NWEN 242

Pipelined Datapath and Control

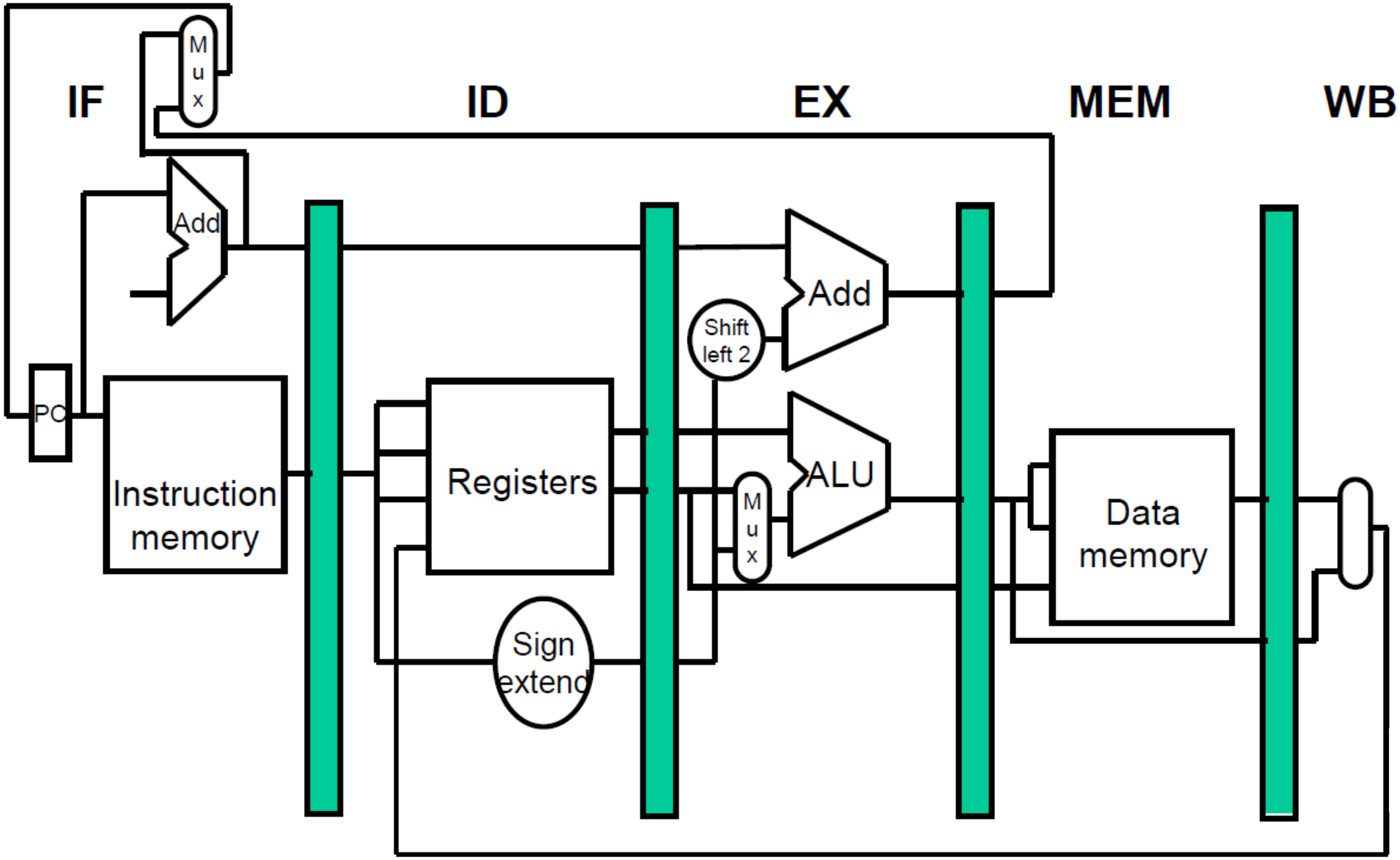
Agenda

- Single-cycle vs. pipelined performance
- Pipeline register and pipeline operation
- Control for the pipeline

Introduction to pipelining

- Involves having several instructions in the processor all at once
  - Each at a different stage of its execution
  - Individual instructions take the same time (five steps)

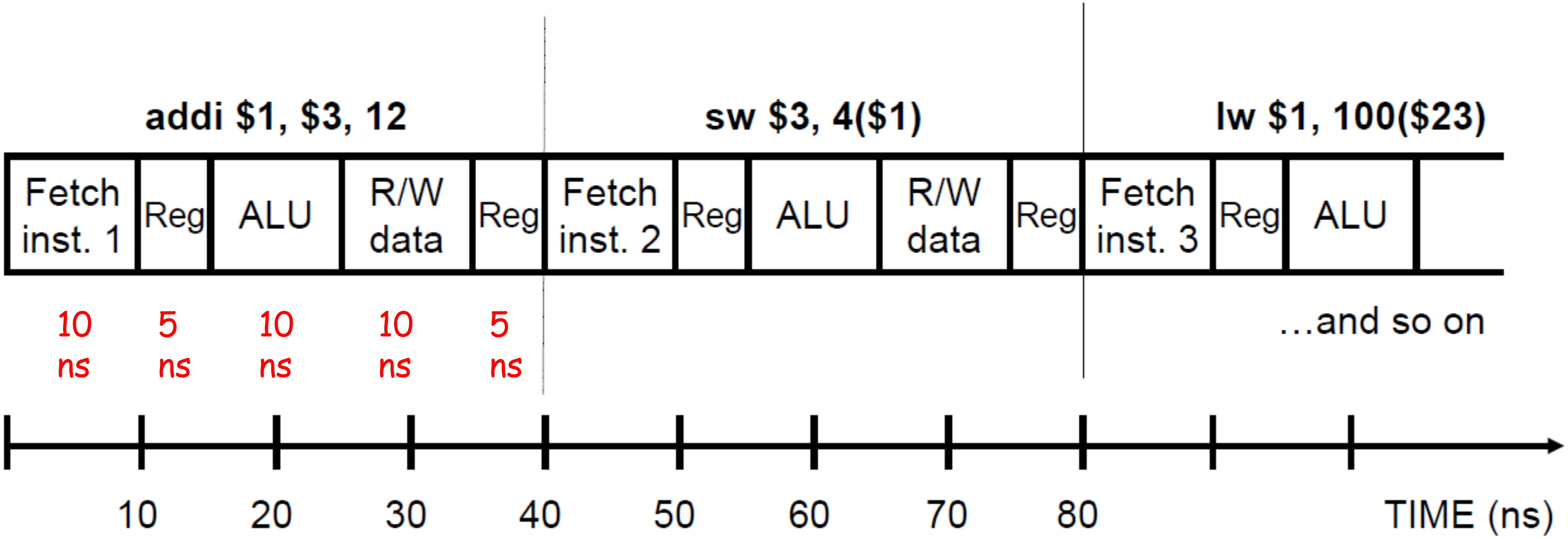
5-stage pipelined datapath



Benefits of pipelining

- Pipelining gives better performance than single-cycle design
- Improved throughput

Throughput for single-cycle datapath



addi $1, $3, 12

sw $3, 4($1)

lw $1, 100($23)

Fetch inst. 1 | Reg | ALU | R/W data | Reg | Fetch inst. 2 | Reg | ALU | R/W data | Reg | Fetch inst. 3 | Reg | ALU

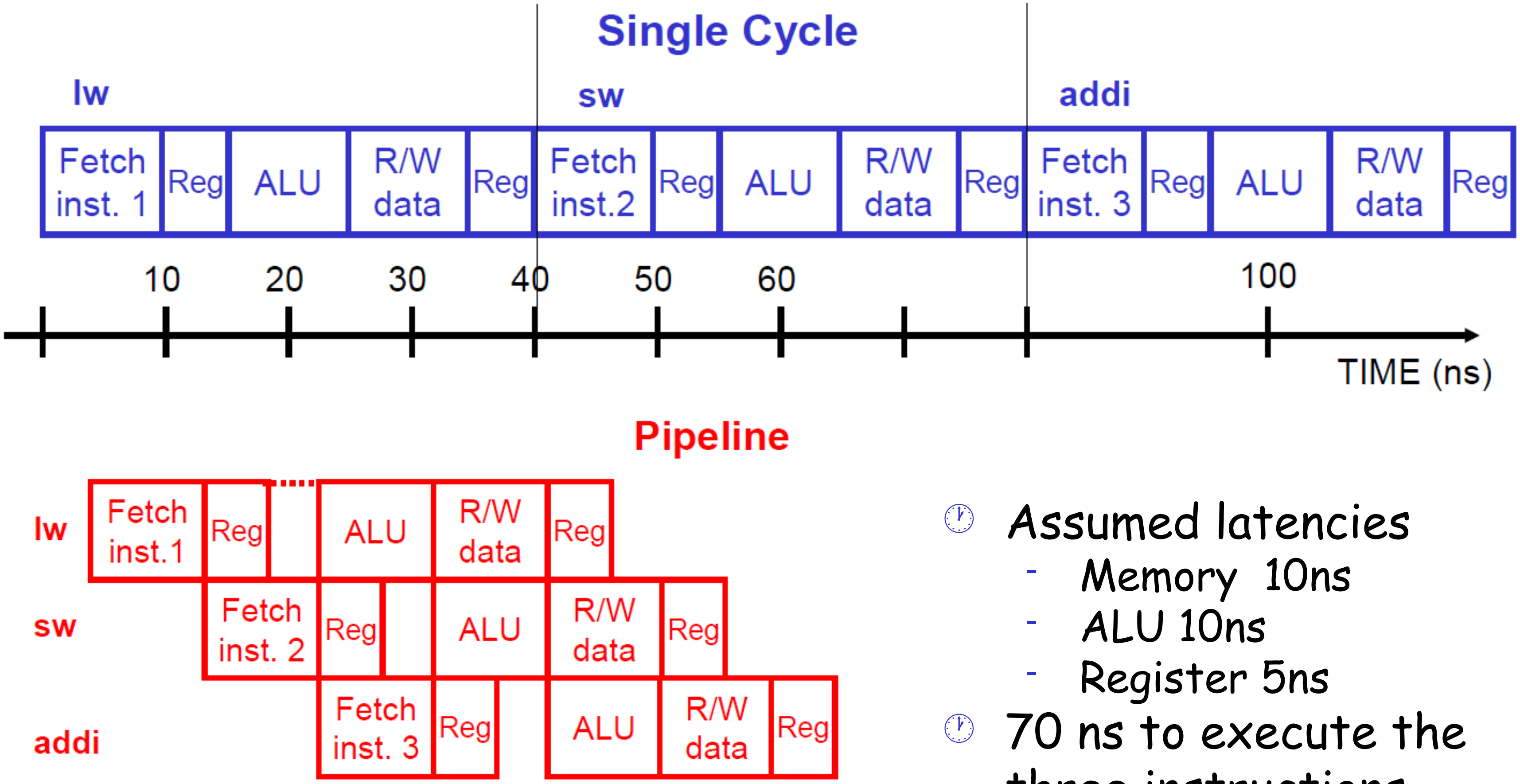
10 ns | 5 ns | 10 ns | 10 ns | 5 ns

…and so on

10 20 30 40 50 60 70 80 TIME (ns)

- Assumed latencies
  - Memory 10ns
  - ALU 10ns
  - Register 5ns
- Suitable clock speed
  - 25 MHz
  - 40ns period
- 120 ns to execute the three instructions

Throughput for pipelined datapath



- Assumed latencies
  - Memory 10ns
  - ALU 10ns
  - Register 5ns
- 70 ns to execute the three instructions

Total clock cycles = 5 + (n-1)

What is the performance gain?

- Non-pipelined
  - Each instruction takes 40ns to complete
  - e.g. completing 3 instructions takes 120ns
- Pipelined with 5 stages
  - Each instruction takes 50ns to complete
  - e.g. third instruction finishes after 50+20=70ns

Time to complete the first instruction

Time to complete next two instructions

- Thus pipelined processor is 120/70 = 1.71 faster (for three instructions) than non-pipelined processor
- How many times faster is the pipelined processor for executing 1000 instructions?

- Single-cycle processor: (4 x 1000) stages x 10ns = 40,000ns
- Pipelined processor: (5 + (1000-1)) stages x 10ns = 10,040ns
- Pipelined processor is almost four times faster than non-pipelined counterpart.

- Single-cycle processor: (4 x 1000) stages x 10ns = 40,000ns
- Pipelined processor: (5 + (1000-1)) stages x 10ns = 10,040ns
- Pipelined processor is almost four times faster than non-pipelined counterpart.

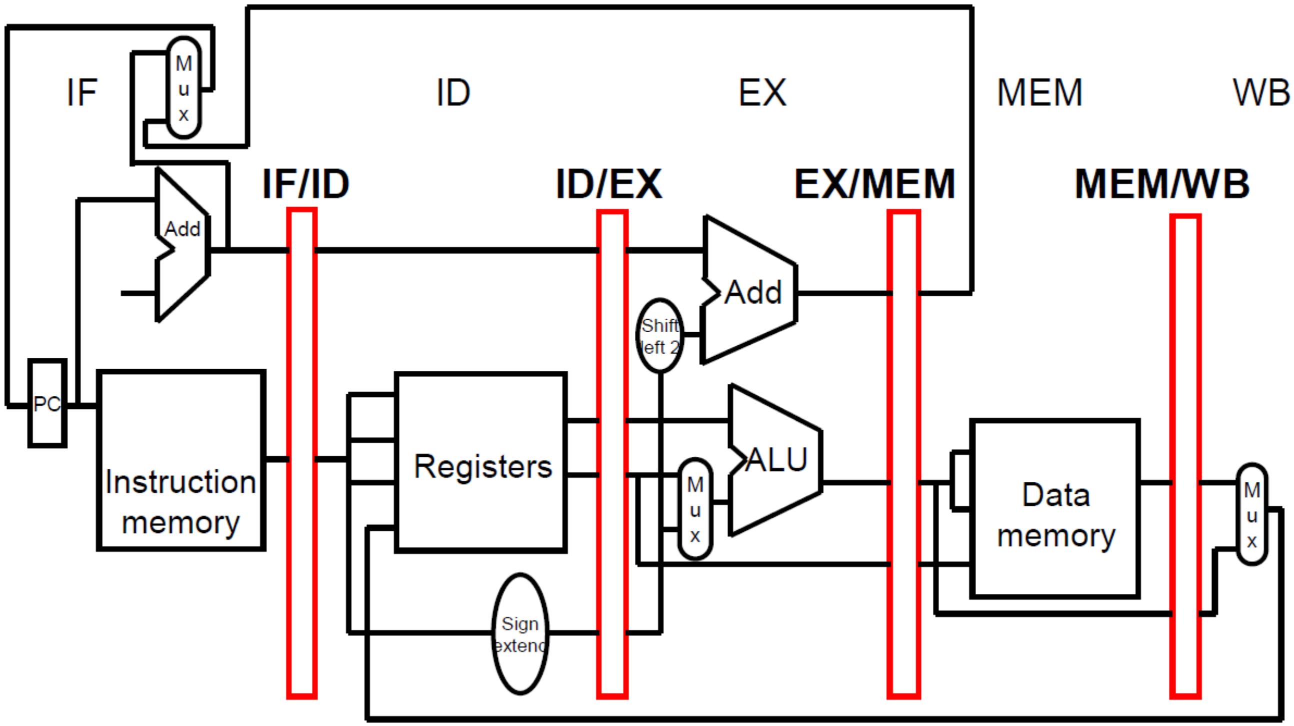
Performance Gain = single-cycle cycle time / pipelined cycle time

How pipeline works

- New datapath element: Pipeline Registers

Pipeline registers

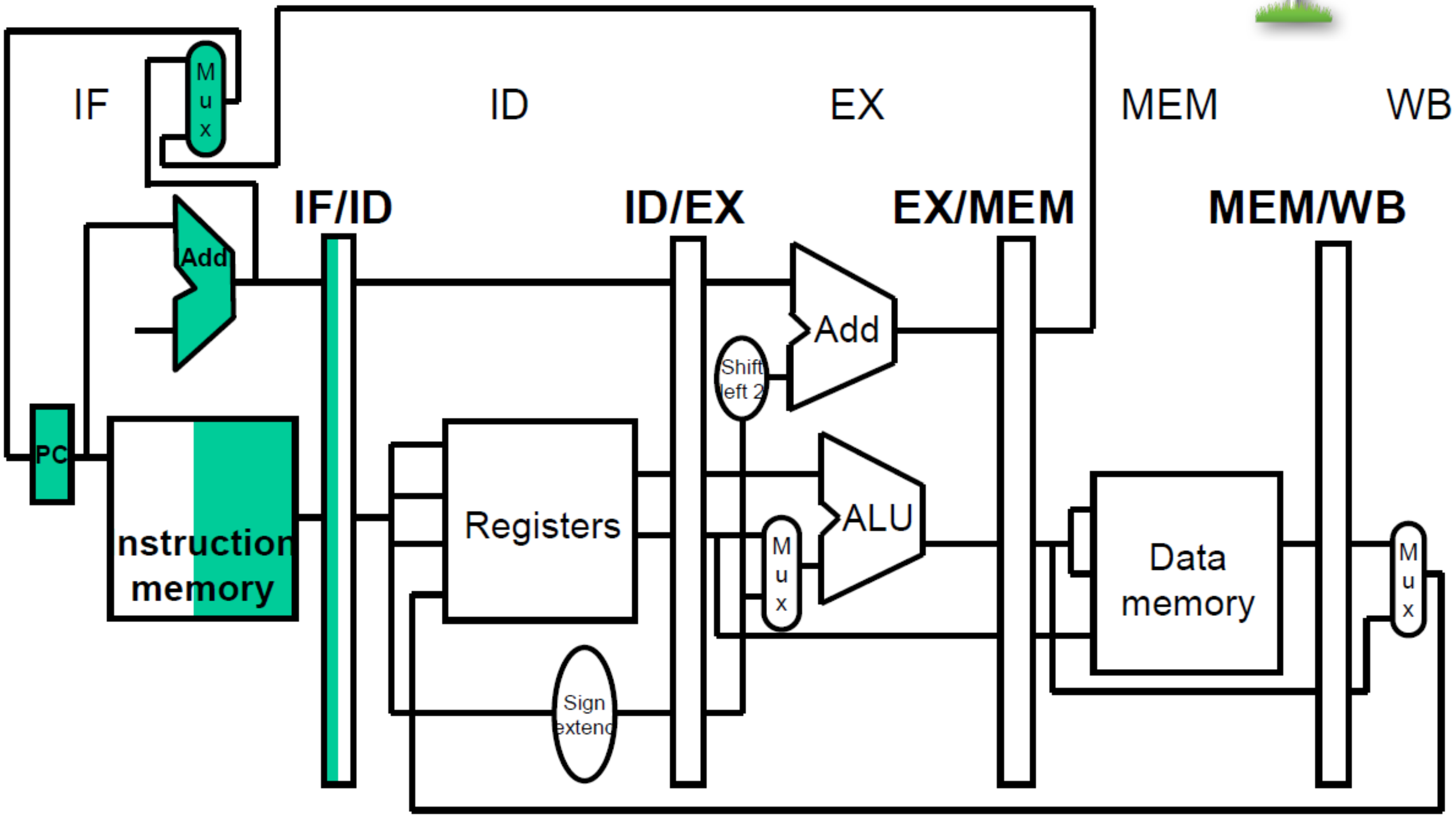
- Need registers between stages
  - to hold information produced in previous cycle
  - get written for every cycle



Pipeline operation

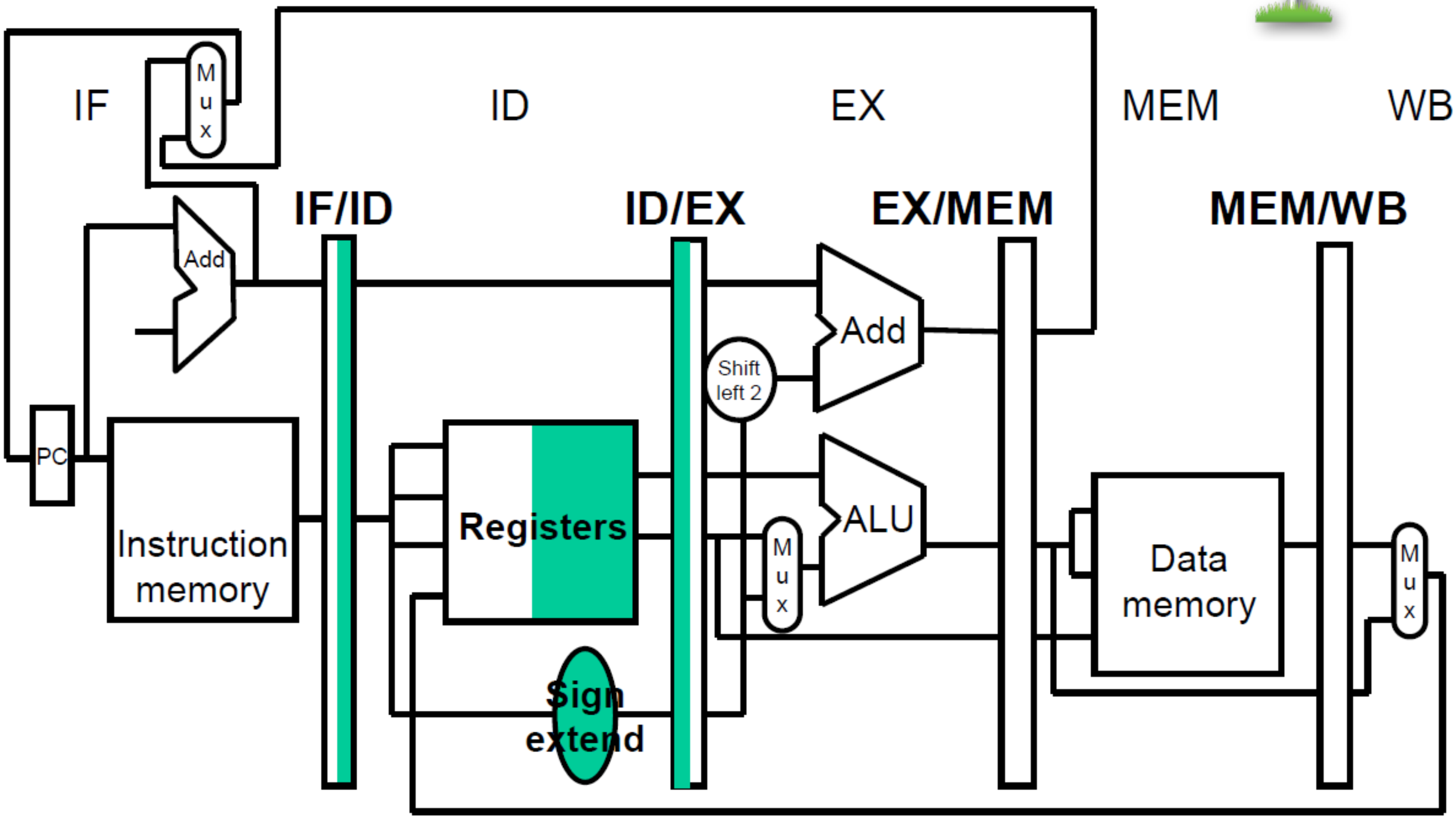
- Cycle-by-cycle flow of instructions through the pipelined datapath

lw stage 1 (IF = instruction fetch)

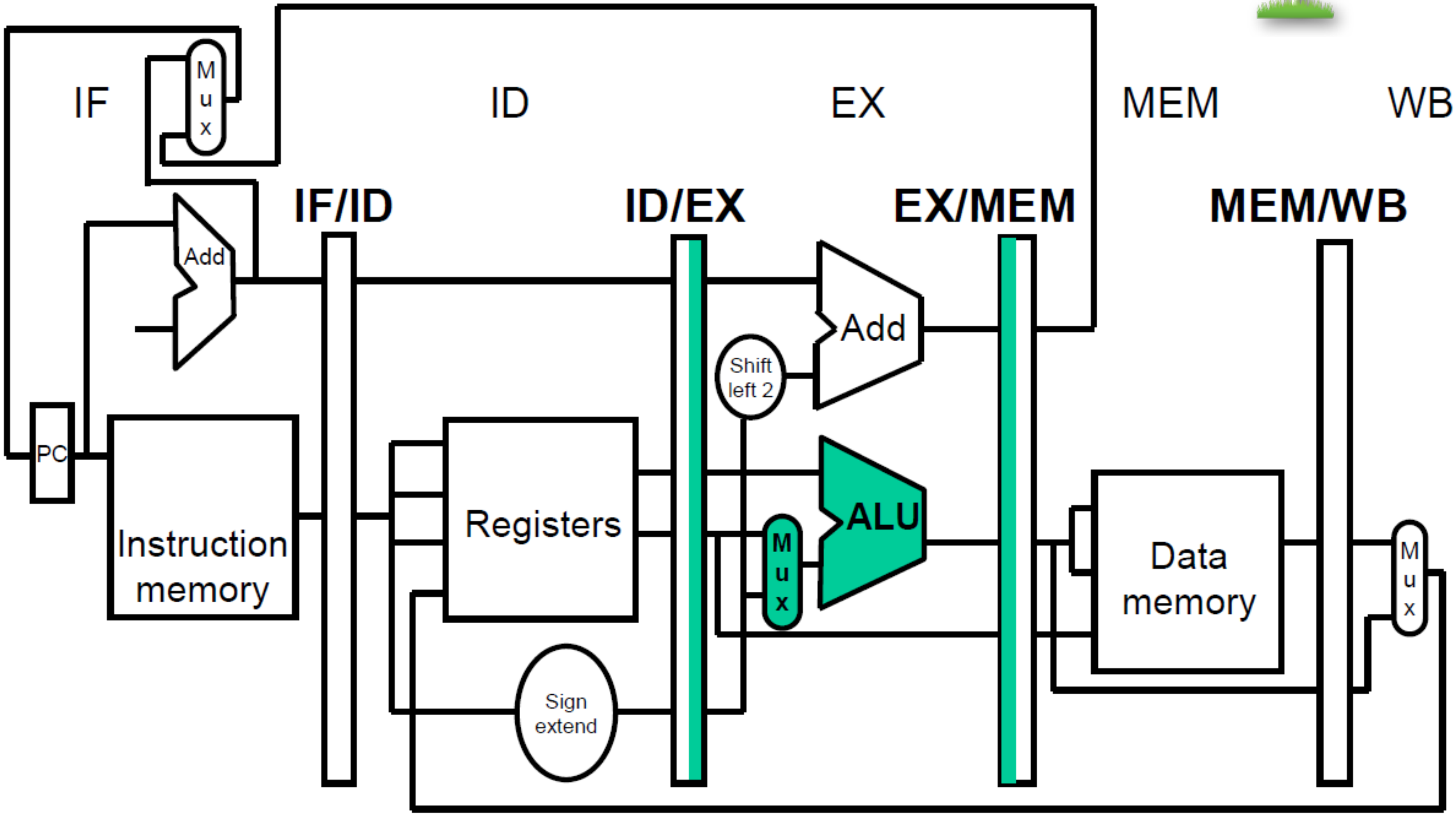


- Right half of a block colored means it's being read
- Left half of a block colored means it's being written

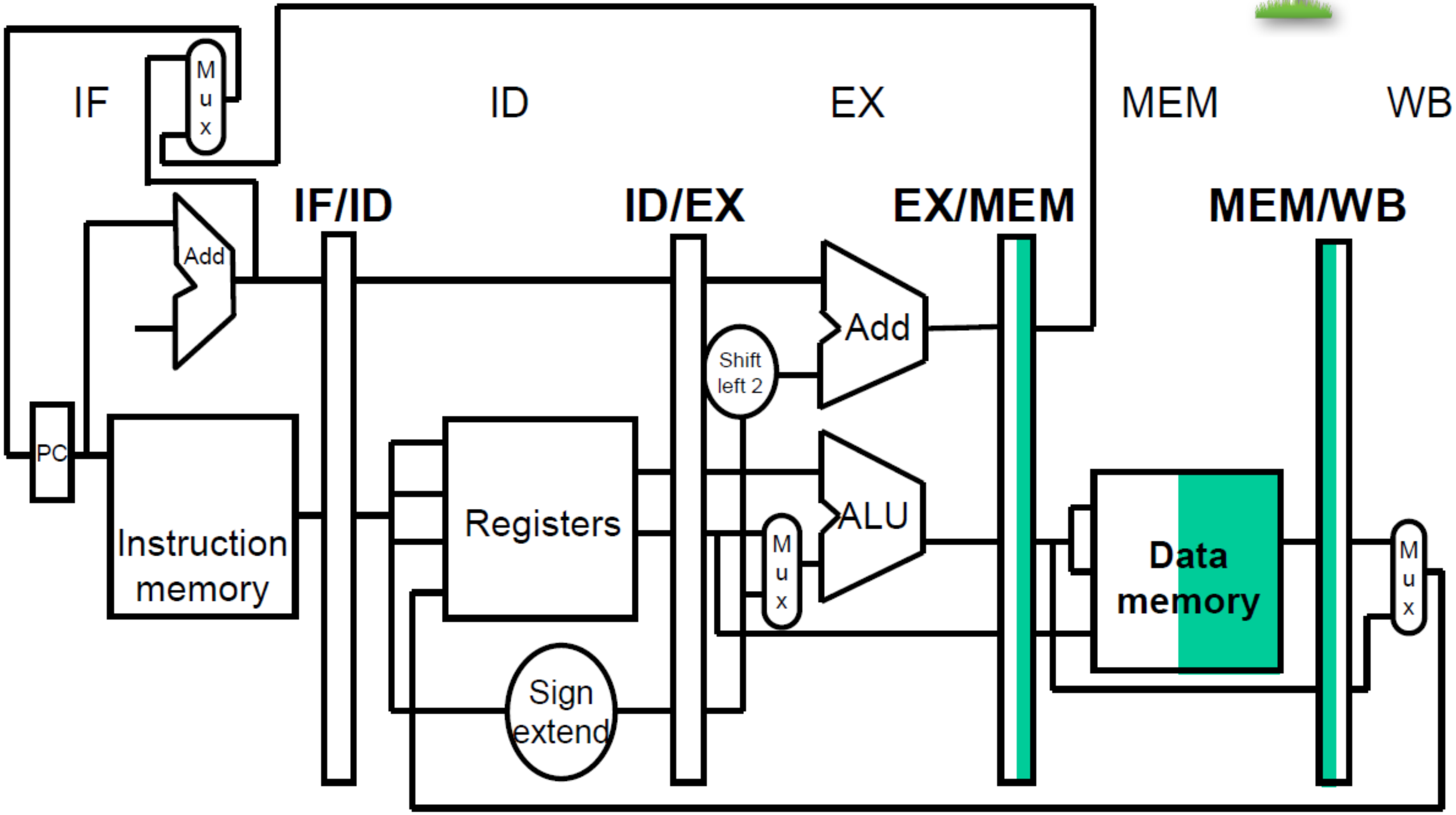
lw stage 2 (ID = instruction decode)



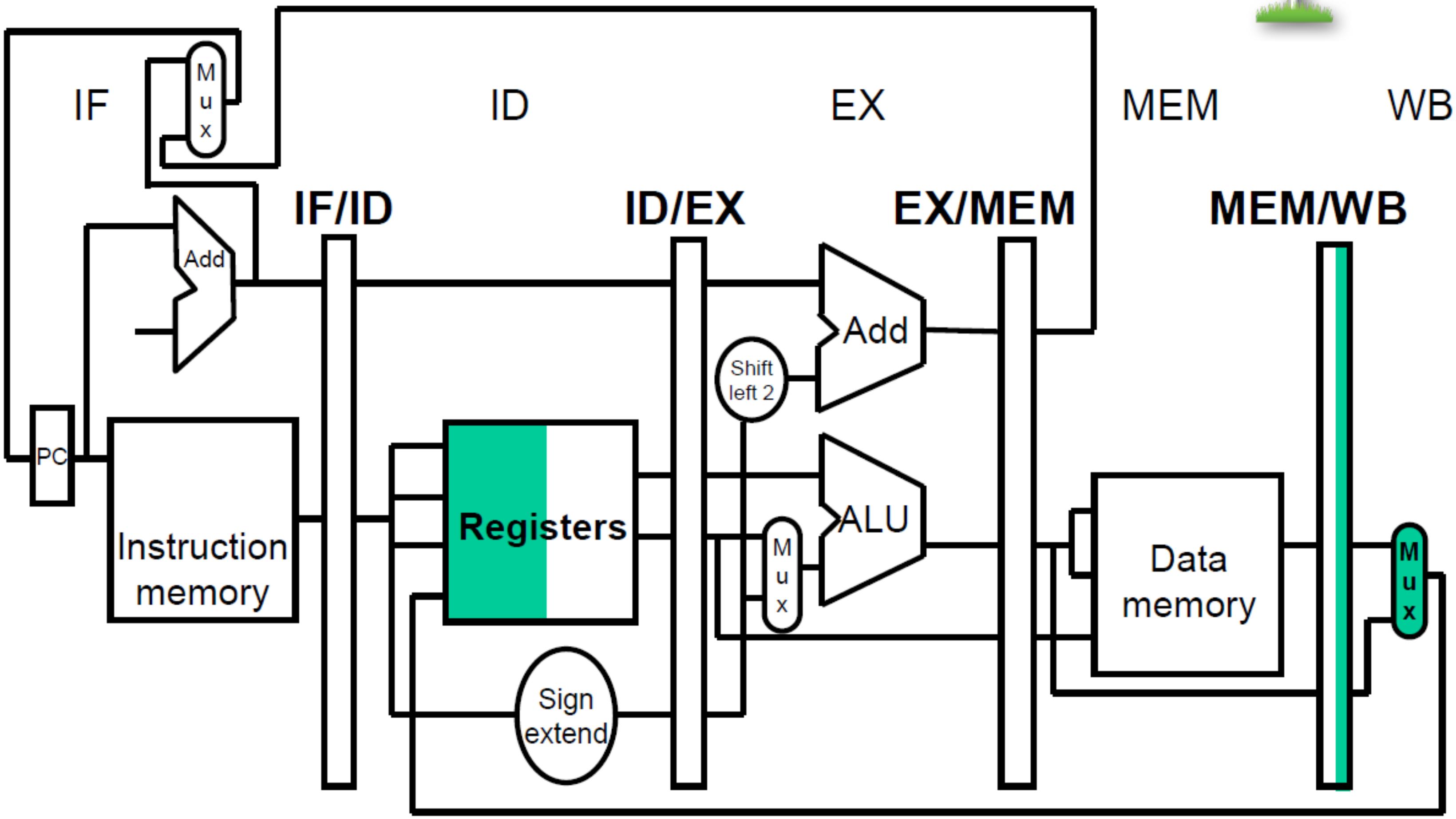
lw stage 3 (EX = execute)



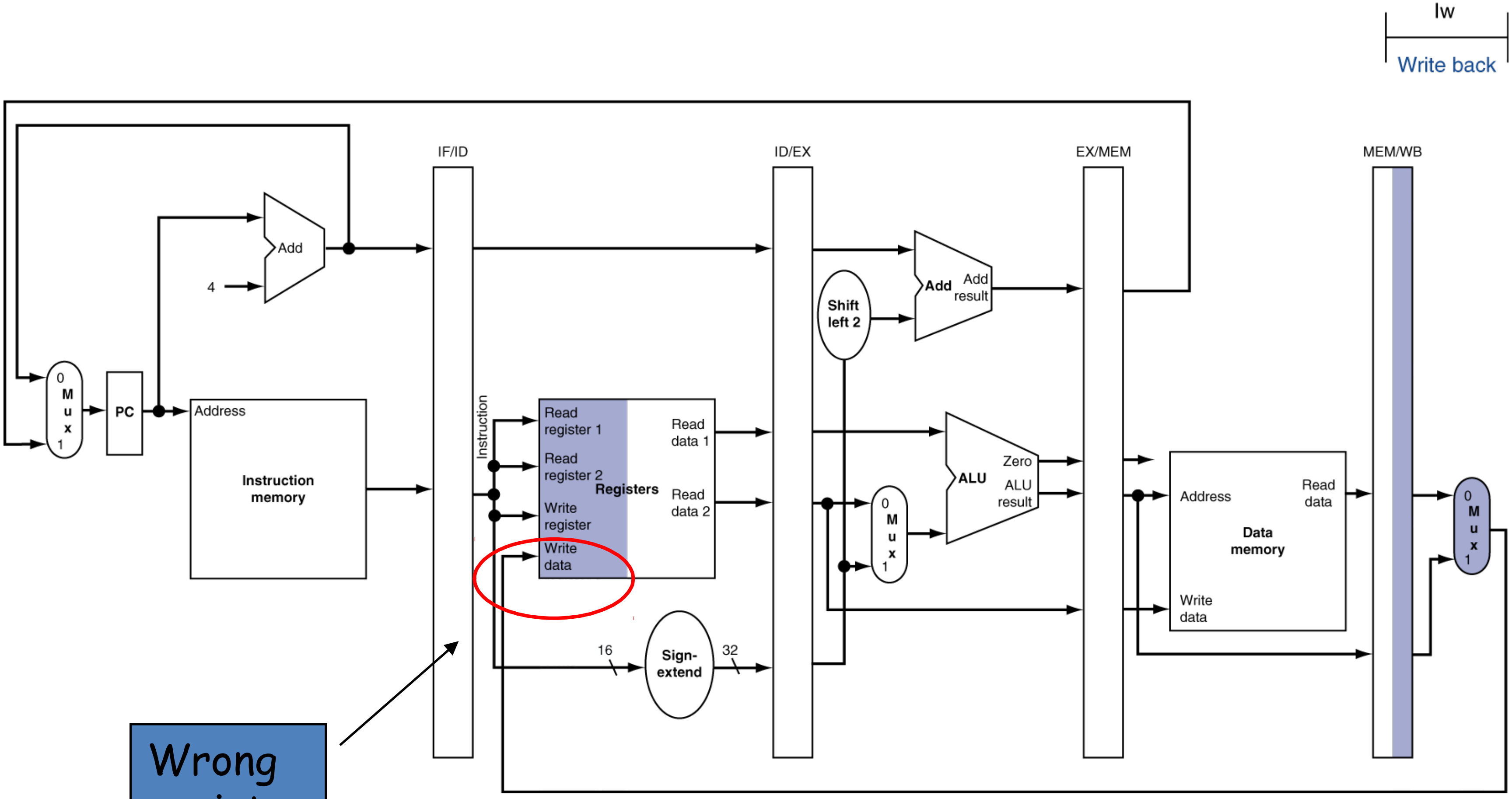
lw stage 4 (MEM = R/W memory)



lw stage 5 (WB = Write-Back)



lw stage 5 (WB = Write-Back)



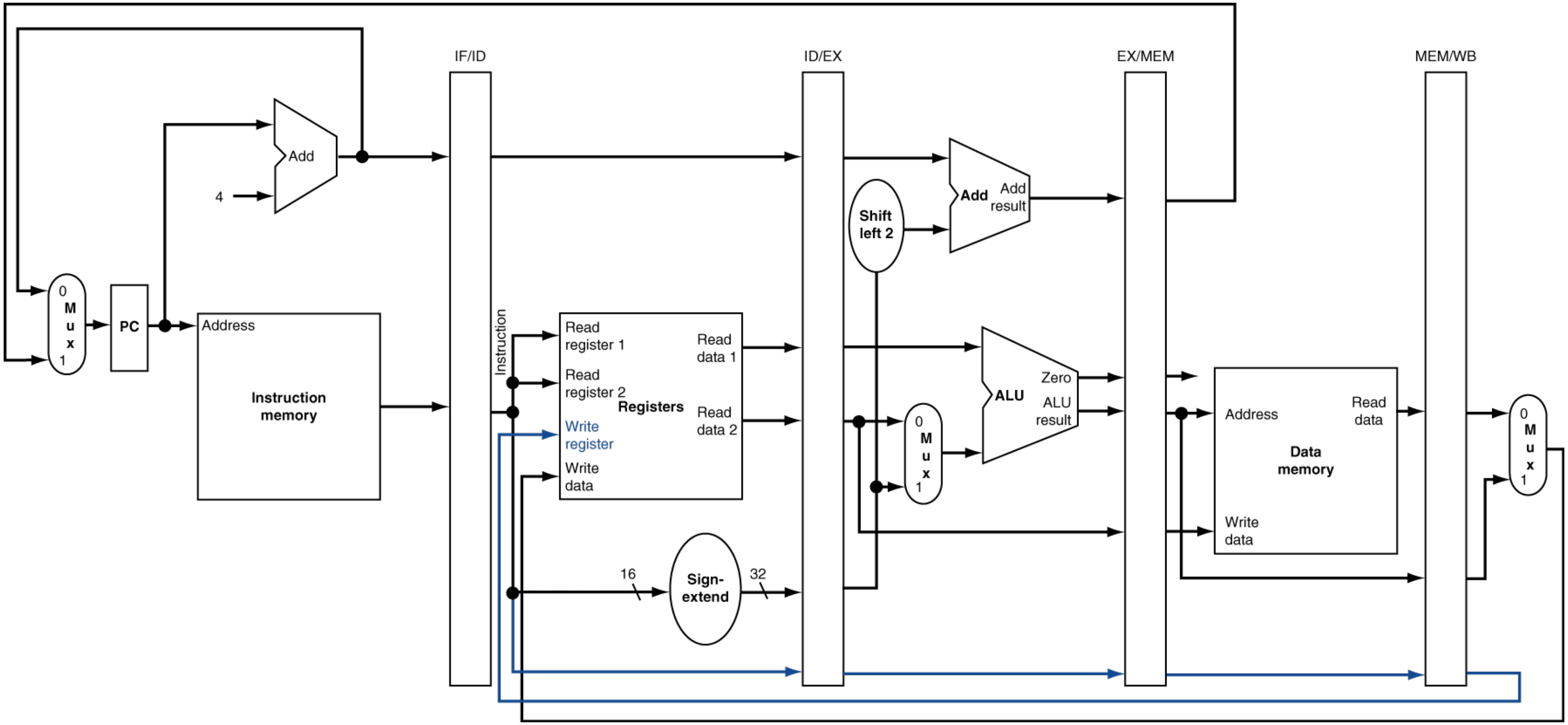
Wrong register number

The write register has been changed

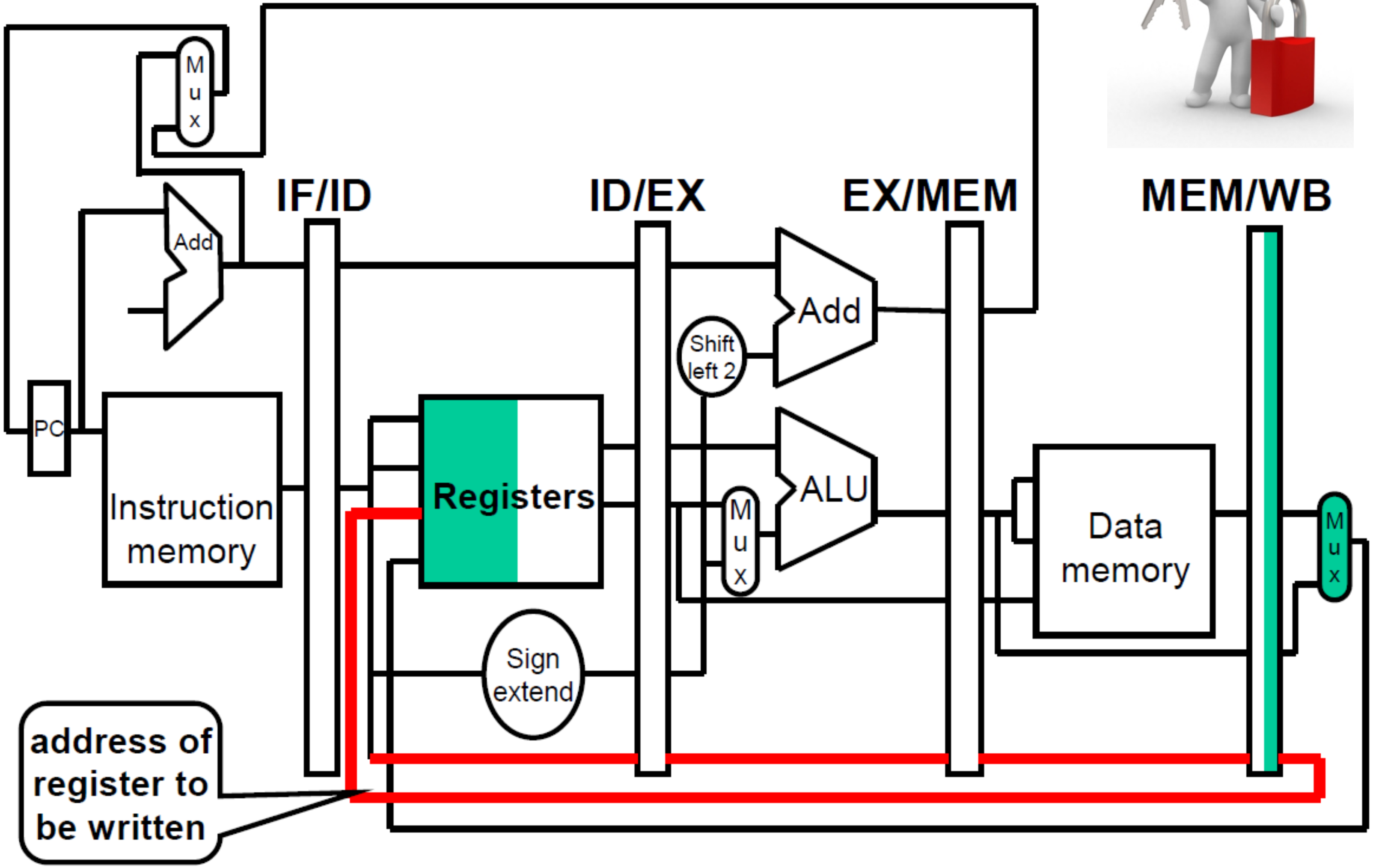
- As soon as we decode a new instruction, we will overwrite the contents of the Read Register 1 and 2 and the write register

- Hence, we have to pass along the register address to write through pipeline registers

Corrected Datapath for Load



Corrected Datapath for Load

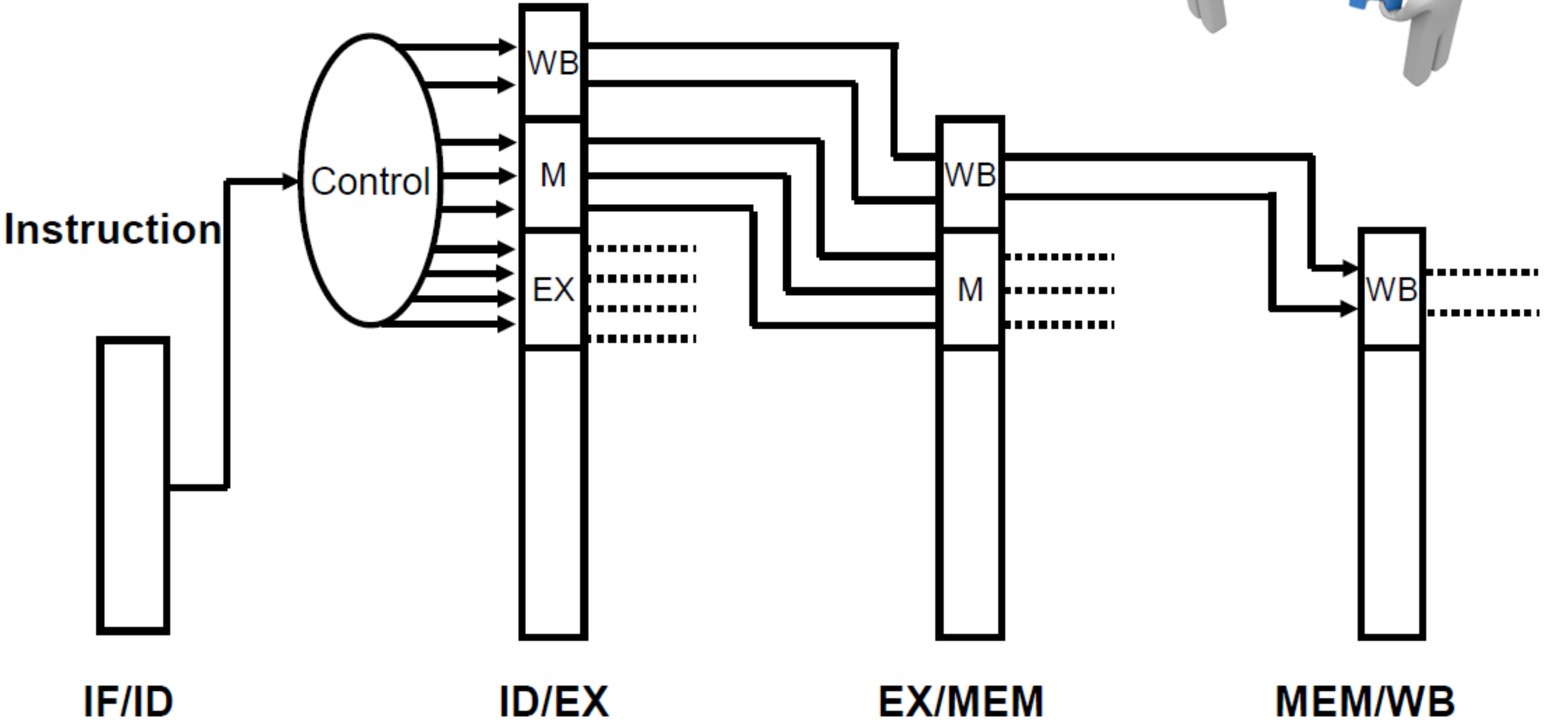


address of register to be written

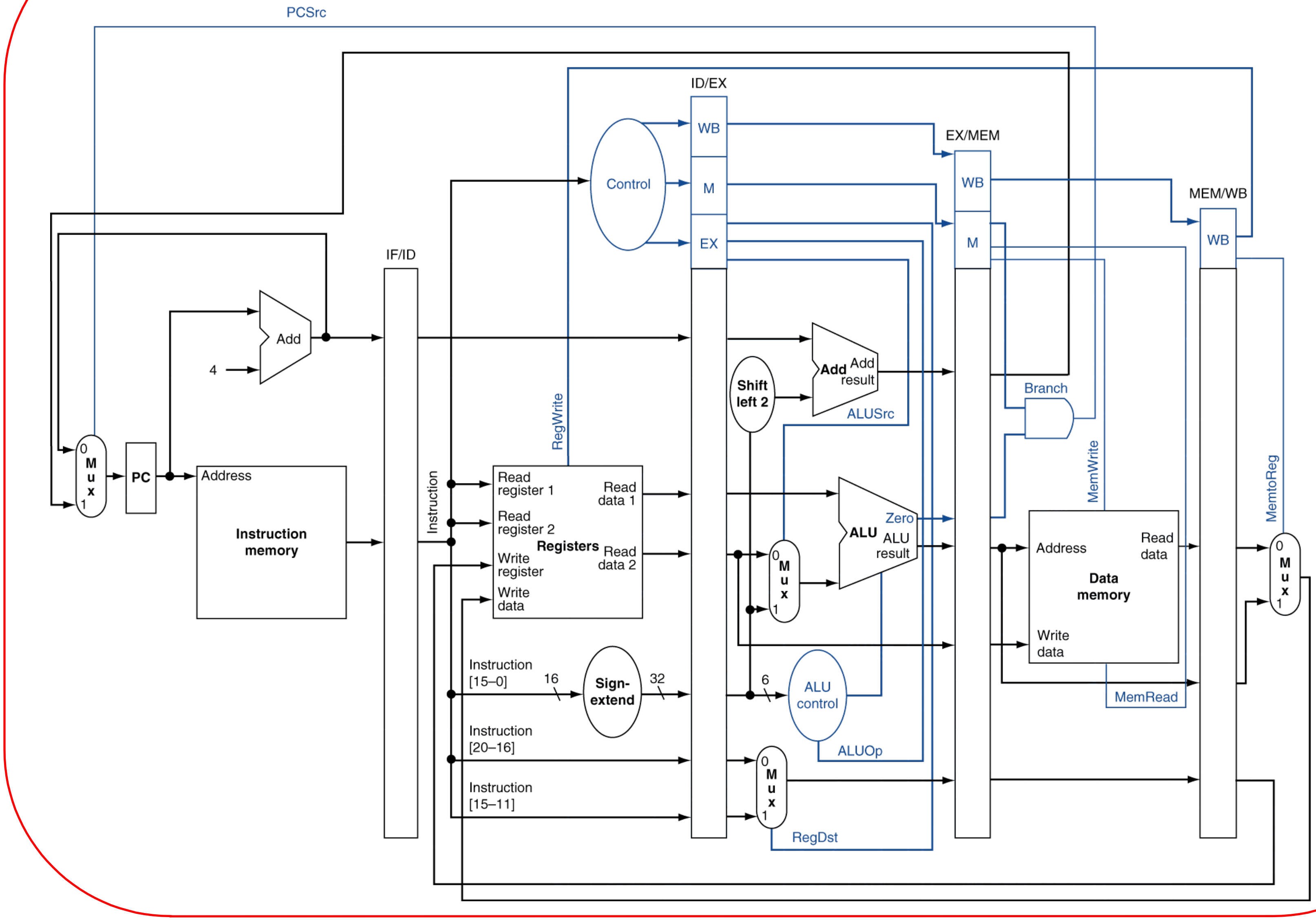
Control for the pipelined datapath

- Mostly same as for single-cycle datapath
  - Control signals are needed at corresponding stage of execution
  - No new control signals required
- But control for stage x depends on instruction at stage x
  - Control for x is decided in ID
  - Control signal is passed to pipeline registers with the instruction

Passing control down the pipeline



Passing control down the pipeline (full version)



Summary

- Pipelined processors are generally faster than single-cycled processors
  - Provided the pipeline is kept full
  - The latency of an individual instruction may be greater
- Pipeline registers
  - isolate stages
  - hold information produced in previous cycle
  - get written for every cycle
  - and control hand-over of partially completed instructions
- Pipeline operation
- Control signals (values)
  - are determined in ID
  - pass through the pipeline registers as the instruction moves down the pipeline
  - are grouped and used by pipeline stage