads
- If they learned pipelining, how long would laundry take?

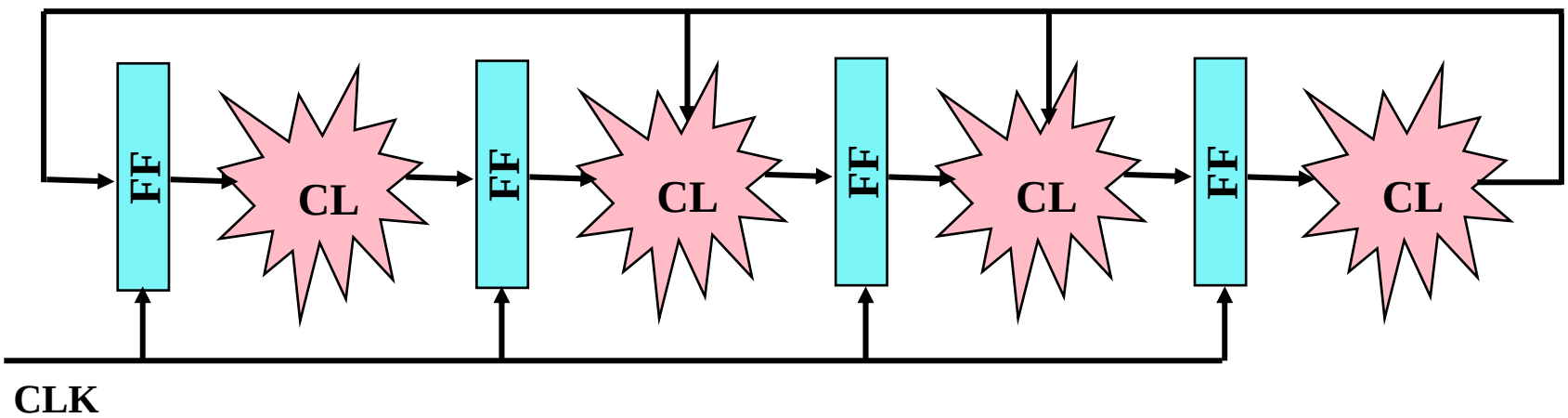
Pipelined Laundry: Start work ASAP



- Pipelined laundry takes 3.5 hours for 4 loads!

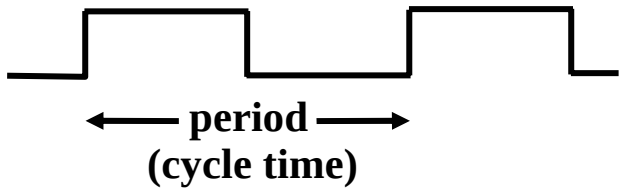
Sequential Logic

- State elements
  - flip flops, registers, memories
- Common clock
- Separated by combination logic
- Feedback allowed

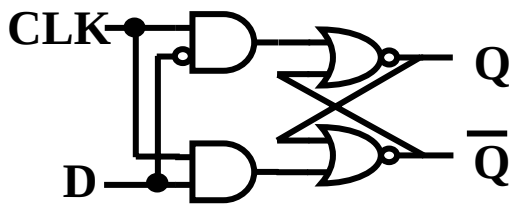


Clocking and Clocked Elements

- Typical Clock
  - 1Hz = 1 cycle per second

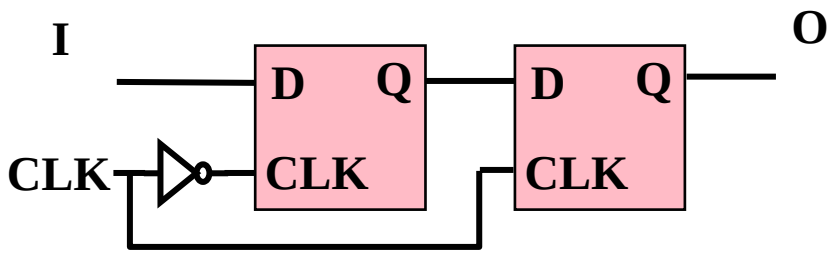


- Transparent Latch

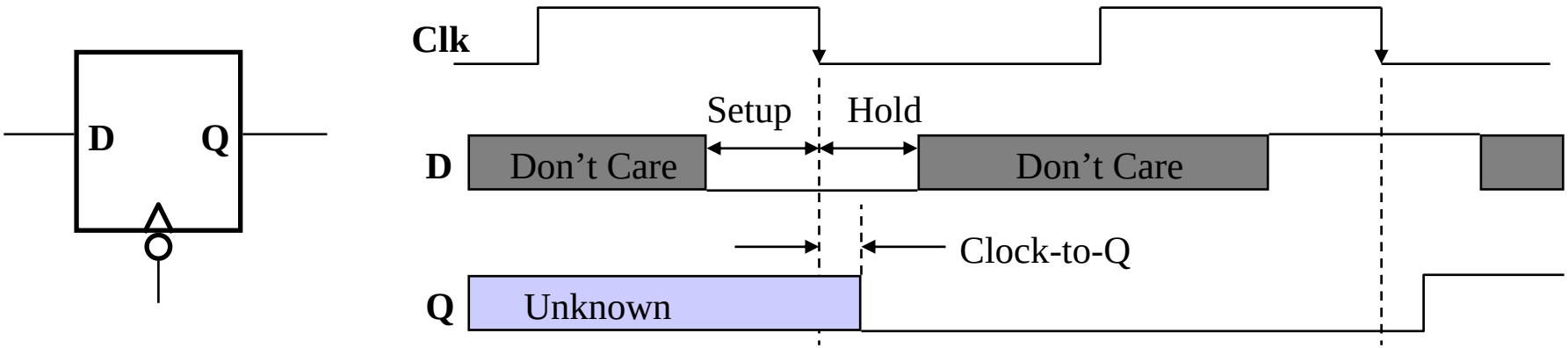


CLK=0, Q=oldQ
CLK=1, Q=D

- Edge Triggered Flip-Flop

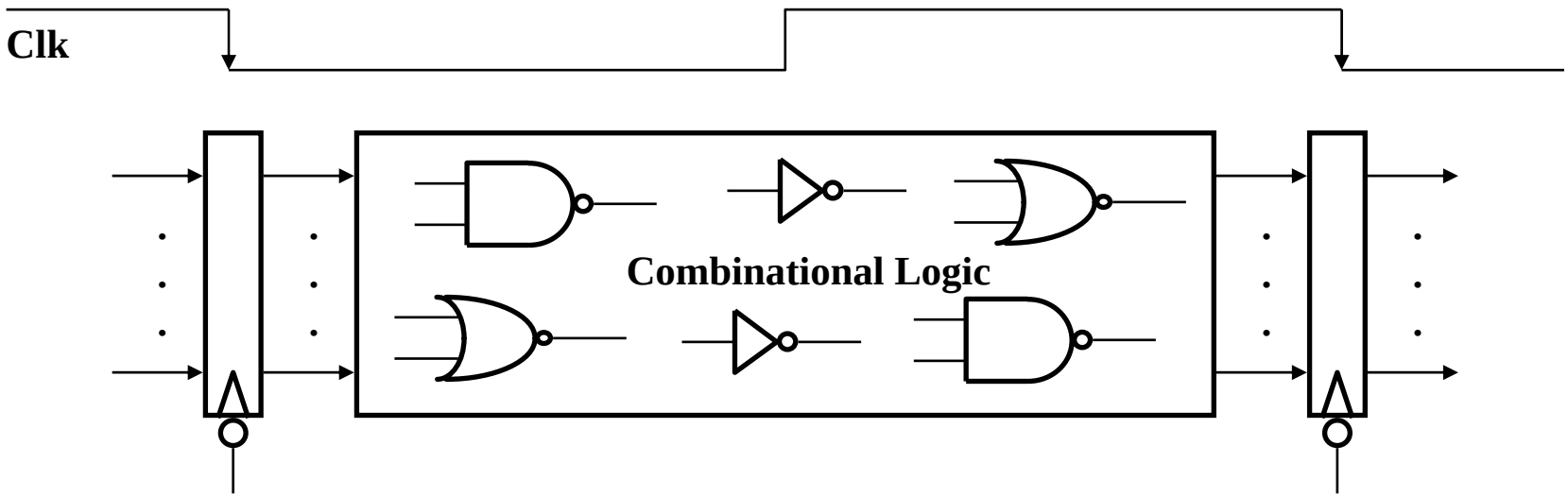


Storage Element's Timing Model



- Setup Time: Input must be stable BEFORE the trigger clock edge
- Hold Time: Input must REMAIN stable after the trigger clock edge
- Clock-to-Q time:
  - Output cannot change instantaneously at the trigger clock edge
  - Similar to delay in logic gates, two components:
    - Internal Clock-to-Q
    - Load dependent Clock-to-Q

Clocking Methodology

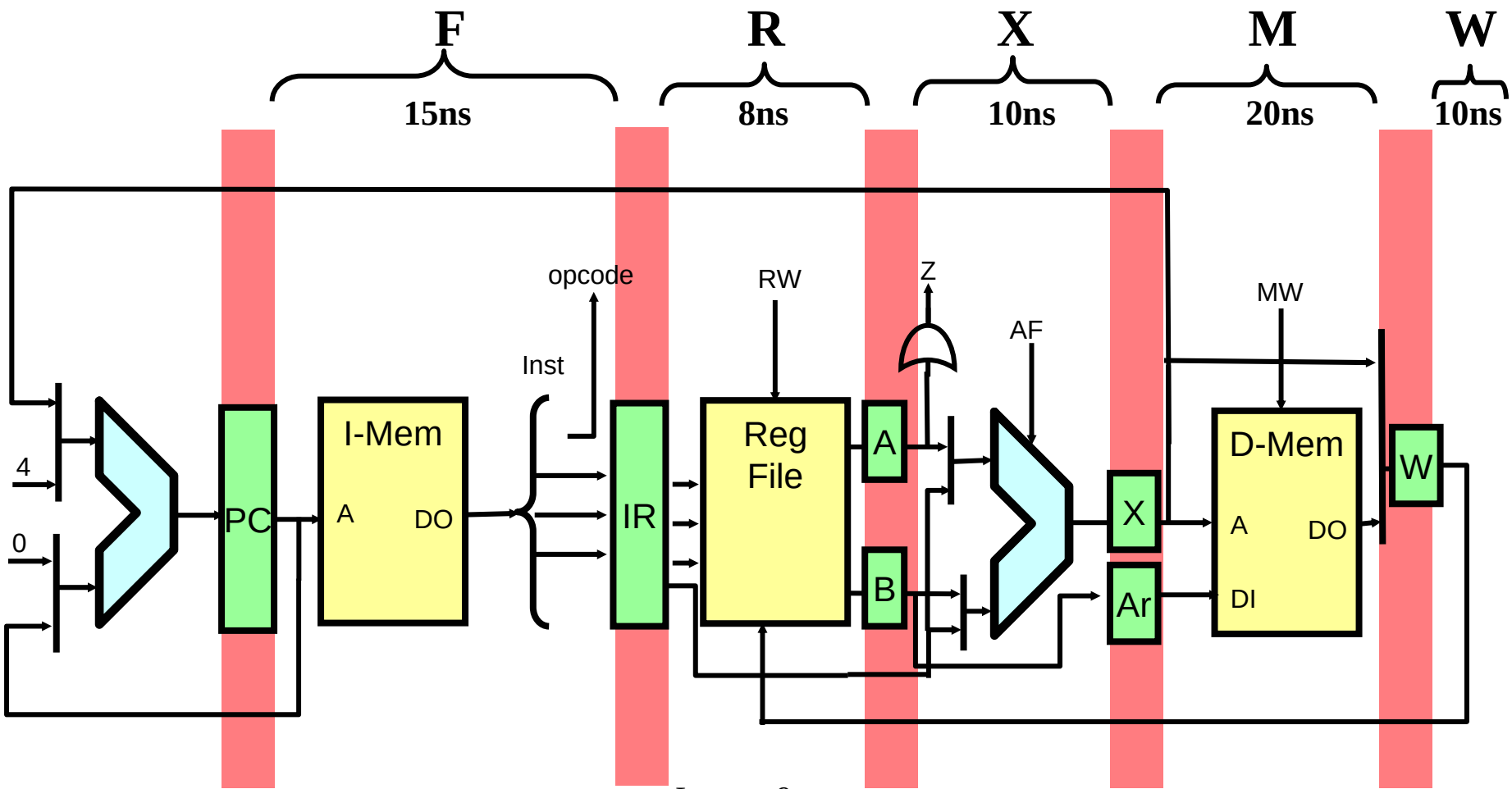


- All storage elements are clocked by the same clock edge
- The combination logic block's:
  - Inputs are updated at each clock tick
  - All outputs MUST be stable before the next clock tick

Can Re-insert Datapath Flip-Flops

"Multicycle Execution"

Instruction execution time = ? Clock frequency = ?



Summary

- Datapath organization
- Control: hardwired and microcode
- Datapath optimization
- Pipelining introduction

- Next Time
  - Pipeline hazards

- Reading assignment – P&H 6.1-6.2