Datapath

- This module describes how instructions are executed using the MIPS datapath
- The datapath includes the ALU, control unit, register file and the pathways that connect the various components
- The ALU and control unit are implemented using the logic circuits described in the previous module

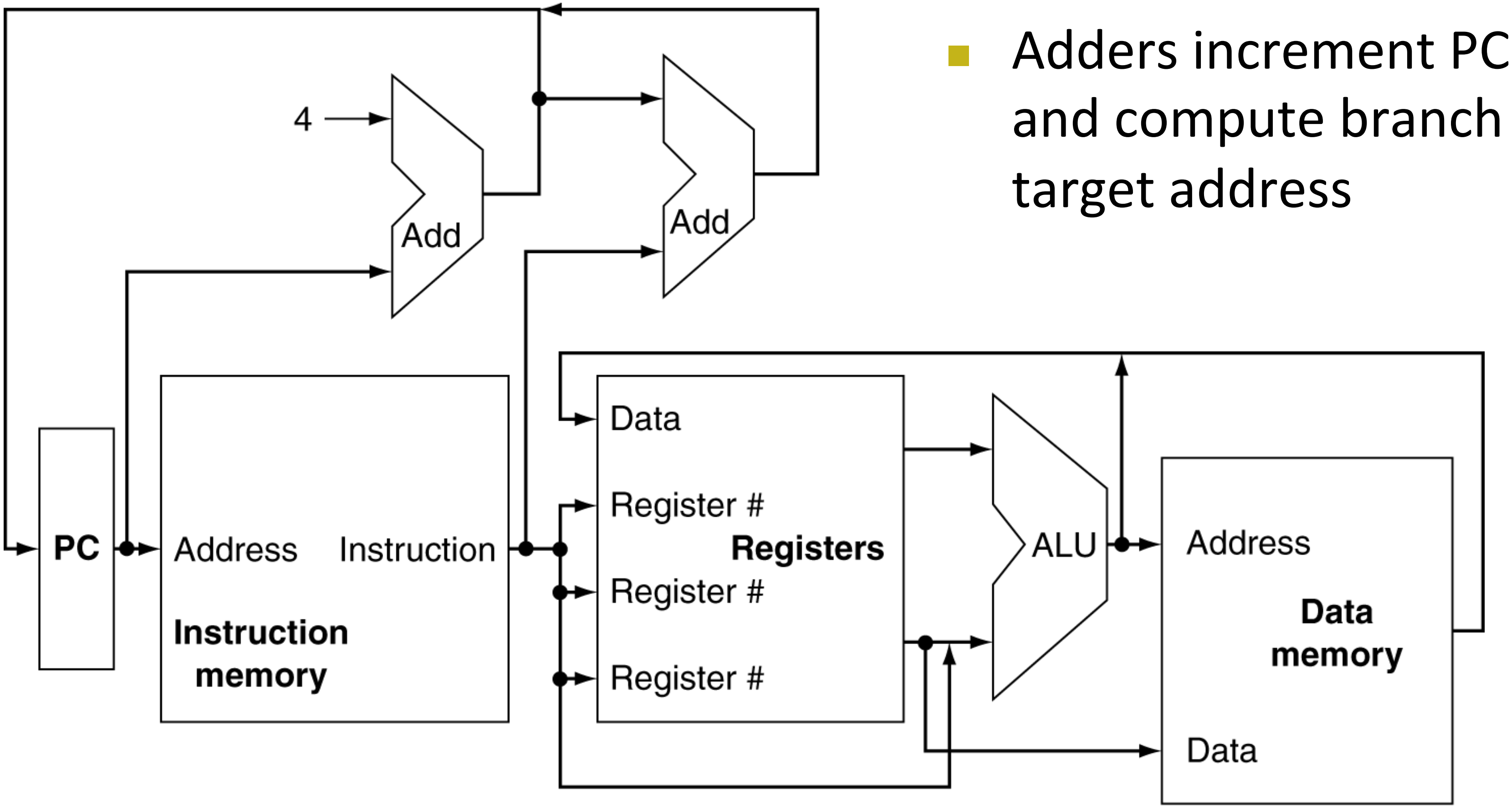
Datapath

- CPU performance factors
  - Instruction count
    - Determined by ISA and compiler
  - CPI and Cycle time
    - Determined by CPU hardware
- We will examine two MIPS implementations
  - A simplified single-cycle version
  - And later, a more realistic pipelined version
- A simple instruction subset will be used
  - Memory reference: `lw`, `sw`
  - Arithmetic/logical: `add`, `sub`, `and`, `or`, `slt`
  - Control transfer: `beq`, `j`

Instruction Execution

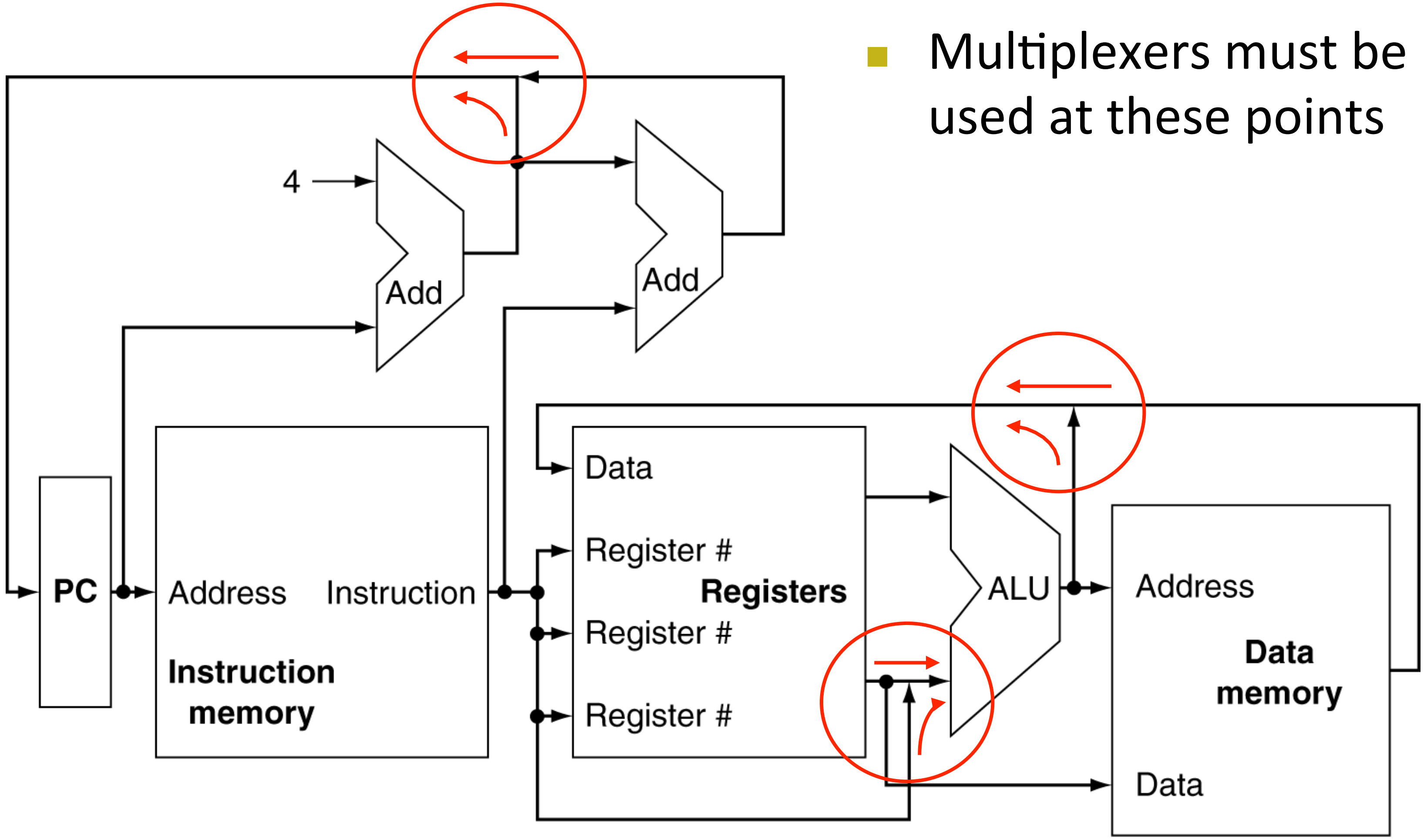
- PC → instruction memory, fetch instruction
- Register numbers → register file, read registers
- Depending on instruction class
  - Use ALU to
    - Calculate result
    - Compute memory address for load/store
    - Evaluate branch condition
  - Access data memory for load/store
  - PC ← target address or PC + 4

Datapath Overview



- Adders increment PC and compute branch target address

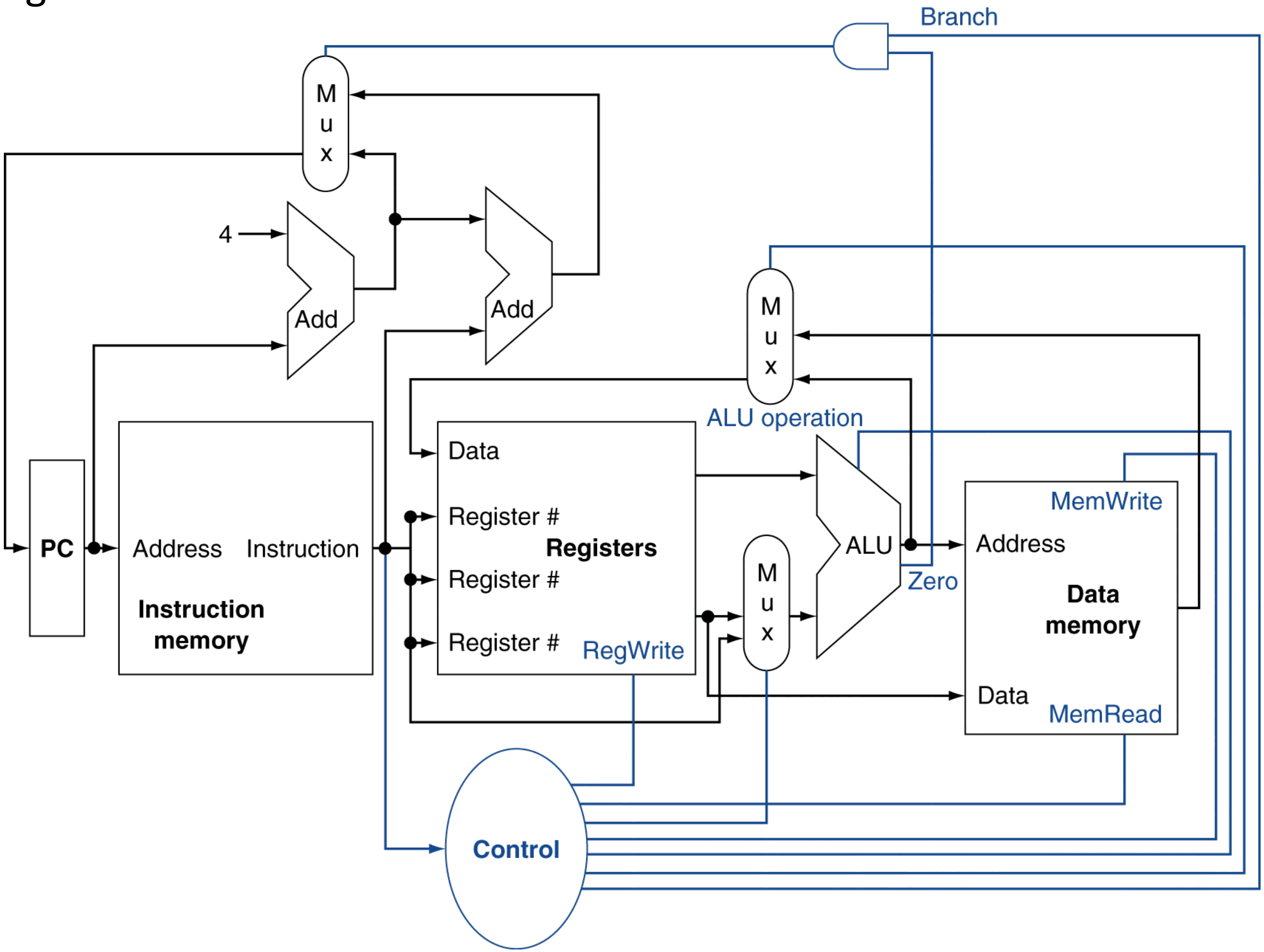
Datapath Overview



- Multiplexers must be used at these points

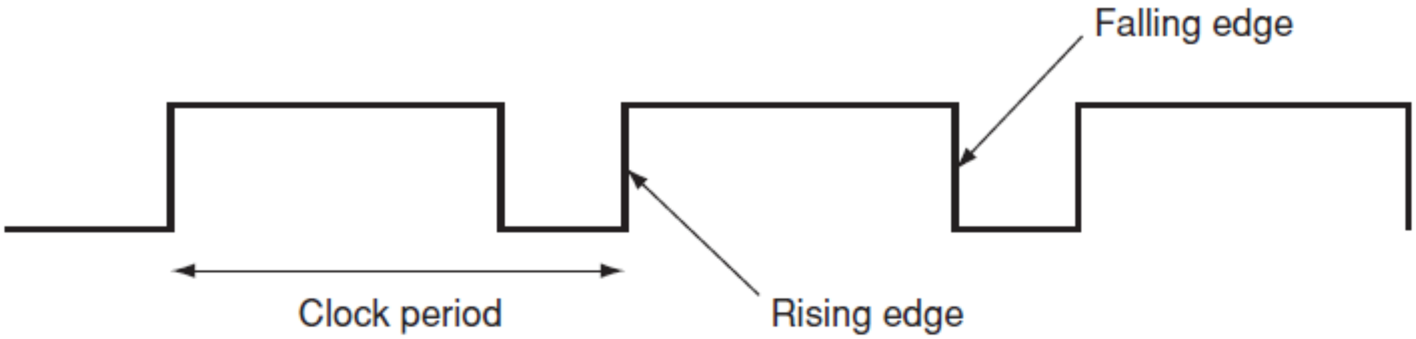
Datapath Control

Control signals determine what actions are taken



Datapath Timing

- Operations are synchronized to a clock
  - For example, when a register is written
  - Instructions complete at clock edges



- Clock signal oscillates between high and low values
- Clock period is one full clock cycle
- State changes only on clock edge (either rising of falling edge)