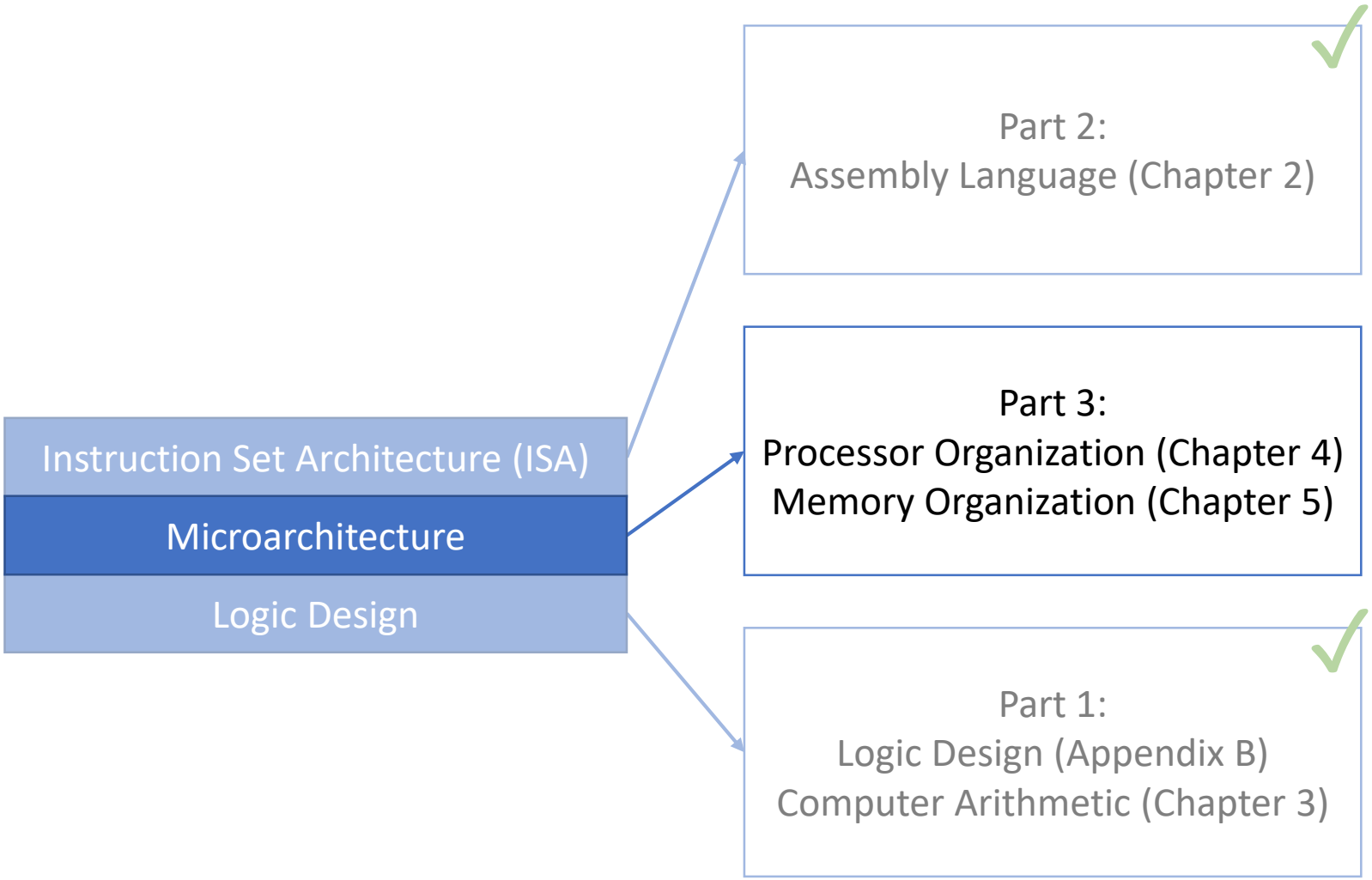
Lecture 23: Pipelining the Datapath

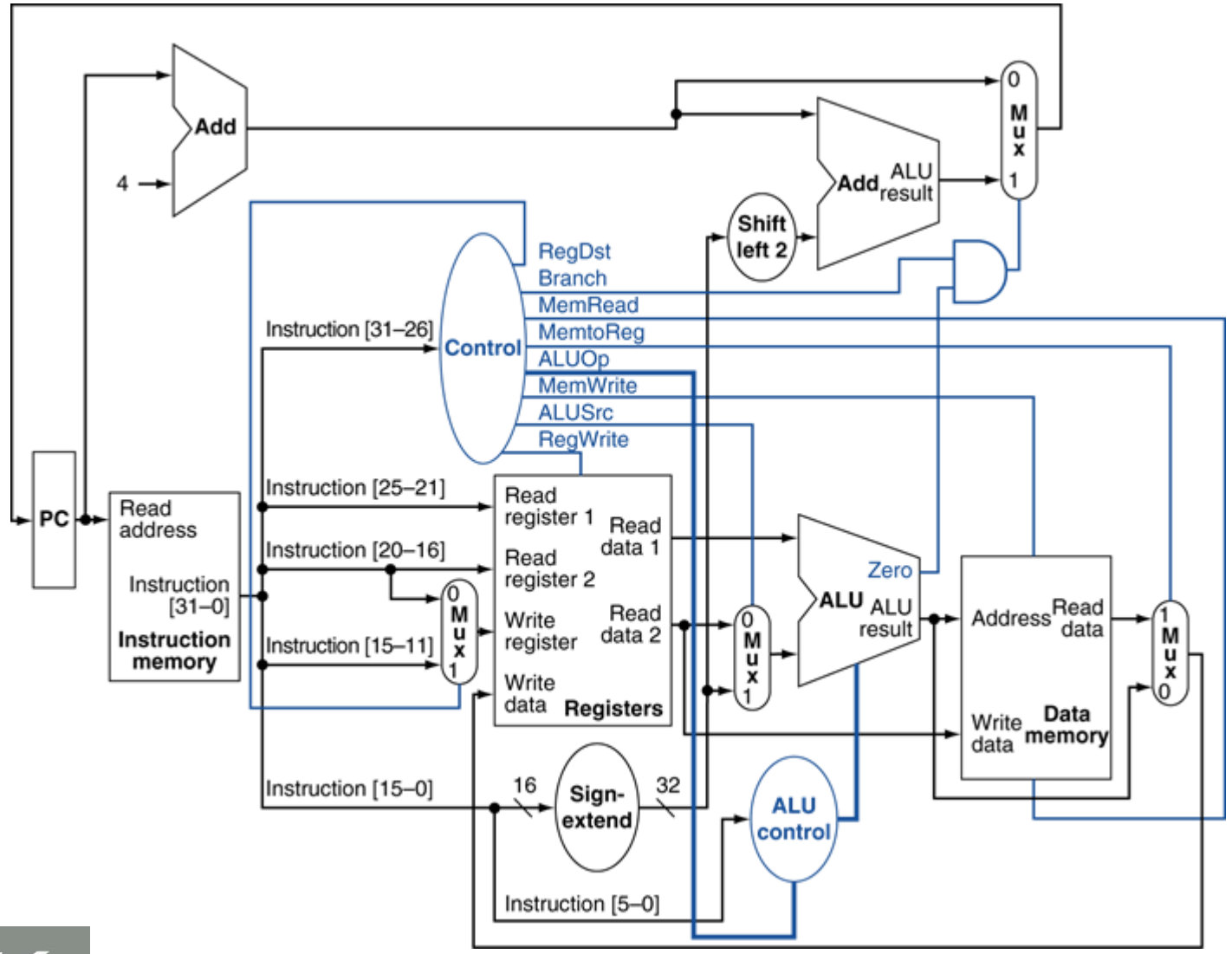
CMPS 221 – Computer Organization and Design

Adapted by Izzat El Hajj from textbook slides for Computer Organization and Design, by David Patterson and John Hennessy, Morgan Kaufmann.

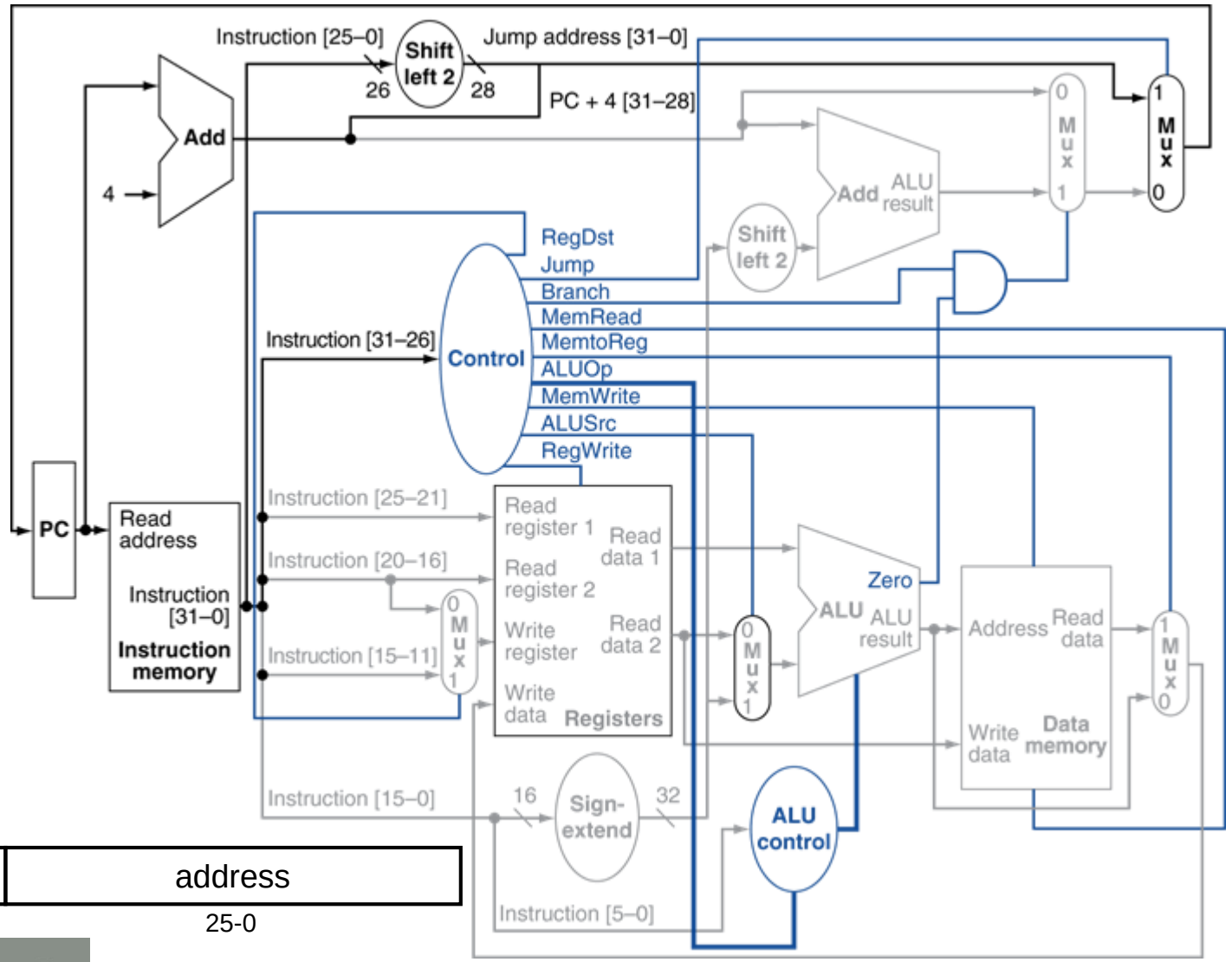
Last Time: Datapath Control Signals



Datapath with Control Signals

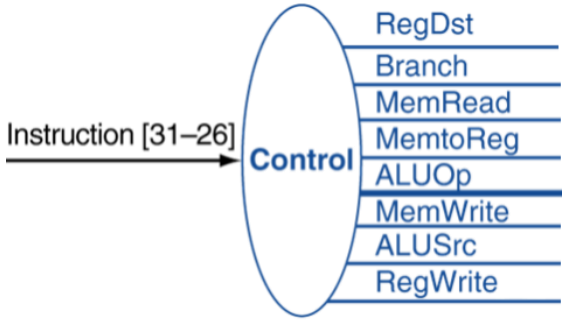


Jump Instructions



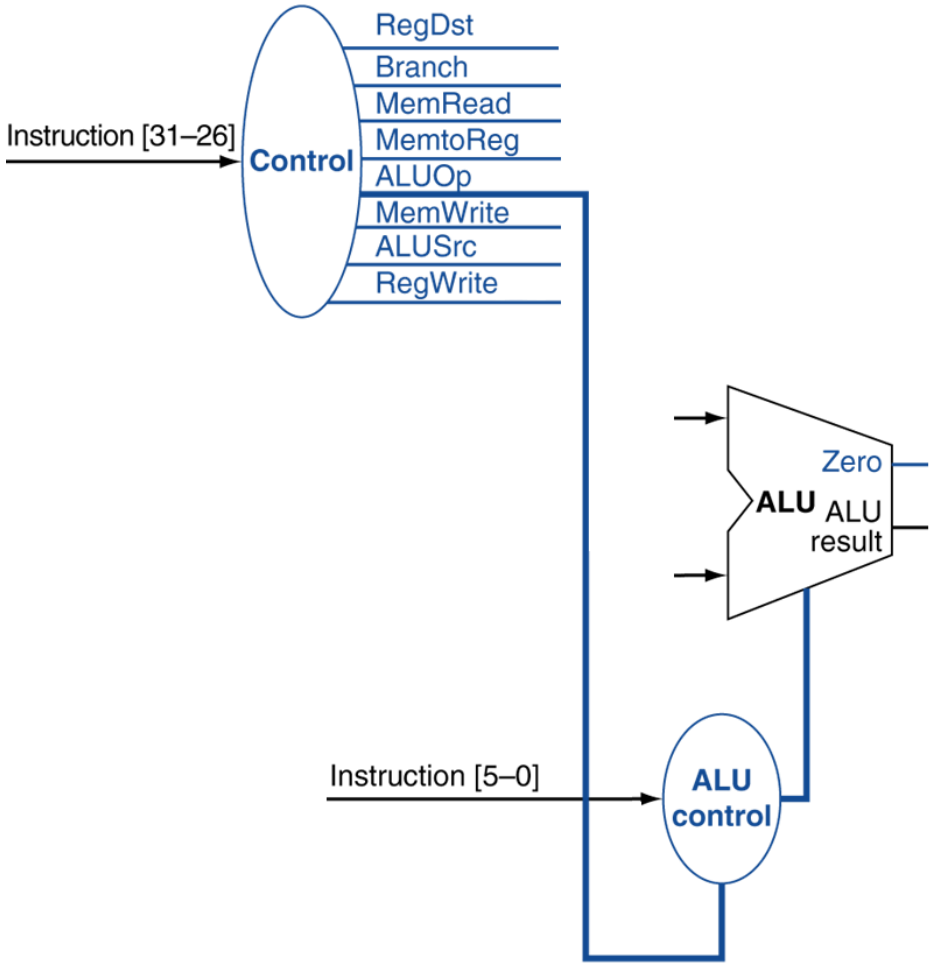
Jump address = PC+4 [31-28] : address × 4

Main Control Unit



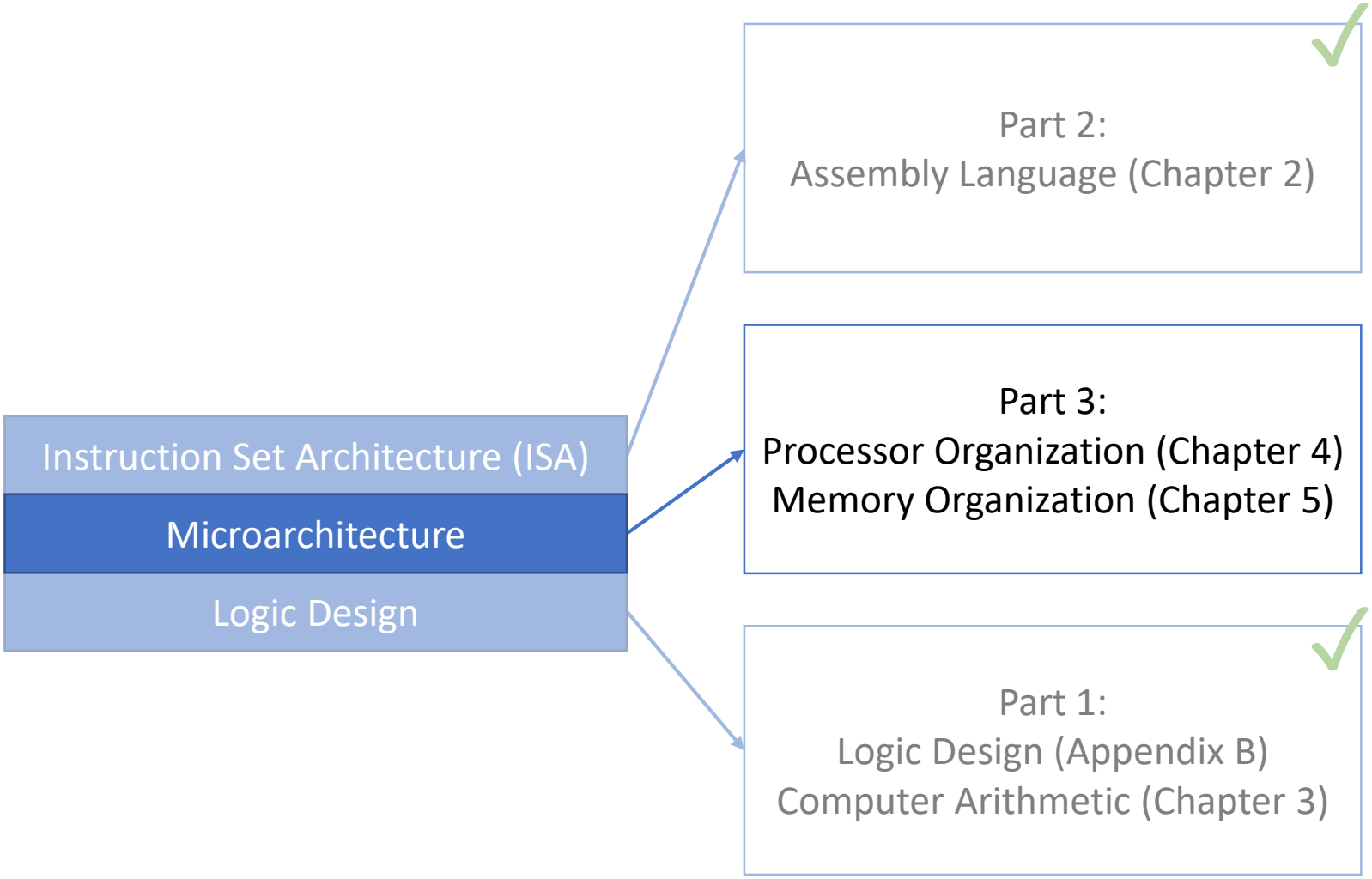
Instr[31-26]	RegDst	RegWrite	ALUSrc	ALUOp	MemRead	MemWrite	MemtoReg	Branch	Jump
R-type (000000)	1	1	0	funct (10)	0	0	0	0	0
addi (001000)	0	1	1	add (00)	0	0	0	0	0
lw (100011)	0	1	1	add (00)	1	0	1	0	0
sw (101011)	d	0	1	add (00)	0	1	d	0	0
beq (000100)	d	0	0	sub (01)	0	0	d	1	0
j (000010)	d	0	d	d	0	0	d	d	1

ALU Control Unit



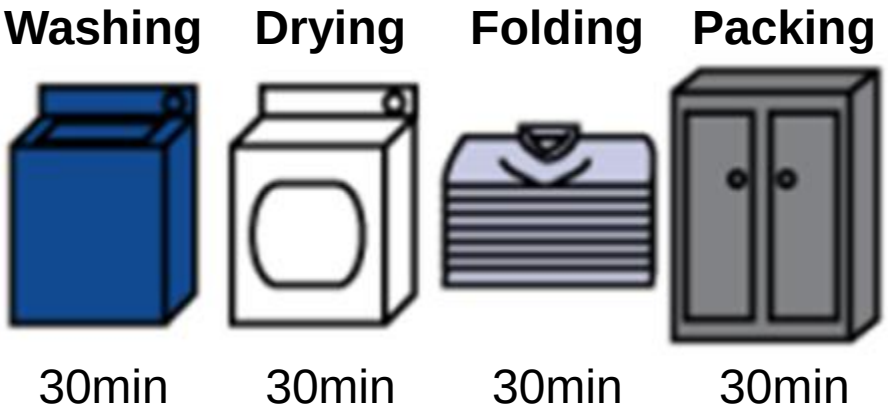
ALUOp	Instr[5-0]	ALU Control Line
funct (10)	add (100000)	add (0010)
	sub (100010)	sub (0110)
	slt (101010)	slt (0111)
add (00)	d	add (0010)
sub (01)	d	sub (0110)

Today: Pipelining



Laundry Analogy

- Steps for doing laundry:

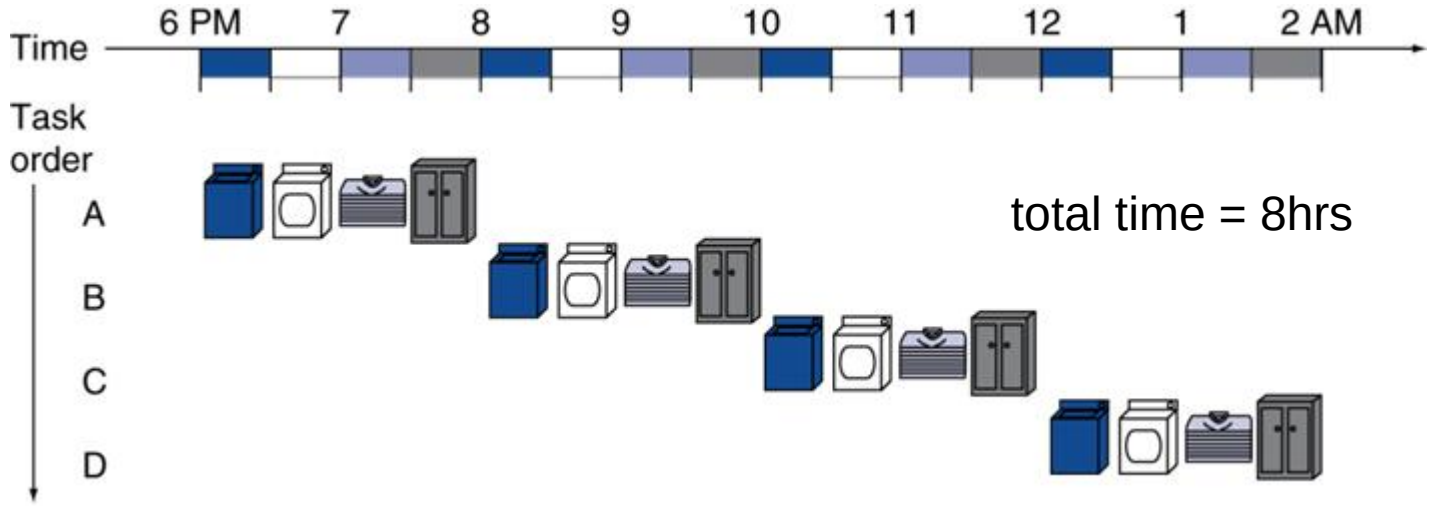


  - Time for one person to do their laundry: 2hrs

- How long does it take for 4 roommates to do their laundry?

Laundry without Pipelining

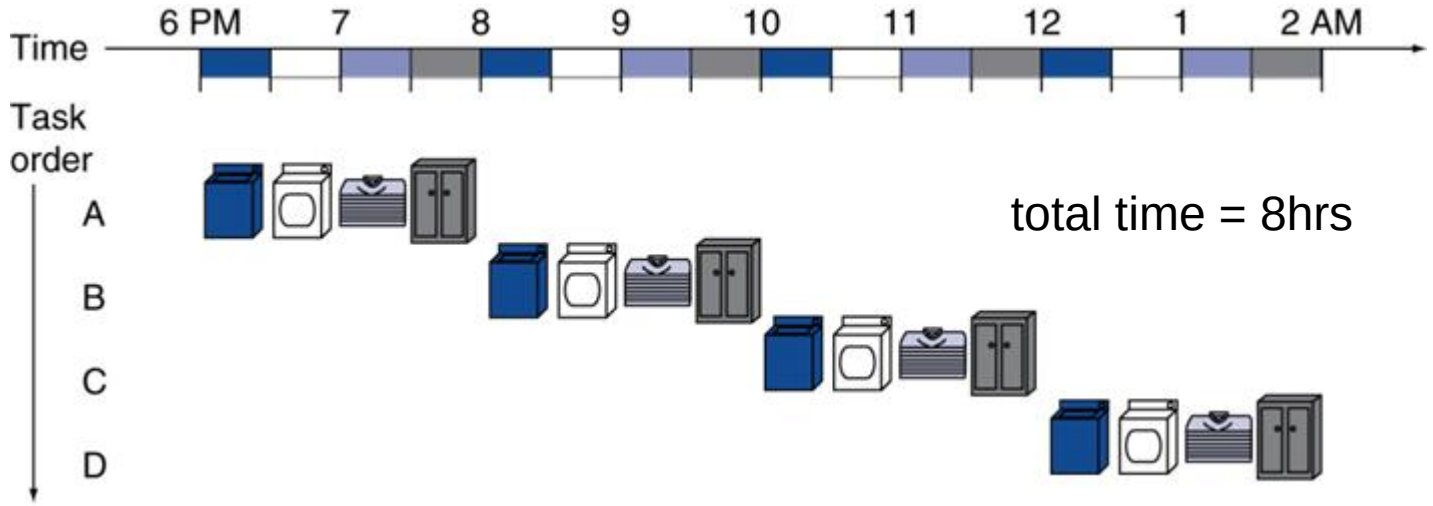
Without Pipelining



total time = 8hrs

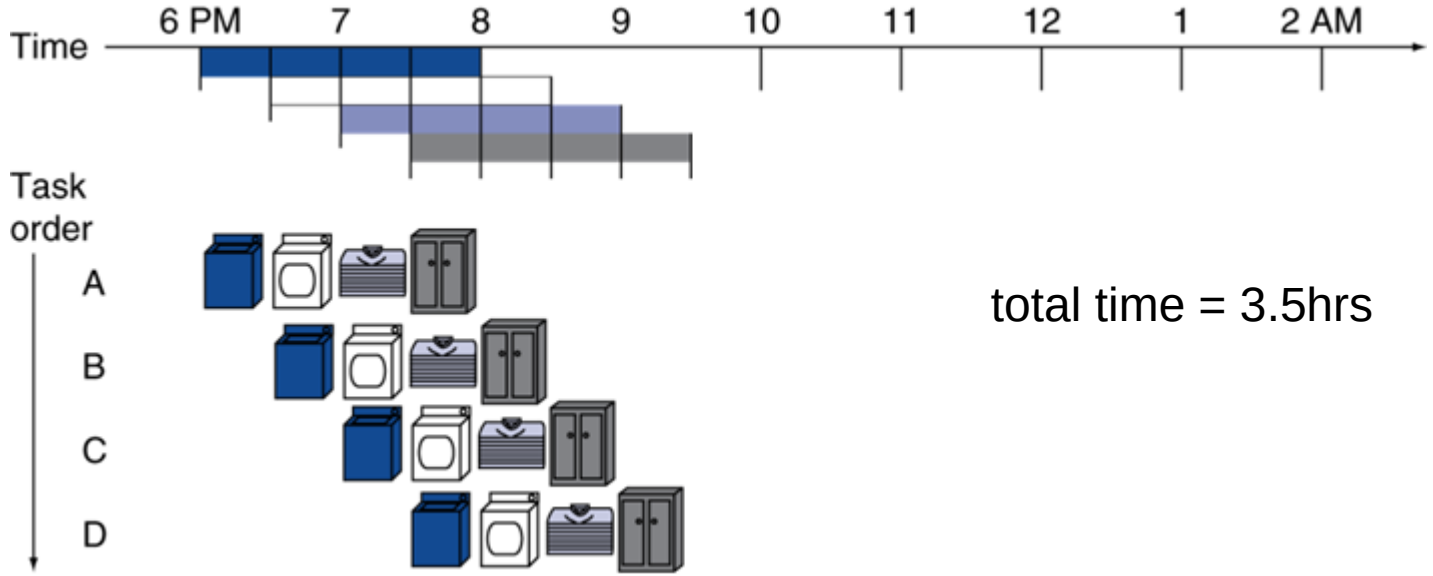
Laundry with Pipelining

Without Pipelining



total time = 8hrs

With Pipelining



total time = 3.5hrs

Pipelining Speedup

- In general, if pipeline stages are balanced
  - i.e., all *S* stages take the same time t_s
  - Assume *N* tasks
  - Time without pipelining:

$$t_{nonpipelined} = N \cdot S \cdot t_s$$

  - Time with pipelining:

$$t_{pipelined} = (N + S - 1) \cdot t_s$$

  - Speedup from pipelining:

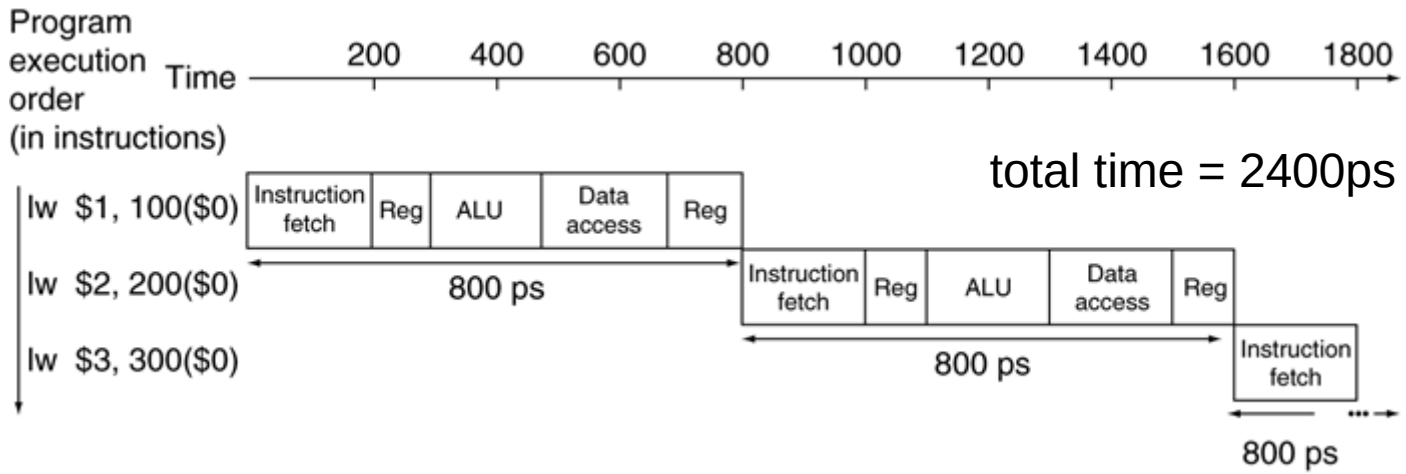
$$speedup = \frac{N \cdot S \cdot t_s}{(N + S - 1) \cdot t_s} \xrightarrow{large\ N} S$$

MIPS Pipeline

- MIPS datapath can be pipelined with five stages:
  1. IF: Instruction fetch from memory
  2. ID: Instruction decode & register read
  3. EX: Execute operation or calculate address
  4. MEM: Access memory operand
  5. WB: Write result back to register
- Stages are not balanced. Assume:
  - 100ps for ID and WB
  - 200ps for other stages

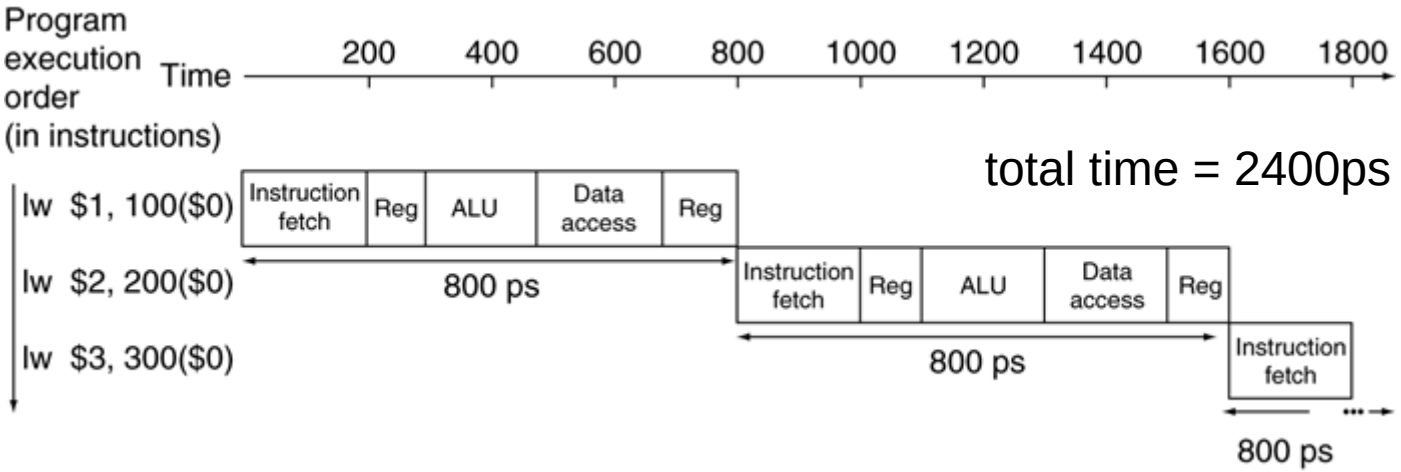
Performance without Pipelining

Without Pipelining

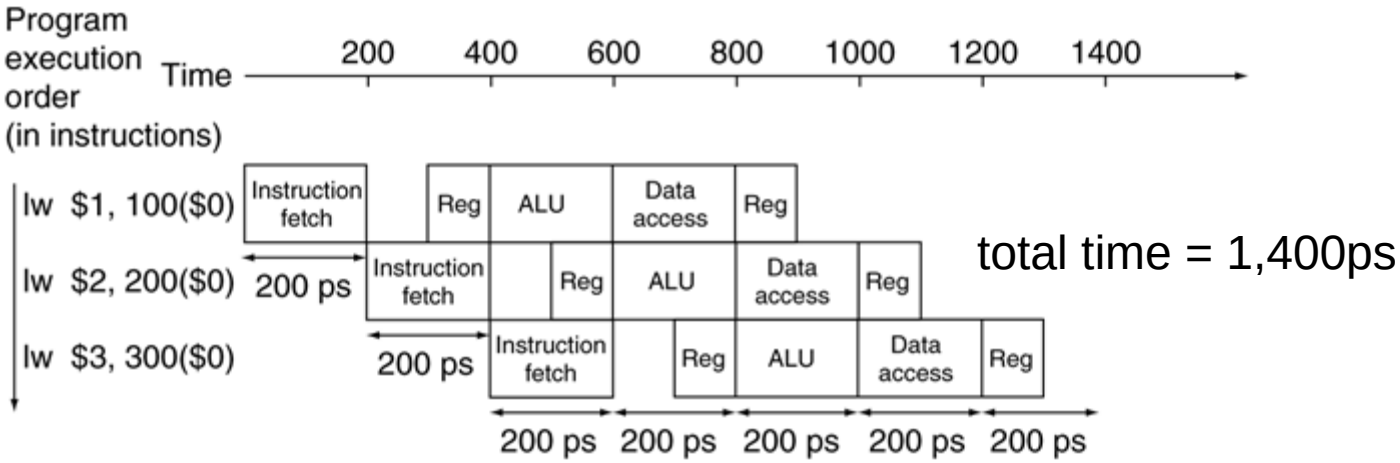


Performance with Pipelining

Without Pipelining



With Pipelining



MIPS Pipeline Speedup

- Stages are not balanced
  - Assume *N* instructions
  - Time without pipelining:

$$t_{nonpipelined} = N \cdot 800ps$$

  - Time with pipelining:

$$t_{pipelined} = (N + 5 - 1) \cdot 200ps$$

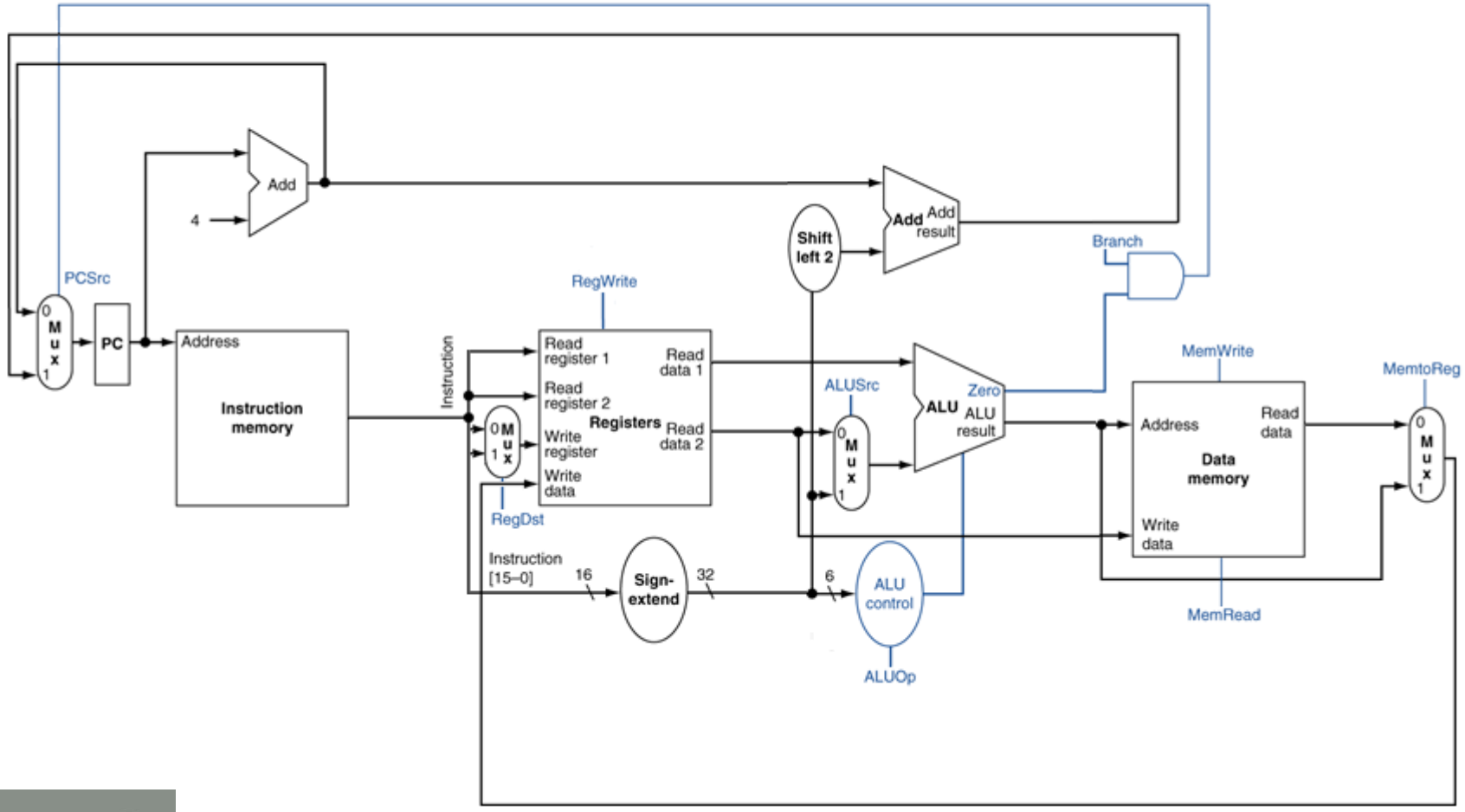
  - Speedup from pipelining:

$$speedup = \frac{N \cdot 800ps}{(N + 5 - 1) \cdot 200ps} \xrightarrow{large\ N} 4\times$$

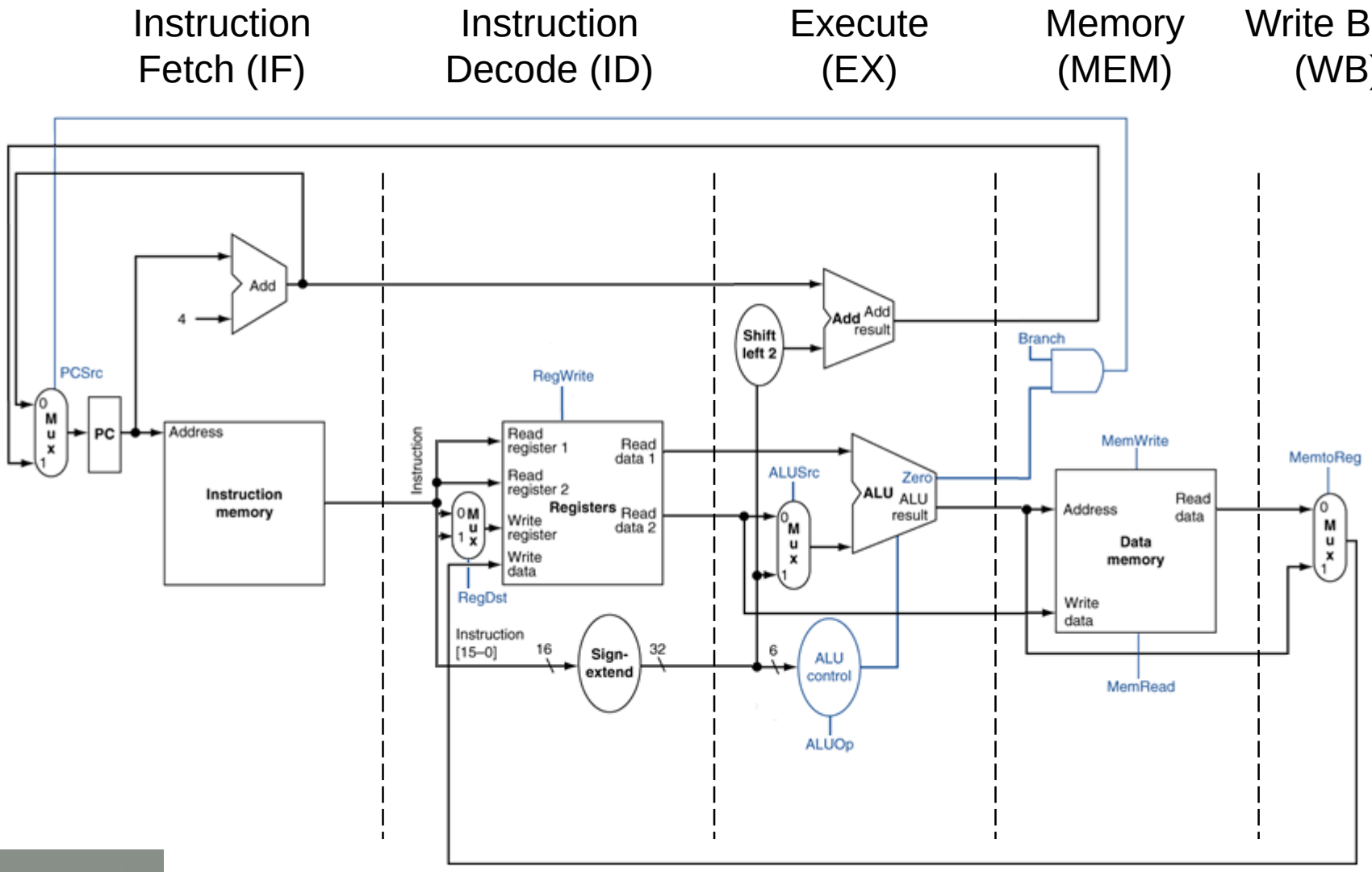
Pipelining and ISA Design

- MIPS ISA designed for pipelining
  - All instructions are 32-bits
    - Easier to fetch and decode in one cycle
    - c.f. x86: 1- to 17-byte instructions
  - Few and regular instruction formats
    - Can decode and read registers in one step
  - Load/store addressing
    - Can calculate address in 3rd stage, access memory in 4th stage
  - Alignment of memory operands
    - Memory access takes only one cycle

Recall: MIPS Datapath

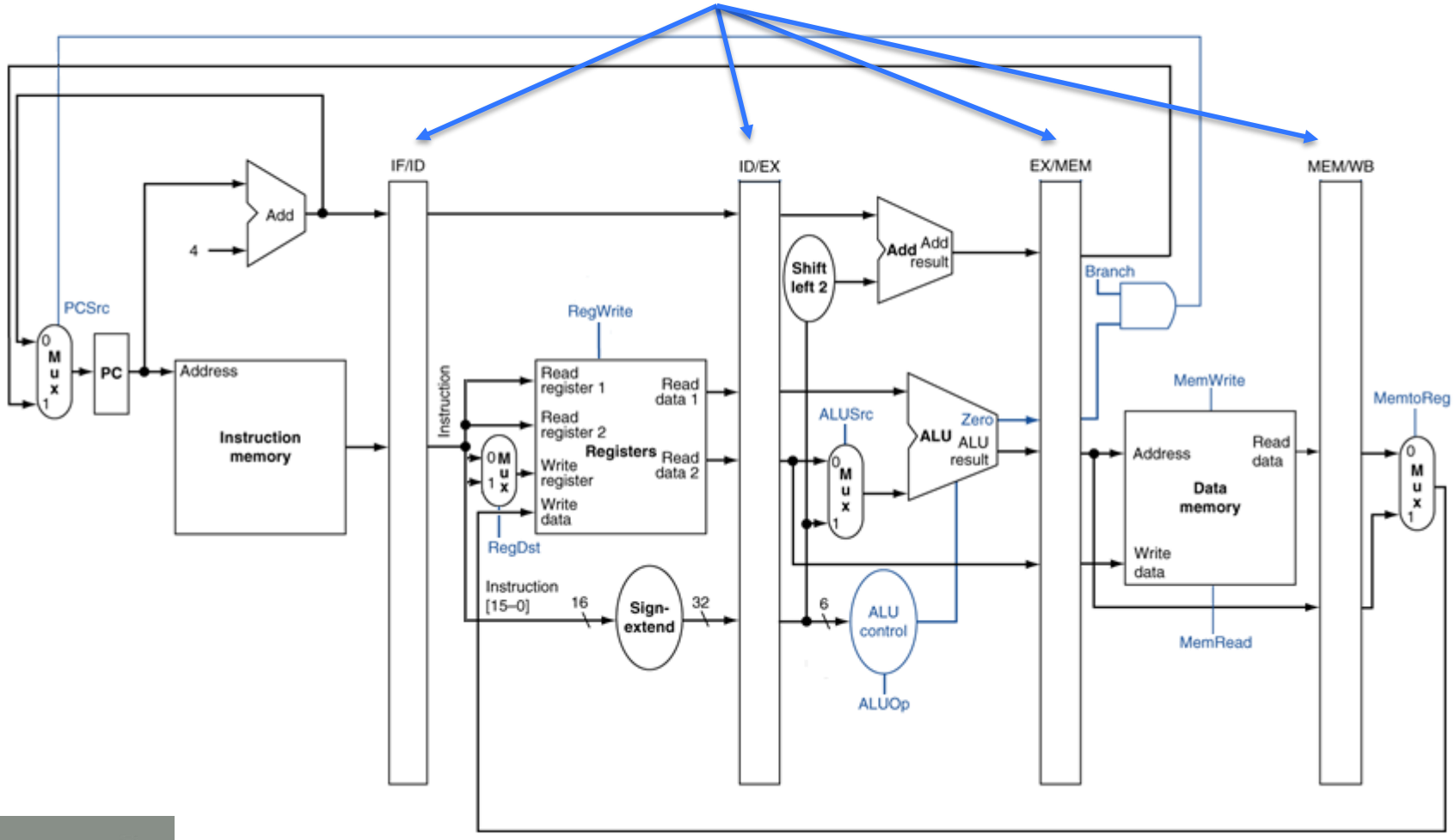


Datapath Stages

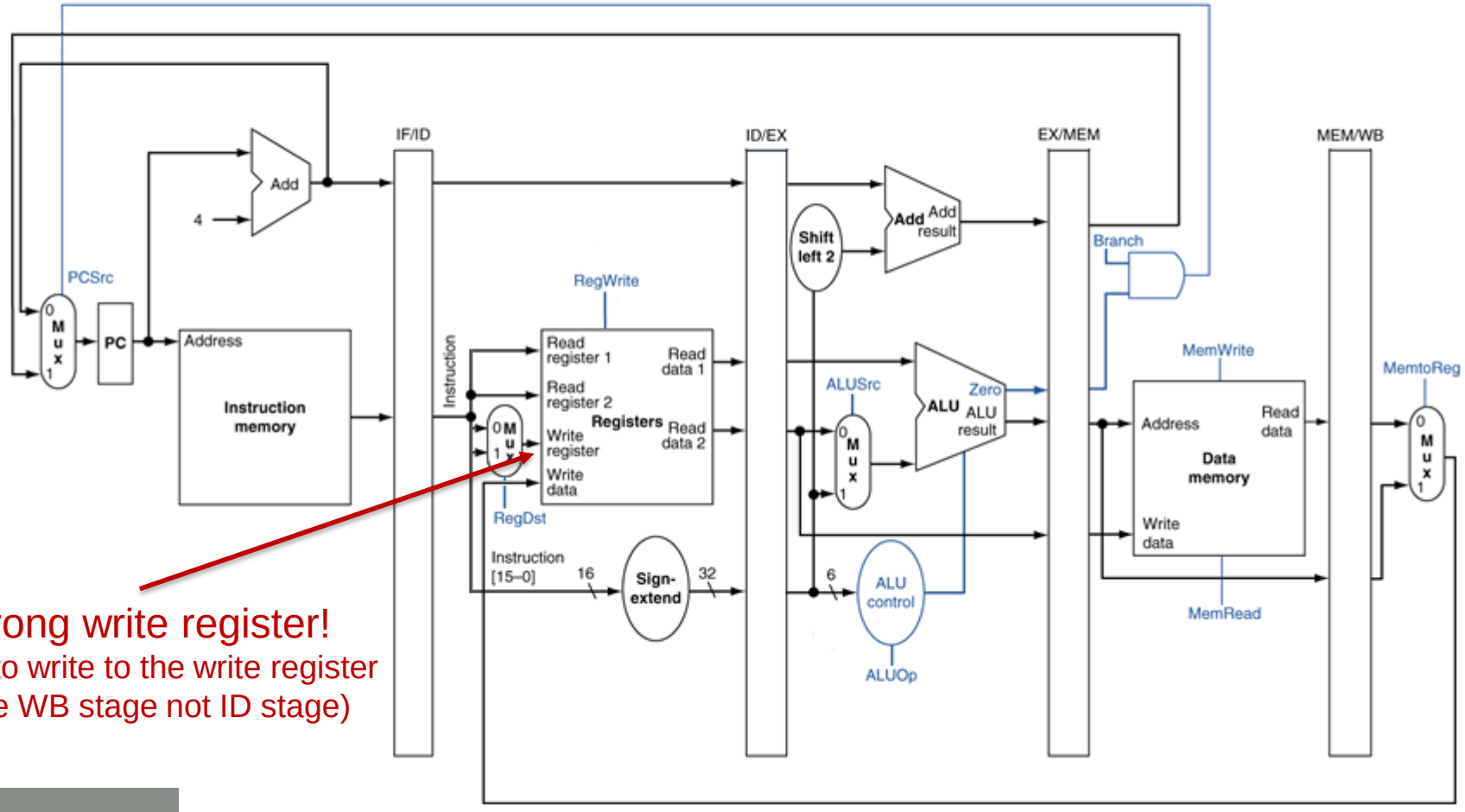


Pipelined Datapath

Pipeline registers ensure that the input to a stage is held for the entire cycle and updated on the next cycle

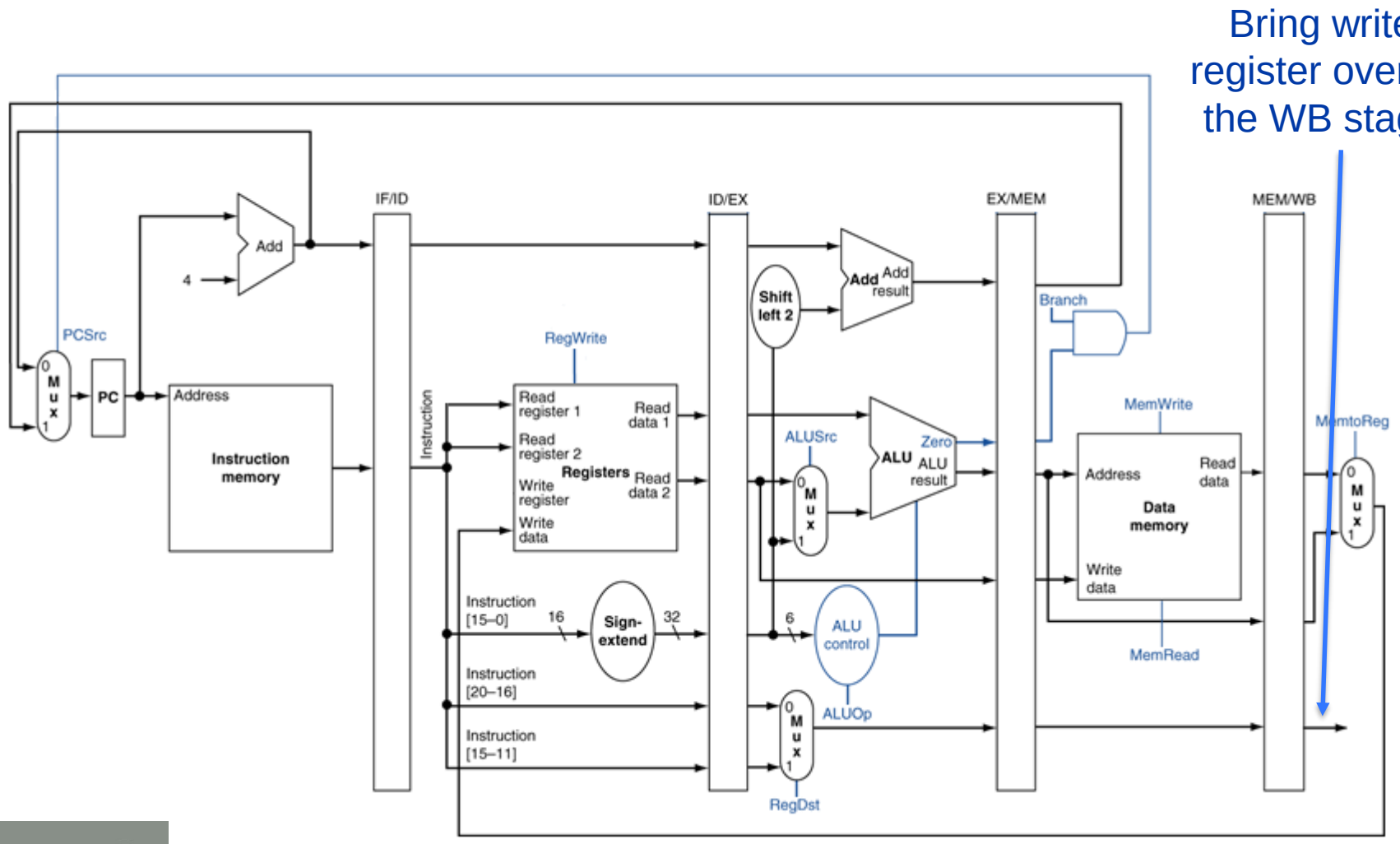


Fixing Writeback Destination

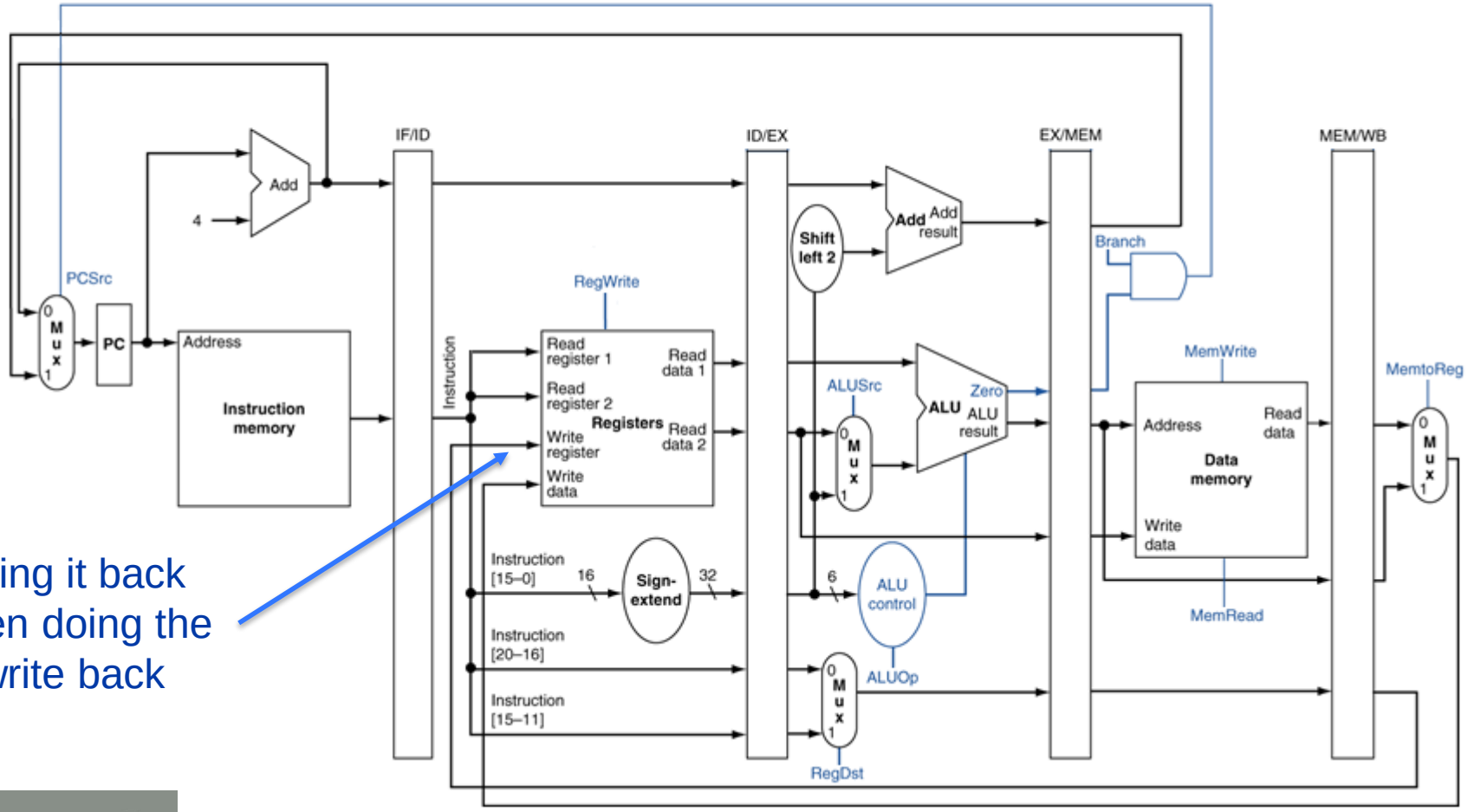


Wrong write register!
(need to write to the write register of the WB stage not ID stage)

Fixing Writeback Destination

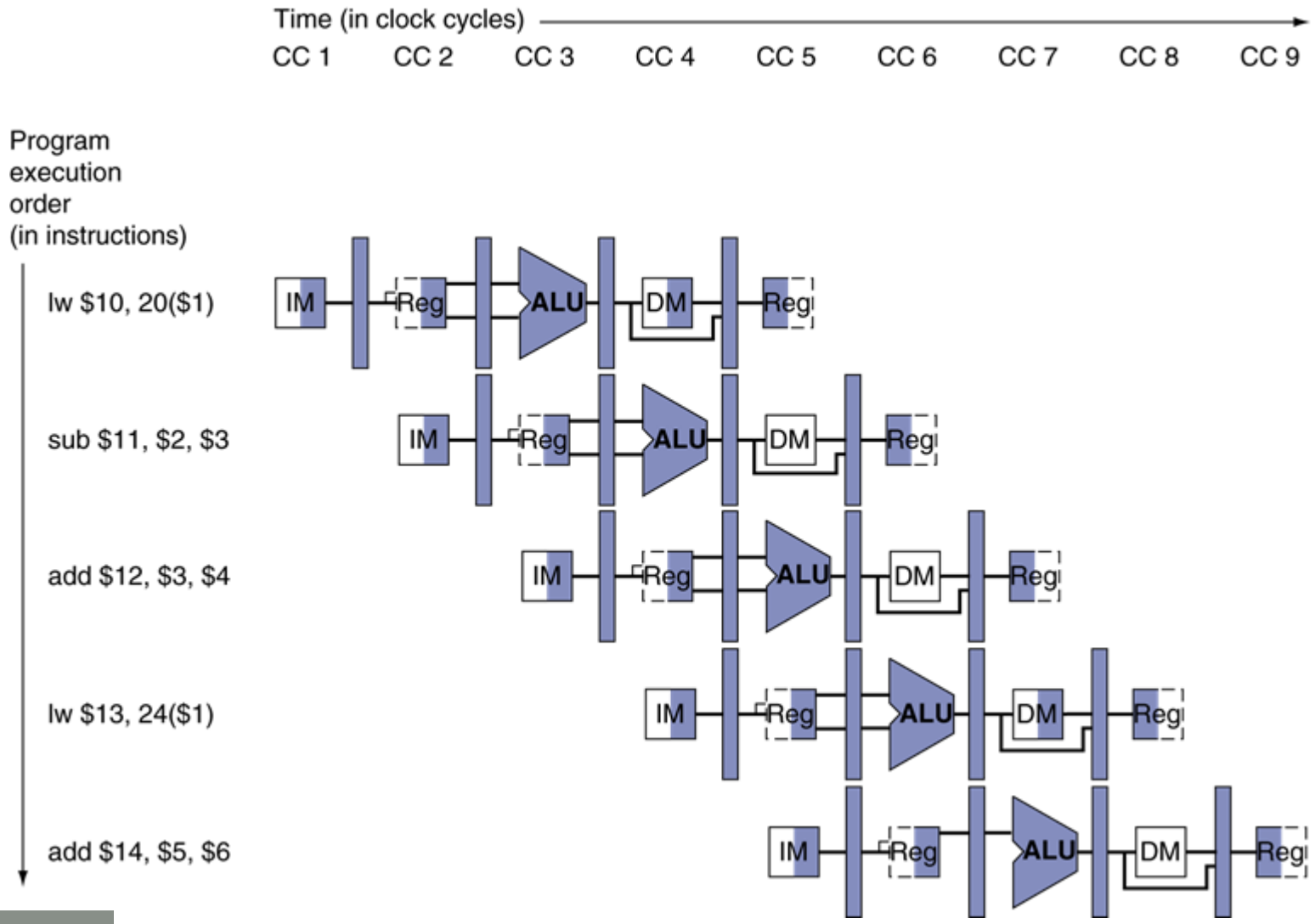


Fixing Writeback Destination

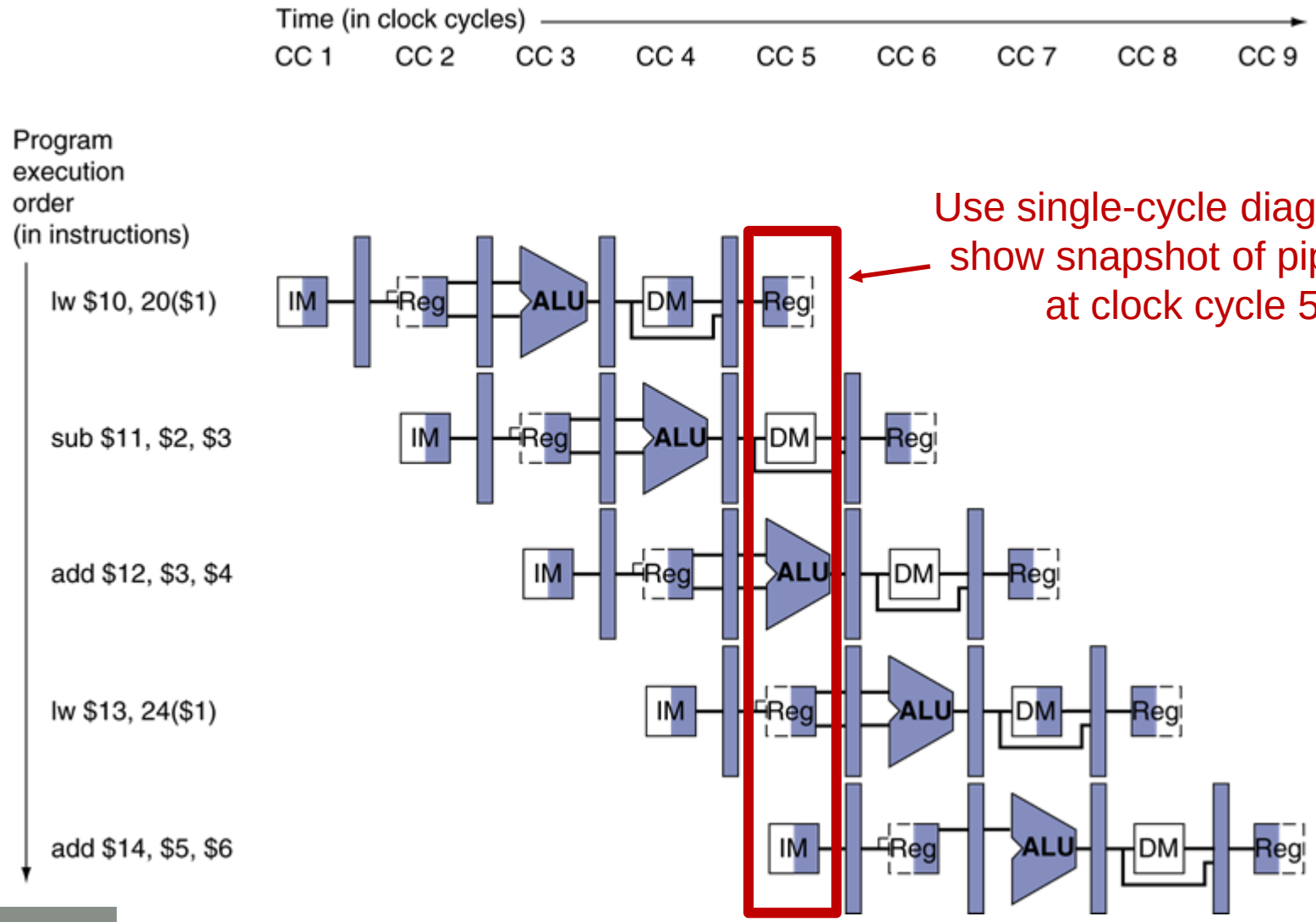


Bring it back when doing the write back

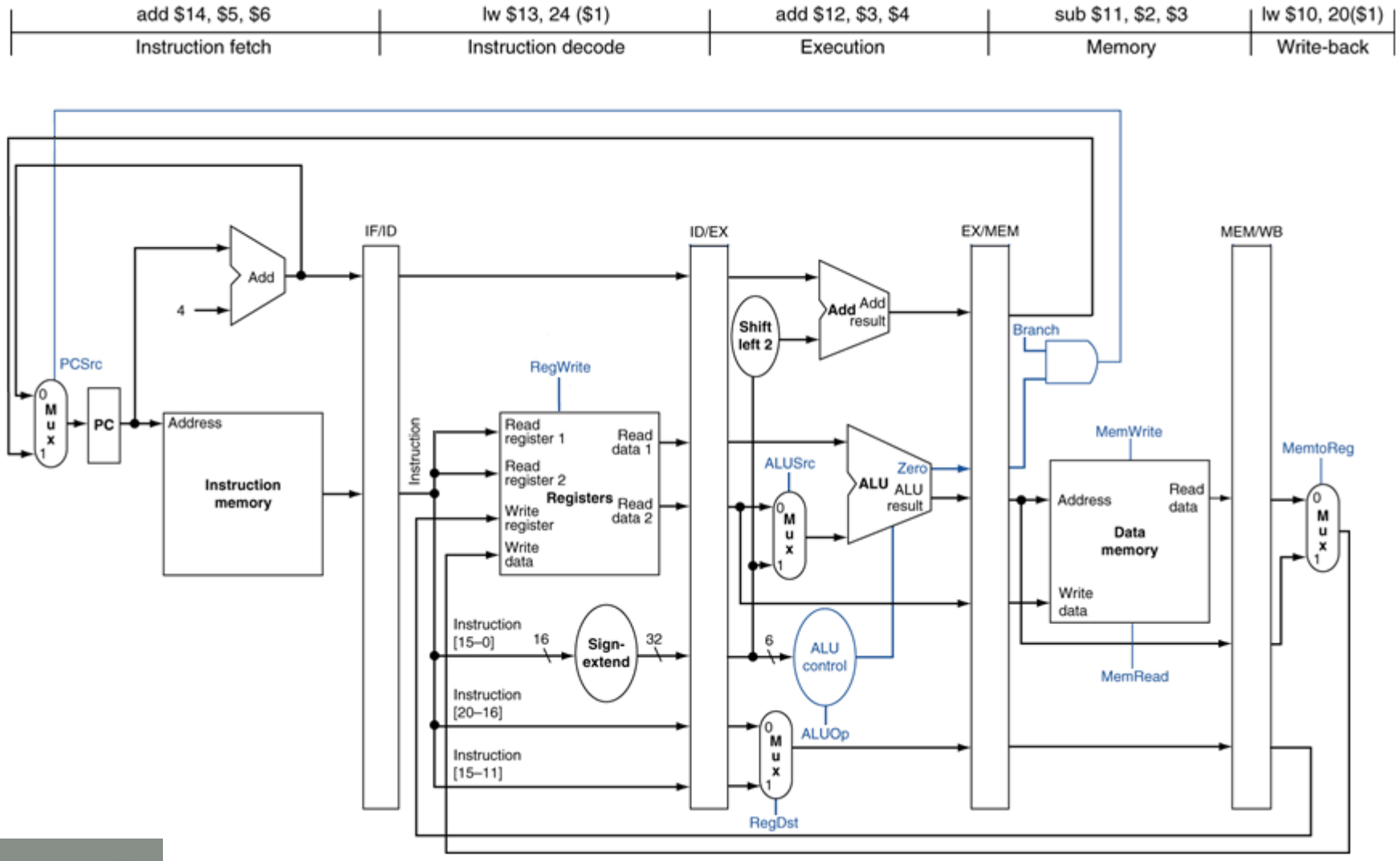
Multi-Cycle Pipeline Diagram



Multi-Cycle Pipeline Diagram

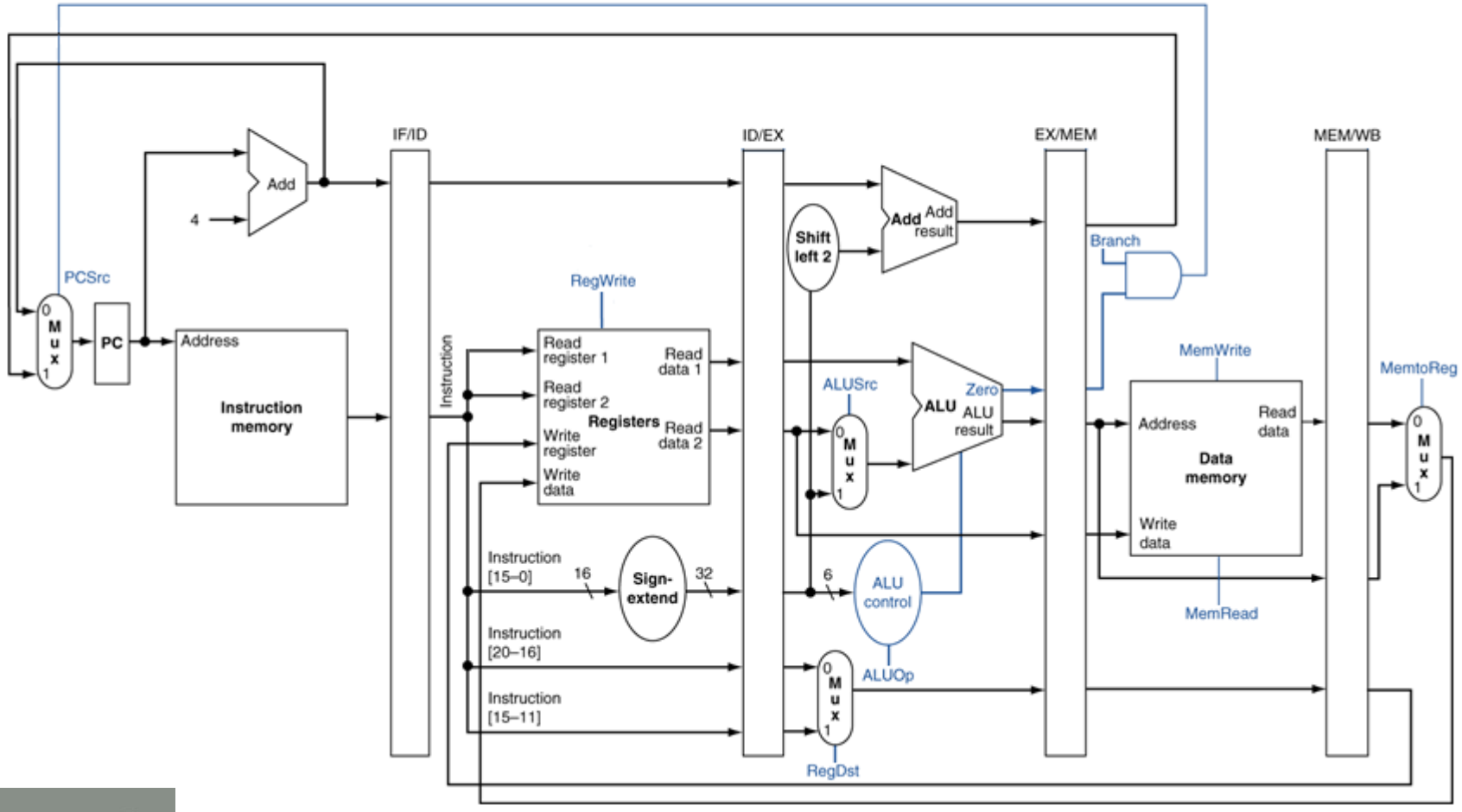


Single-Cycle Pipeline Diagram

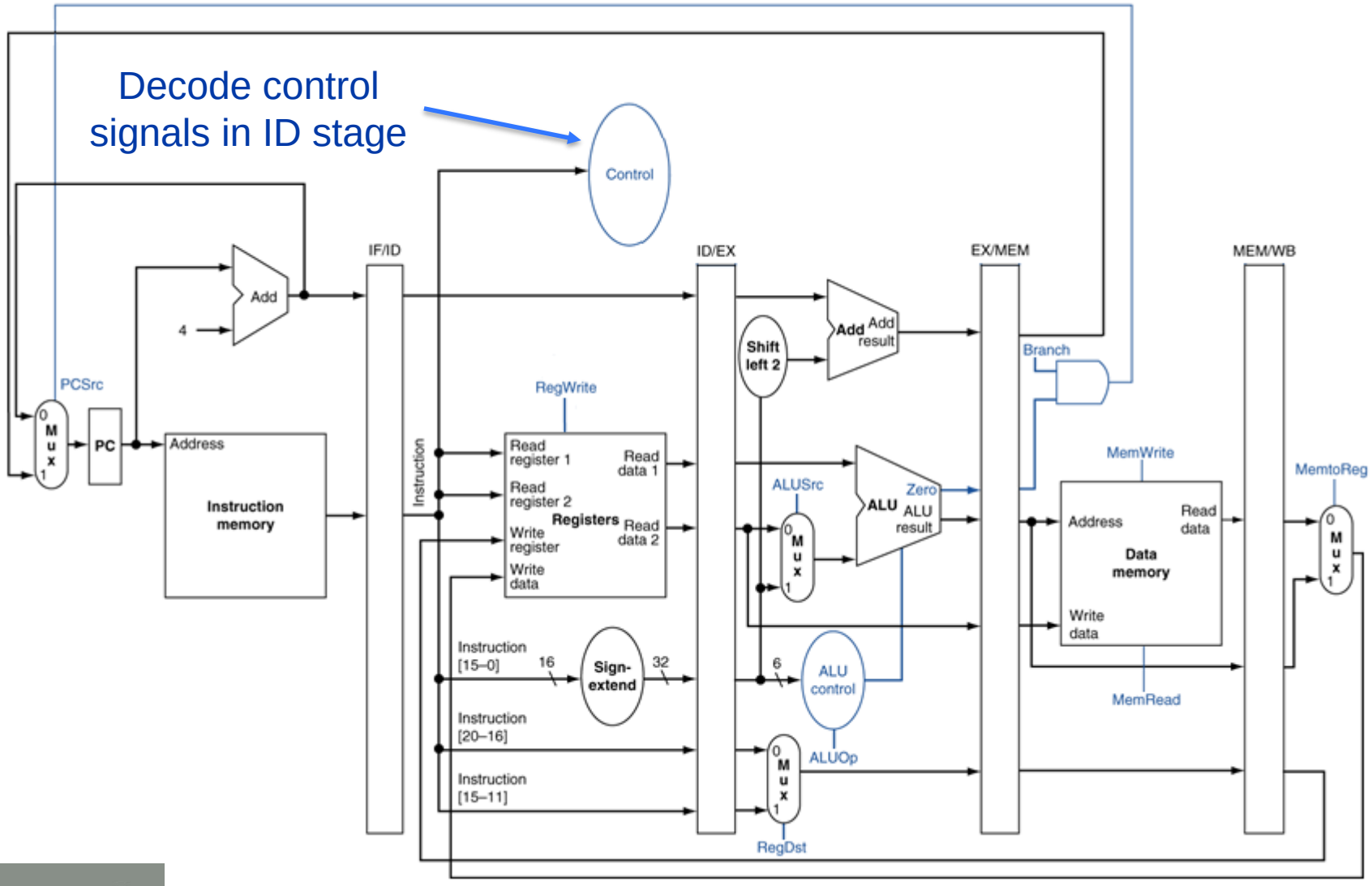


Pipeline with Control Signals

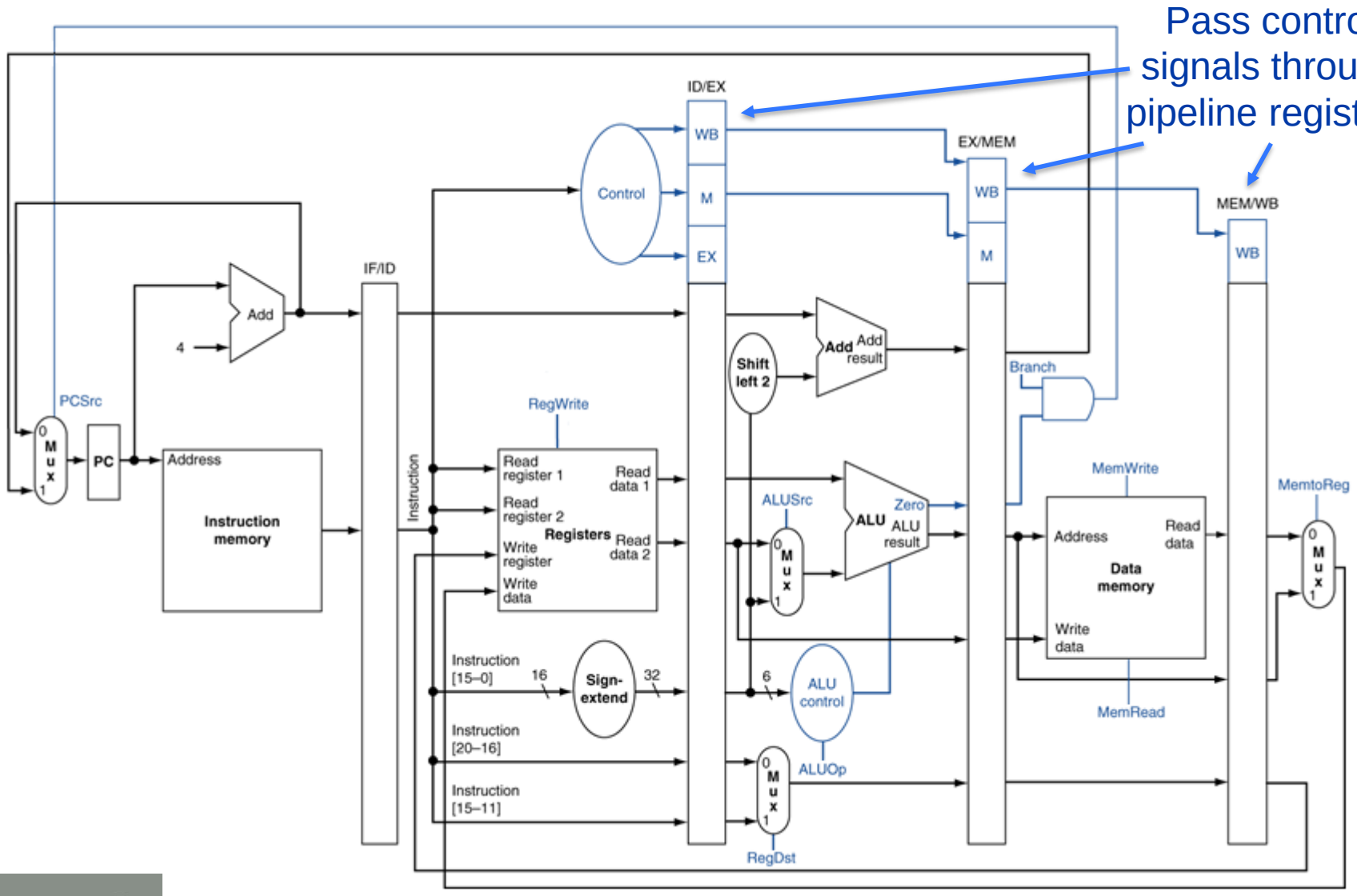
Where to get the control signals from?



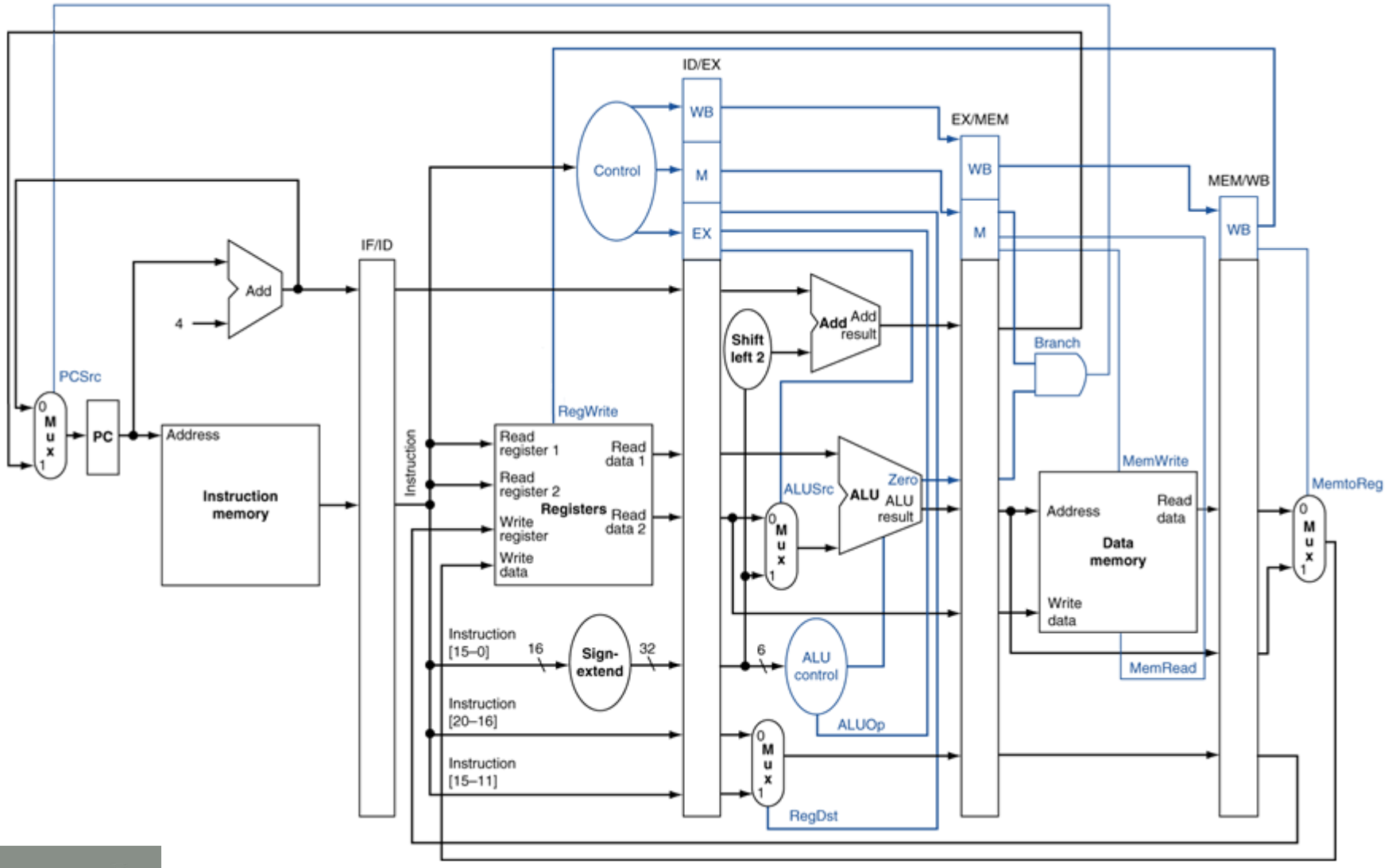
Pipeline with Control Signals



Pipeline with Control Signals



Pipeline with Control Signals



Textbook Sections

- The content in these slides corresponds to:
  - Textbook:
    - *Computer Organization and Design, 5th Edition by David Patterson and John Hennessy, Morgan Kaufmann, 2014.*
  - Sections:
    - 4.5, 4.6