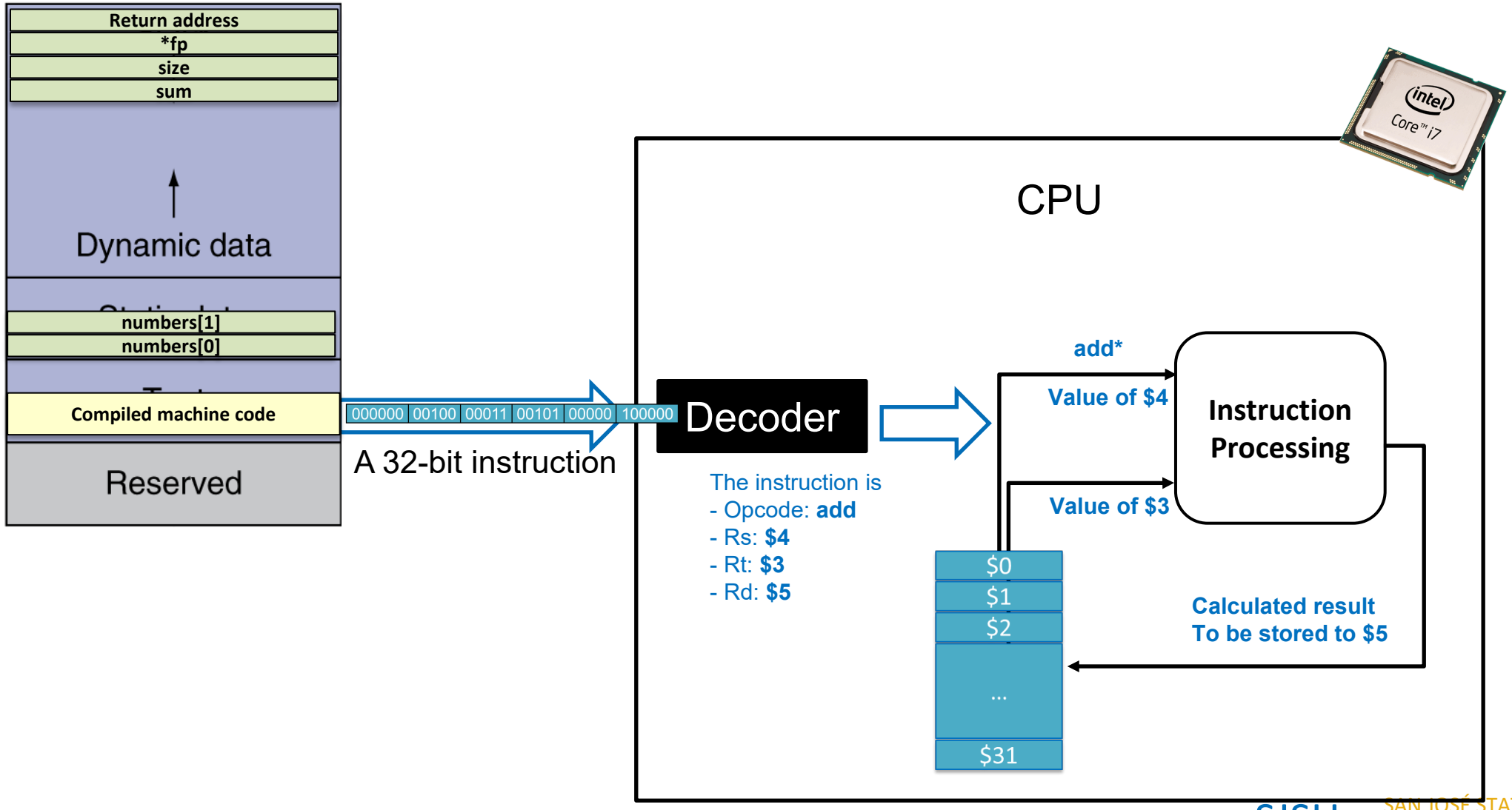
CMPE 200
Computer Architecture & Design

Lecture 3.
Processor Microarchitecture and Design (1)

Haonan Wang

SAN JOSÉ STATE
UNIVERSITY

Instruction Processing



Instruction Execution on CPU

- To run a software, CPU
  - **Fetches** instructions from memory in the right order
  - **Decodes** the instruction and **Reads** the operand values from register file
  - **Executes** the computation by using ALU
  - **Loads/store** data from/to memory
  - **Writes back** the computation results to register file

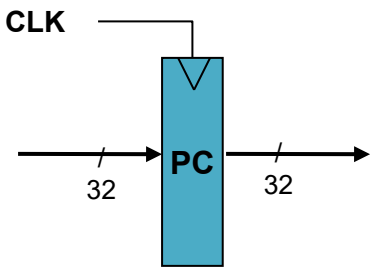
Implementing a Single-Cycle CPU

- **All instructions execute in a single clock cycle**
  - Instructions to Implement: LW, SW; ADD, SUB, AND, OR, SLT; BEQ
  - CPI = 1
- **Different instructions need different execution times**
  - e.g., LW needs to access memory *vs.* ADD does not need to access memory
- **Single-cycle CPU's clock cycle period should be set to the longest instruction's execution time**

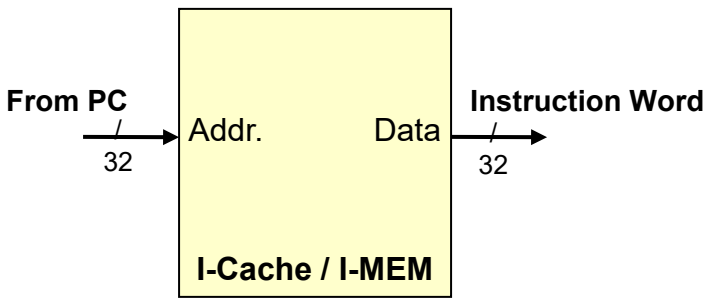
Fetch Components

- **Required operations**
  - Taking address from PC and reading instruction from memory
  - Incrementing PC to point at next instruction

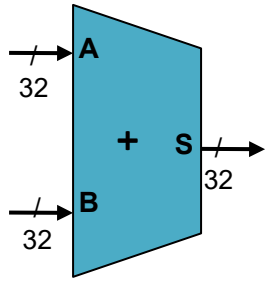
- **Components**
  - PC register
  - Instruction Memory / Cache
  - Adder to increment PC value



Register

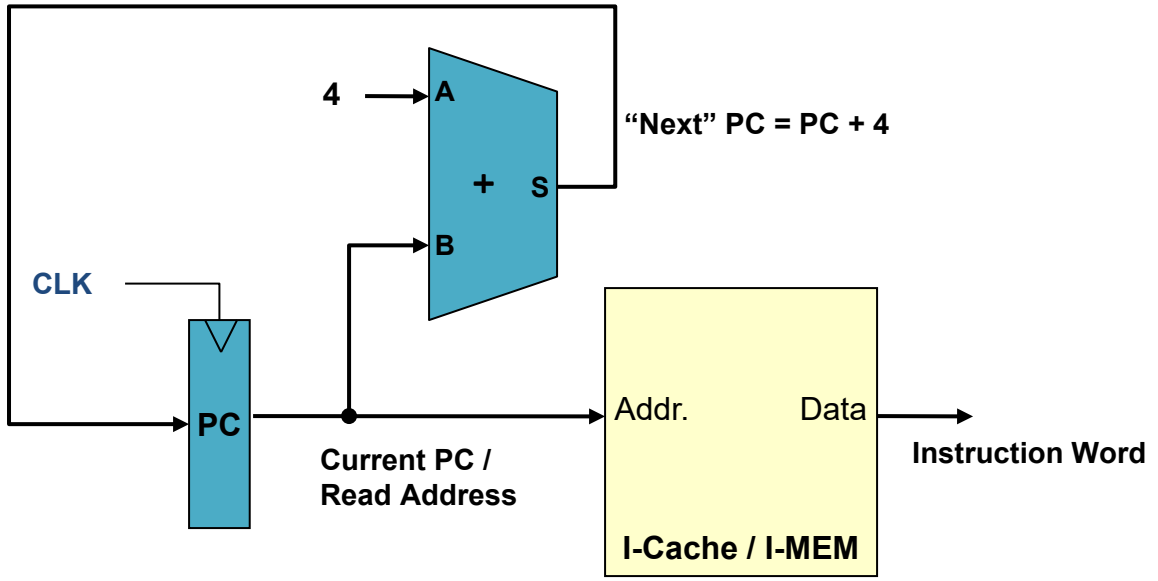


Memory



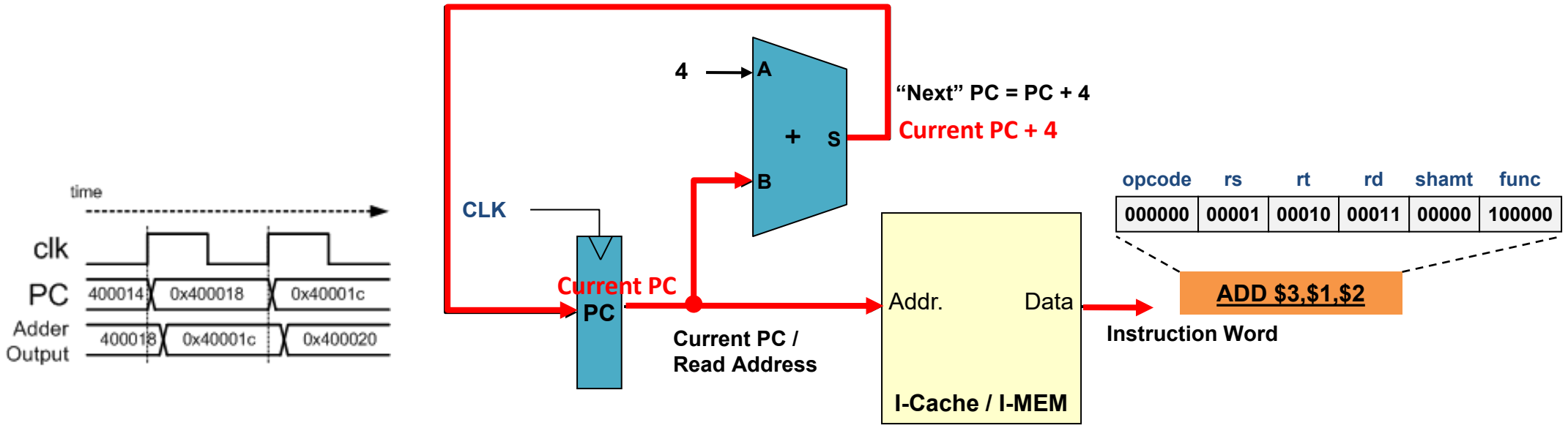
Adder

Fetch Data Path



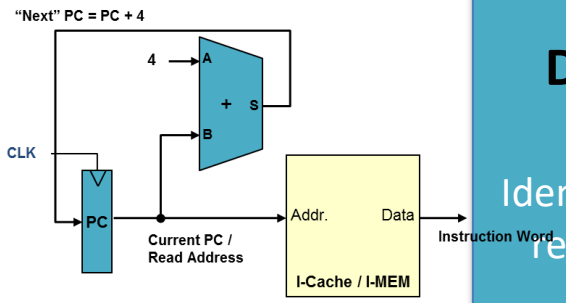
Fetch Data Path

- PC value serves as address to instruction memory while also being incremented by 4 using the adder
- Instruction word is returned by memory after some delay
- New PC value is clocked into PC register at the end of clock cycle



Decode

- **What is the instruction and operands?**



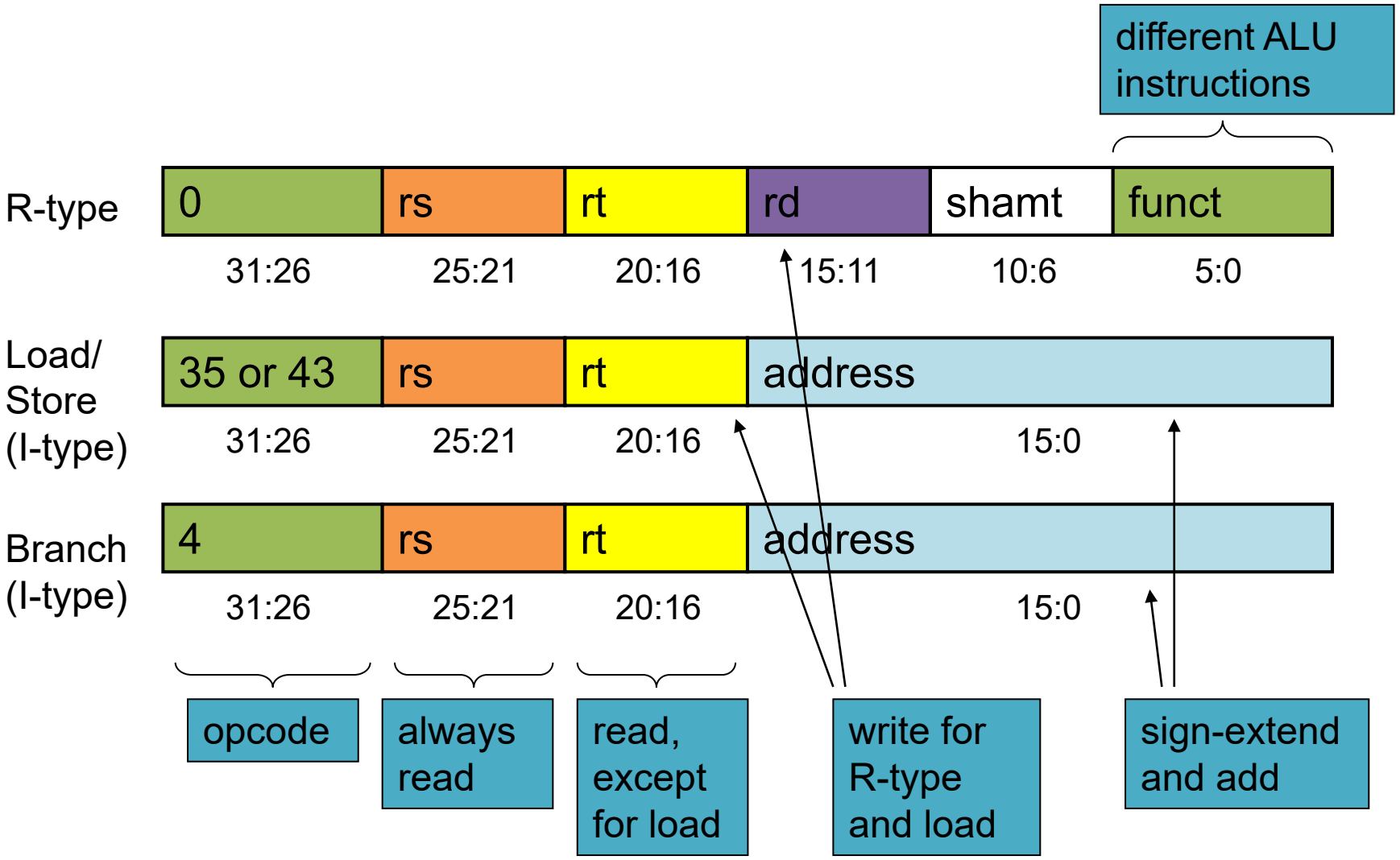
Decode & Register
Identify opcode & read operands

Execute

Mem

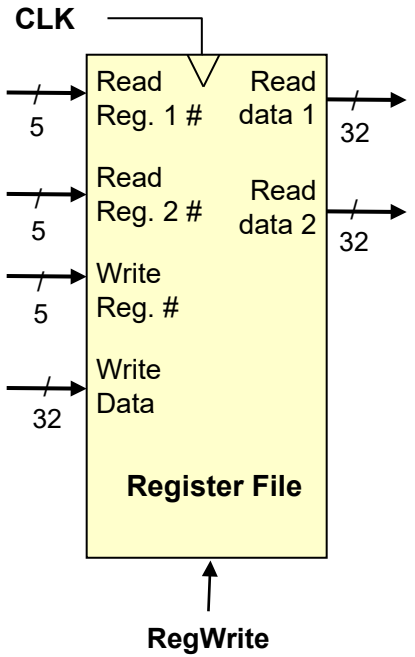
Write Result
Write to dst register

Instruction Formats



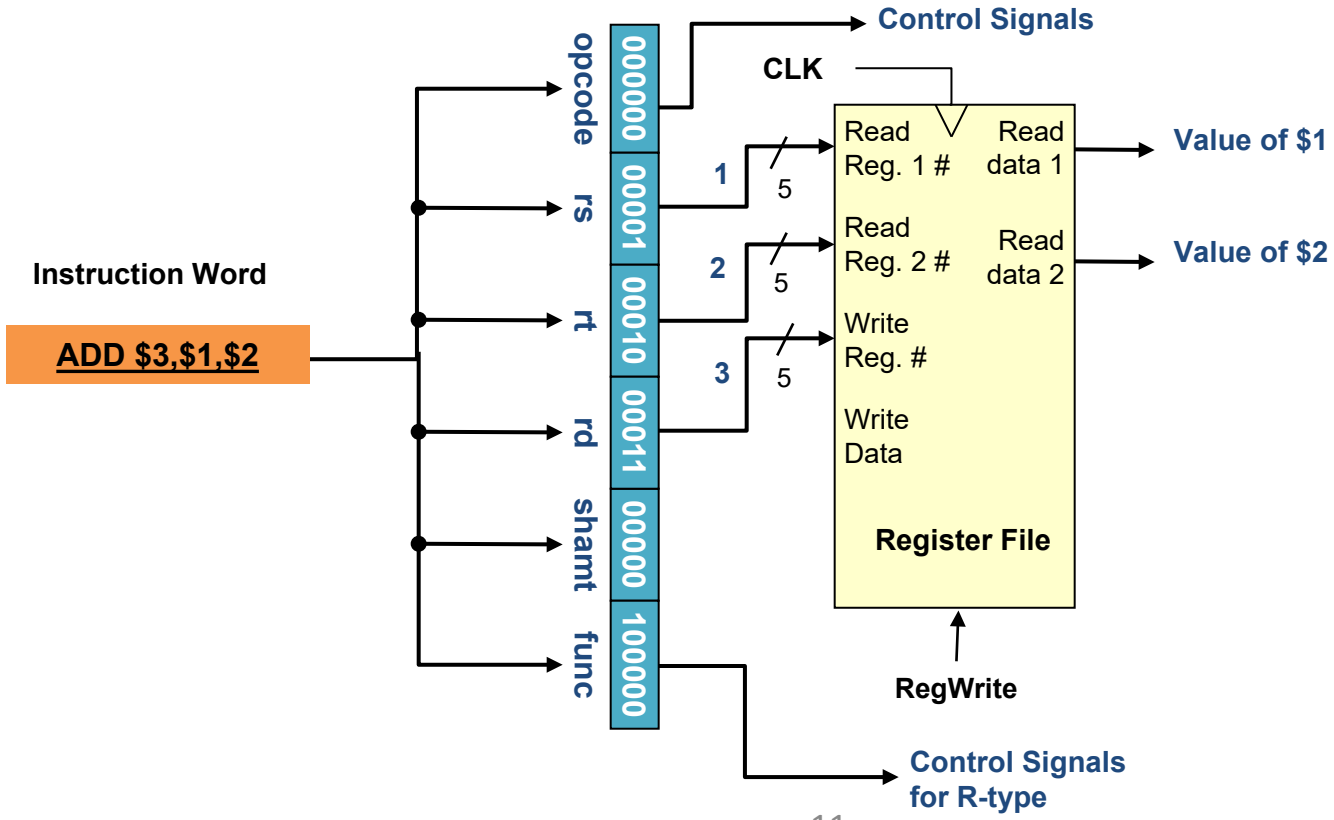
Decode Components

- **Register file is a collection of registers**
- **To support R-type instruction,**
  - 4 Input ports: two src register ids, a dst register id and the result value
  - 2 Output ports: two operand values



Decode Data Path

- **R-type**
  - Opcode and func. field are decoded to produce control signals
  - Three register ids are passed to Register file



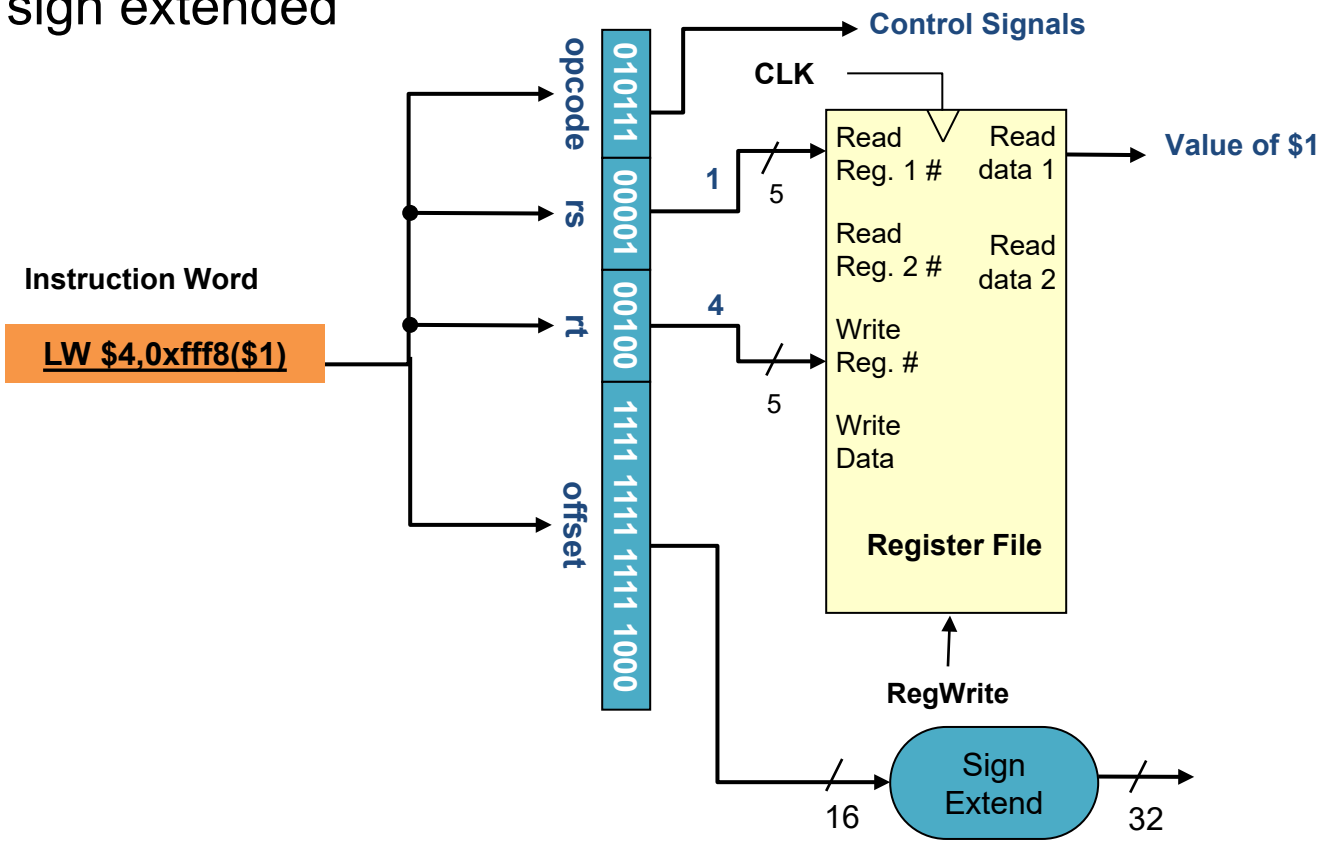
Sign Extension Unit

- **In a 'LW' or 'SW' instructions with their base register + offset format, the instruction only contains the offset as a 16-bit value**
  - Example: LW $4,-8($1)
  - Machine Code: 0x8c24fff8

- **A 16-bit offset must be extended to 32-bits to add to the base register**

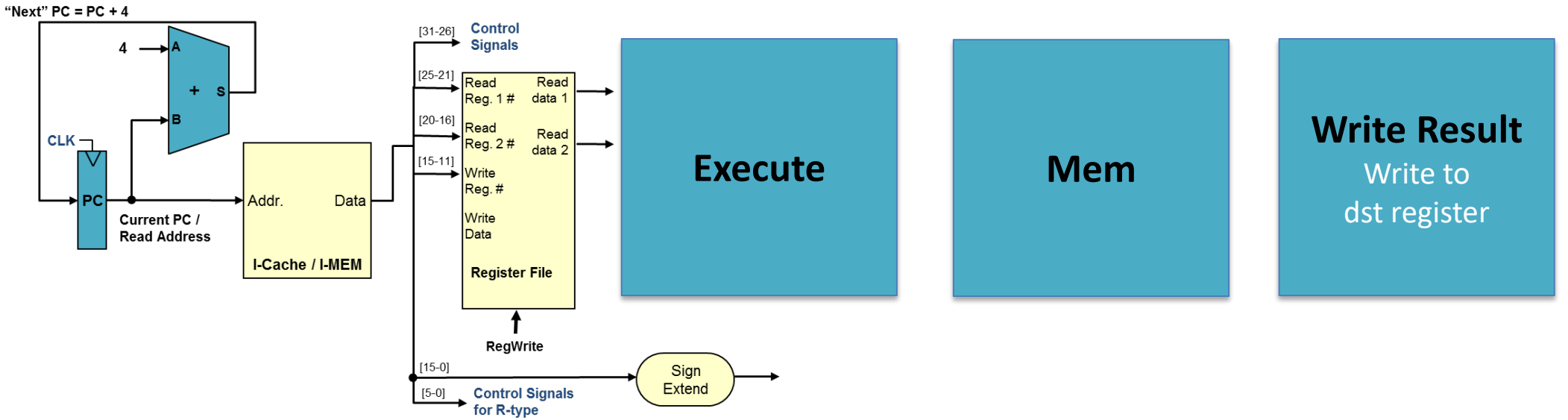
Decode Data Path

- **I-type**
  - Opcode field is decoded to produce control signals
  - Two register ids are passed to Register file
  - Offset is sign extended



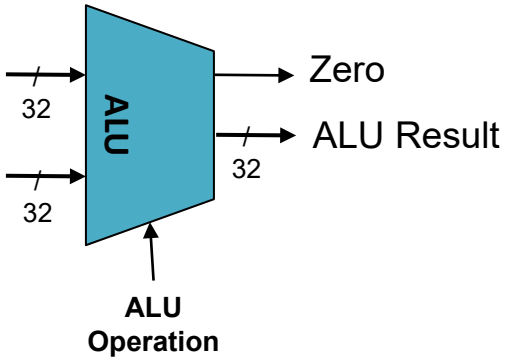
Execution & Write Result

- **Let's execute the instruction!**

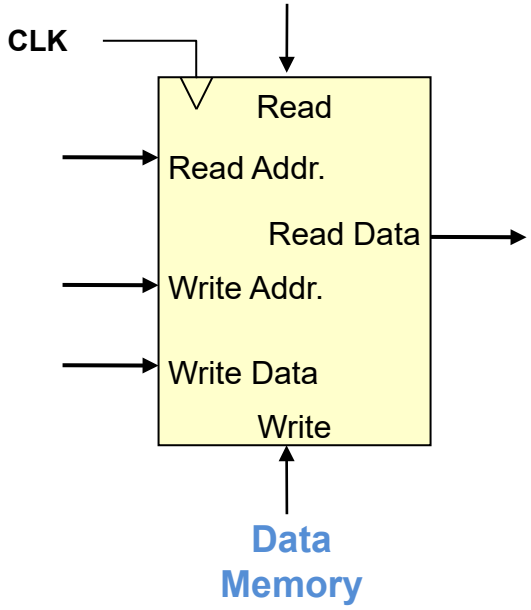


Execution Components

- **ALU is used for**
  - R-type instructions to operate various arithmetic operations
  - LW/SW instructions to operate (base address + offset) to generate the target memory address
  - Branch instructions to compare (subtract) the two operand values
- **Data Memory for LW/SW instructions**
- **Shift-left-2 and another Adder for Branch instructions to calculate new PC value**

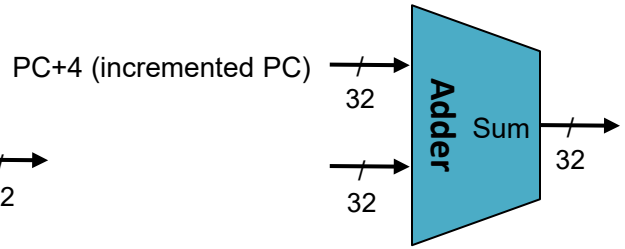


ALU



Data Memory

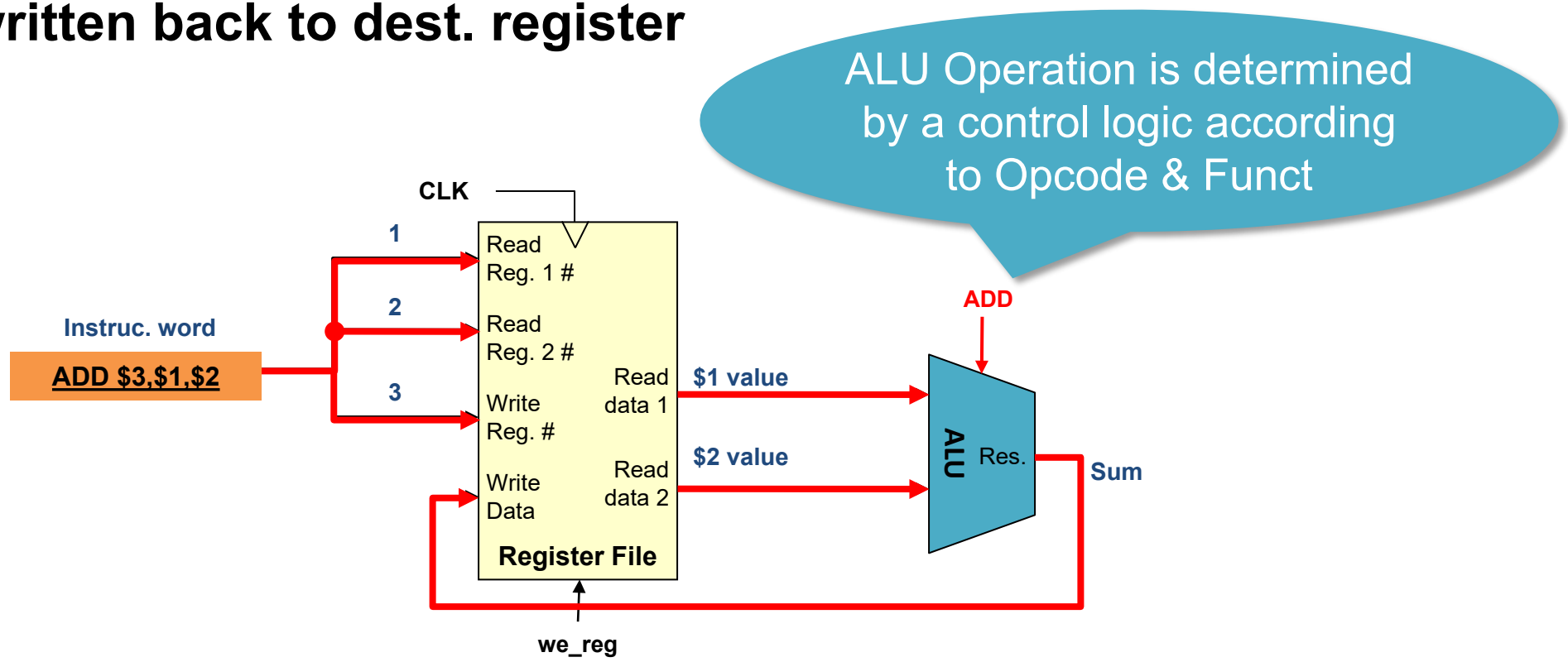
Shift-left-2



Adder

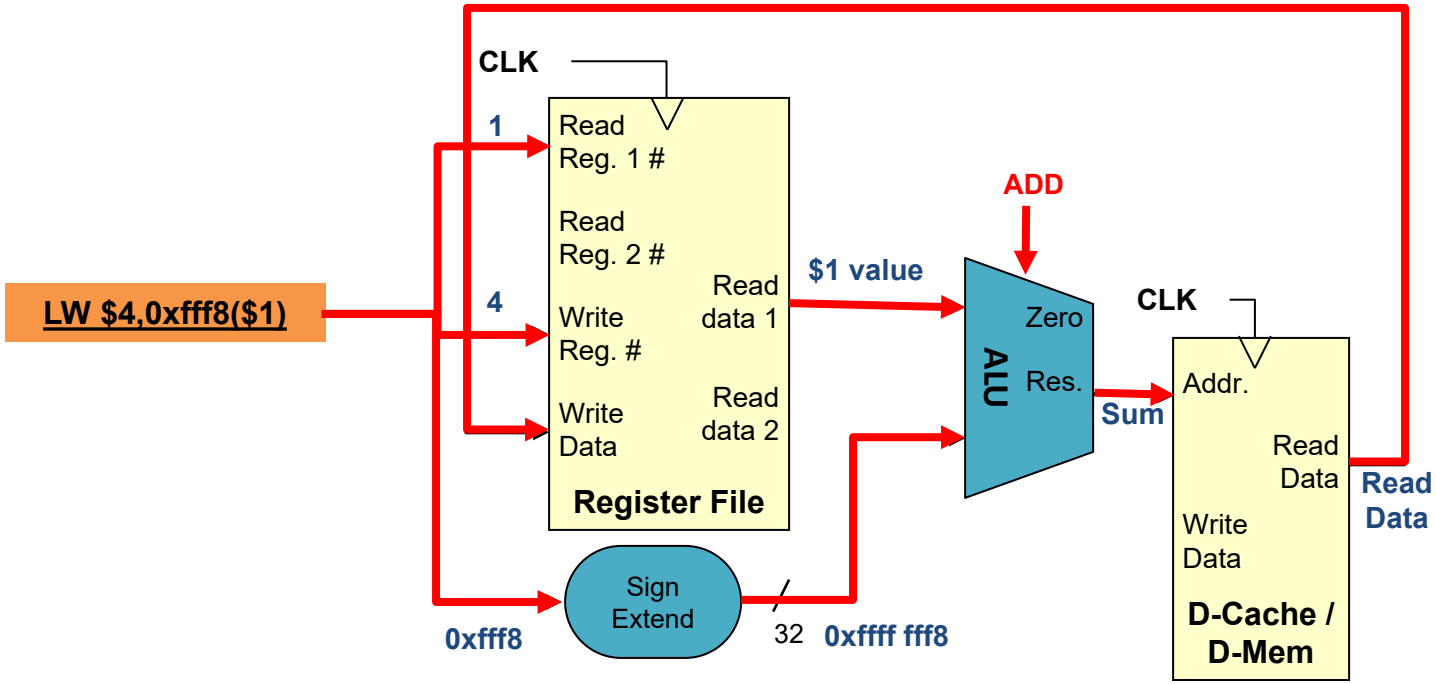
Datapath for R-type Instructions

- **ALU takes inputs from register file and performs the add, sub, and, or, slt, operations**
- **Result is written back to dest. register**



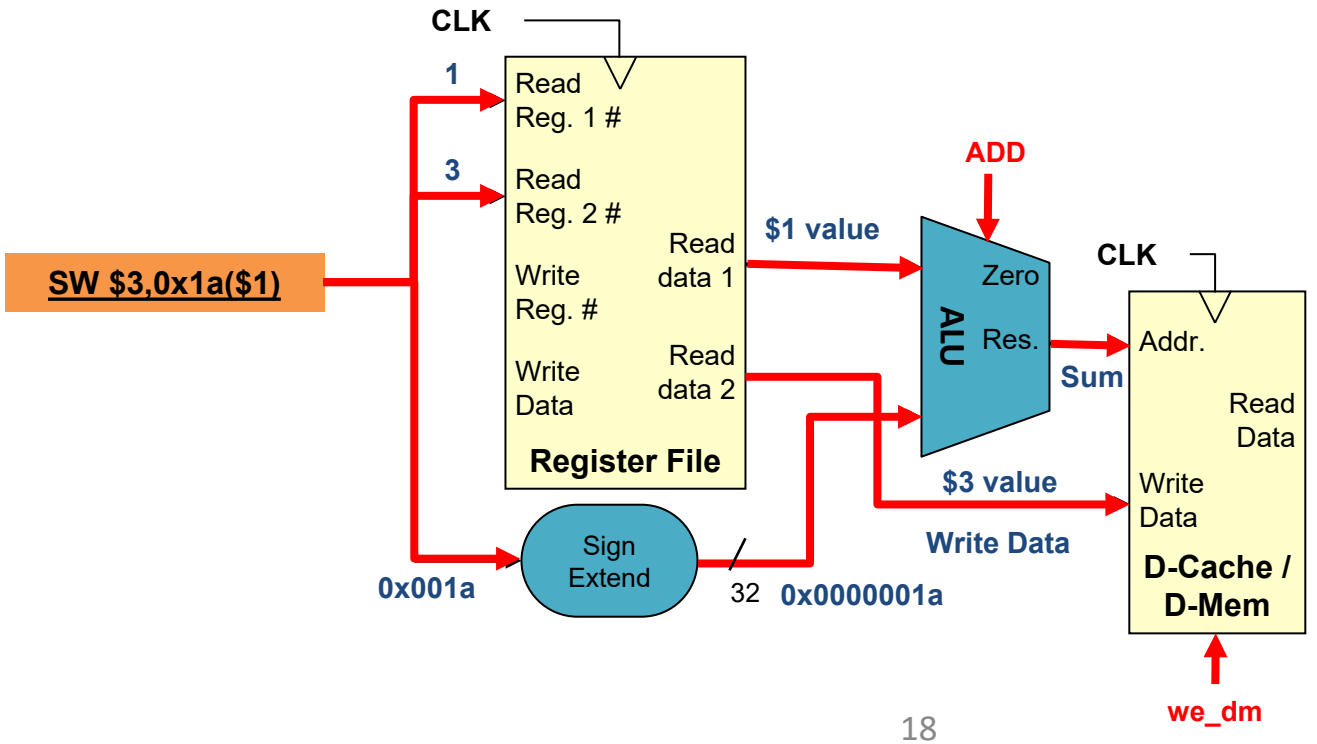
Memory Access Datapath: LW

- Operands are read from register file while offset is sign extended
- ALU calculates target memory address
- Memory access is performed
- If LW, read data is written back to register



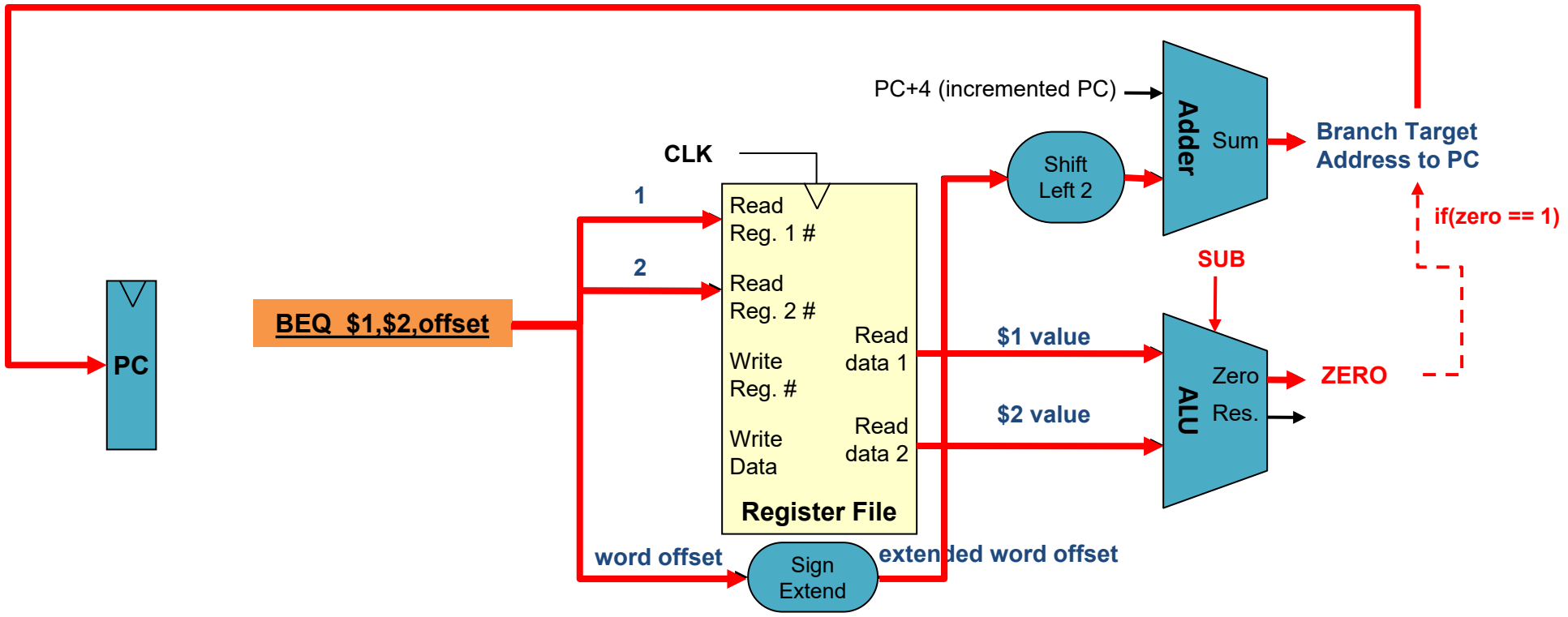
Memory Access Datapath: SW

- Operands are read from register file while offset is sign extended
- ALU calculates target memory address
- If SW, data is written to the memory



Branch Datapath: BEQ

- ALU for comparison (examine 'zero' output)
- Sign extension unit for branch offset
- Adder to add PC and offset



Conclusion Time

What are the stages of single-cycle CPU execution?

- **Fetch, Decode, Execute, Memory, Writeback**

→ IF, ID, EX, MEM, WB

What are the two different ways to calculate addresses in the CPU?

- **ALU** **→ Byte addresses, data access.**
- **Shift-left-2 & Adder** **→ Label, PC.**

SAN JOSÉ STATE UNIVERSITY *powering* SILICON VALLEY