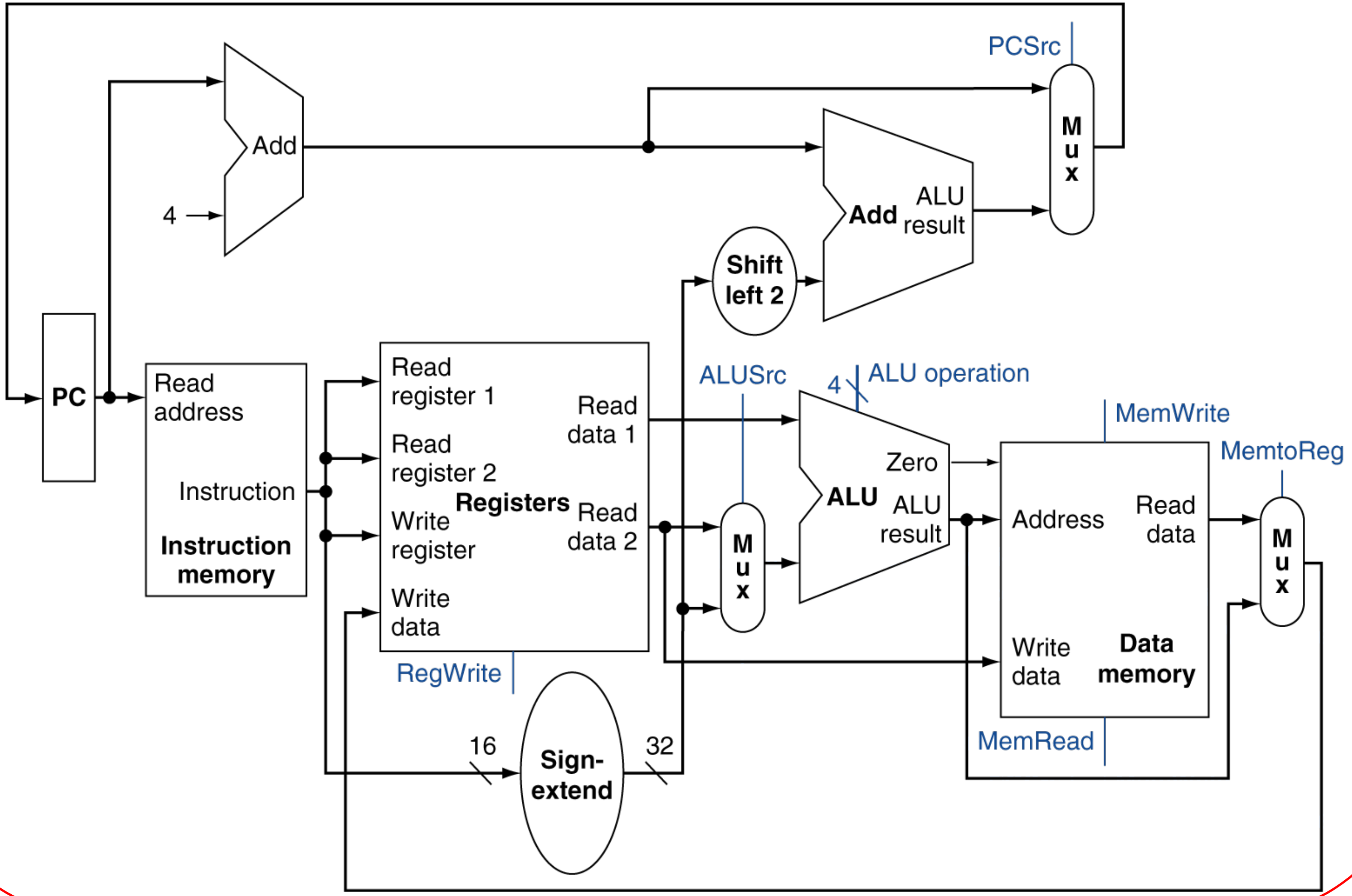
NWEN 242

Pipelining Overview and Datapath Review

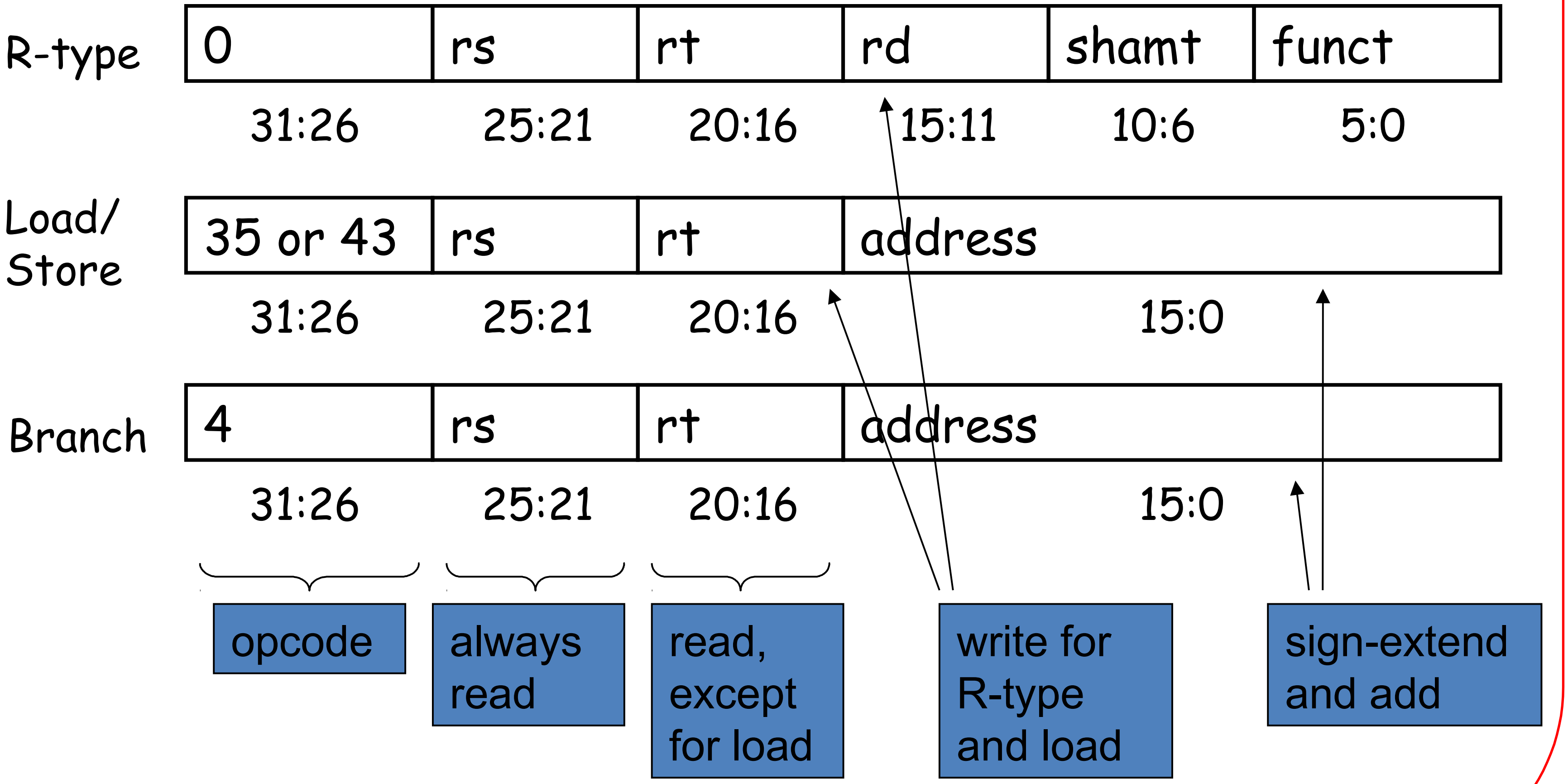
Agenda

- Review of datapath
- What is pipelining?
- Why do pipelining?
    - Single-cycle vs. pipelined performance
- What the pipeline registers do

Full Datapath



R-type and I-type Instruction Classes



R-type	0	rs	rt	rd	shamt	funct
	31:26	25:21	20:16	15:11	10:6	5:0
Load/ Store	35 or 43	rs	rt	address		
	31:26	25:21	20:16	15:0		
Branch	4	rs	rt	address		
	31:26	25:21	20:16	15:0		

opcode

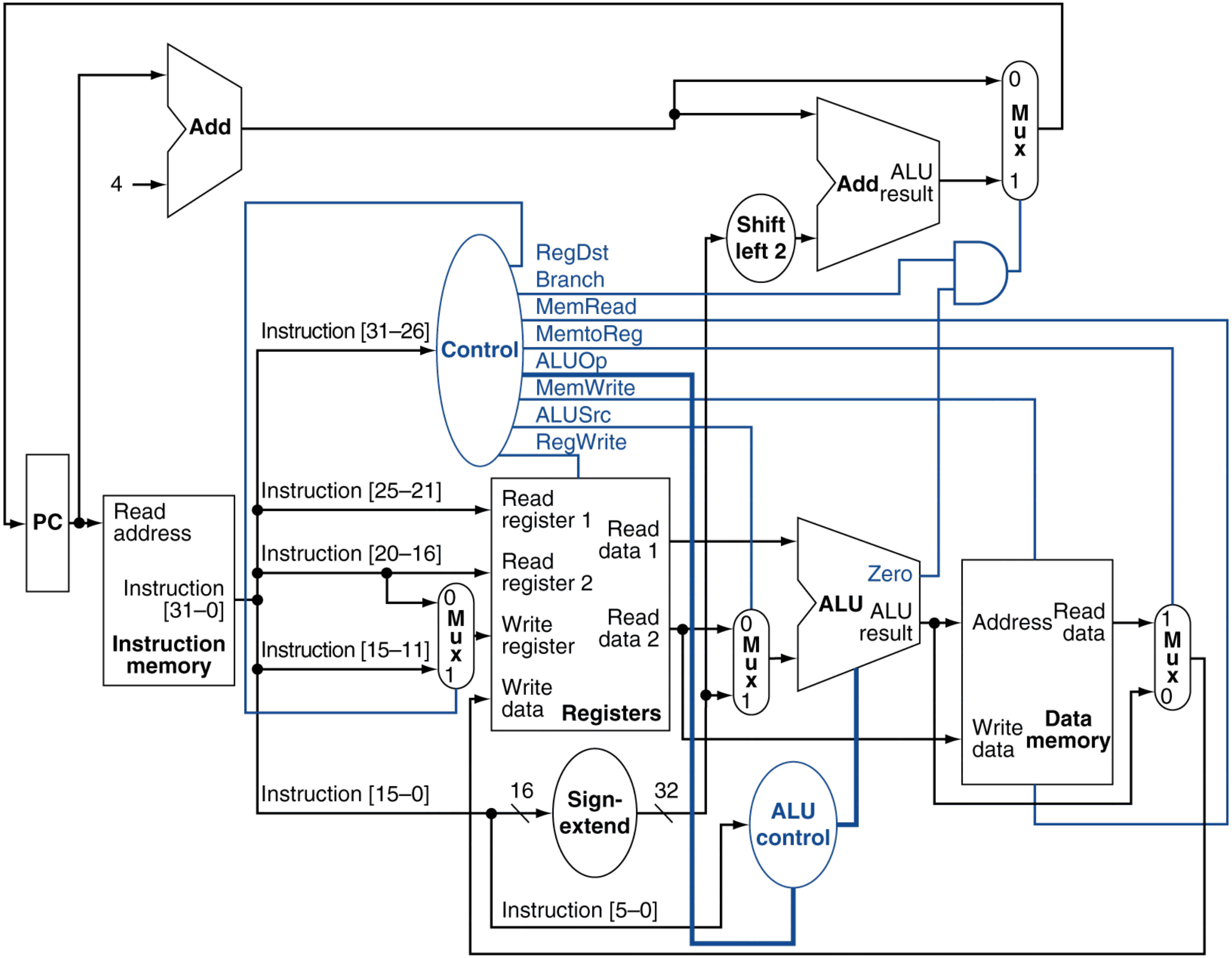
always read

read, except for load

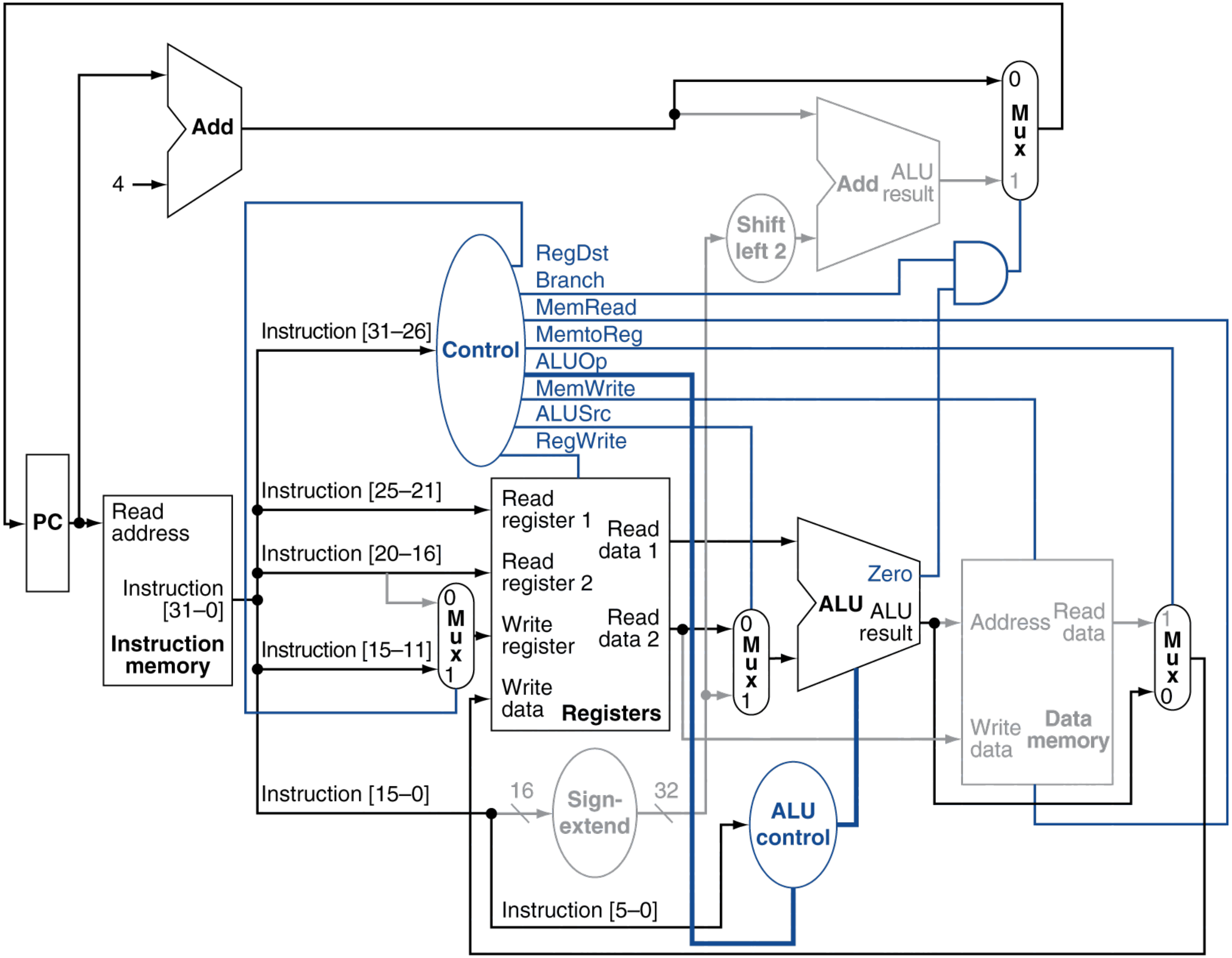
write for R-type and load

sign-extend and add

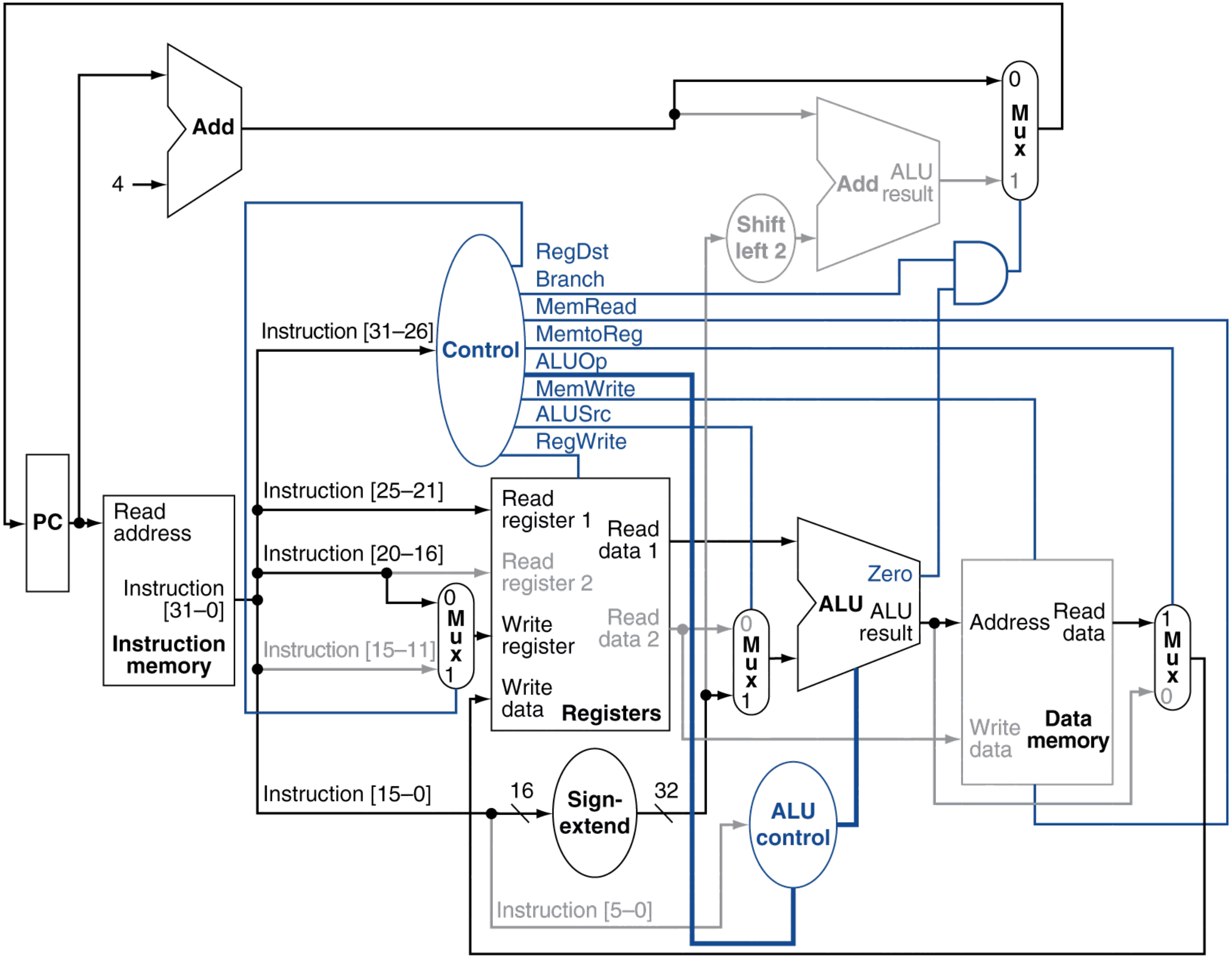
Datapath With Control



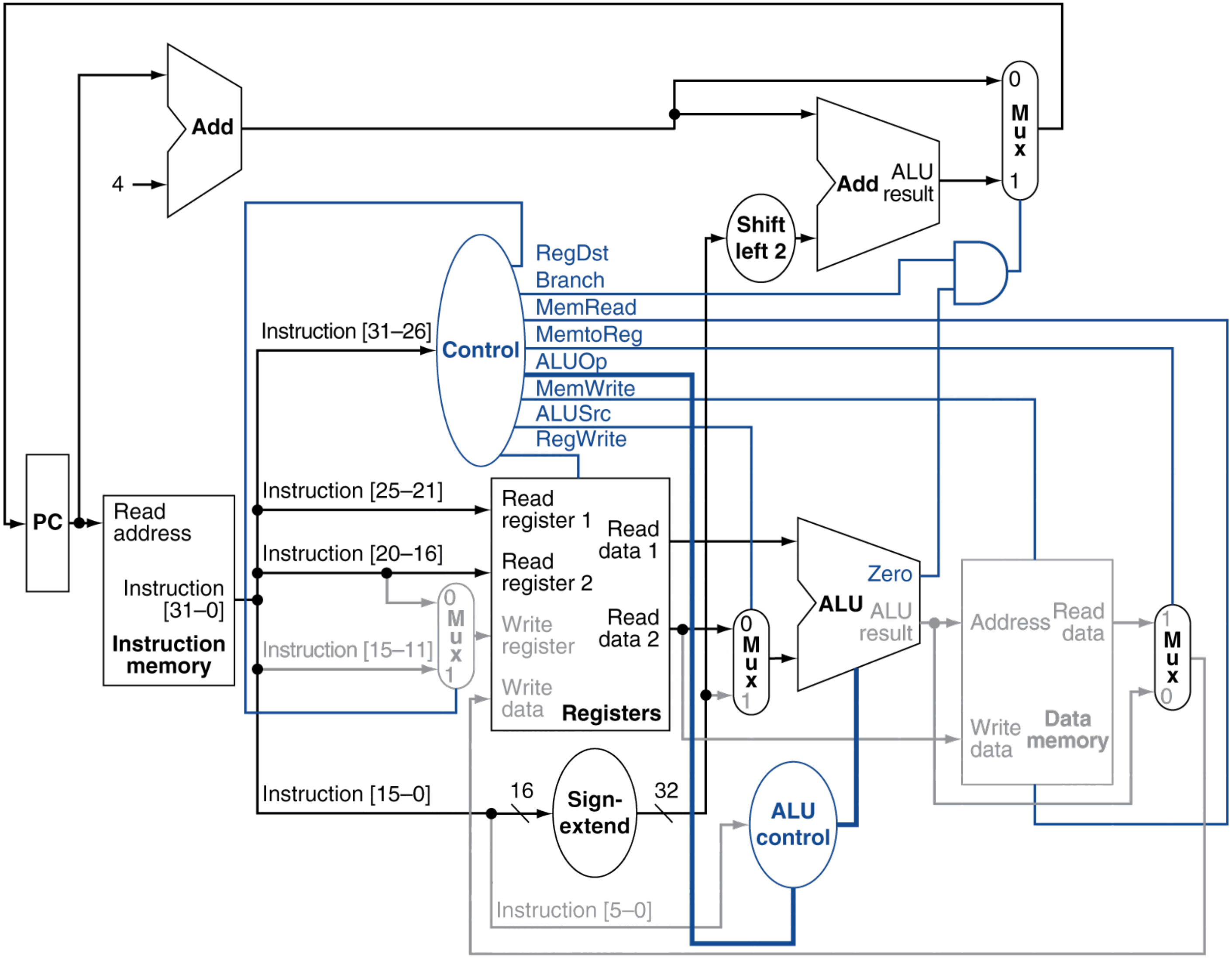
R-Type Instruction



Load Instruction



Branch-on-Equal Instruction



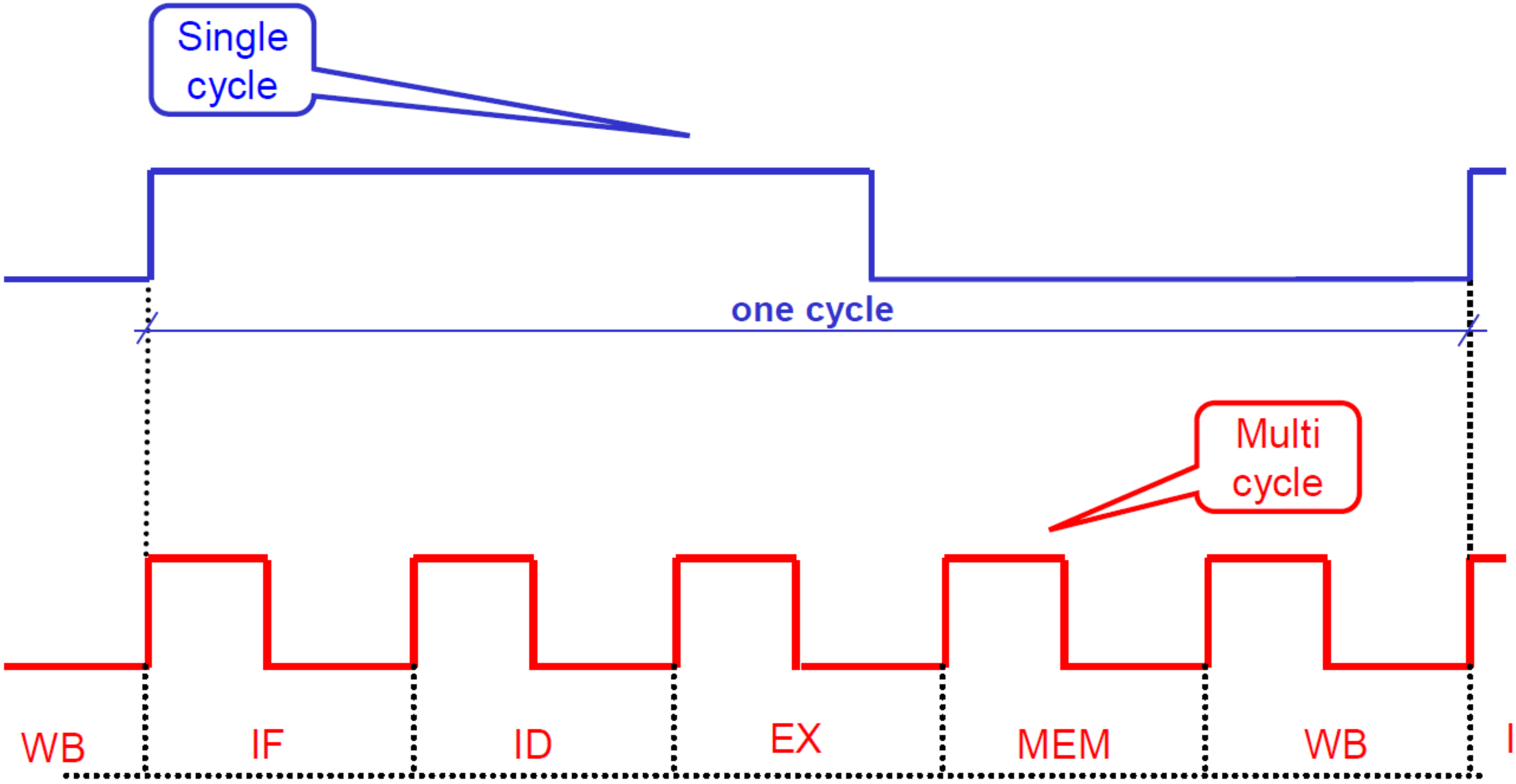
Single-cycle design OR ???

- Single-cycle processors
  - Easy to design
  - Everything happens during one processor clock period
  - But cycle length must accommodate longest instruction
    - e.g. lw takes 5 functional steps, beq takes 3 or 4 functional steps

- Multi-cycle processors
  - Several cycles are used for each instruction and longer instructions take more clock cycles
  - Require more complex control, but avoid idling

- **Pipelining** supports **multi-cycle design** and achieves **instruction-level parallelism**

One to multi-cycle conversion

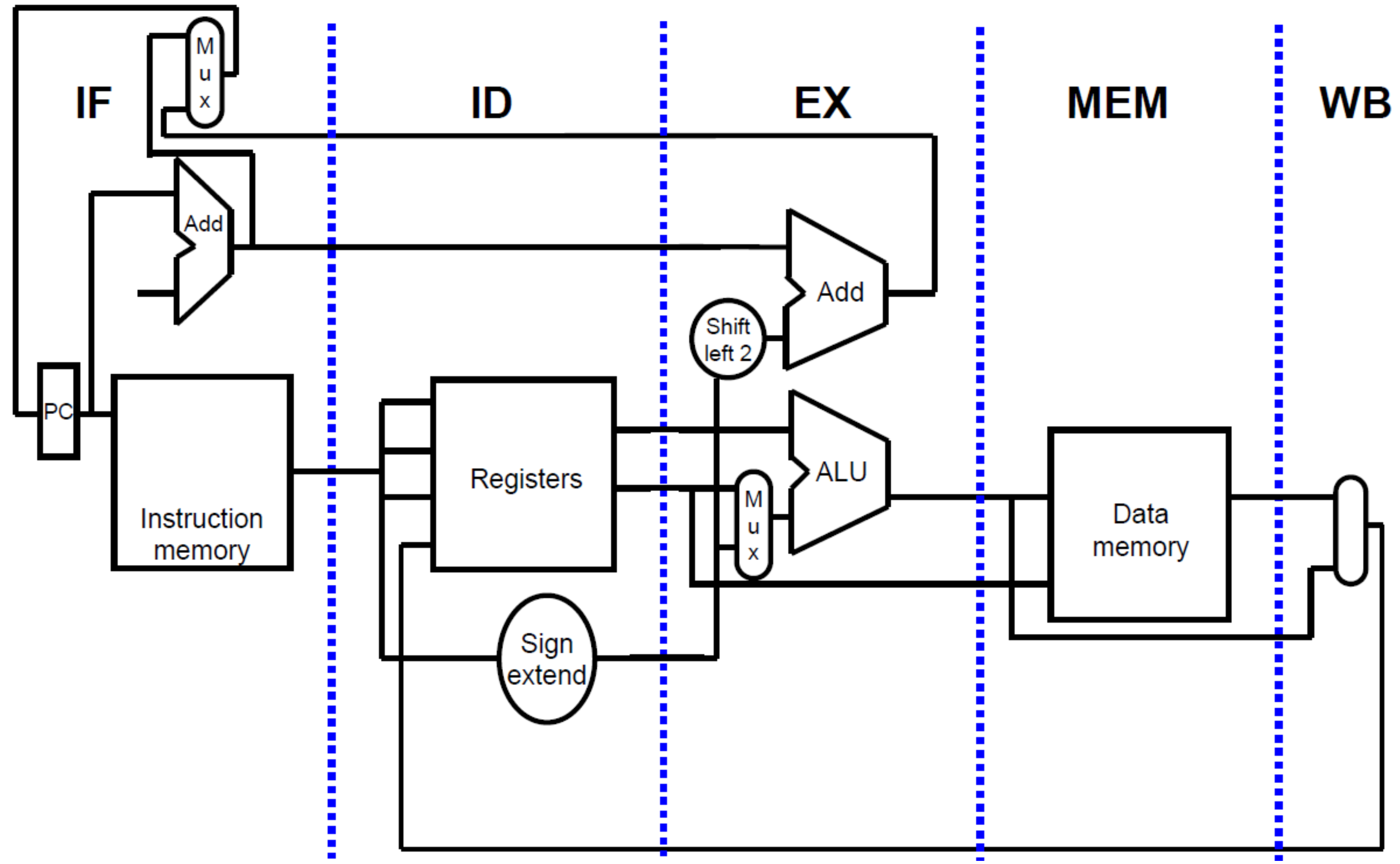


- How to decide the clock cycle length for a multi-cycle design?

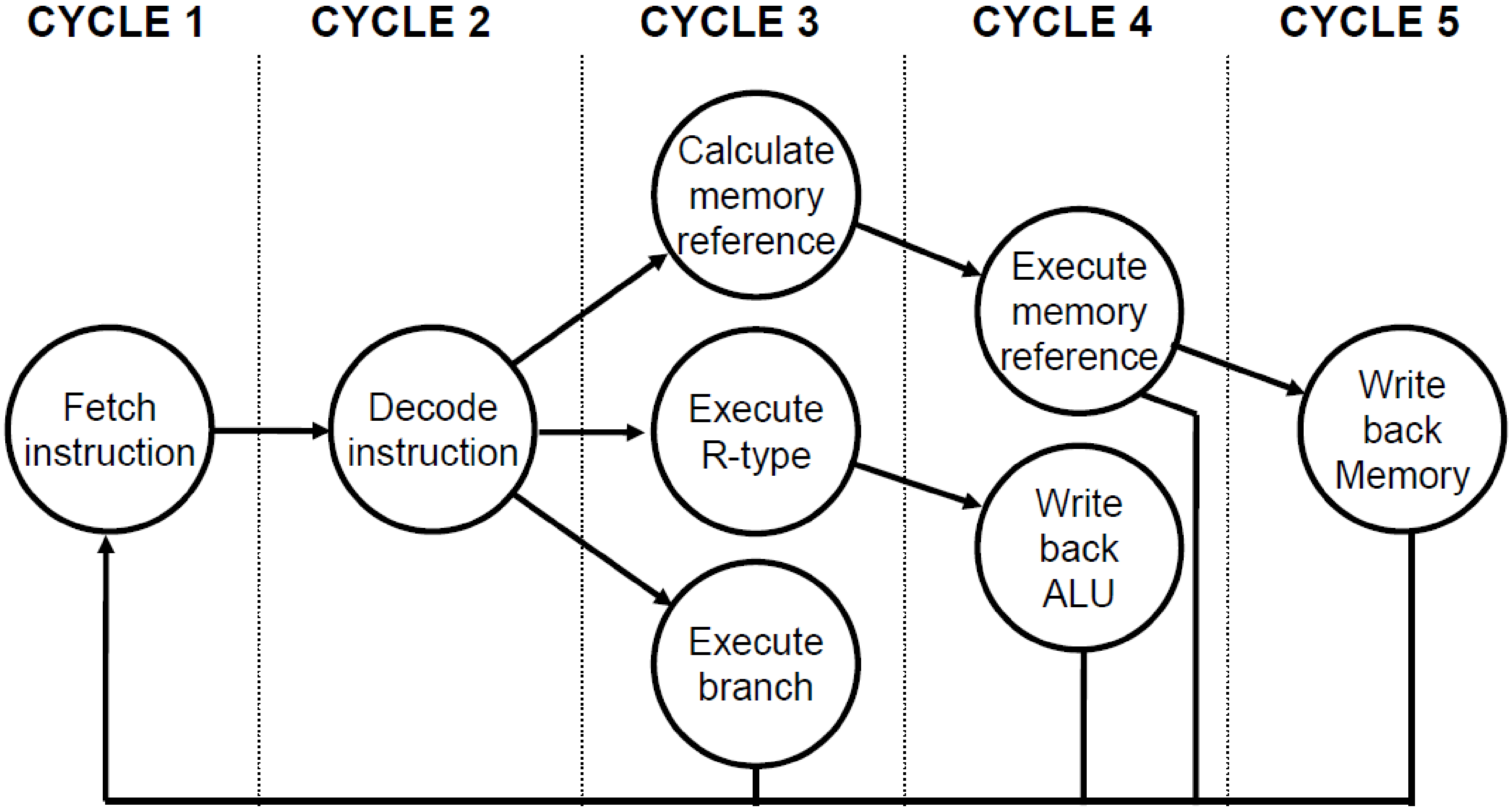
Overview of the cycles

- To execute an R-type, I-type, or J-type instruction
  - Cycle 1: instruction fetch (IF)
  - Cycle 2: instruction decode (ID)
    - decode instruction, read registers, sign extension of the immediate field
  - Cycle 3: execution (EX)
    - Compute branch target, test condition, and GOTO cycle 1
    - Compute memory address
    - Execute R-type
  - Cycle 4: memory access (MEM)
    - Perform memory access (if sw GOTO cycle 1)
    - Write result of R-type to register and GOTO cycle 1
  - Cycle 5: write back (WB)
    - Write result of memory access to register and GOTO cycle 1
    - lw only

Multiple cycles on single cycle datapath



Operations in multi-cycle time division

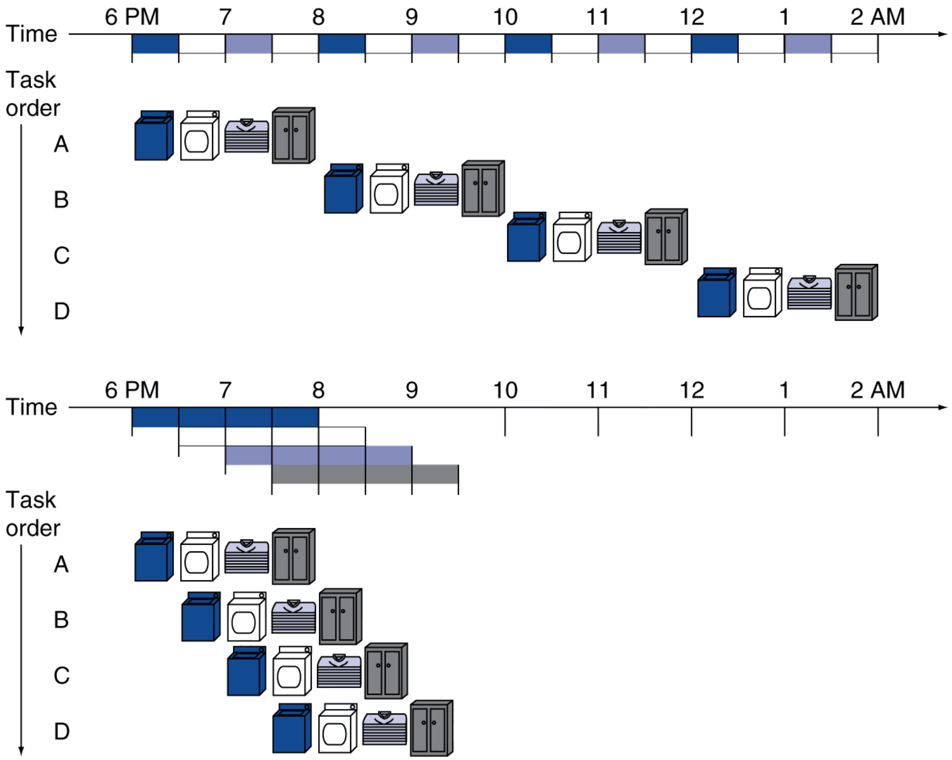


Introduction to pipelining

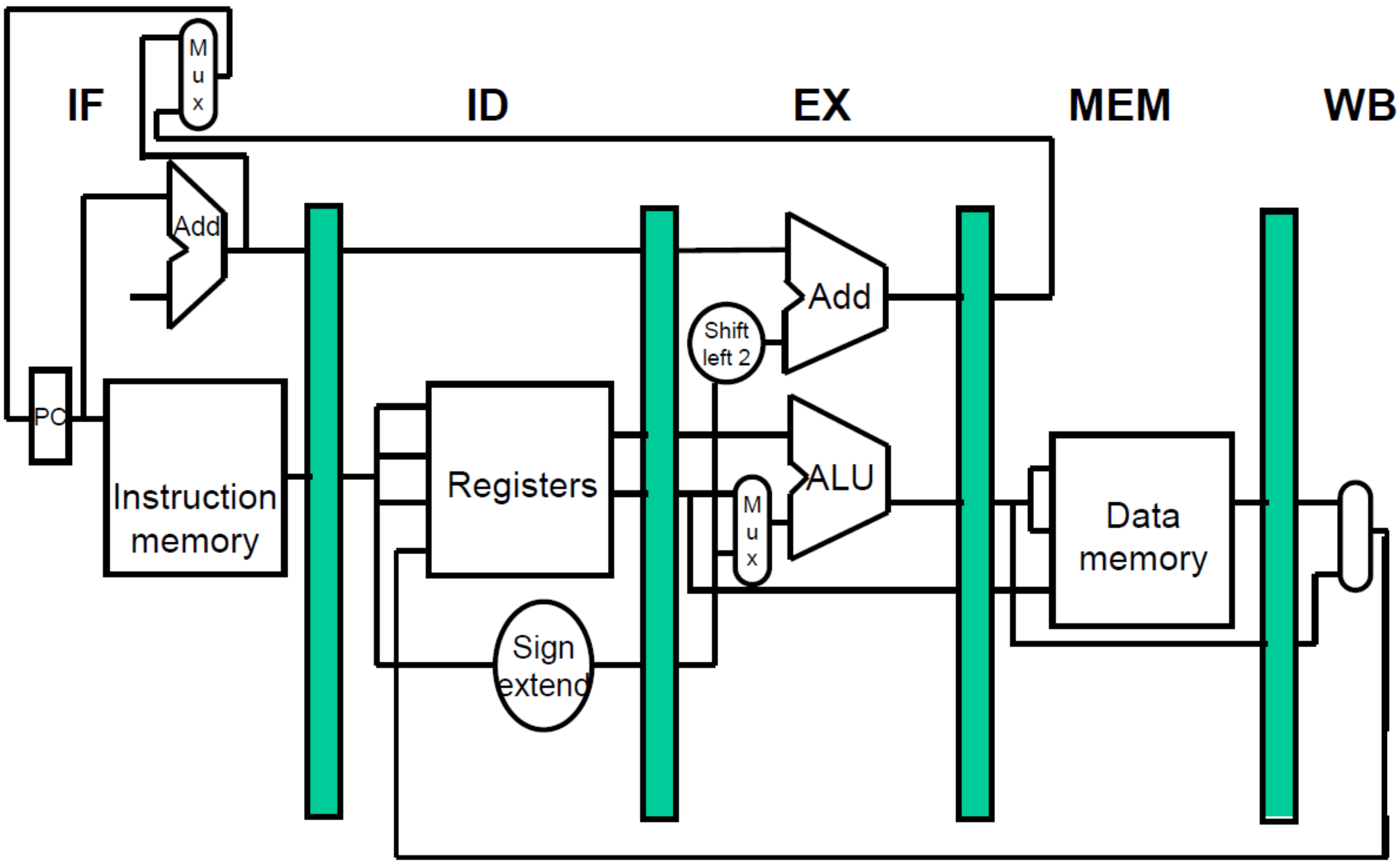
- Involves having several instructions in the processor all at once
  - Each at a different stage of its execution
  - Analogous to a pipelined laundry
    - Several tasks in the laundry at the same time
    - Each task is using a different facility and therefore at a different stage

Introduction to pipelining

- Involves having several instructions in the processor all at once
  - Analogous to a pipelined laundry
    - Several tasks in the laundry at the same time
    - Each task is using a different facility and therefore at a different stage



5-stage pipelined datapath



Benefits of pipelining

- Pipelining gives better performance than single-cycle design
- Individual instructions take the same time
- Improved throughput

Summary

- A five-stage pipeline (IF/ID/EX/MEM/WB)
- Pipelined processors are generally faster than single-cycled processors
  - For long programs
  - Provided the pipeline is kept full
  - But the latency of an individual instruction may be greater than for a single-cycled datapath