CprE 381: Computer Organization and Assembly Level Programming

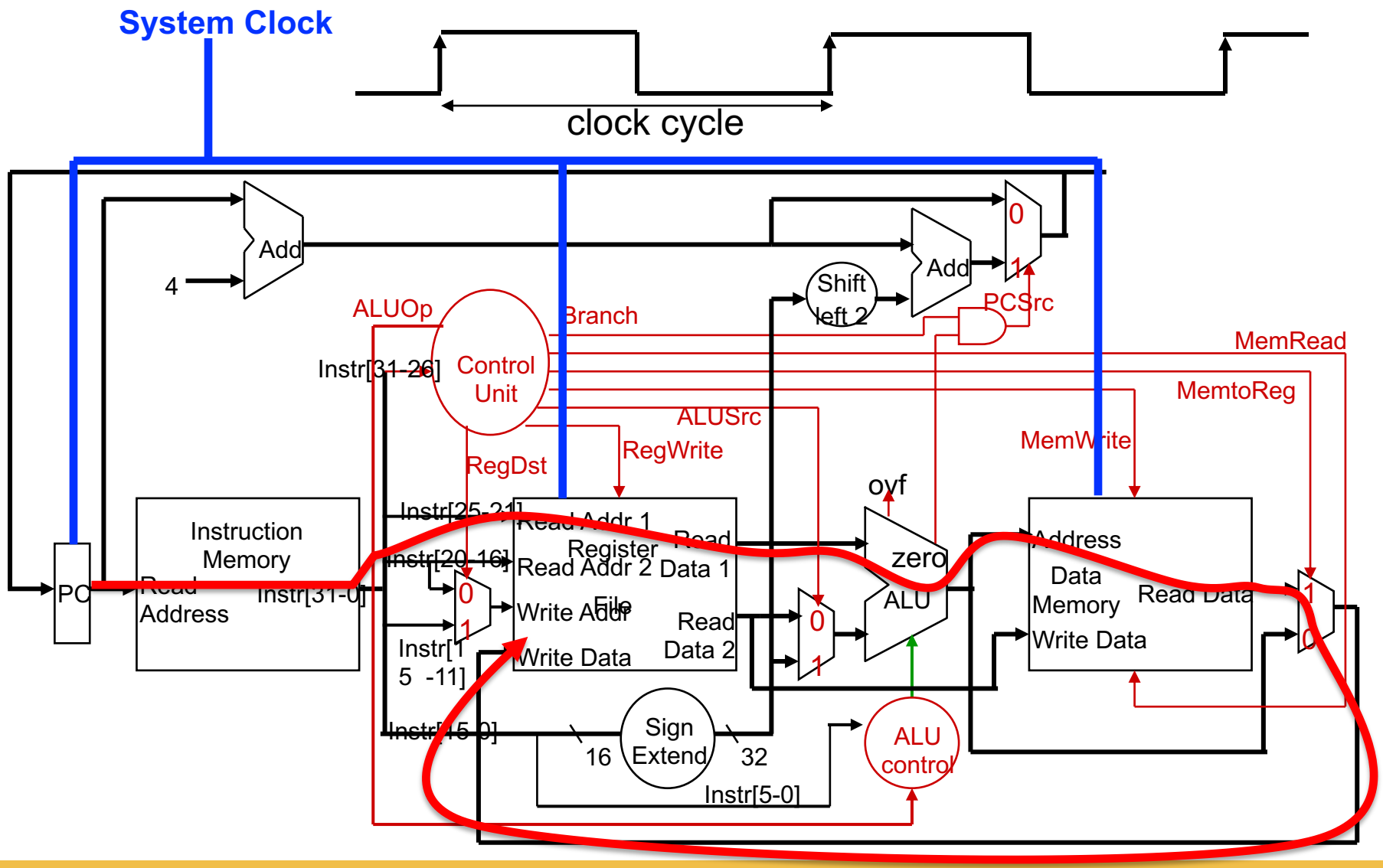
Pipelining

Henry Duwe
Electrical and Computer Engineering
Iowa State University

Administrative

- HW6 due Mar 13
- Term Project
  - Part 2a checkin BEFORE Spring Break
  - Part 2b due by lab week after Spring Break
- Exam 2
  - April 1 (three weeks from today)
- Midterm Grades submitted
  - How should I view my Canvas grade?
    - Snapshot of first 8 weeks of semester

Review: Worst Case Timing (Load Instruction)



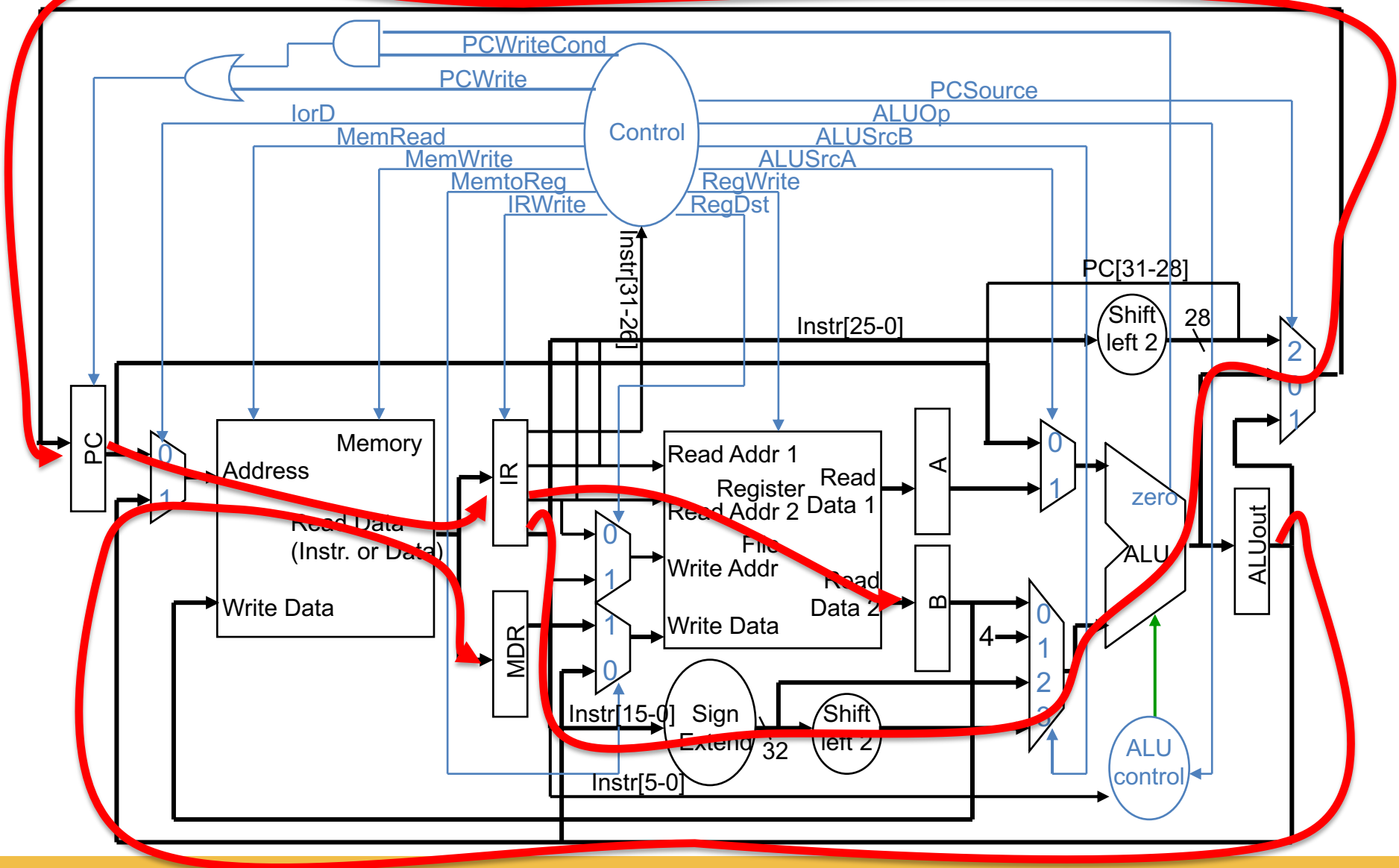
Review: Execution Time

- Drawing on the previous equation:

$$\textit{Execution Time} = \#\ \textit{Instructions} \times \frac{\textit{Cycles}}{\textit{Instruction}} \times \frac{\textit{Seconds}}{\textit{Cycle}}$$

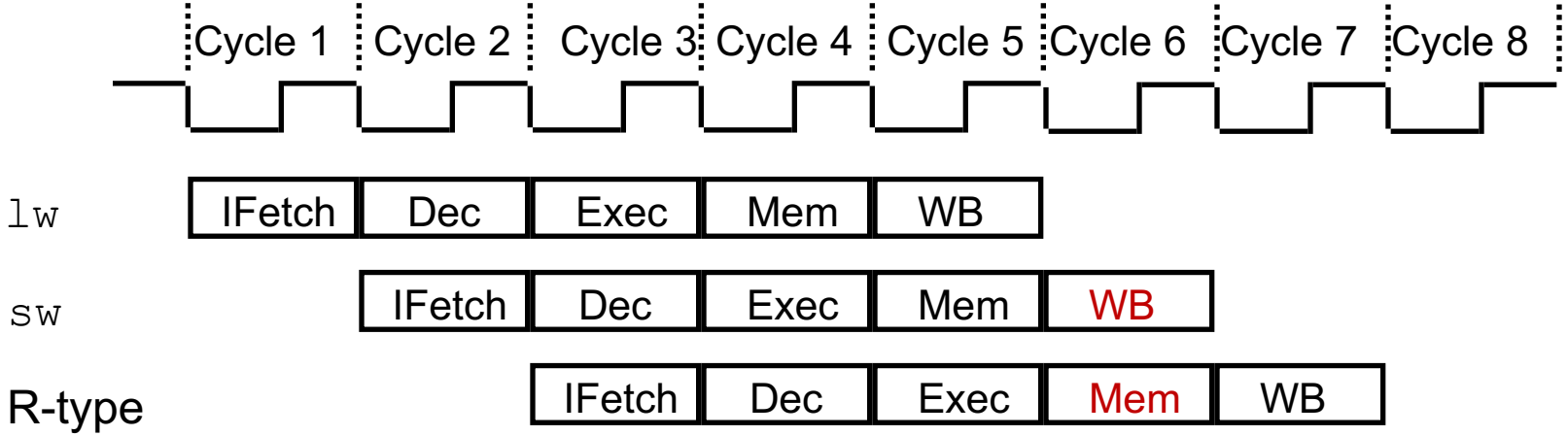
- To improve performance (i.e., reduce execution time)
  - Increase clock rate (decrease clock cycle time) OR
  - Decrease CPI OR
  - Reduce the number of instructions
- Designers balance cycle time against the number of cycles required
  - Improving one factor may make the other one worse...

Review: Multicycle Processor



A Pipelined MIPS Processor

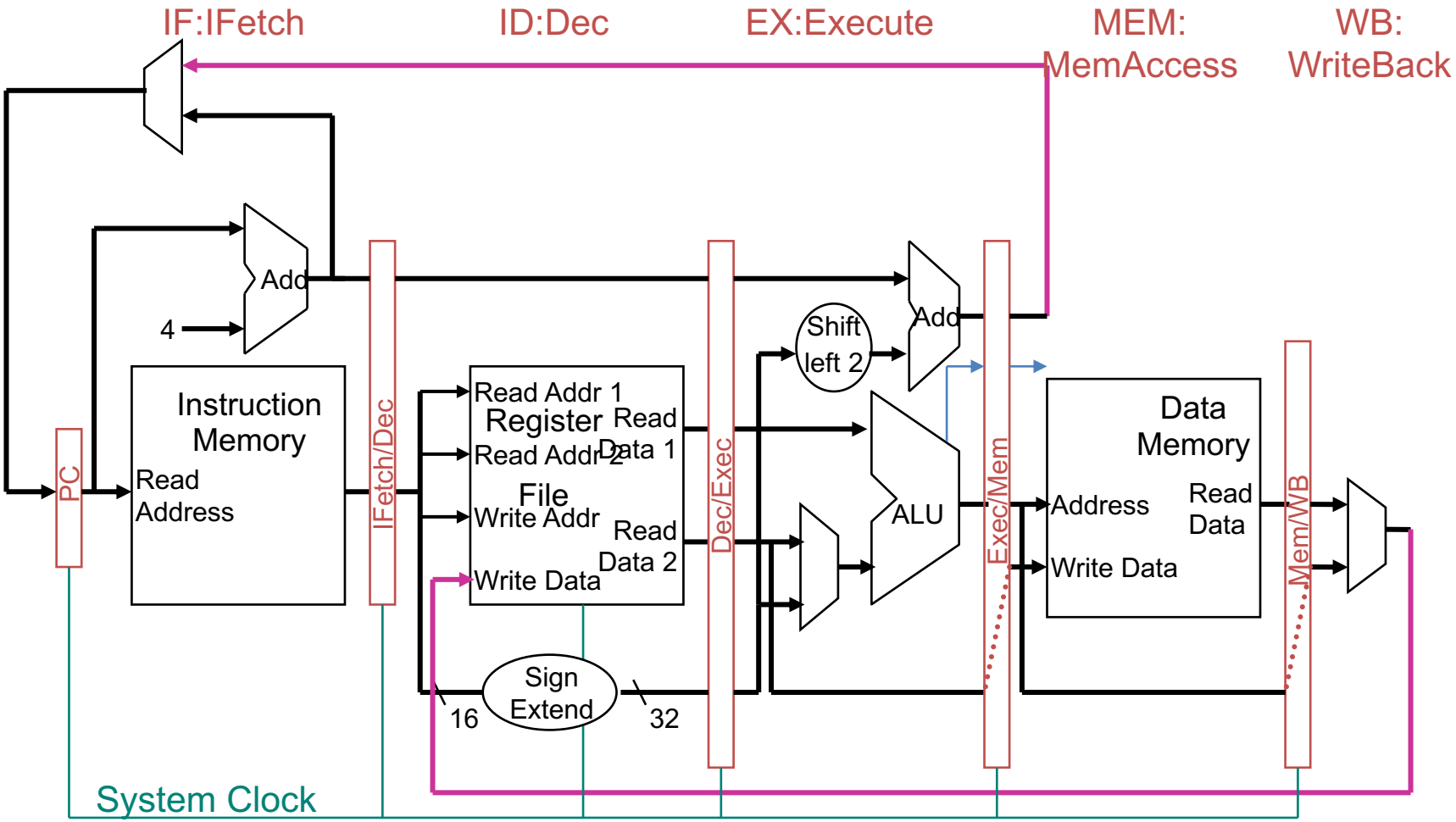
- Start the next instruction before the current one has completed
  - Improves throughput - total amount of work done in a given time
  - Instruction latency (execution time, delay time, response time - time from the start of an instruction to its completion) is *not* reduced



  - Clock cycle (pipeline stage time) is limited by the slowest stage
  - For some instructions, some stages are wasted cycles

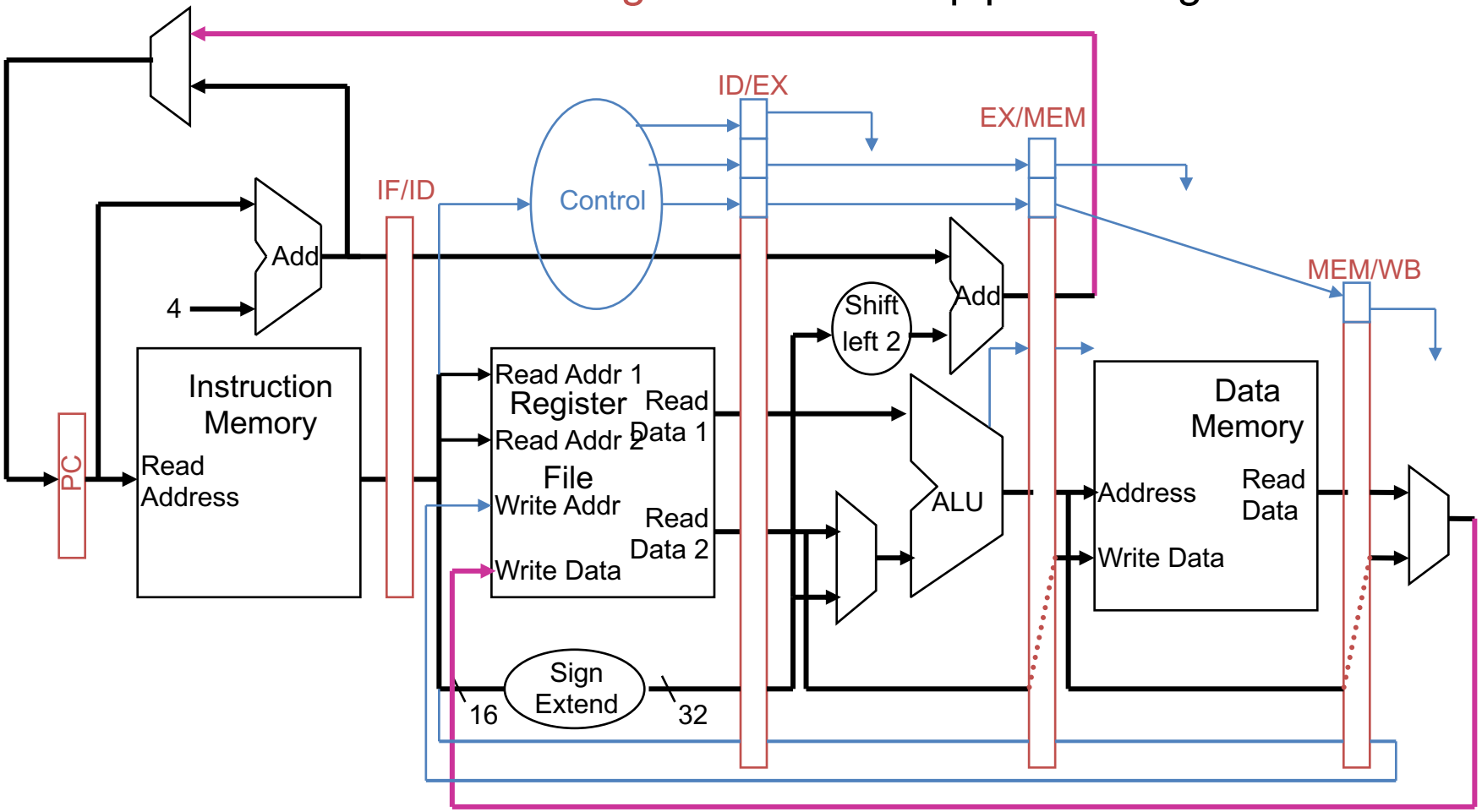
MIPS Pipeline Datapath Modifications

- What do we need to add/modify in our MIPS datapath?
  - State registers between each pipeline stage to isolate them

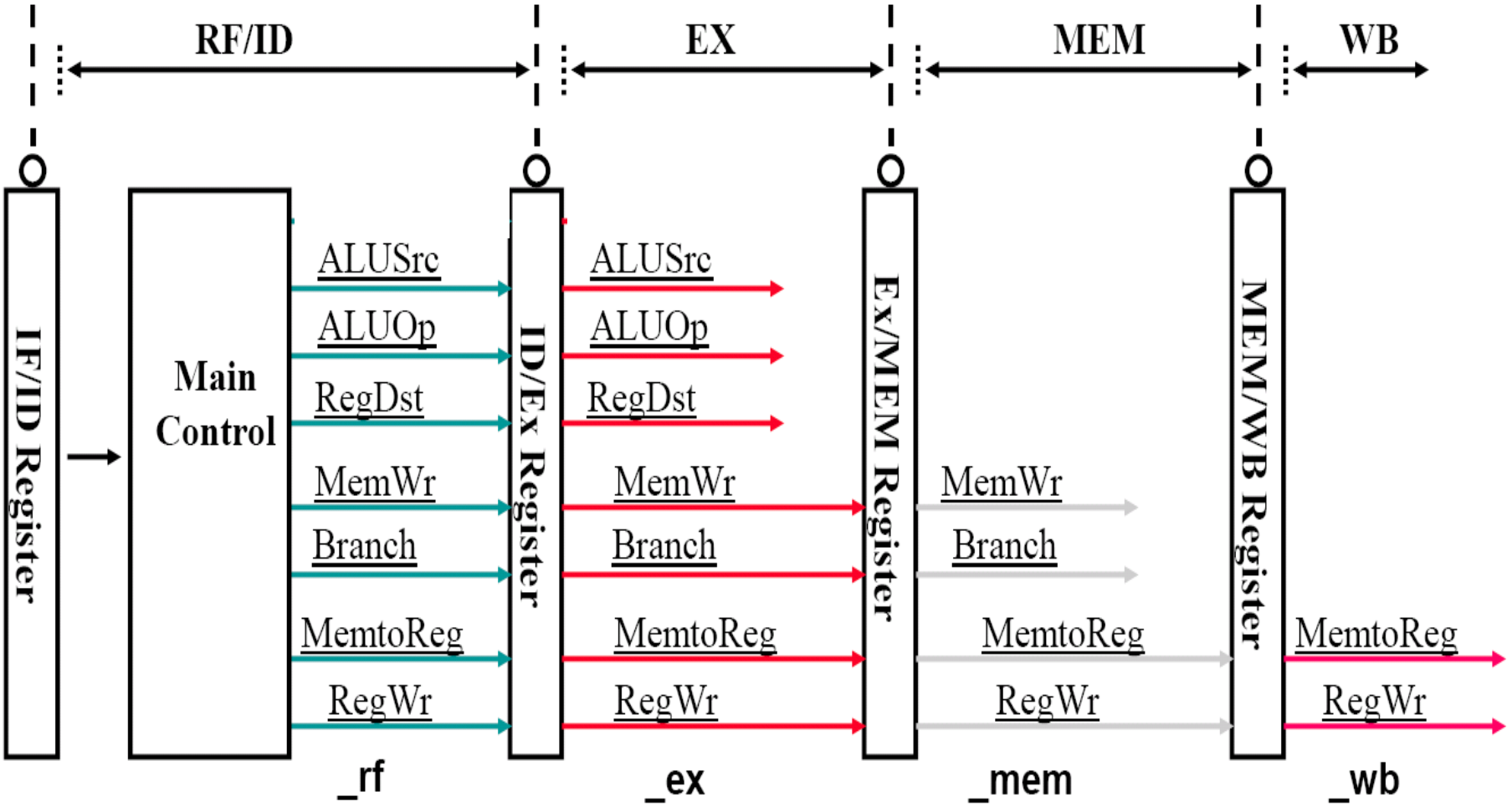


MIPS Pipeline Control Path Modifications

- All control signals can be determined during Decode
  - And held in the state registers between pipeline stages



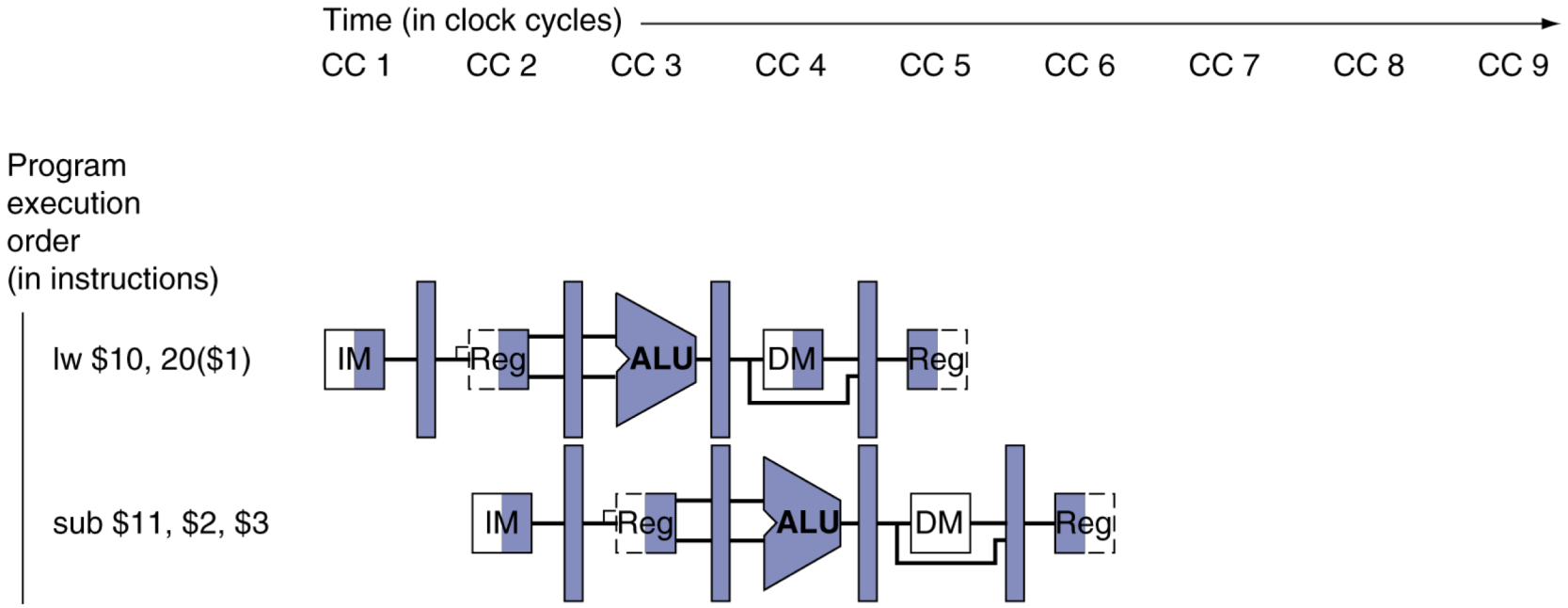
Implementing Control



Pipelining the MIPS ISA

- What makes it easy? (MIPS-specific)
  - All instructions are the same length (32 bits)
    - Can fetch in the 1st stage and decode in the 2nd stage
  - Few symmetric instruction formats (three)
    - Can begin reading register file in 2nd stage
  - Memory operations can occur only in loads and stores
    - Can use the execute stage to calculate memory addresses
    - Can access memory in one stage
  - Each MIPS instruction writes at most one result (i.e., changes the machine state) and does so near the end of the pipeline (MEM and WB)
- What makes it hard? (General)
  - Structural hazards: what if we had only one memory?
  - Control hazards: what about branches?
  - Data hazards: what if an instruction's input operands depend on the output of a previous instruction?

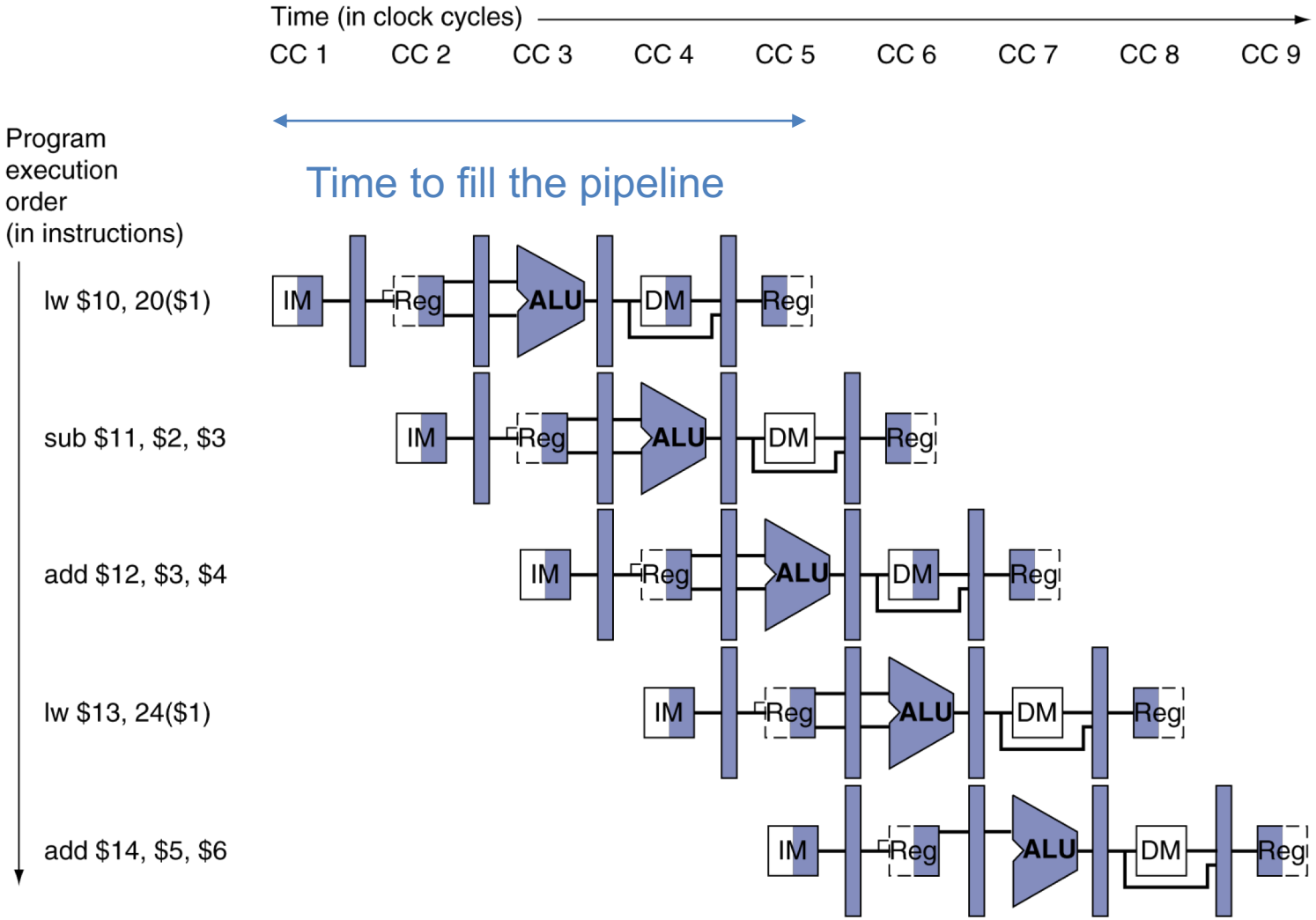
Graphically Representing MIPS Pipeline



- Can help with answering questions like:
  - How many cycles does it take to execute this code?
  - What is the ALU doing during cycle 4?
  - Is there a hazard, why does it occur, and how can it be fixed?

Why Pipeline? For Performance!

- Once the pipeline is full, one instruction is completed every cycle so CPI = 1



A Simple Performance Analysis

- Suppose 2ns for memory access, 2ns for ALU operation, and 1ns for register file read or write
  - Compute instr rate

- Nonpipelined Execution (limited to 1 instruction in proc):
  - **`lw`** : IF + Read Reg + ALU + Memory + Write Reg = 2 + 1 + 2 + 2 + 1 = 8 ns
  - **`add`**: IF + Read Reg + ALU + Write Reg = 2 + 1 + 2 + 1 = 6 ns
  - Single-cycle: 1/8ns
  - Multi-cycle: 1/9ns

- Pipelined Execution (steady-state full pipeline):
  - Cycle time: Max(IF,Read Reg,ALU,Memory,Write Reg) = 2 ns
  - 1/2ns

A Simple Performance Analysis

- Suppose 2ns for memory access, 2ns for ALU operation, and 1ns for register file read or write
  - Compute instr rate

- Nonpipelined Execution (limited to 1 instruction in proc):

In-class Assessment!

Access Code: =====

Note: sharing access code to those outside of classroom or using access while outside of classroom is considered cheating

- Pipelined Execution (steady-state full pipeline):
  - Cycle time: Max(IF,Read Reg,ALU,Memory,Write Reg) = 2 ns
  - 1/2ns

Acknowledgments

- These slides contain material developed and copyright by:
  - Joe Zambreno (Iowa State)
  - David Patterson (UC Berkeley)
  - Mary Jane Irwin (Penn State)
  - Christos Kozyrakis (Stanford)
  - Onur Mutlu (Carnegie Mellon)
  - Krste Asanović (UC Berkeley)