RISC-V Single Cycle Architecture I

Muhammad Tahir

Lecture 7

Electrical Engineering Department
University of Engineering and Technology Lahore

Contents

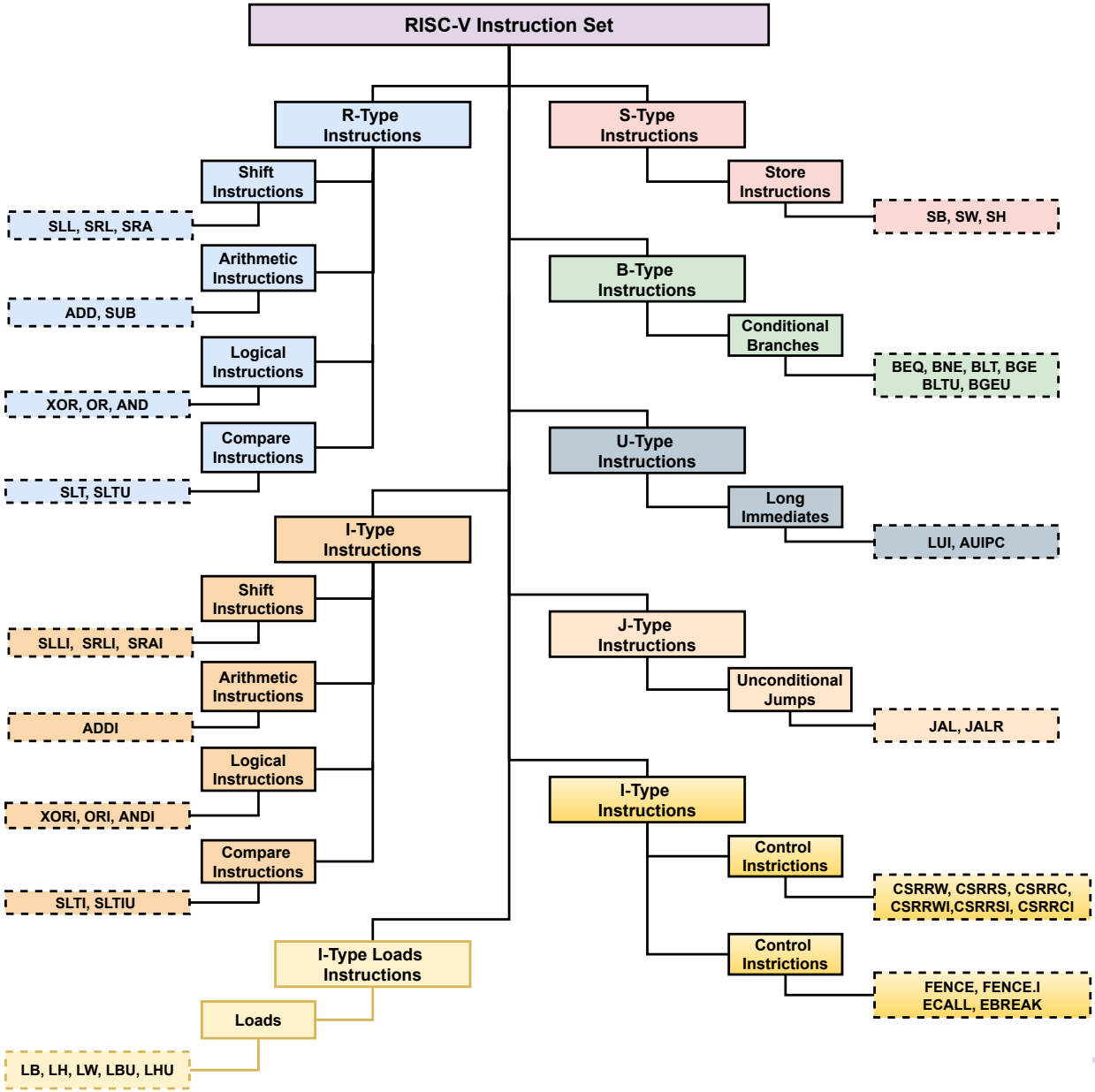
1. The Processor
2. Single Cycle Datapath
3. R-type Datapath
4. I-type Datapath
5. S-type Datapath

The Processor

The instruction processing is divided into two main parts:

- `Datapath` is a collection of functional units wired together to perform a set of specified operations
- `Control` is implemented as a special hardware that tells the *datapath* what needs to be done

Build Processor that supports RISC-V ISA



The Processor Building Blocks

- Combinational Elements
  - Mux
  - Decoder
  - ALU
- Sequential/Synchronous Elements
  - Register
  - Register File
  - Memory (Instruction and Data)

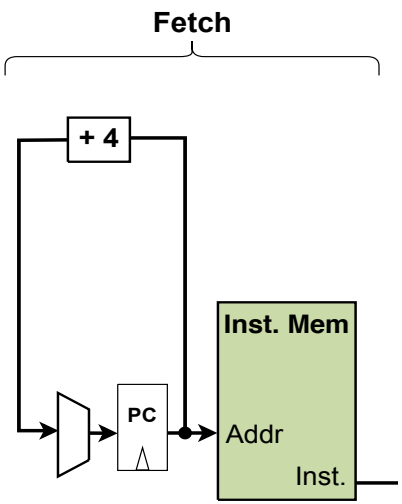
Instruction Processing Steps

Basic phases of instruction execution:

- Instruction Fetch (IF)
- Instruction Decode (ID)
- Execute (EX)
- Memory Access(MEM)
- Writeback to Register File (WB)

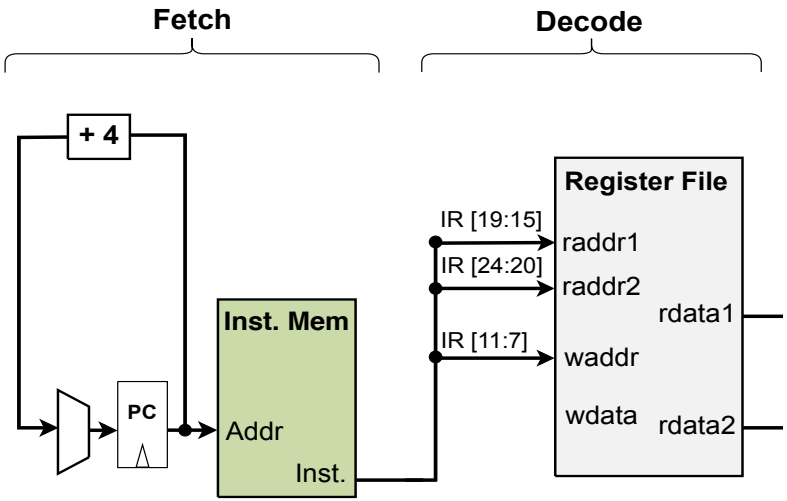
Instruction Fetch (IF)

- Instruction Memory, Program counter and PC Increment form IF unit



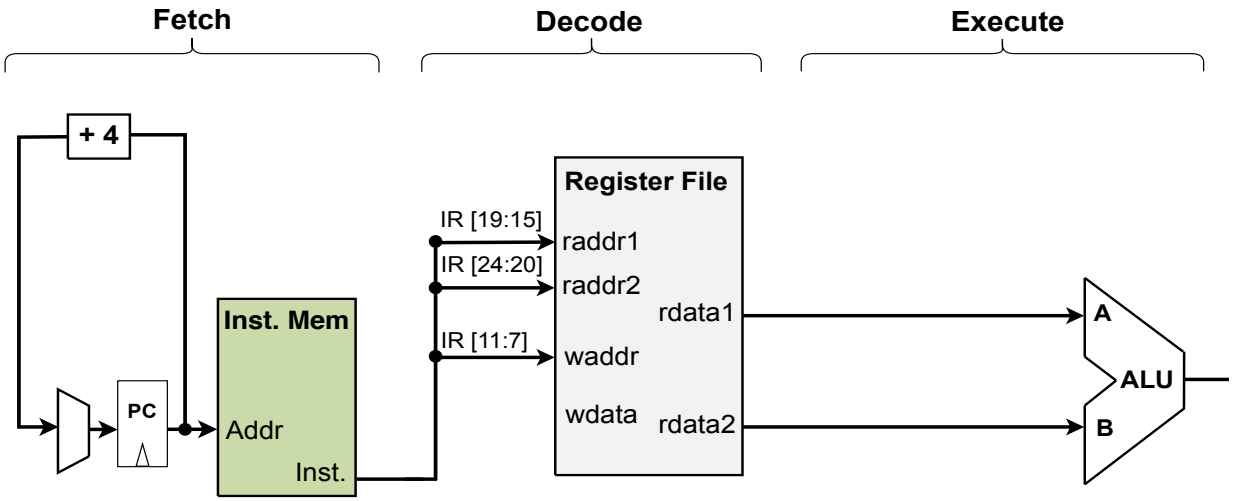
Instruction Decode (ID)

- Instruction fetch and `decode`



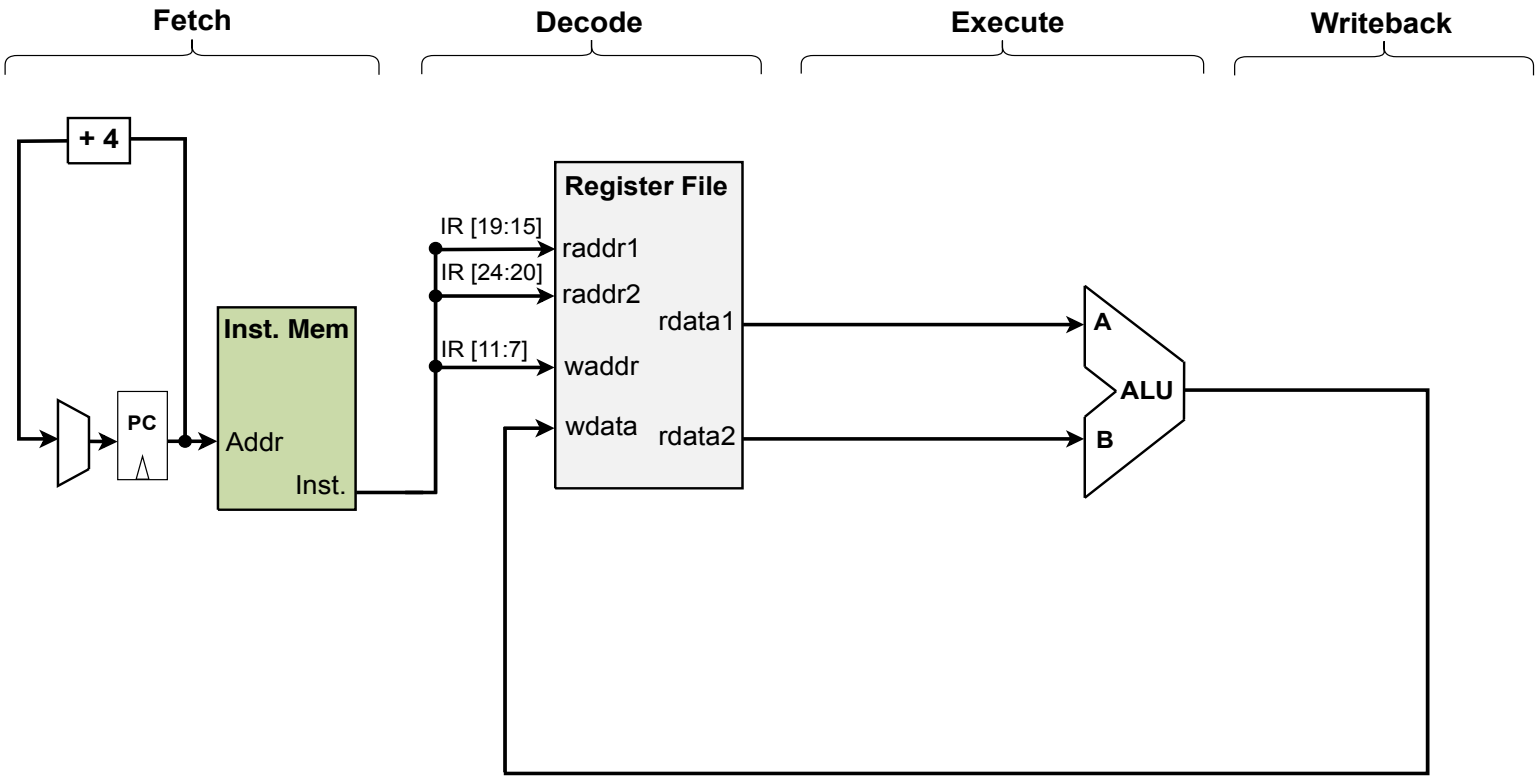
Instruction Execution (EX)

- Instruction fetch, decode and `execute`

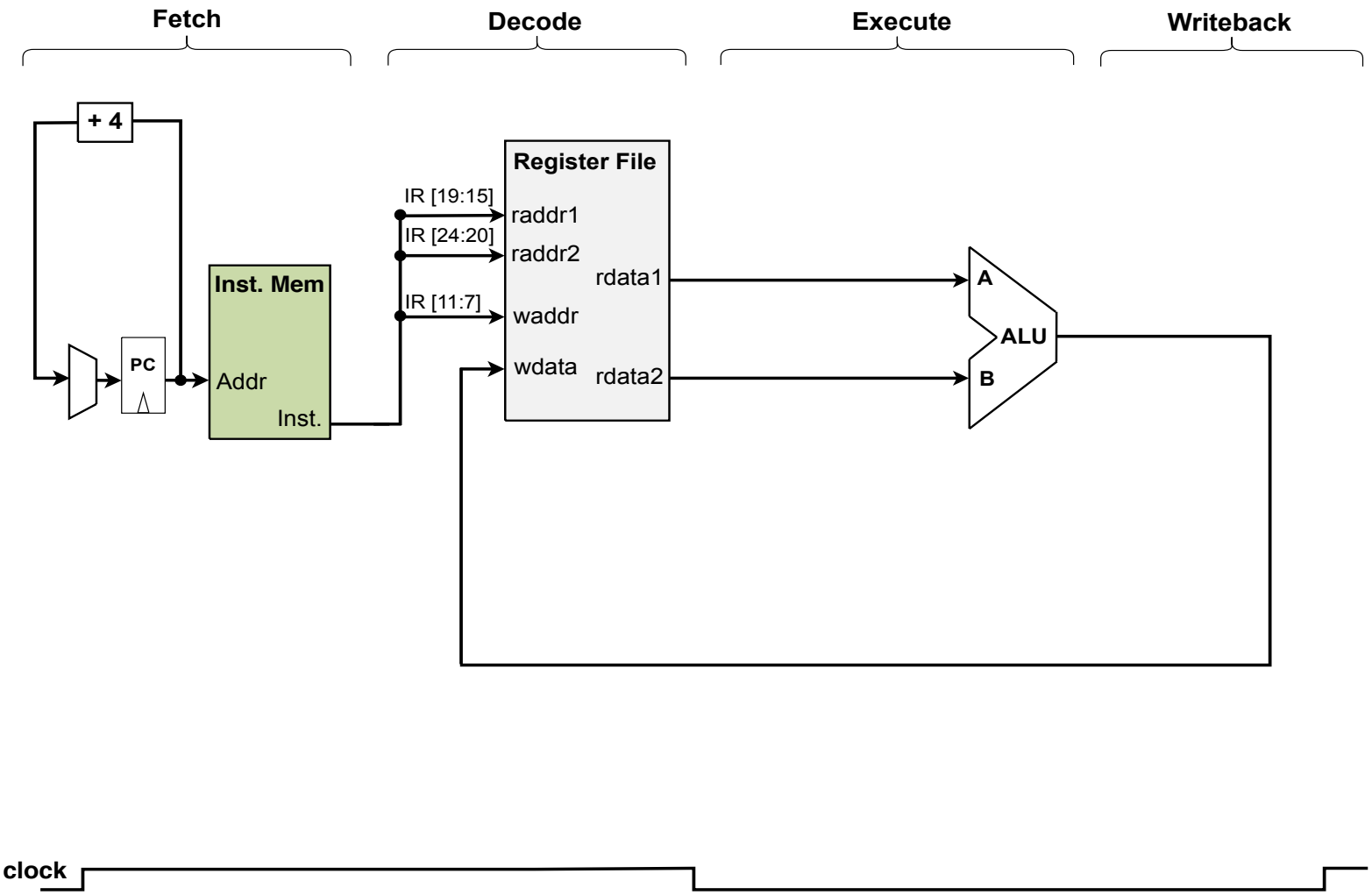


Write Back (WB) to Register File

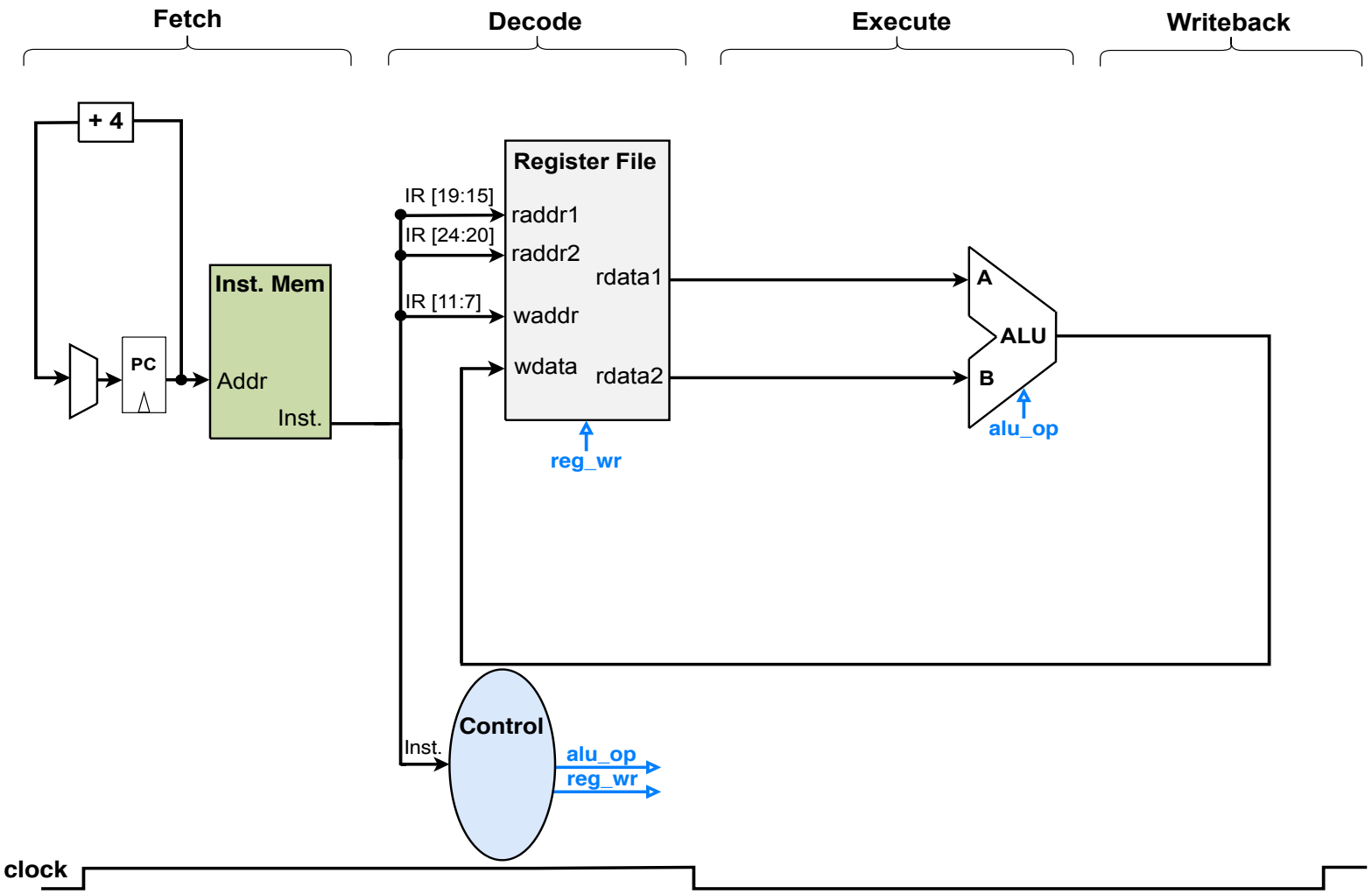
- Instruction fetch, decode, execute and `writeback`



Datapath for R-type Instructions



Data- and Control-path for R-type Instructions



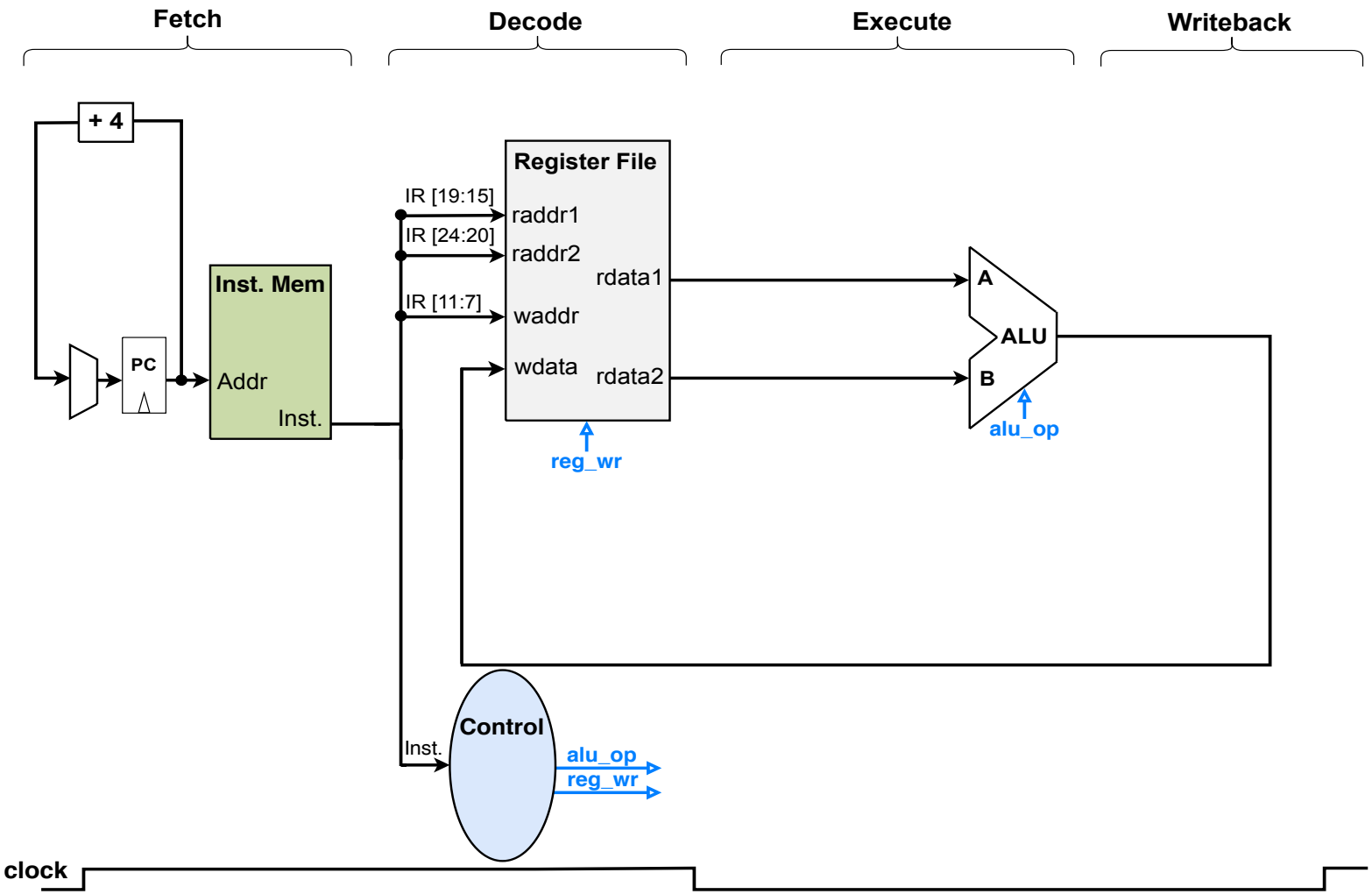
Data Processing Instructions: R-type

- R-type: ADD instruction

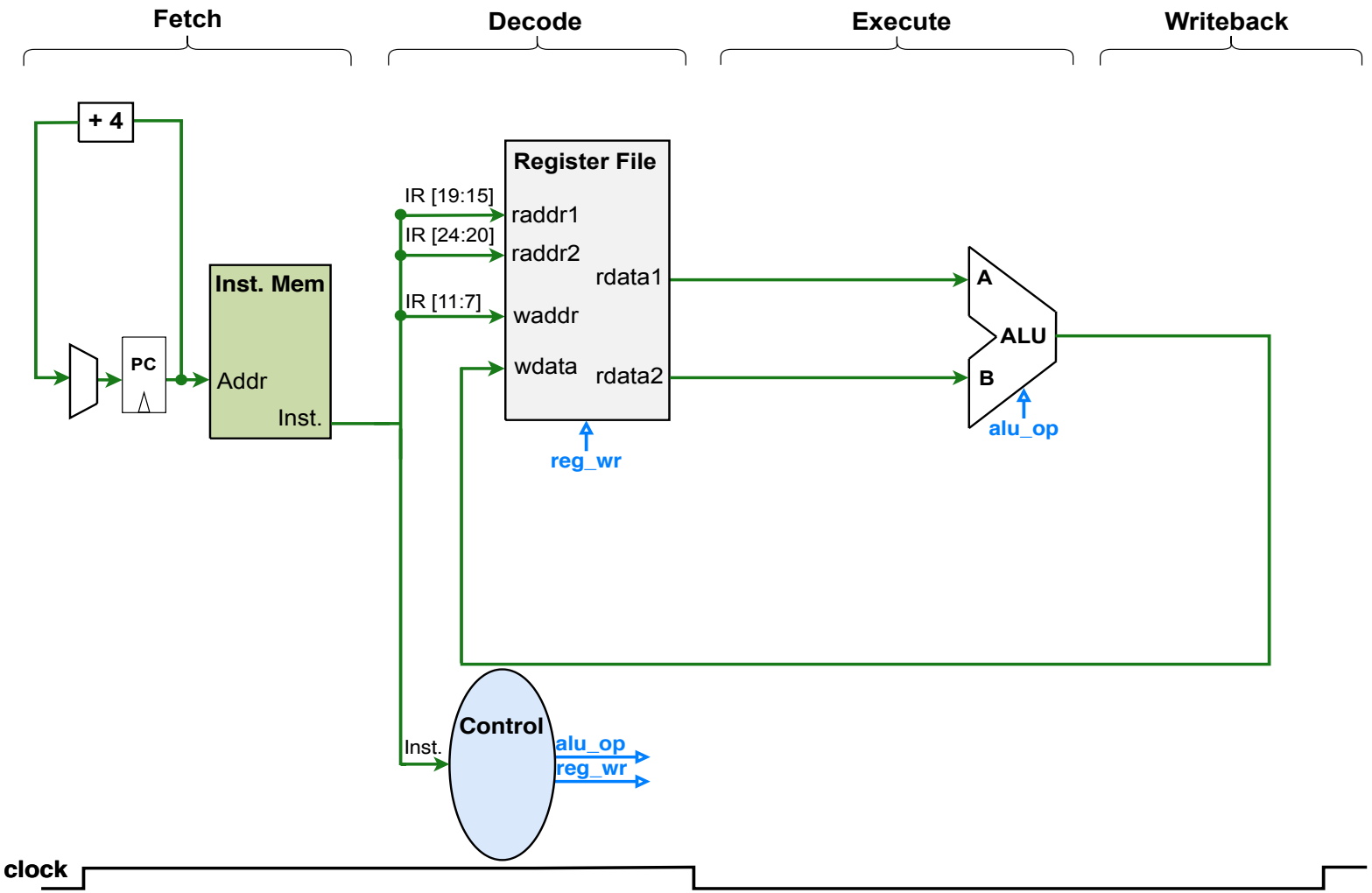
7	5	5	3	5	7
func7	rs2	rs1	func3	rd	opcode
0000000	rs2	rs1	000	rd	0110011

- R-type instructions are register to register operations
- The destination register: `rd` ← rs1(func3, func7)rs2

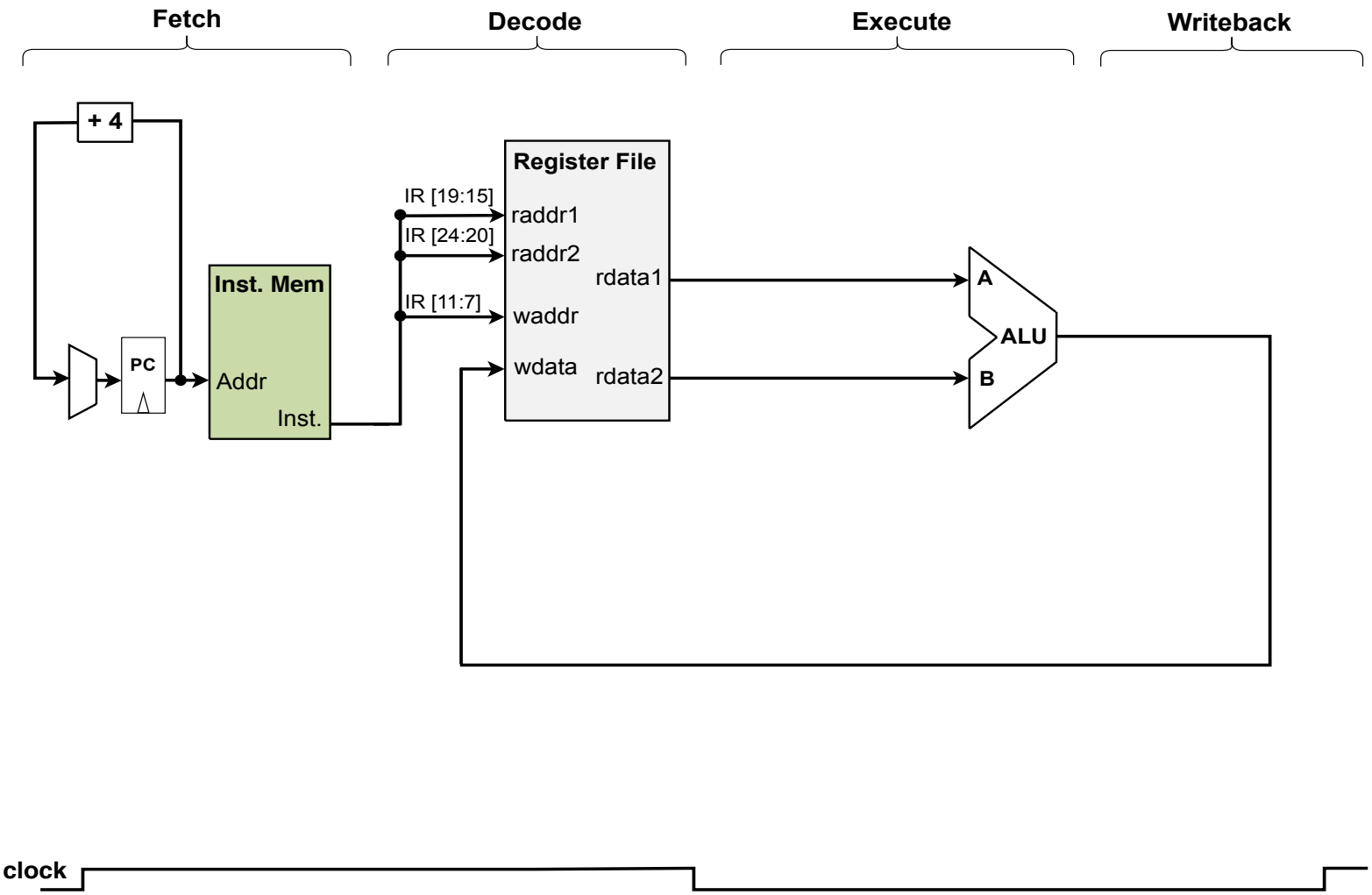
How R-type (ADD) Instruction Execute



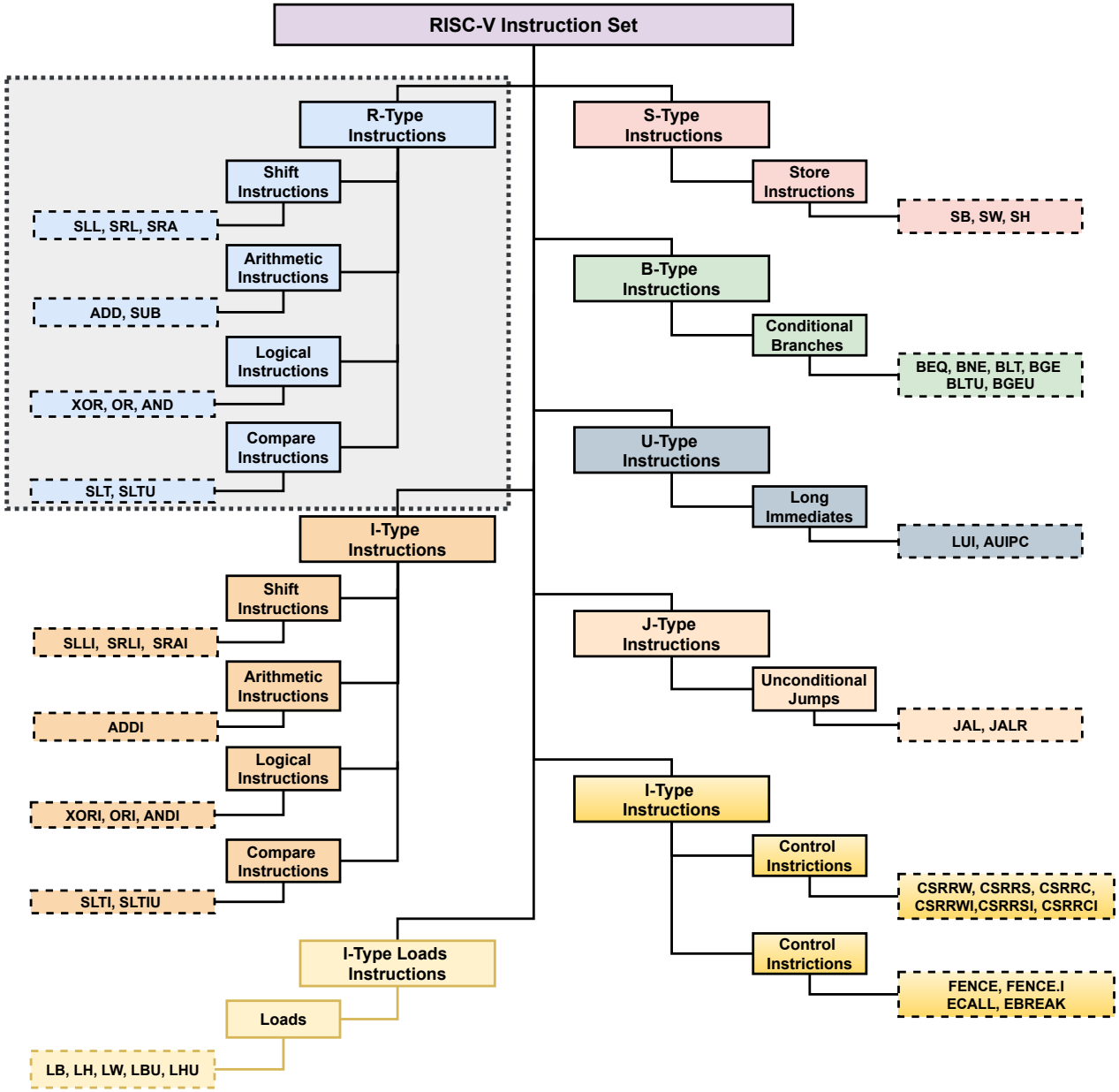
R-type Instruction Execution Example



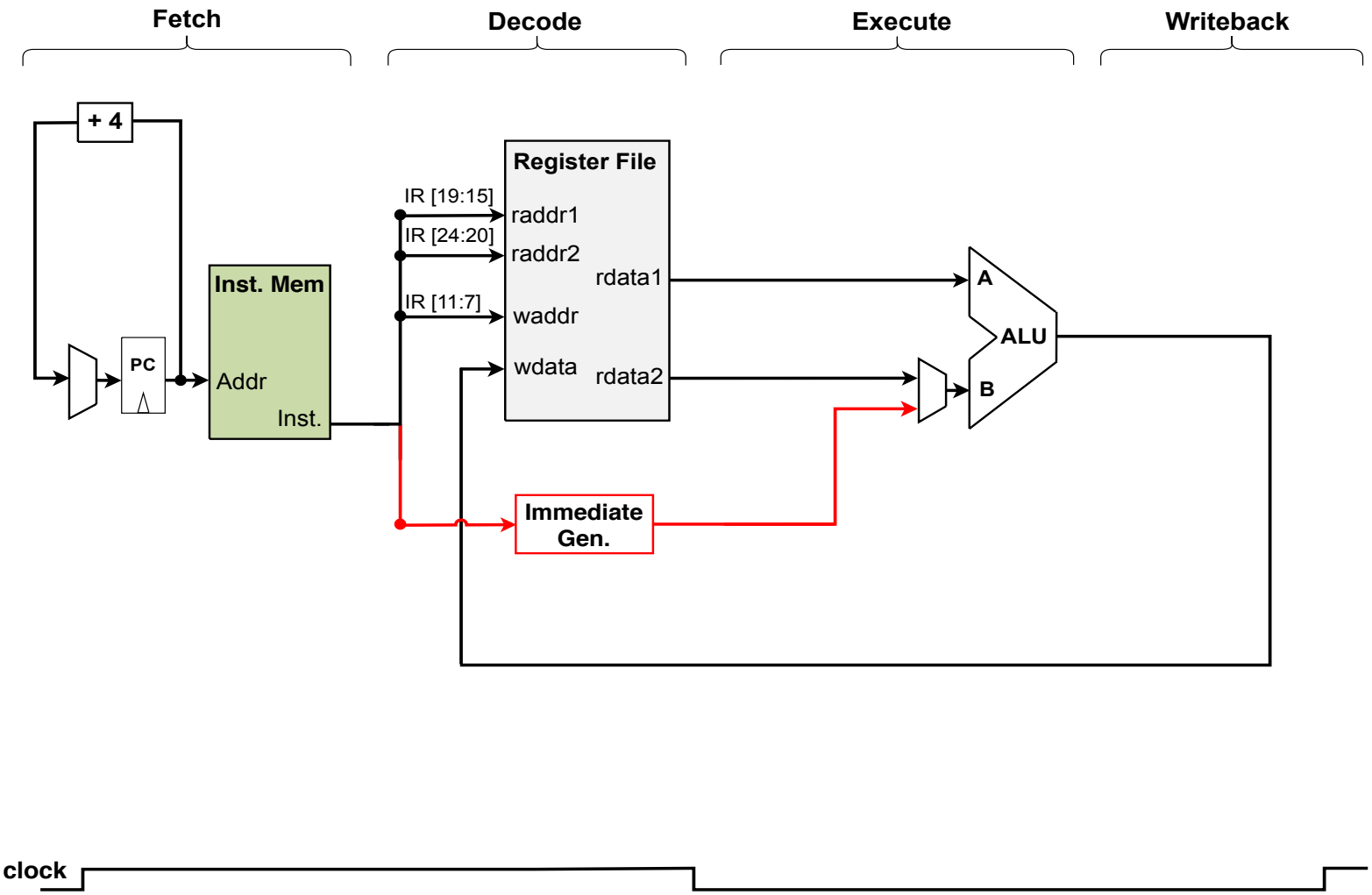
Datapath for R-type Instructions



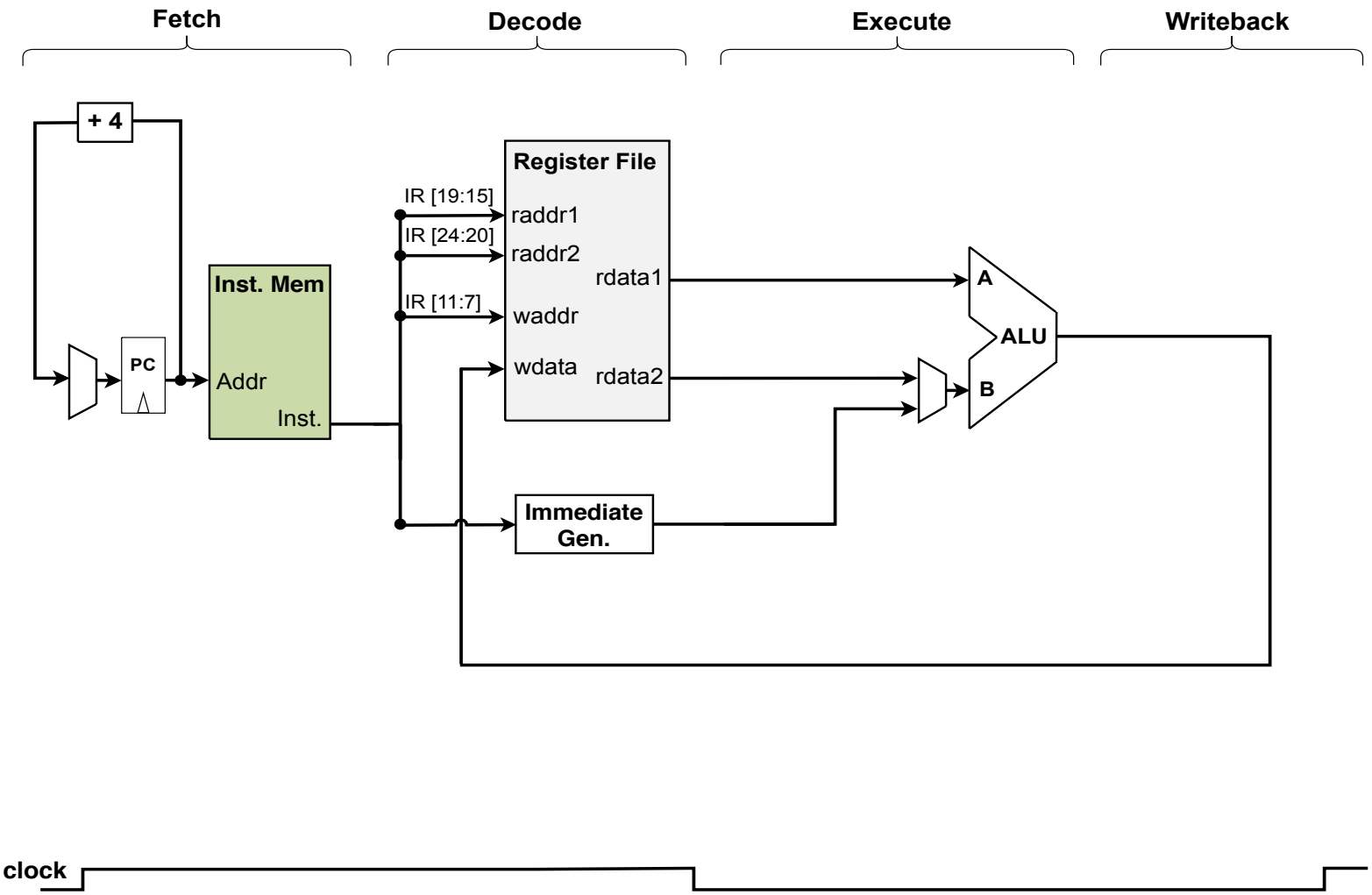
RISC-V ISA Supported by R-type



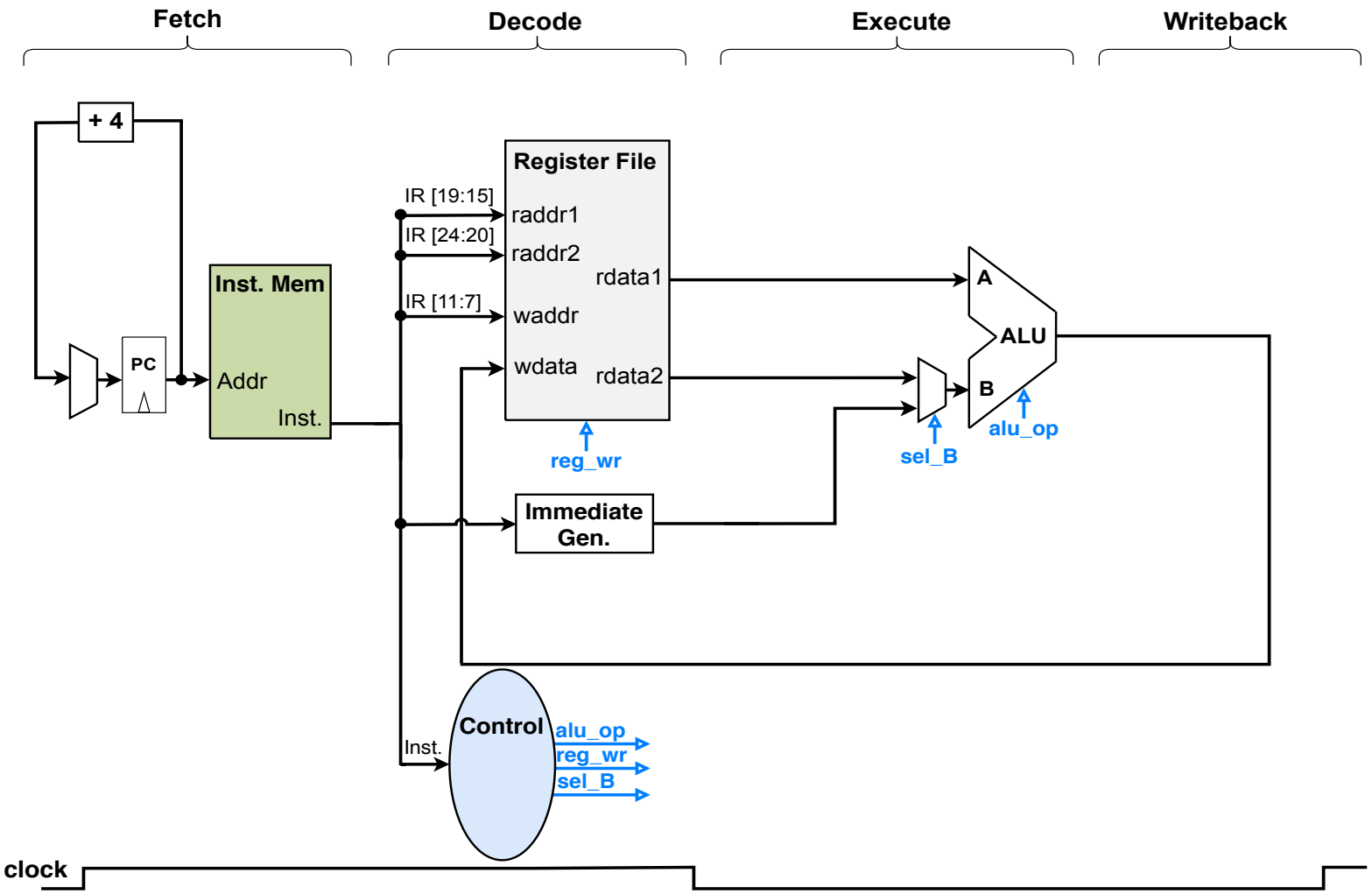
Adding Support for I-type



Adding Support for I-type Cont'd



I-type Datapath with Control



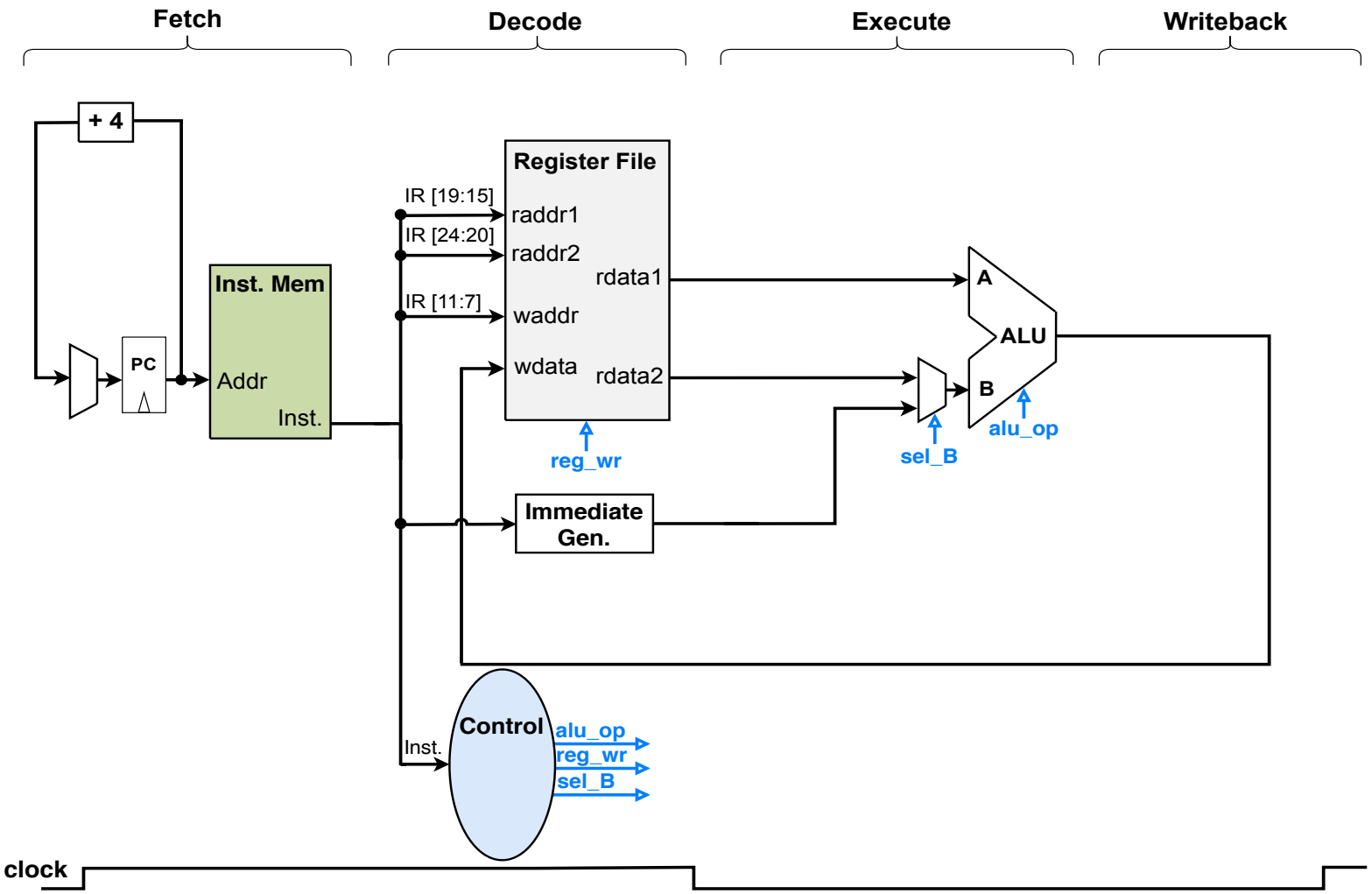
I-type Instructions

- I-type: ADDI instruction

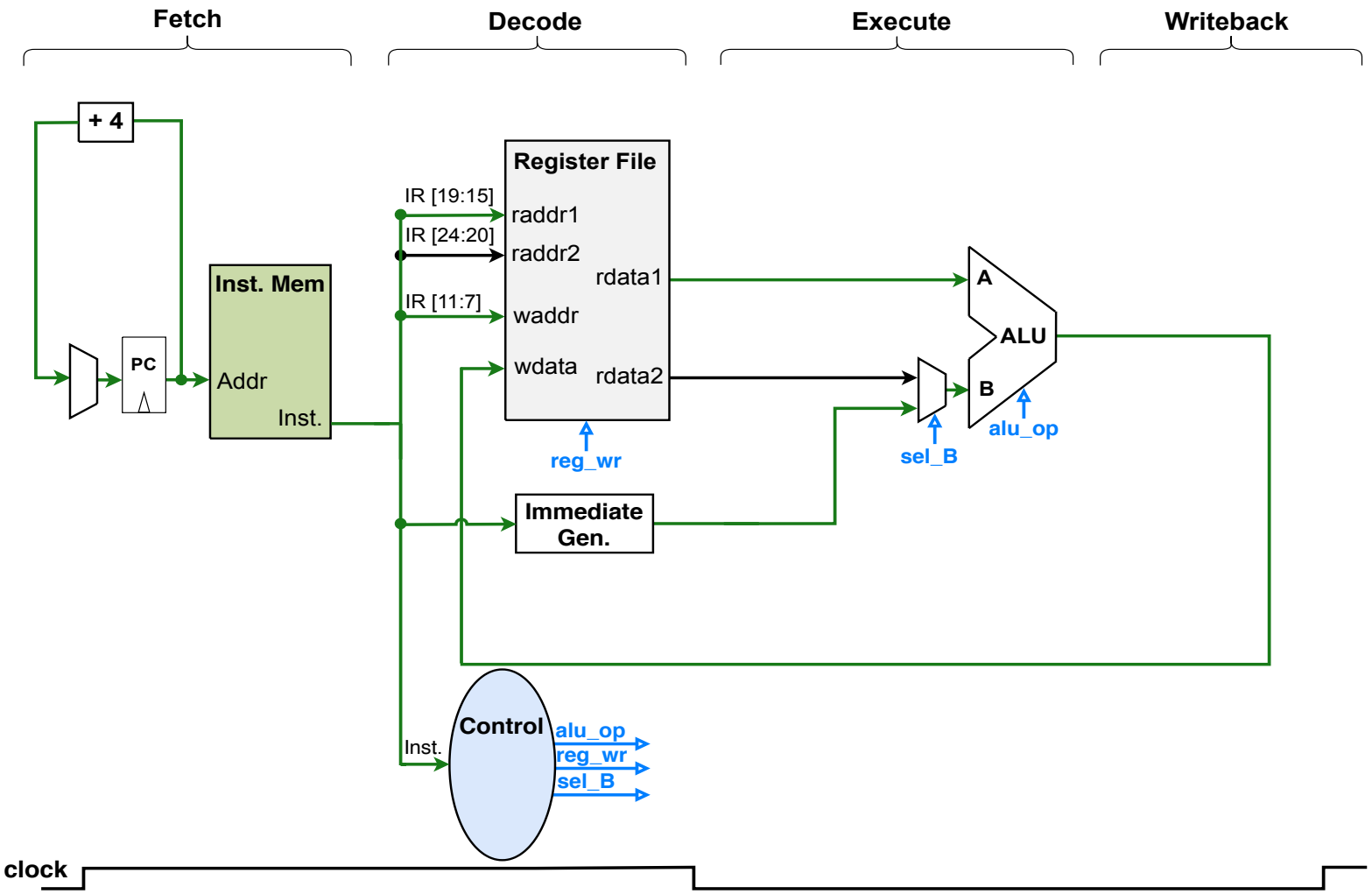
12	5	3	5	7
imm[11:0]	rs1	func3	rd	opcode
immediate	rs1	000	rd	0010011

- I-type instructions use immediate value as an operand
- The destination register: `rd` ← rs1 (func3) imm

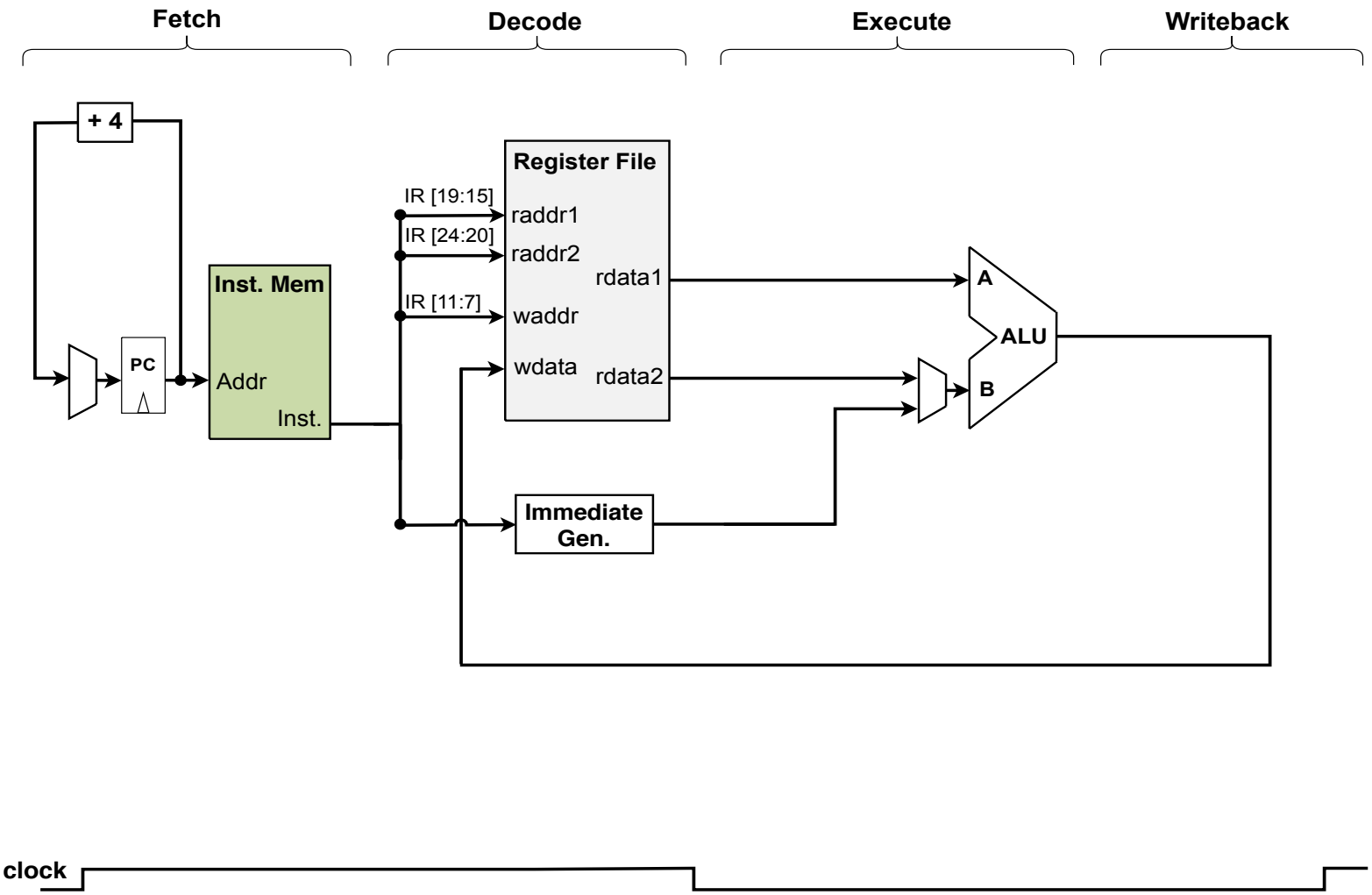
How I-type (ADDI) Instruction Execute



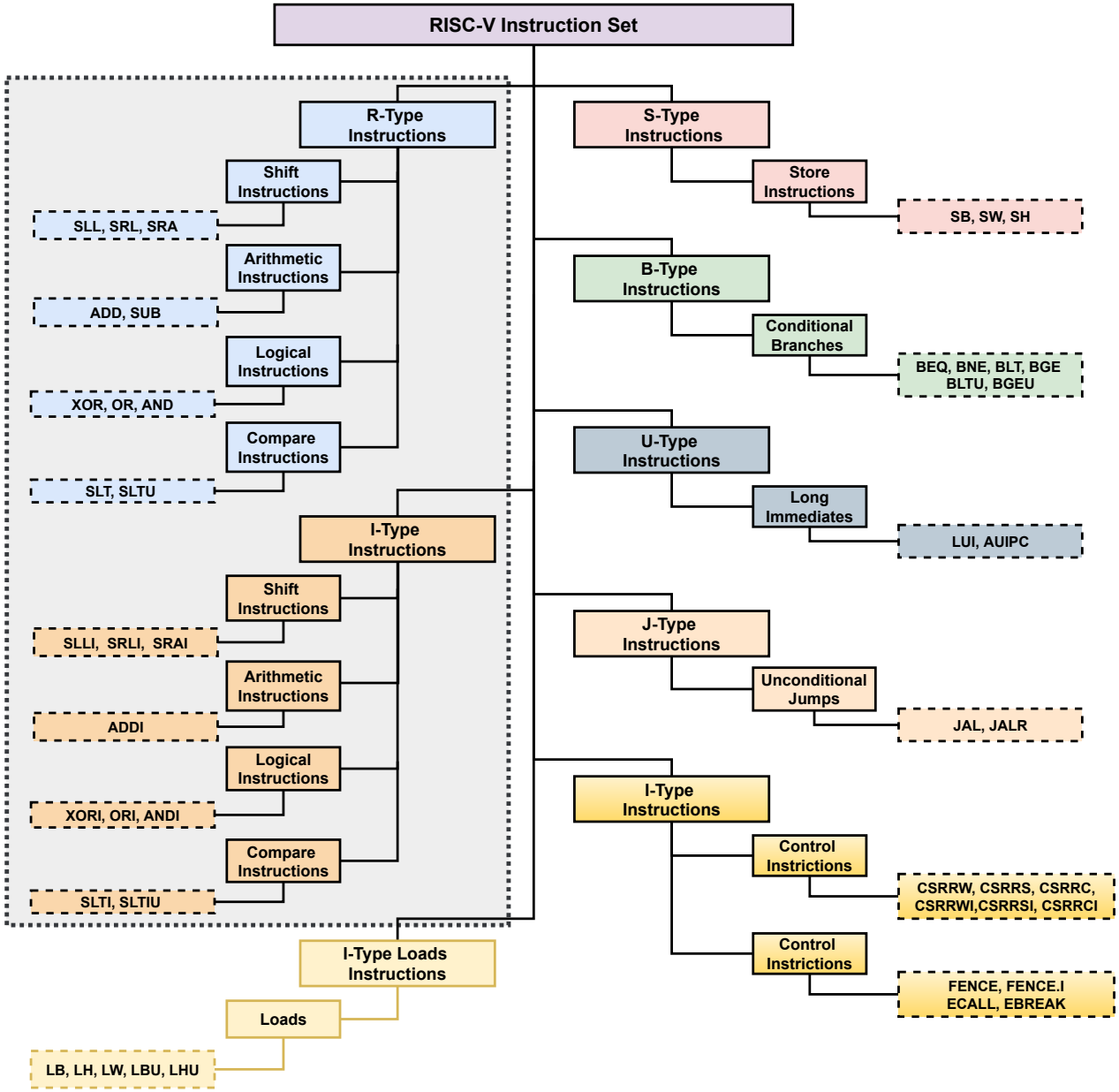
I-type Instruction Execution Example



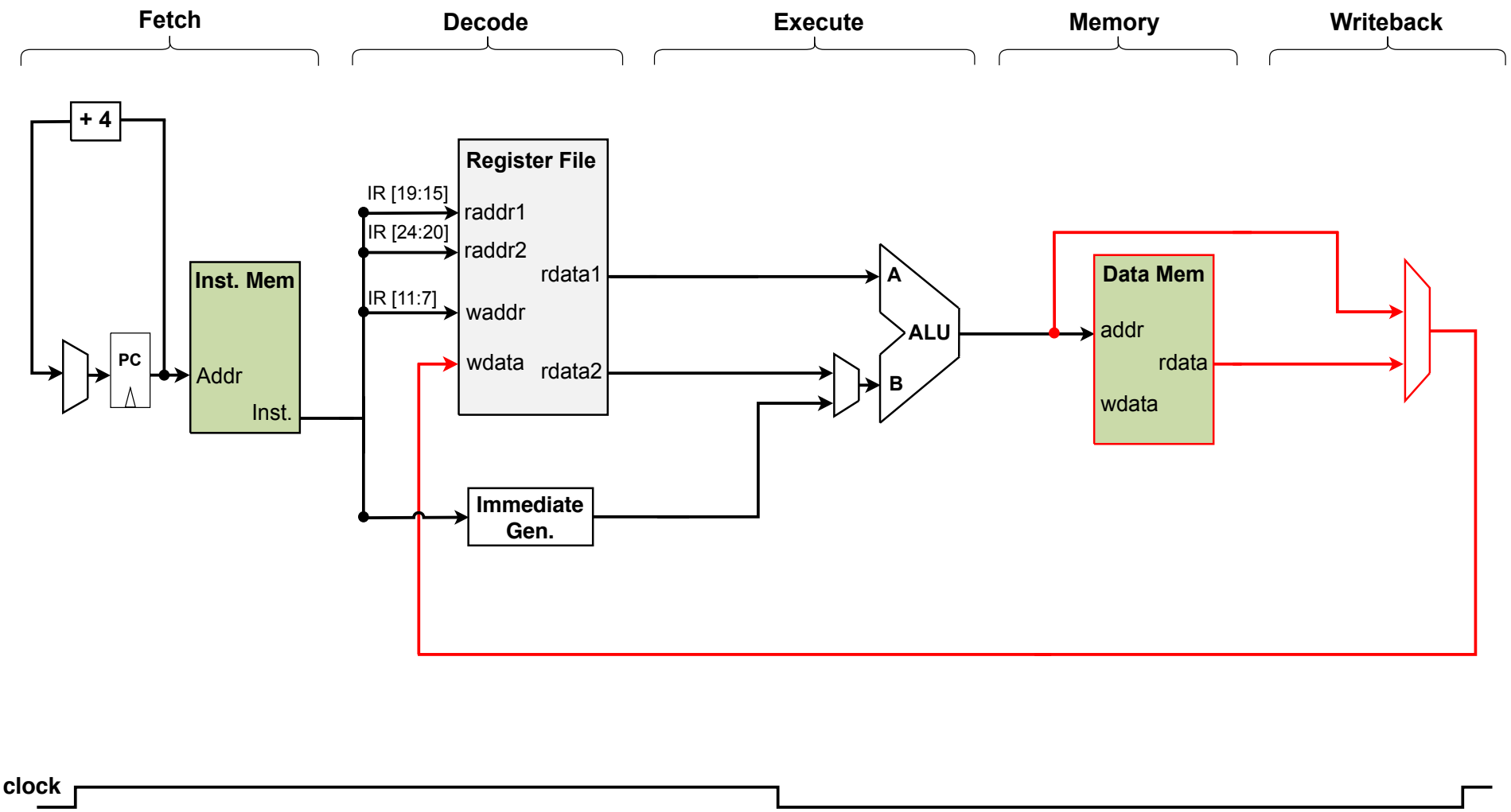
Datapath for R- and I-type



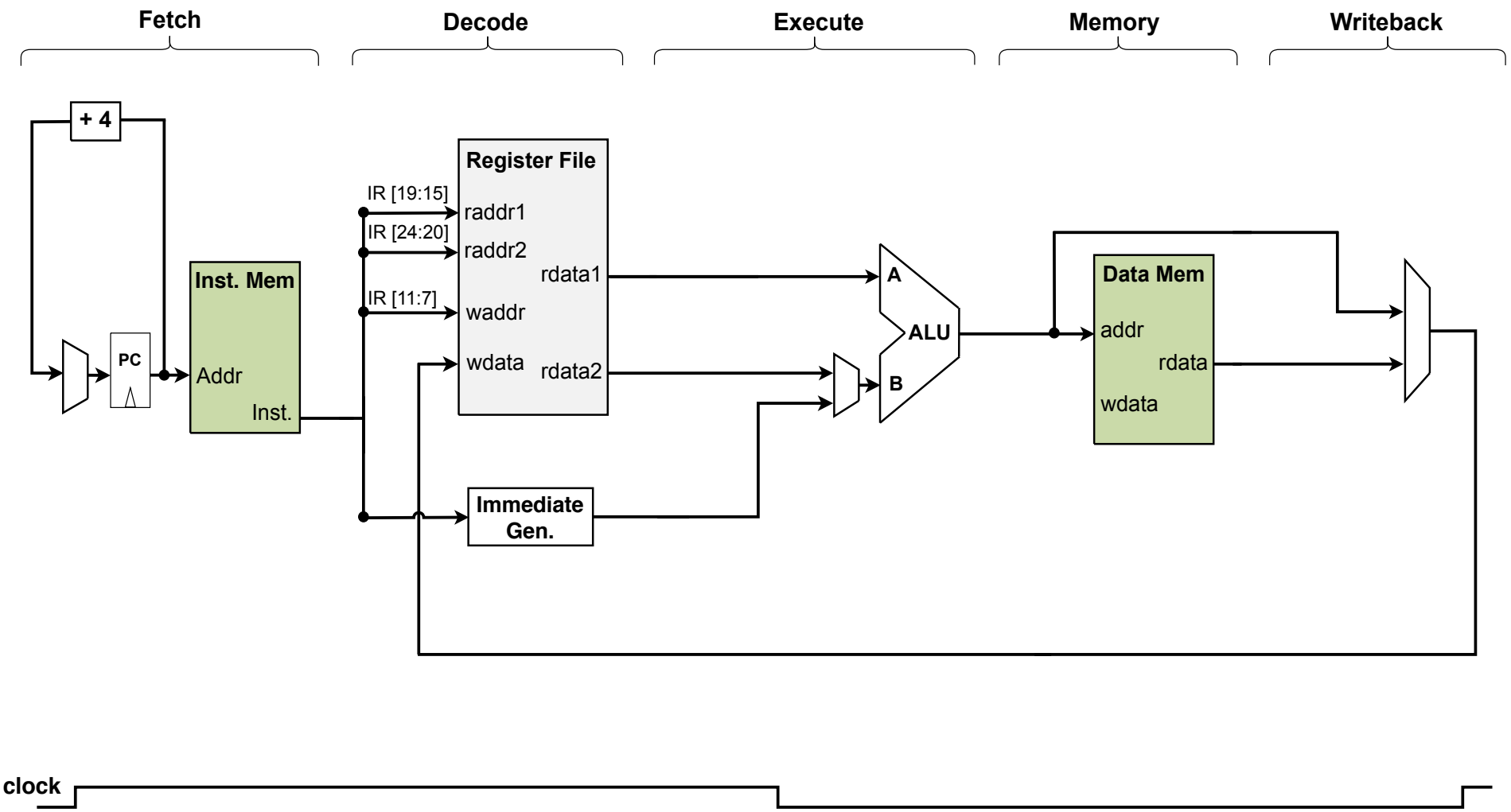
RISC-V ISA Supported by R- and I-type



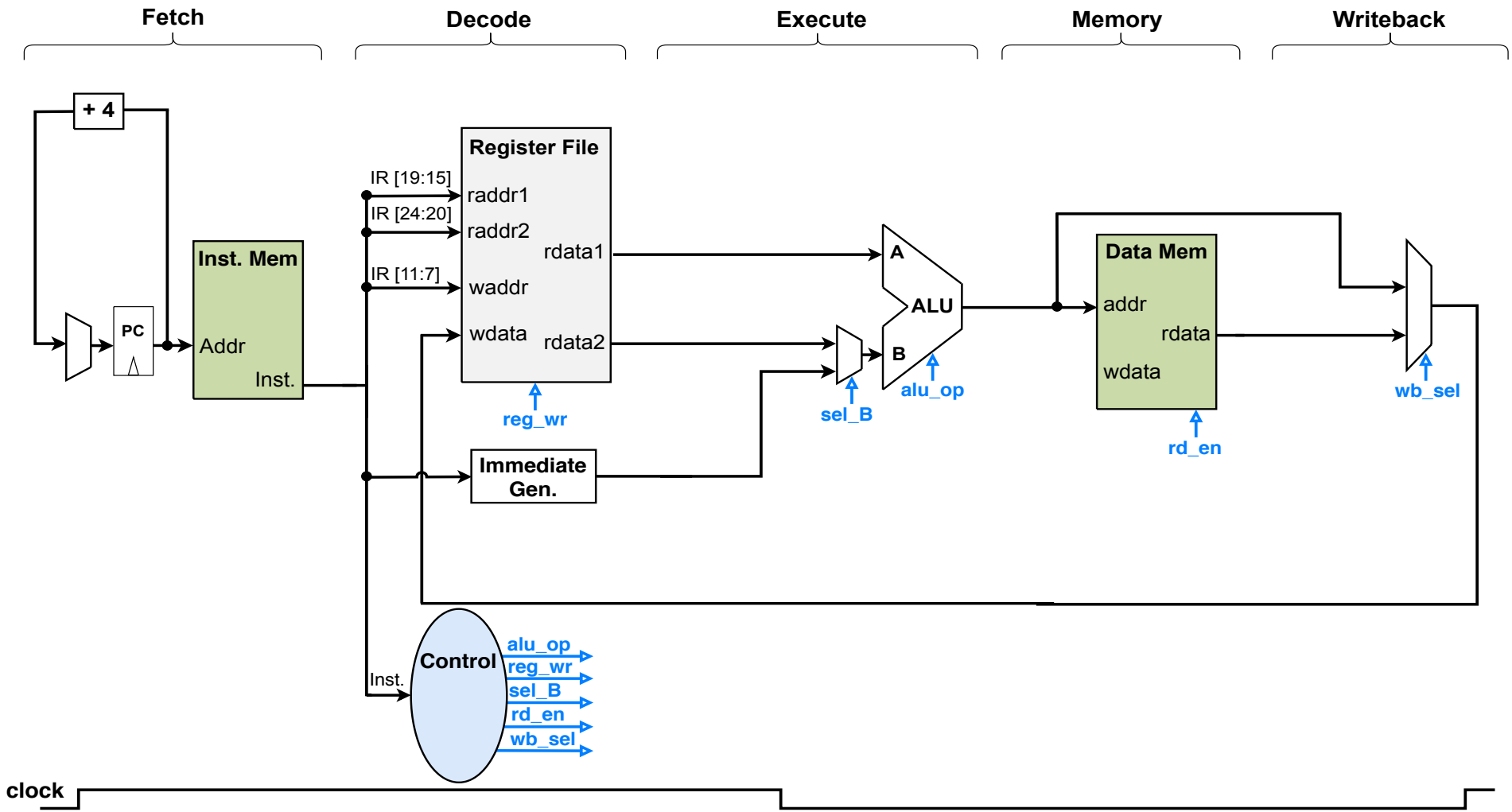
I-type: Load Operation



I-type: Load Operation Cont'd



I-type: Load Operation Data- and Control-path



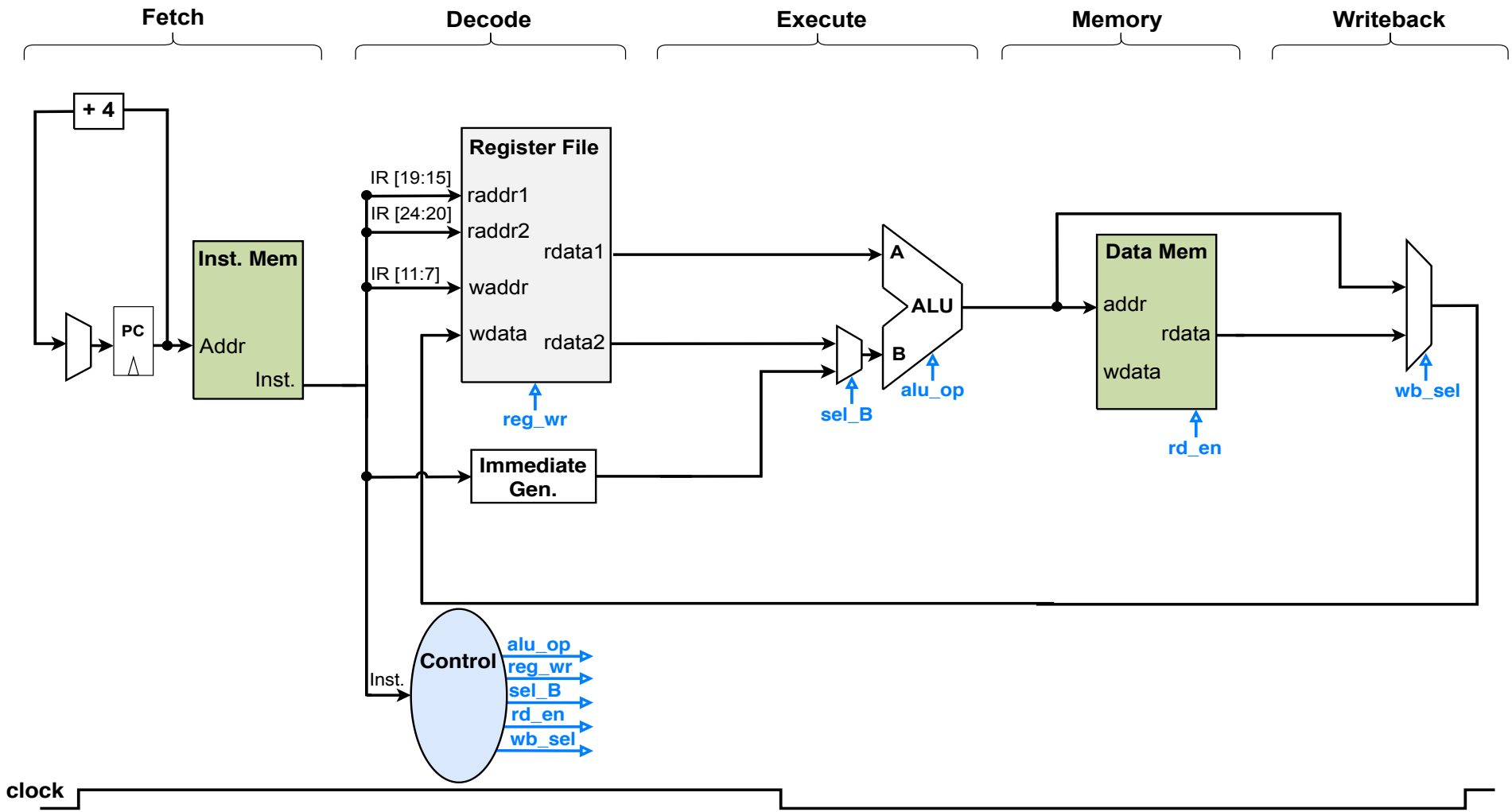
Memory Access Instructions: I-type Cont'd

- I-type: Load word (LW)

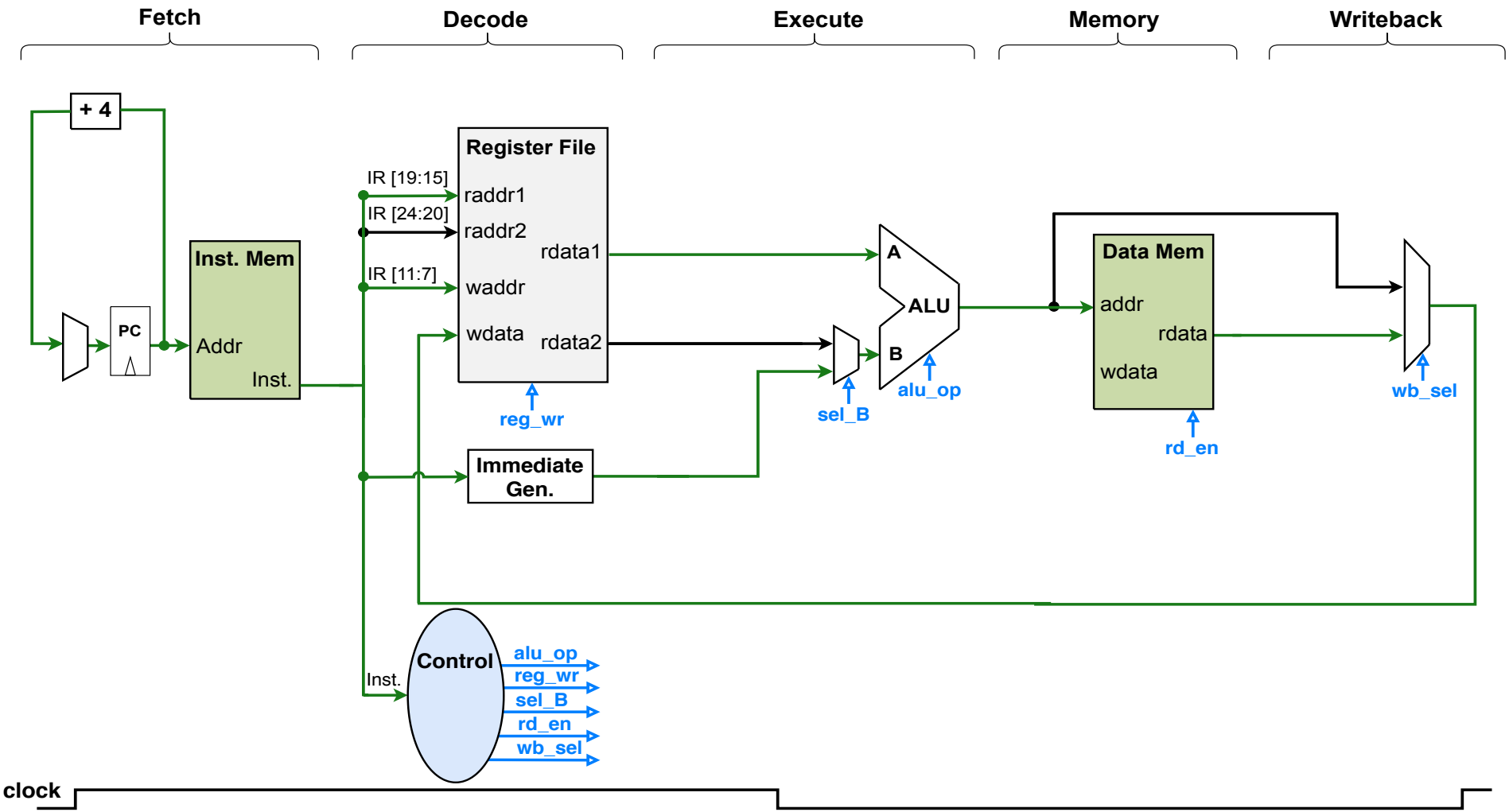
12	5	3	5	7
imm[11:0]	rs1	func3	rd	opcode
immediate	rs1	010	rd	0000011

- `I-imm` = signExtend(inst[31:20])
- `opcode` = LOAD: rd $\leftarrow$ mem[rs1 + I-imm]
- `funct3` = LW/LB/LBU/LH/LHU

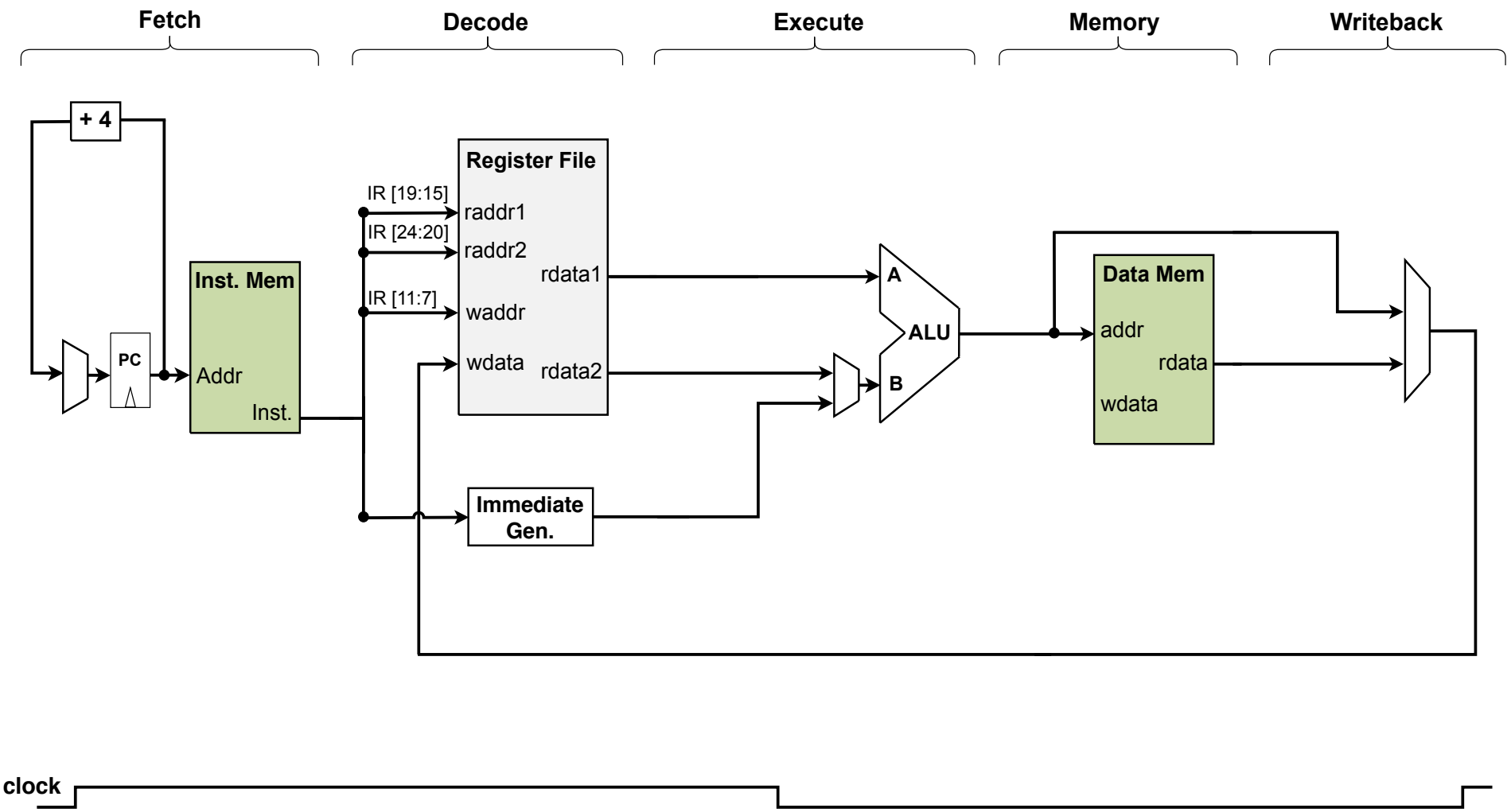
How I-type Load (LW) Instruction Execute



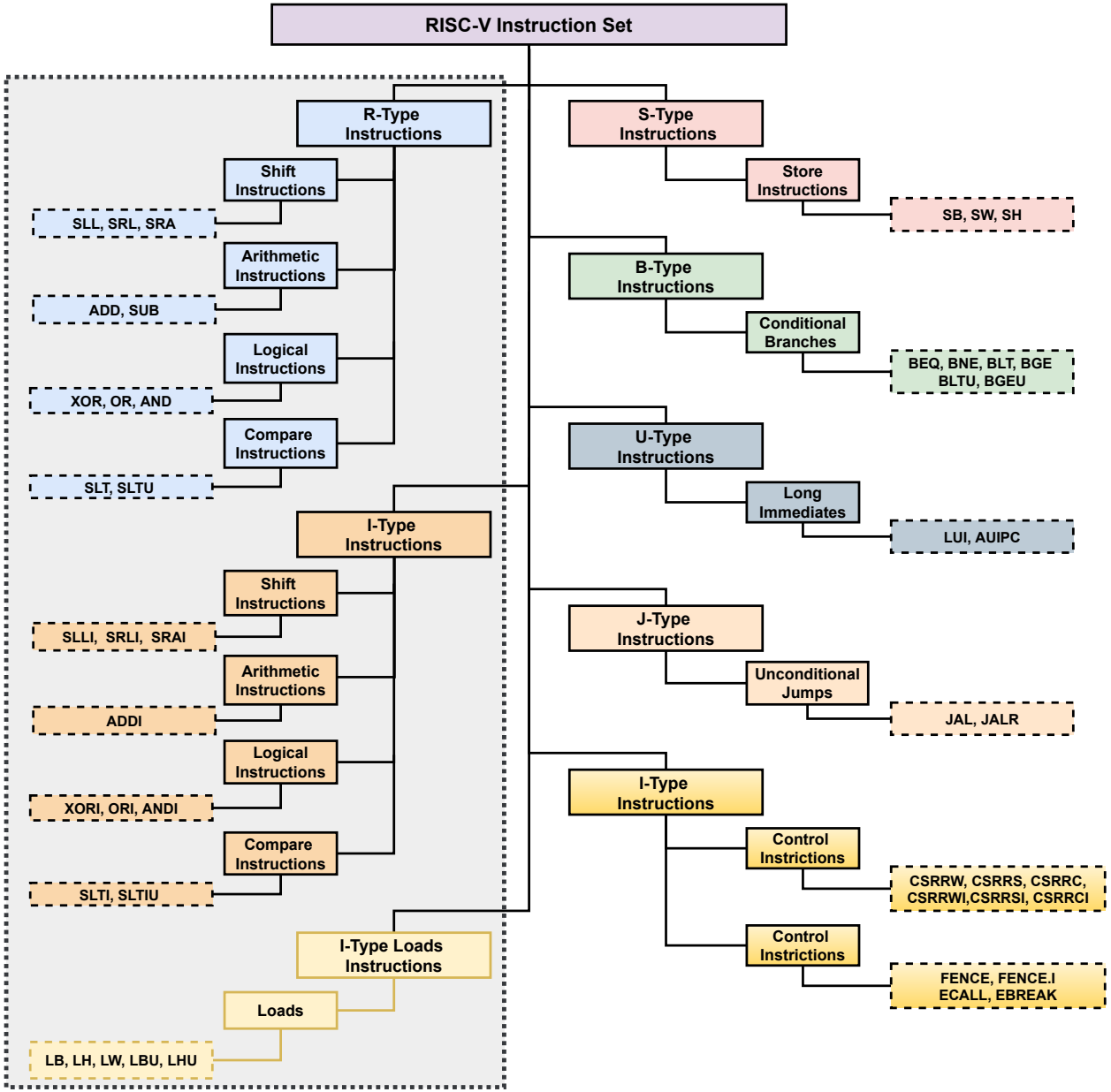
I-type Load Instruction Execution Example



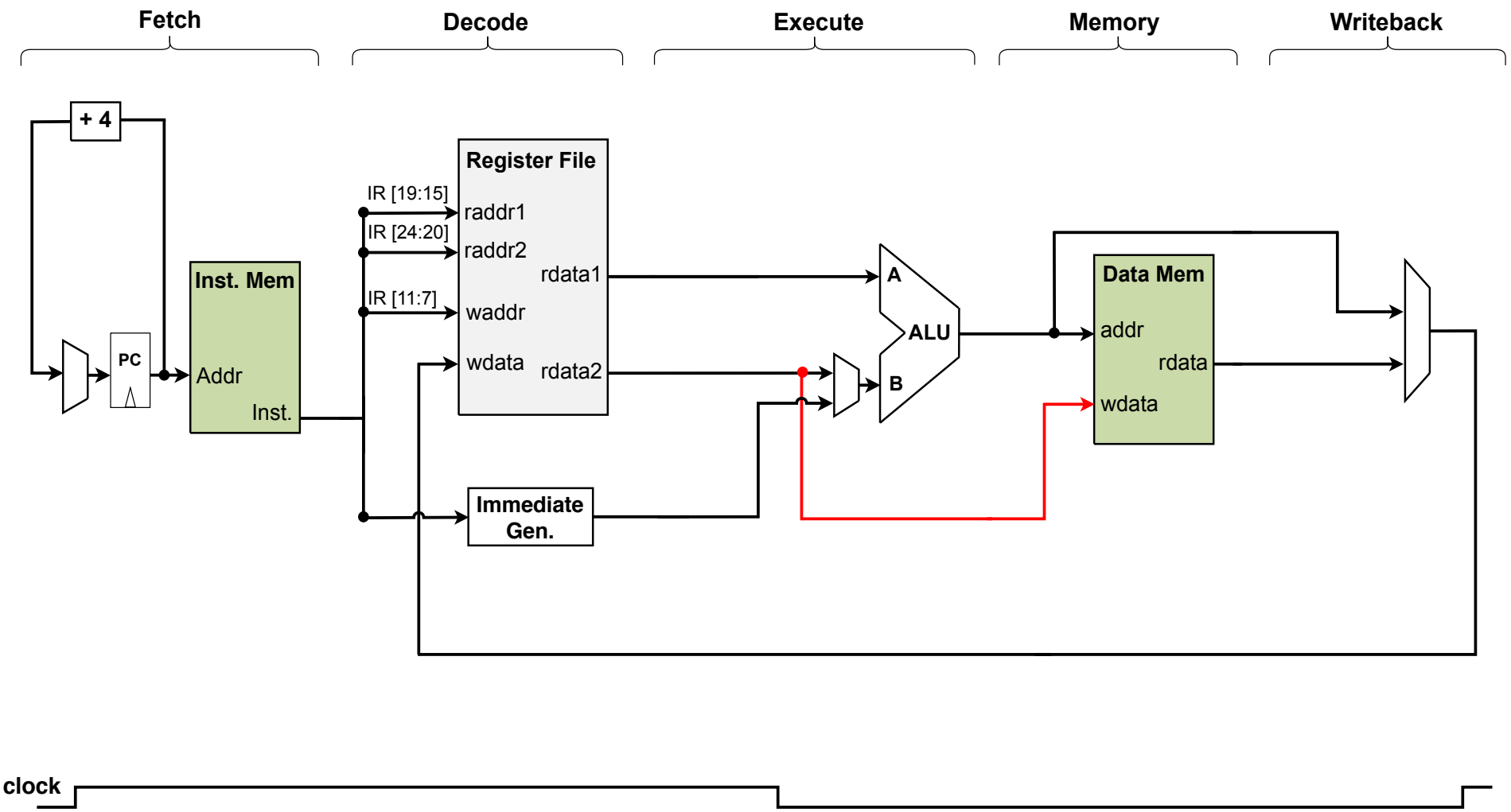
Datapath for R- and I-type (Including Load)



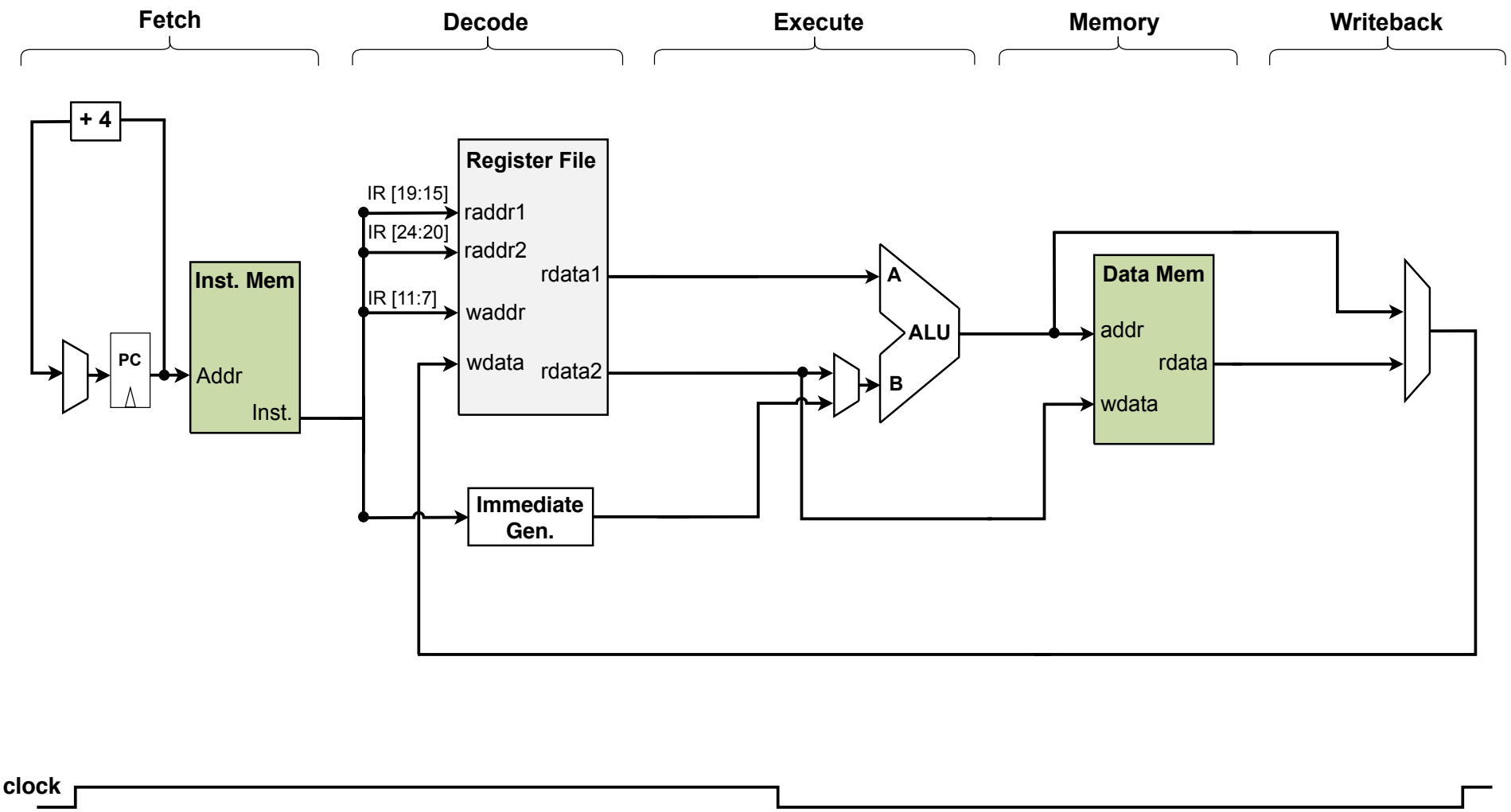
RISC-V ISA Supported by R- and I-type



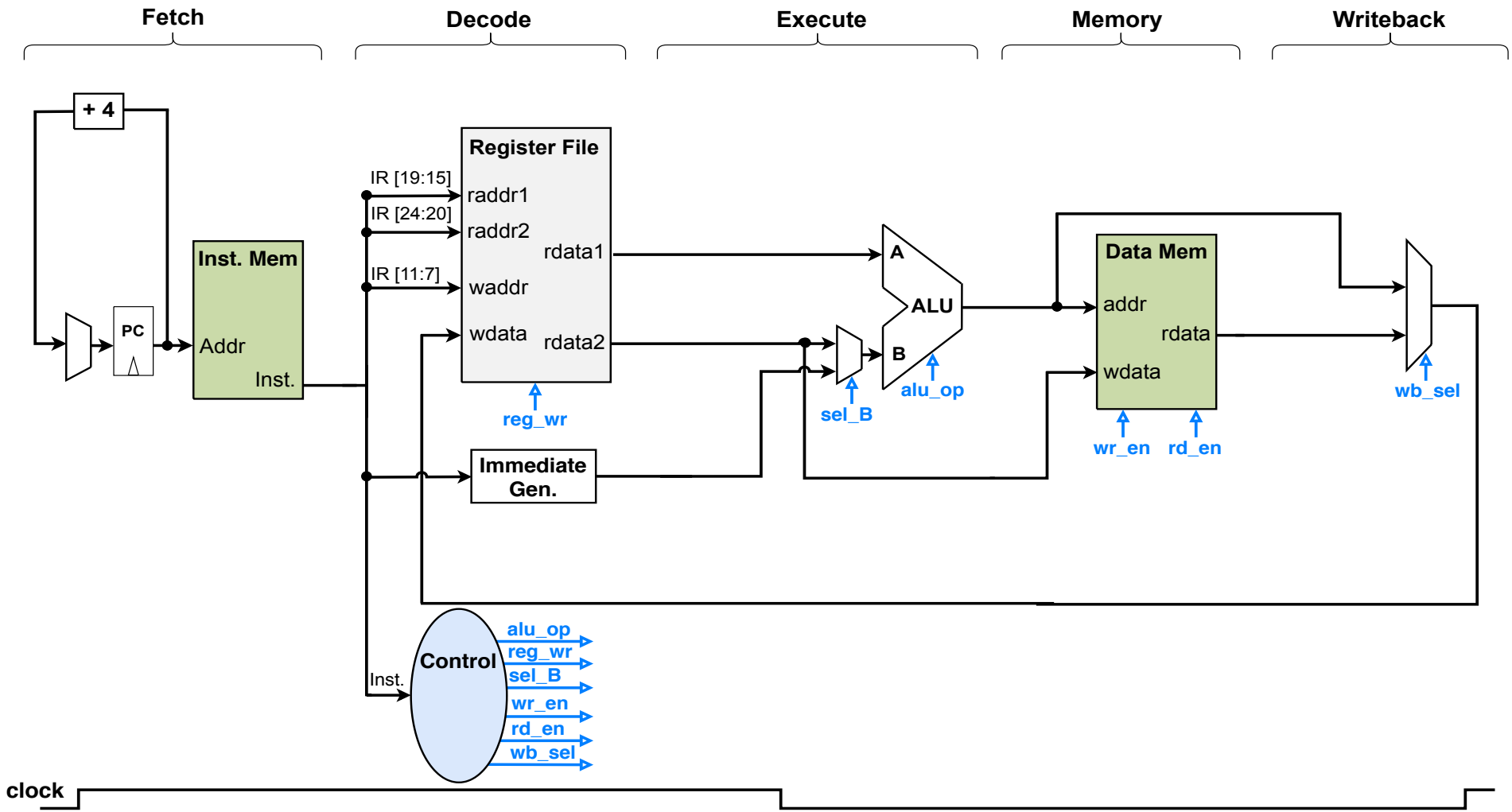
S-type: Store Operation



S-type: Store Operation Cont'd



S-type: Store Data- and Control-path



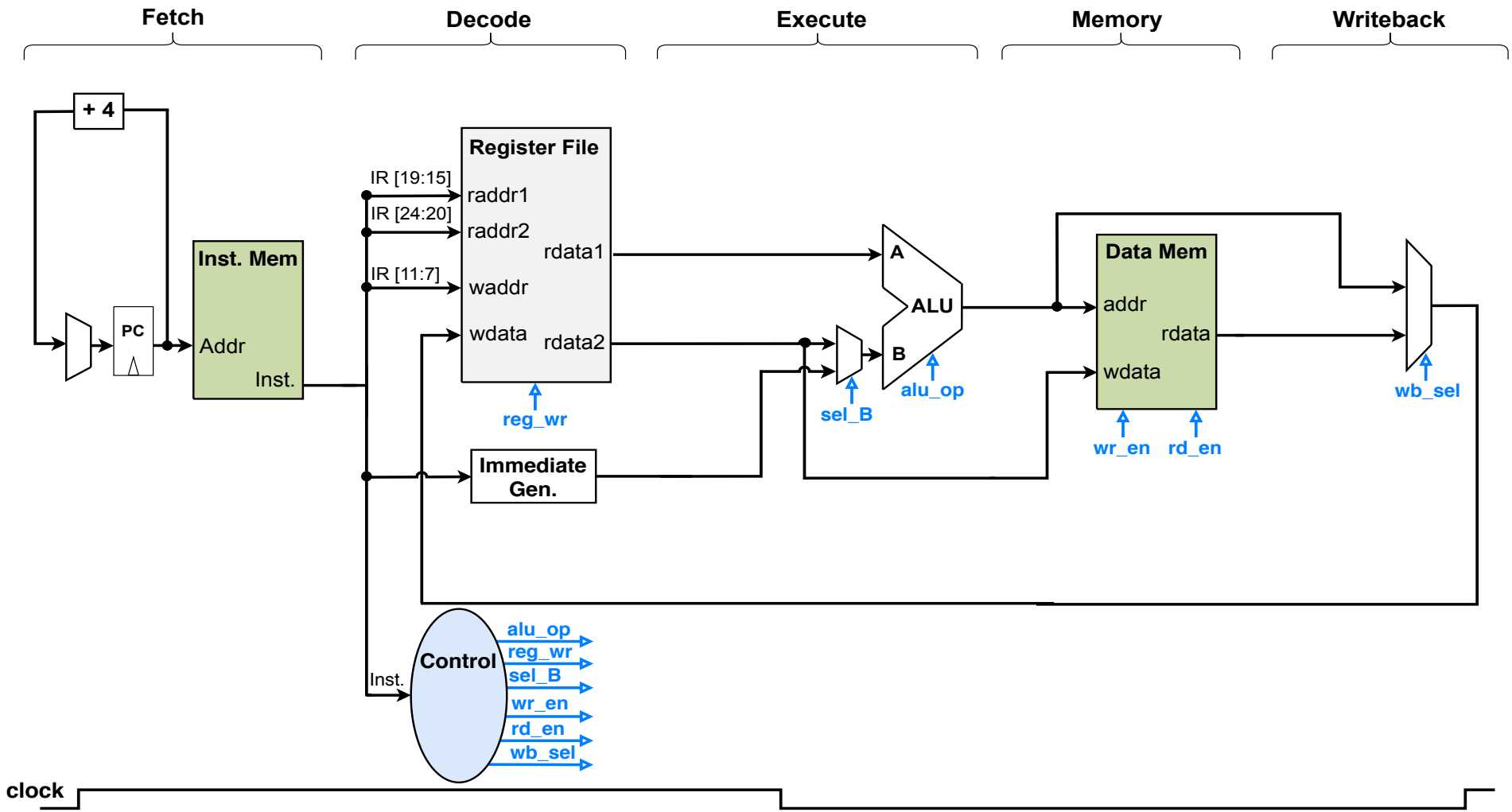
S-type: Store Word

- S-type: Store word (SW)

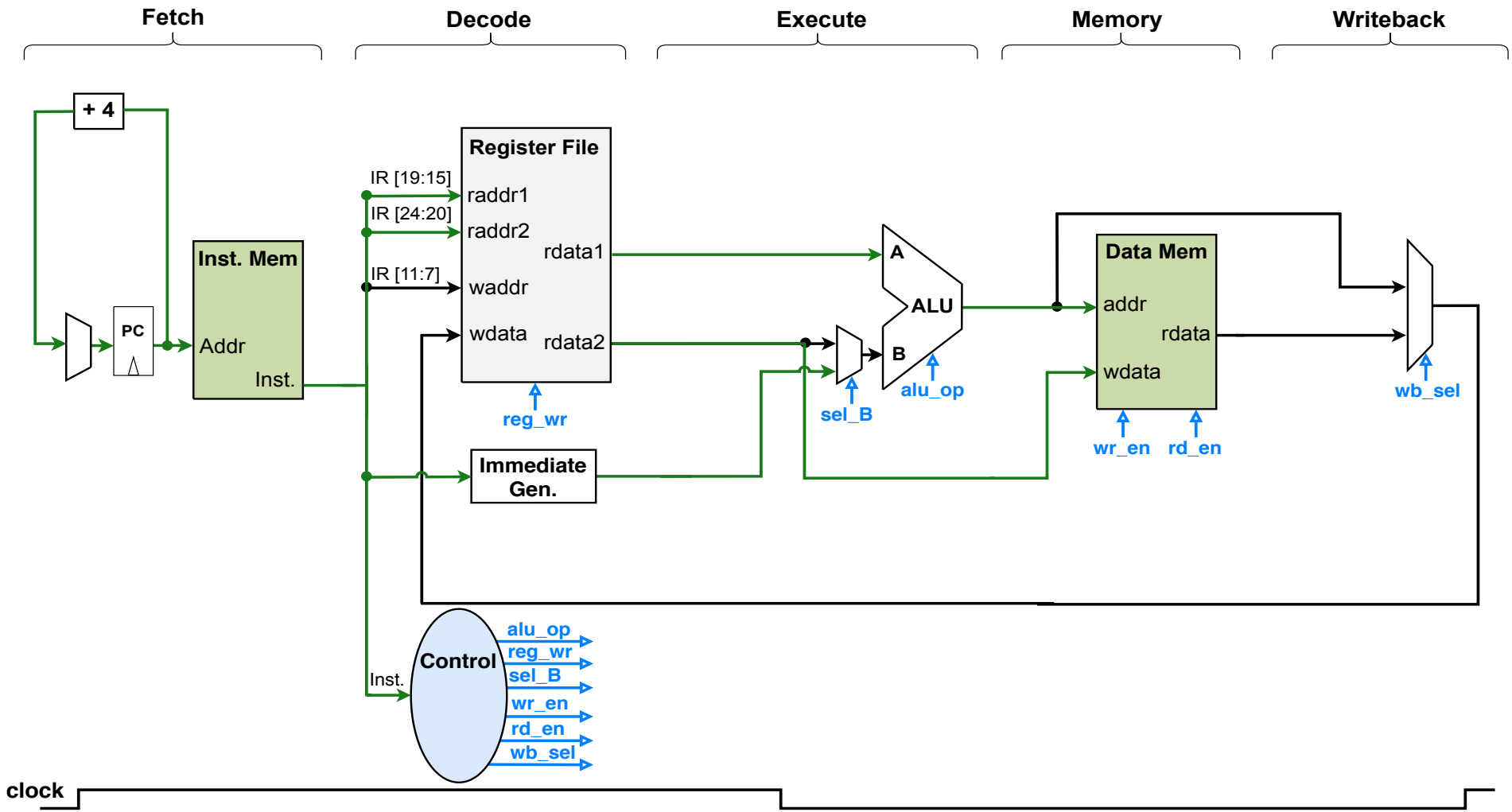
7	5	5	3	5	7
imm[11:5]	rs2	rs1	func3	imm[4:0]	opcode
immediate	rs2	rs1	010	immediate	0100011

- `S-imm` = signExtend(inst[31:25], inst[11:7])
- `opcode` = STORE: rs2 $\rightarrow$ mem[rs1 + S-imm]
- `funct3` = SW/SB/SH

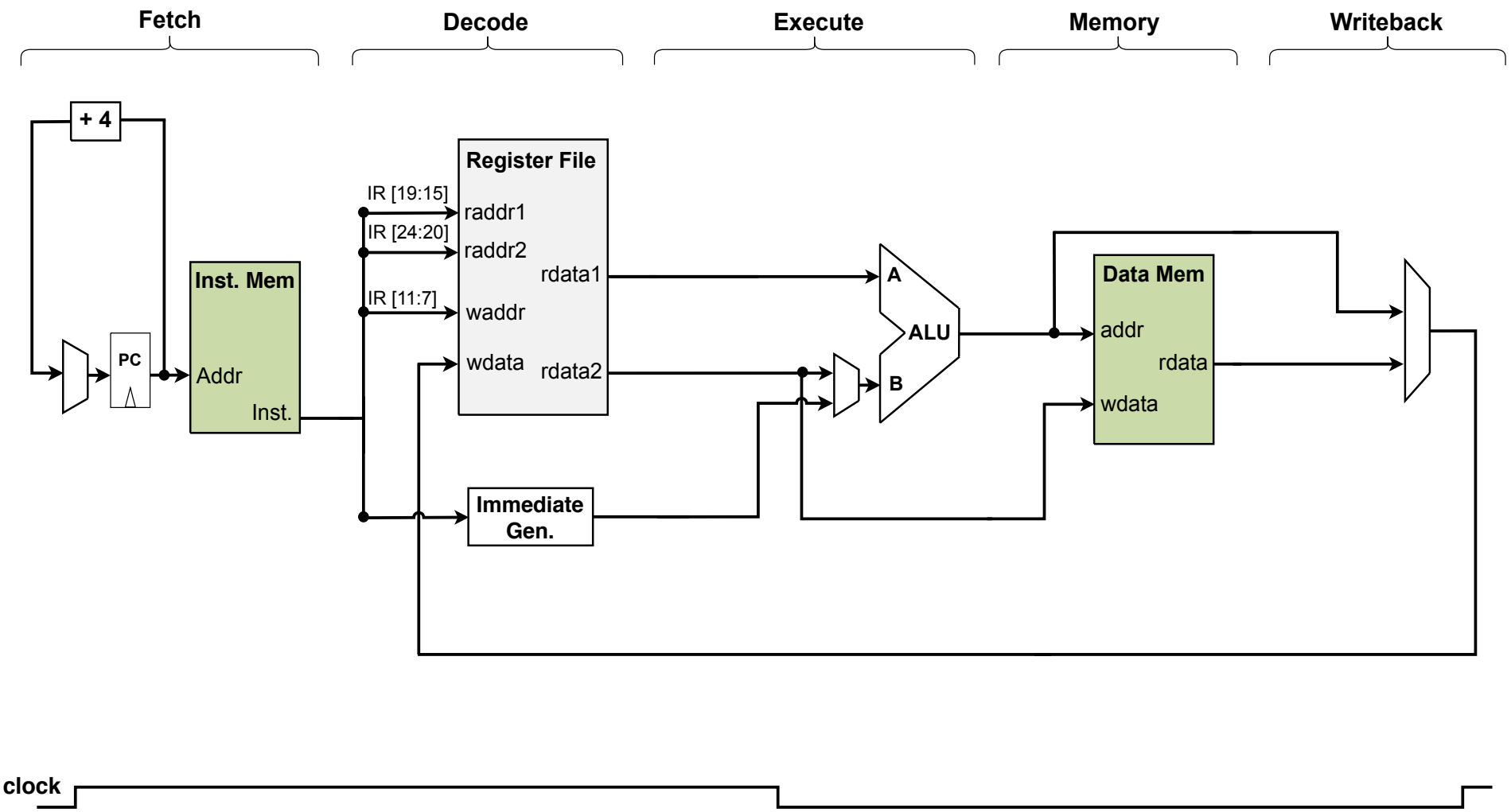
S-type: How Store Word Instruction Execute



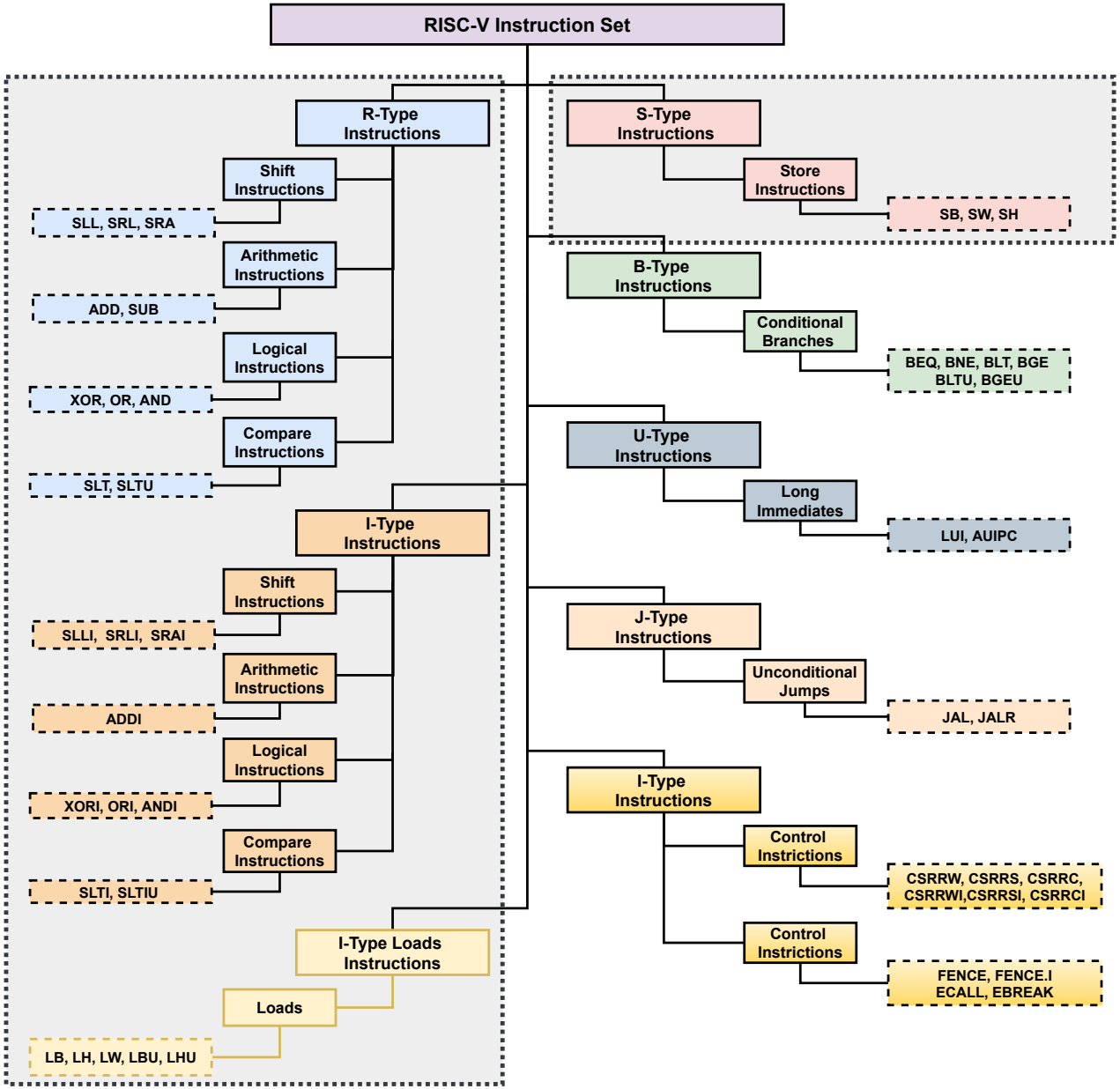
S-type: Store Instruction Execution



Datapath for R-, I- and S-type



RISC-V ISA Supported by R-, I- and S-type



Suggested Reading

- Read relevant sections of Chapters 2 and 4 of [Patterson and Hennessy, 2021].

Acknowledgment

- Preparation of this material was partly supported by Lampro Mellon Pakistan.

References

Patterson, D. and Hennessy, J. (2021).
Computer Organization and Design RISC-V Edition: The Hardware Software Interface, 2nd Edition.
Morgan Kaufmann.