Computer Architecture
Computer Engineering Track
Tutorial 14

MIPS Processor: The Basic Implementation Scheme

Artem Burmyakov, Alexander Tormasov

November 25, 2021

INNOPOLIS UNIVERSITY

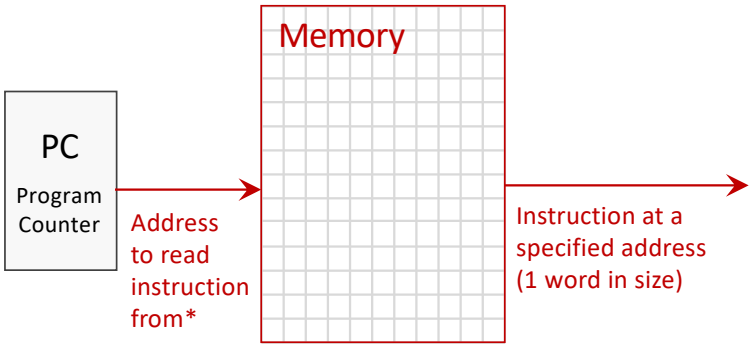
Program Counter

Program Counter - a special-purpose register containing the identifier of the next CPU instruction to be executed

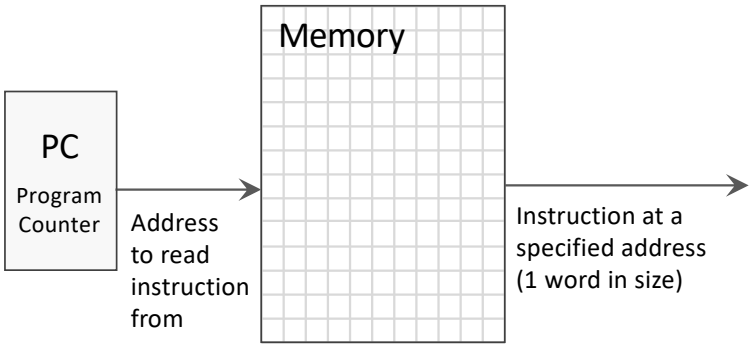
Recap:

- A program is a set of CPU instructions loaded into system memory;
- Every instruction takes 32 bits (1 word, or 4 bytes) of memory, in its binary representation

Program Counter and Memory

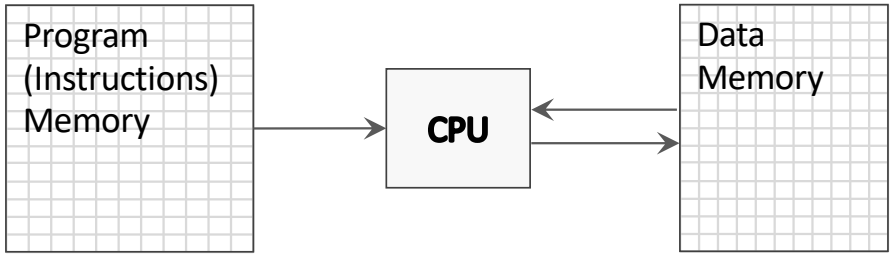


* Arrows indicate the direction of the data flow between CPU components:
E.g. the address of the instruction corresponding to the identifier stored in the PC register is sent to the memory controller, and the content of a corresponding memory cell is output



Harvard vs. Von Neumann Architecture for System Memory

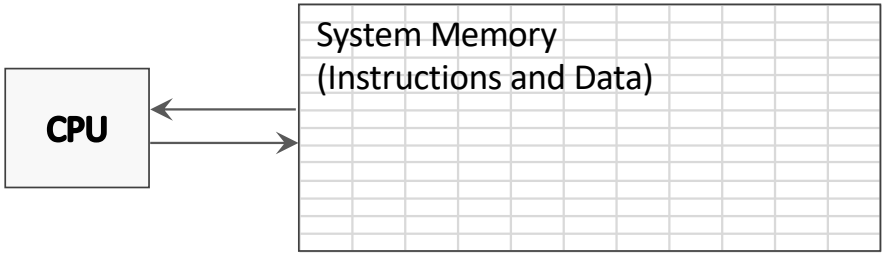
Harvard Architecture:



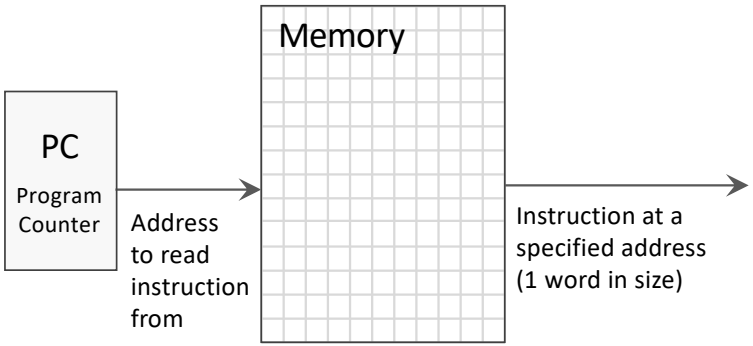
Read only

Read and Write

Von Neumman Architecture:

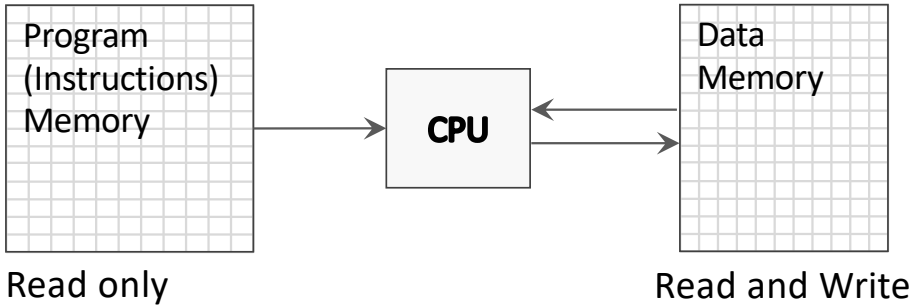


Read and Write

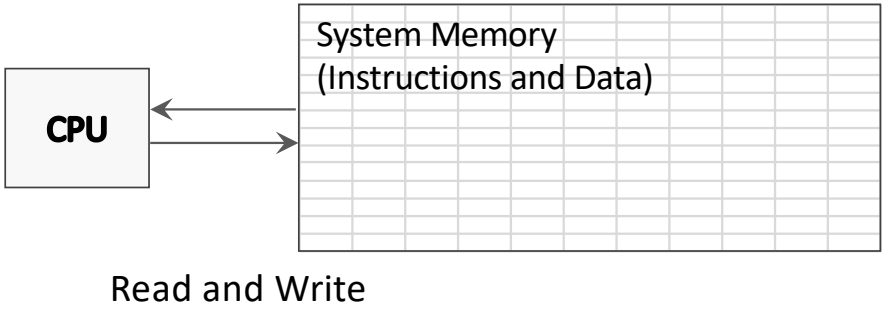


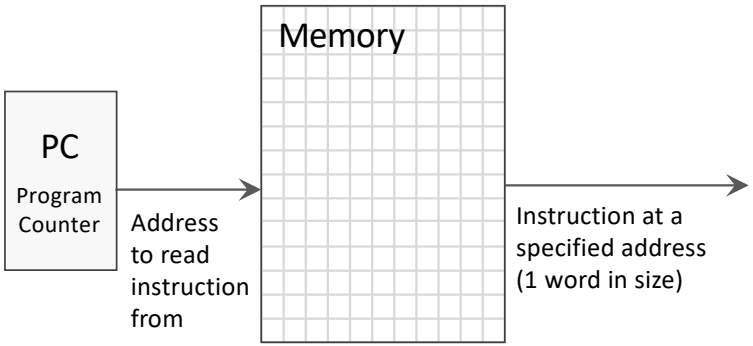
Harvard vs. Von Neumann Architecture for System Memory

Harvard Architecture – not used that much



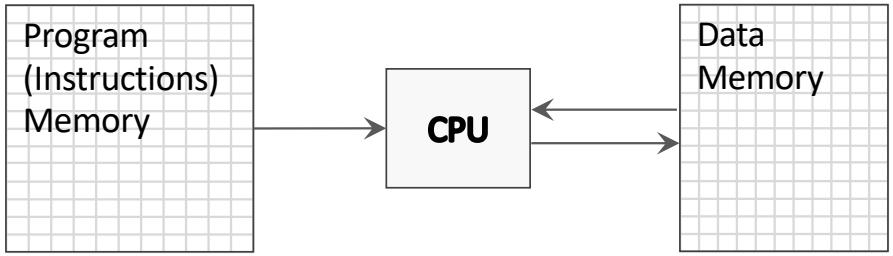
Von Neumman Architecture **– widely used**





Harvard vs. Von Neumann Architecture for System Memory

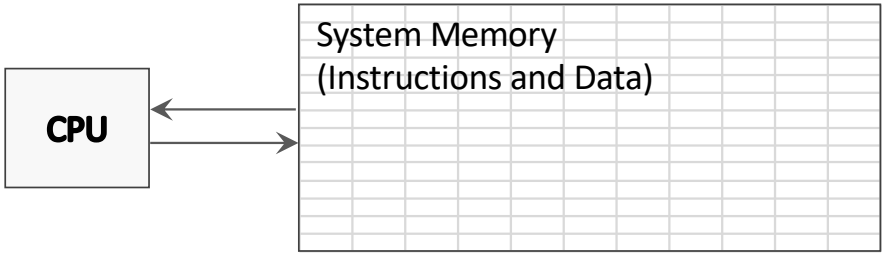
Harvard Architecture – not used that much



Read only

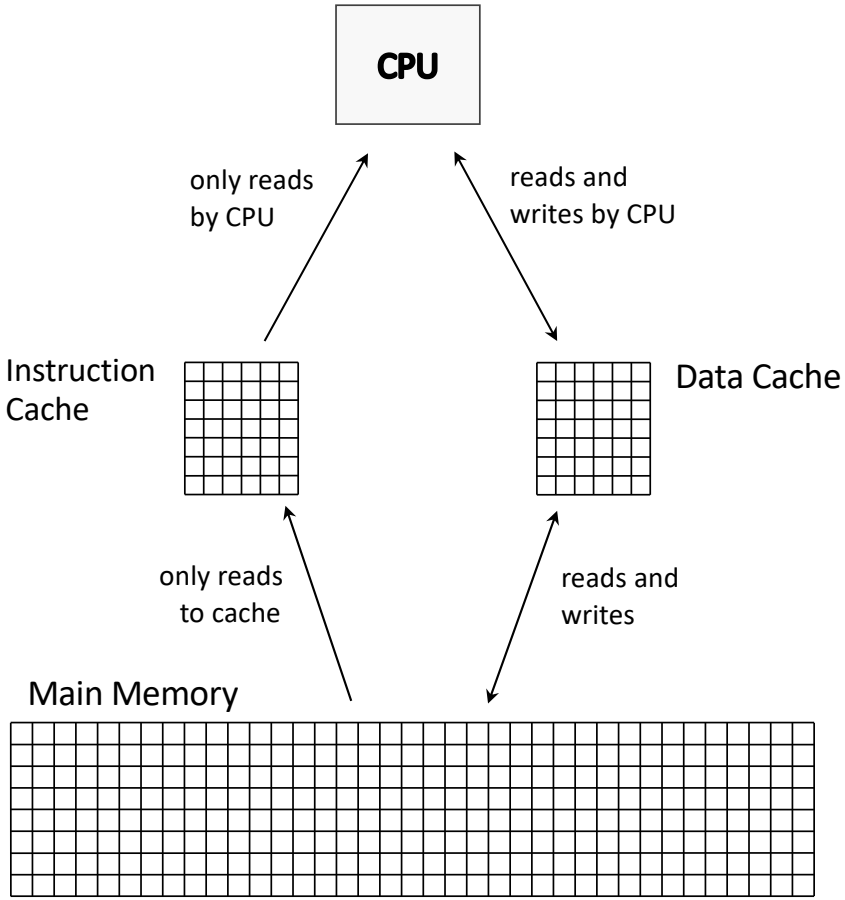
Read and Write

Von Neumman Architecture – widely used

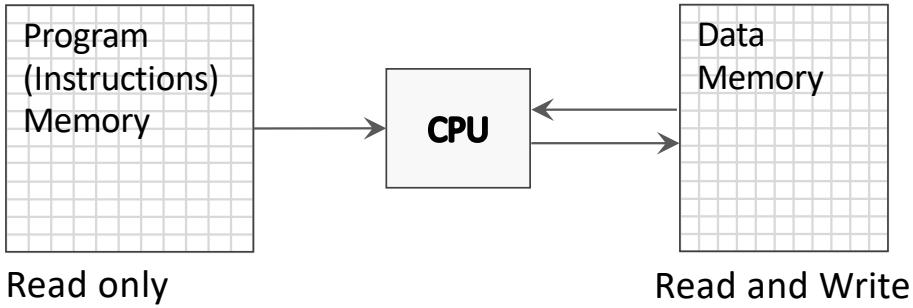


Read and Write

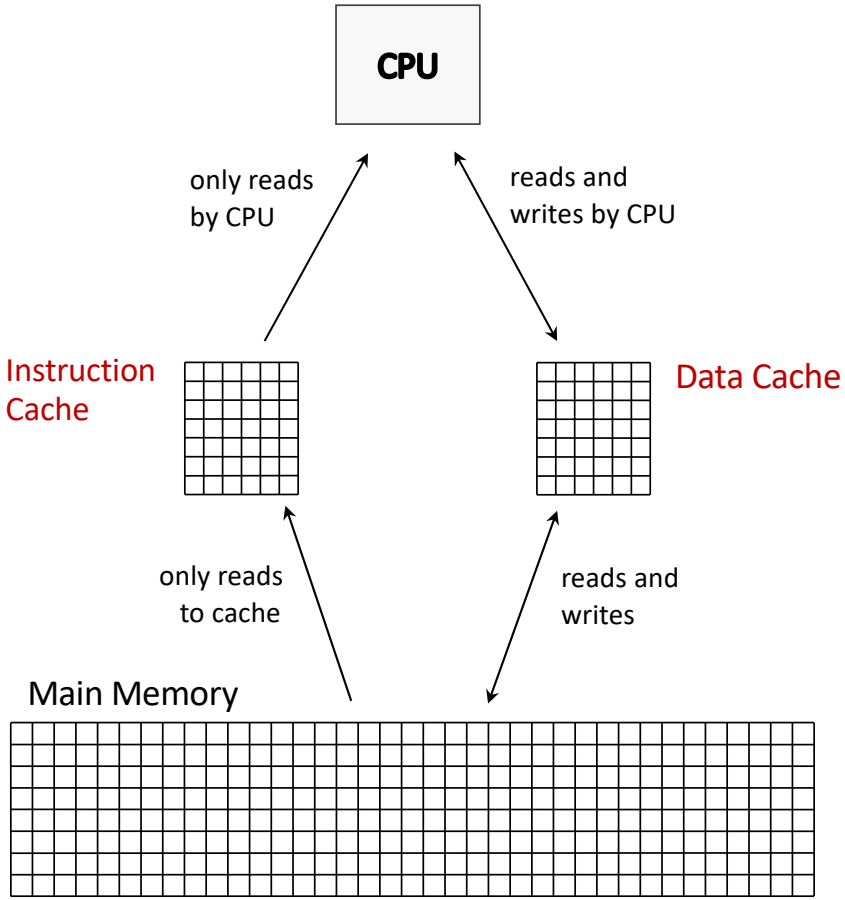
Harvard architecture inspires this cache model:



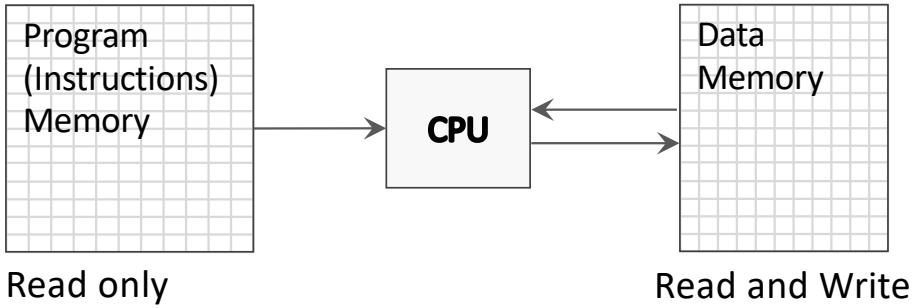
Harvard Architecture for System Memory



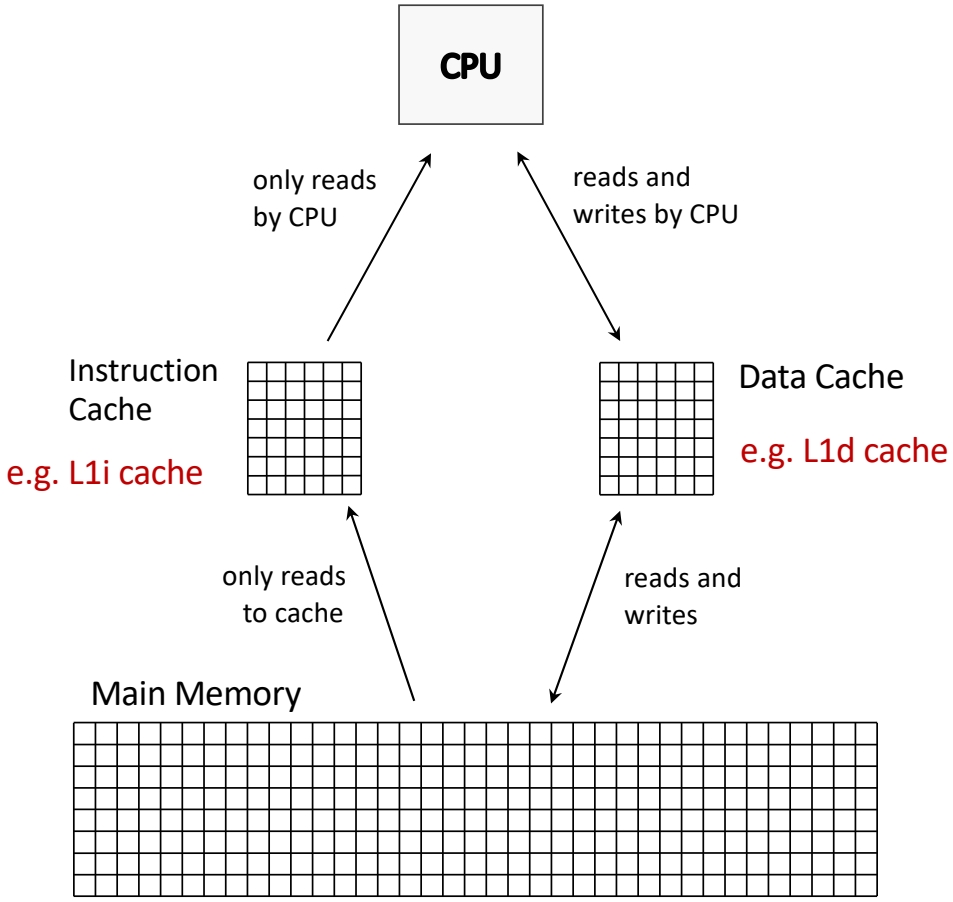
Harvard architecture inspires this cache model:



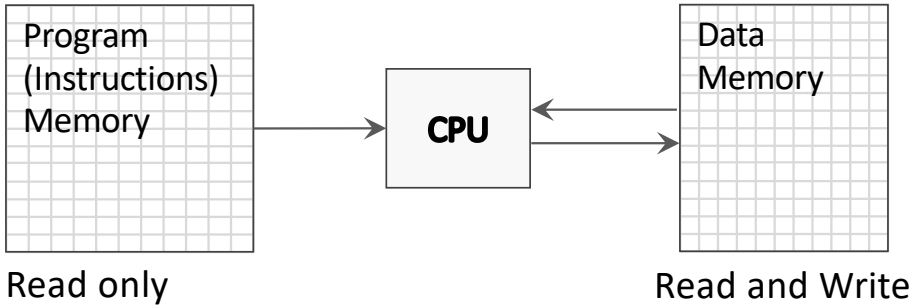
Harvard Architecture for System Memory



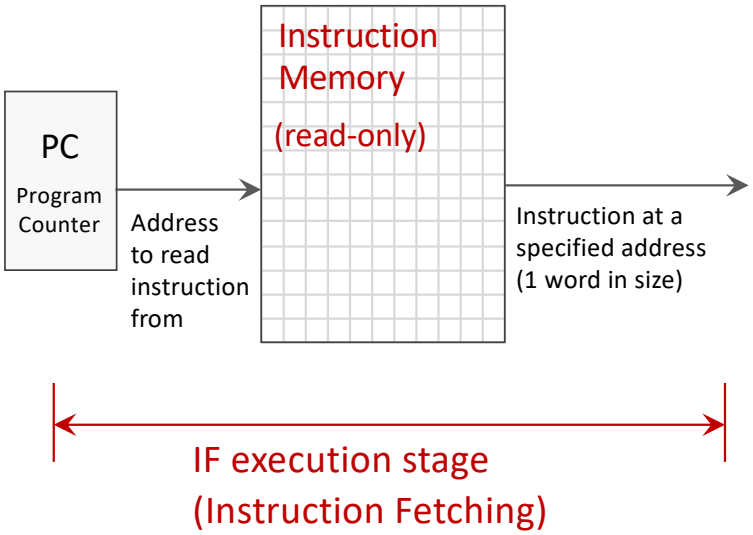
Harvard architecture inspires this cache model:



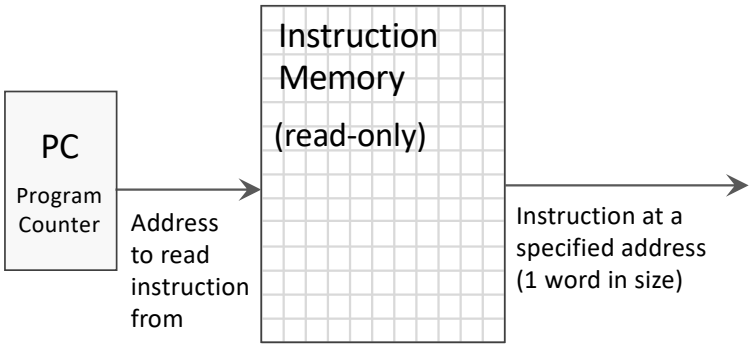
Harvard Architecture for System Memory



Instruction Fetching

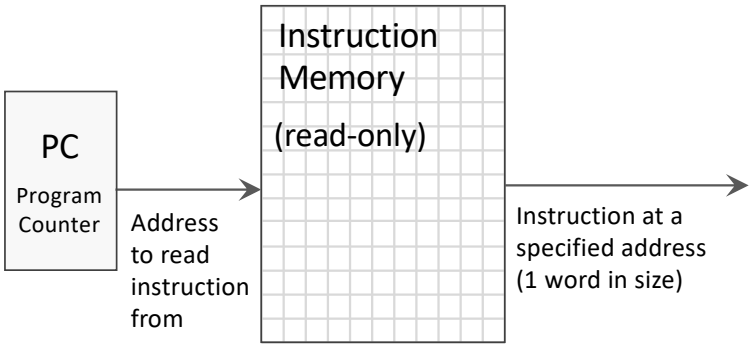


Program Counter Update



In parallel to instruction fetching, CPU computes the address of the next instruction to be executed

Program Counter Update

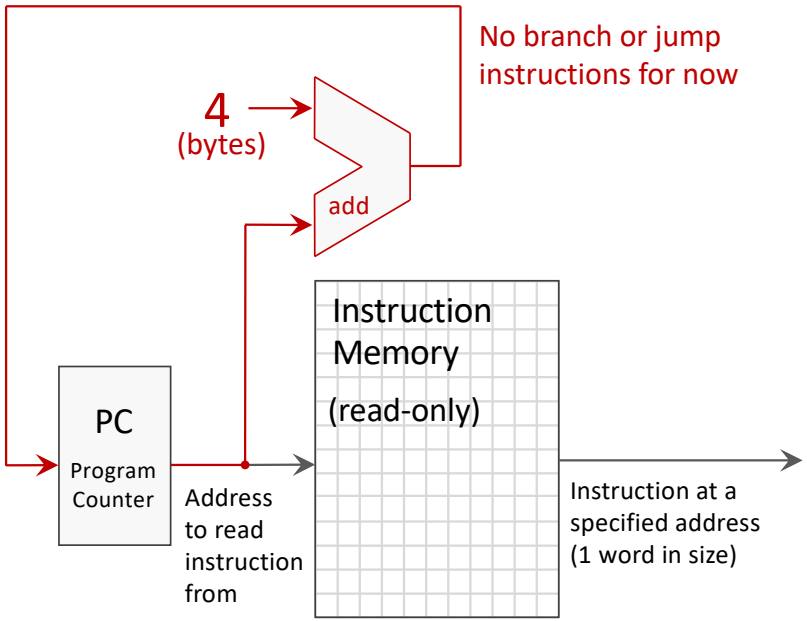


In parallel to instruction fetching, CPU computes the address of the next instruction to be executed

Assumption:

- All program instructions are loaded into a continuous memory block;
- Every instruction takes 4 bytes of memory;
- Consider no branch or jump instructions for now

Program Counter Update (branch or jump instruction excluded)

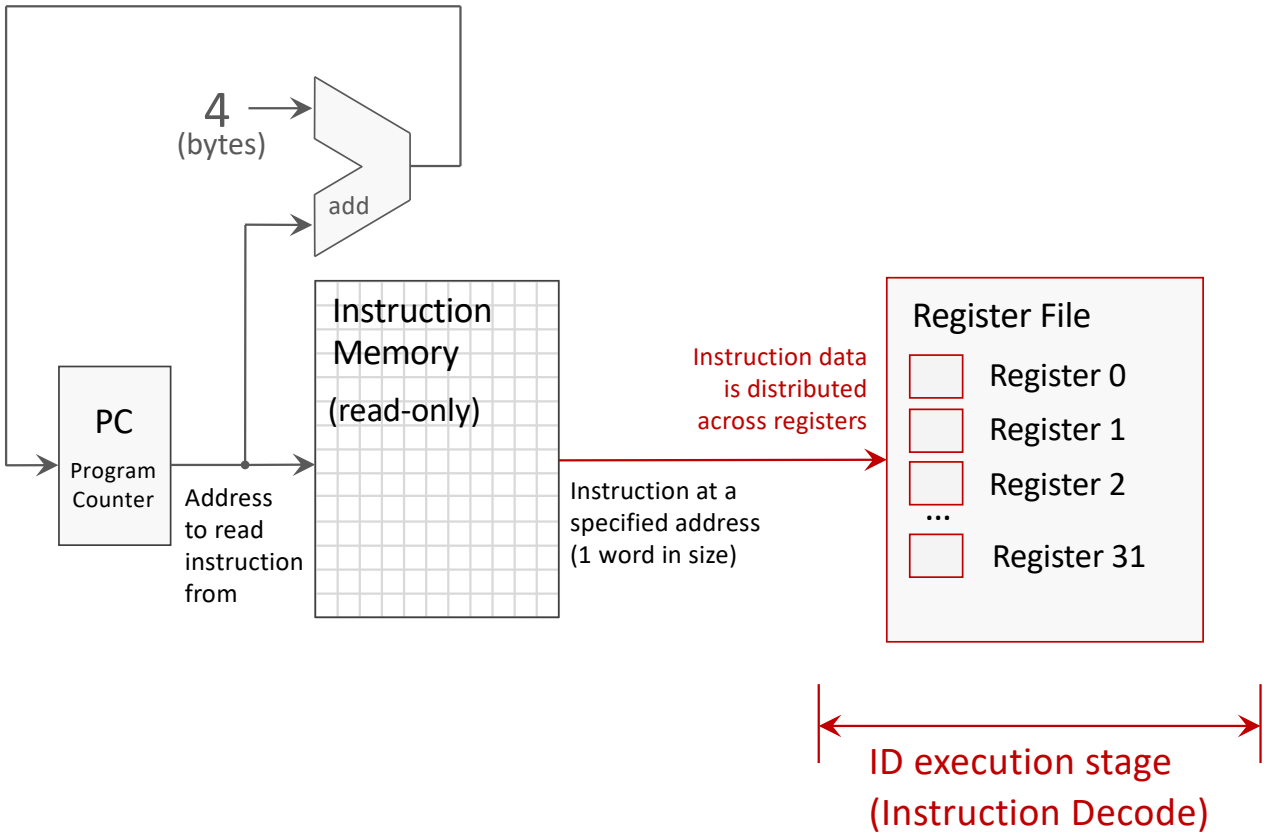


In parallel to instruction fetching, CPU computes the address of the next instruction to be executed

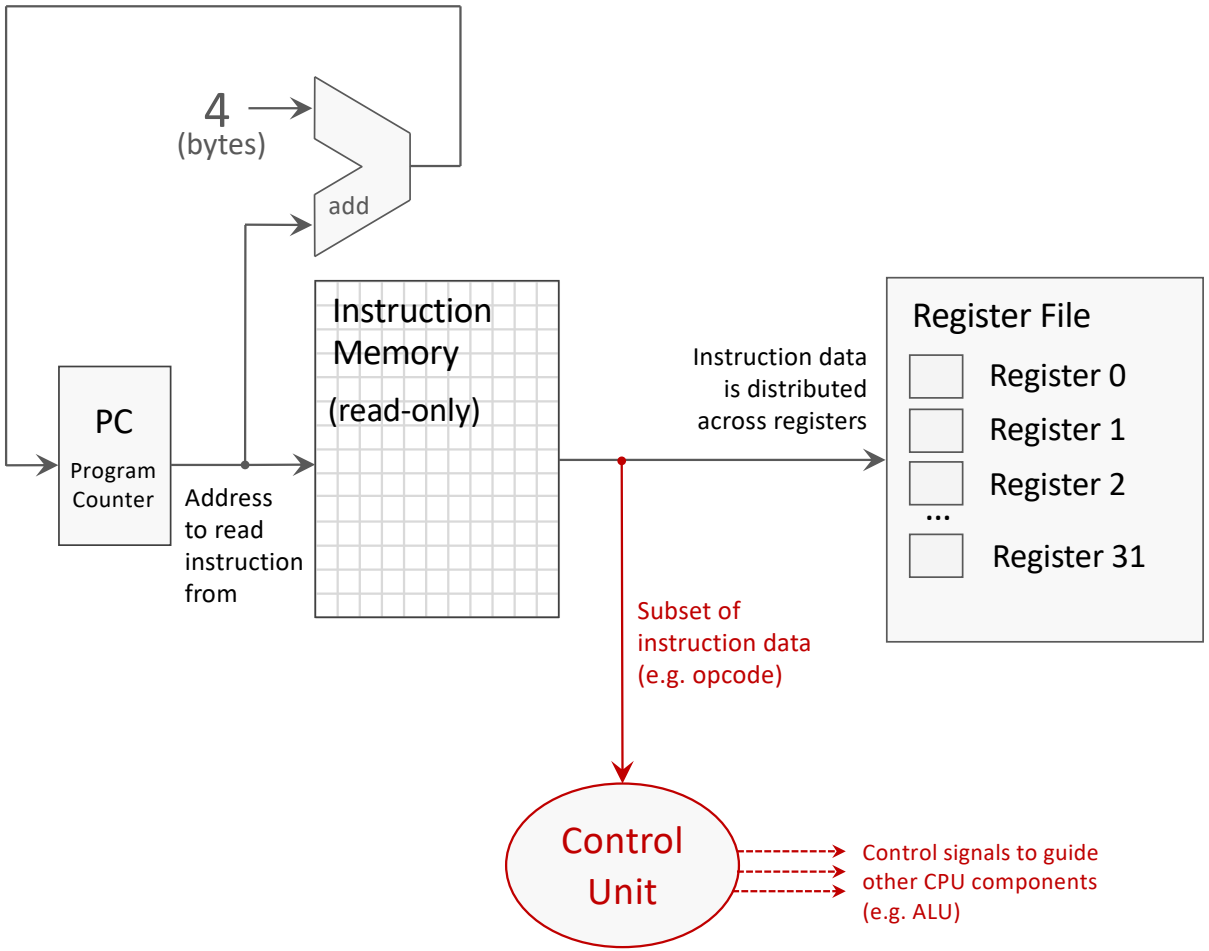
Assumption:

- All program instructions are loaded into a continuous memory block;
- Every instruction takes 4 bytes of memory;
- Consider no branch or jump instructions for now

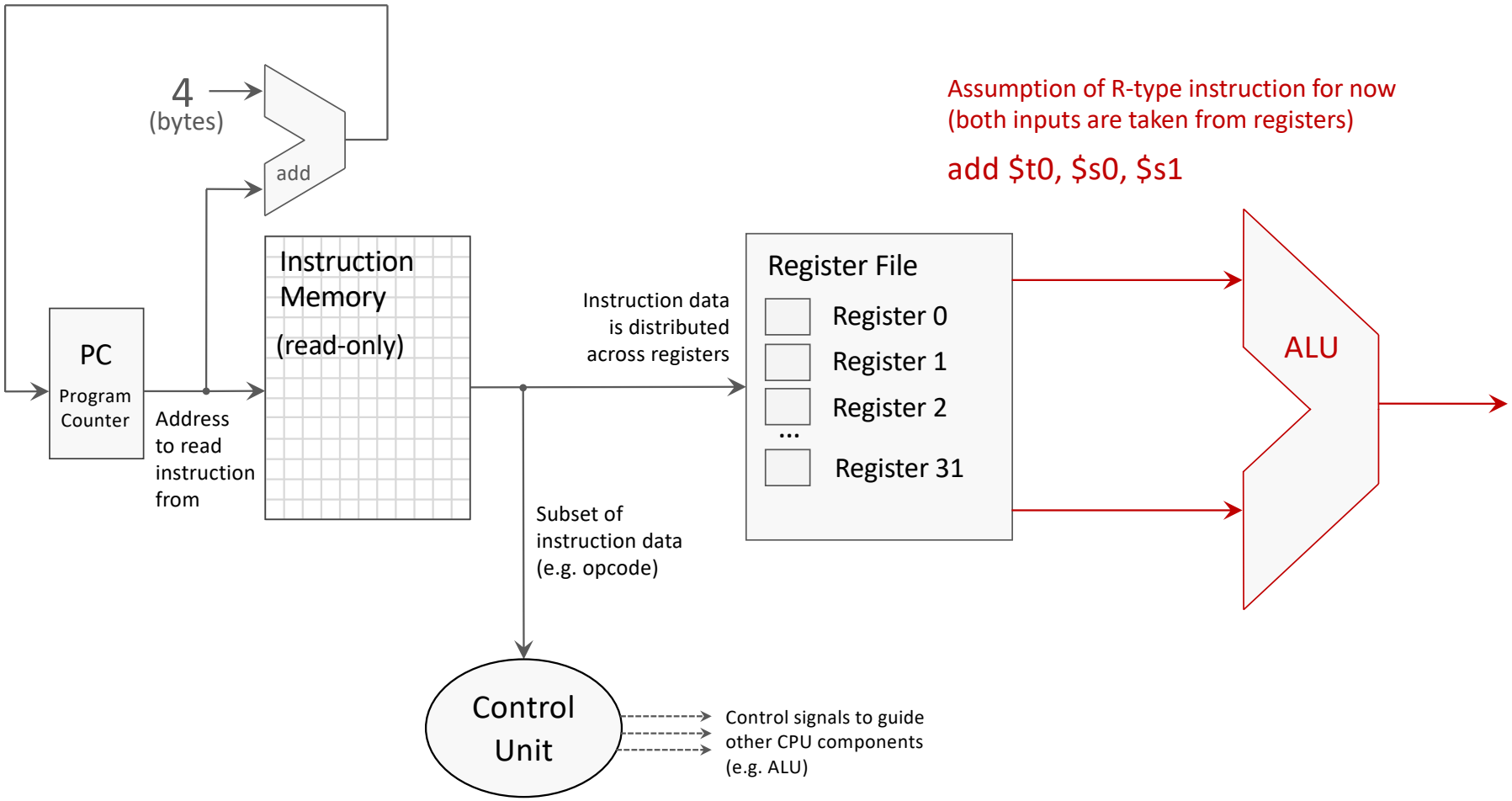
Instruction Decoding over Registers



Data Exchange with Control Unit



Execution of R-type Instructions



Types of MIPS Processor Instructions: R, I, and J

All MIPS instructions are 32 bit long in their binary representation;
The difference is in the number of fields, their meaning, and sizes

R-type

add, sub, sll, srl

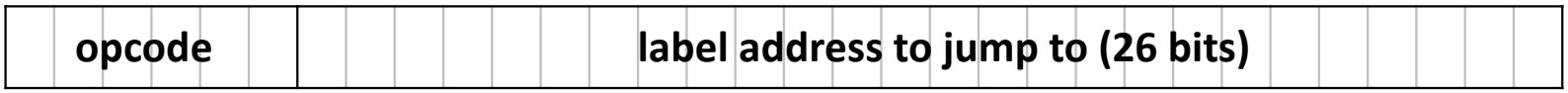
("register", or "regular")

I-type

addi, lw, sw

("immediate")

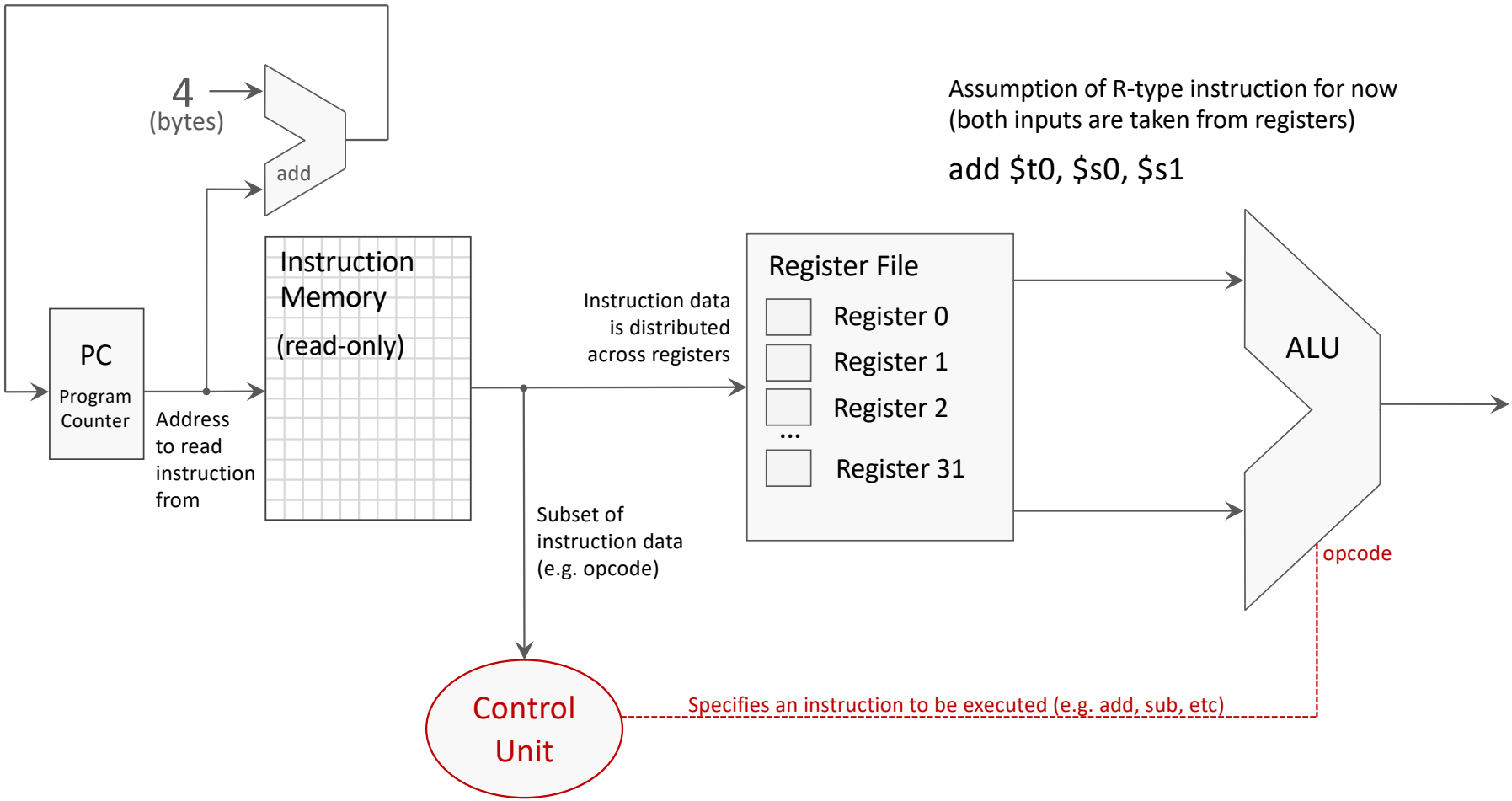
J-type



j, jr, jal

("jump")

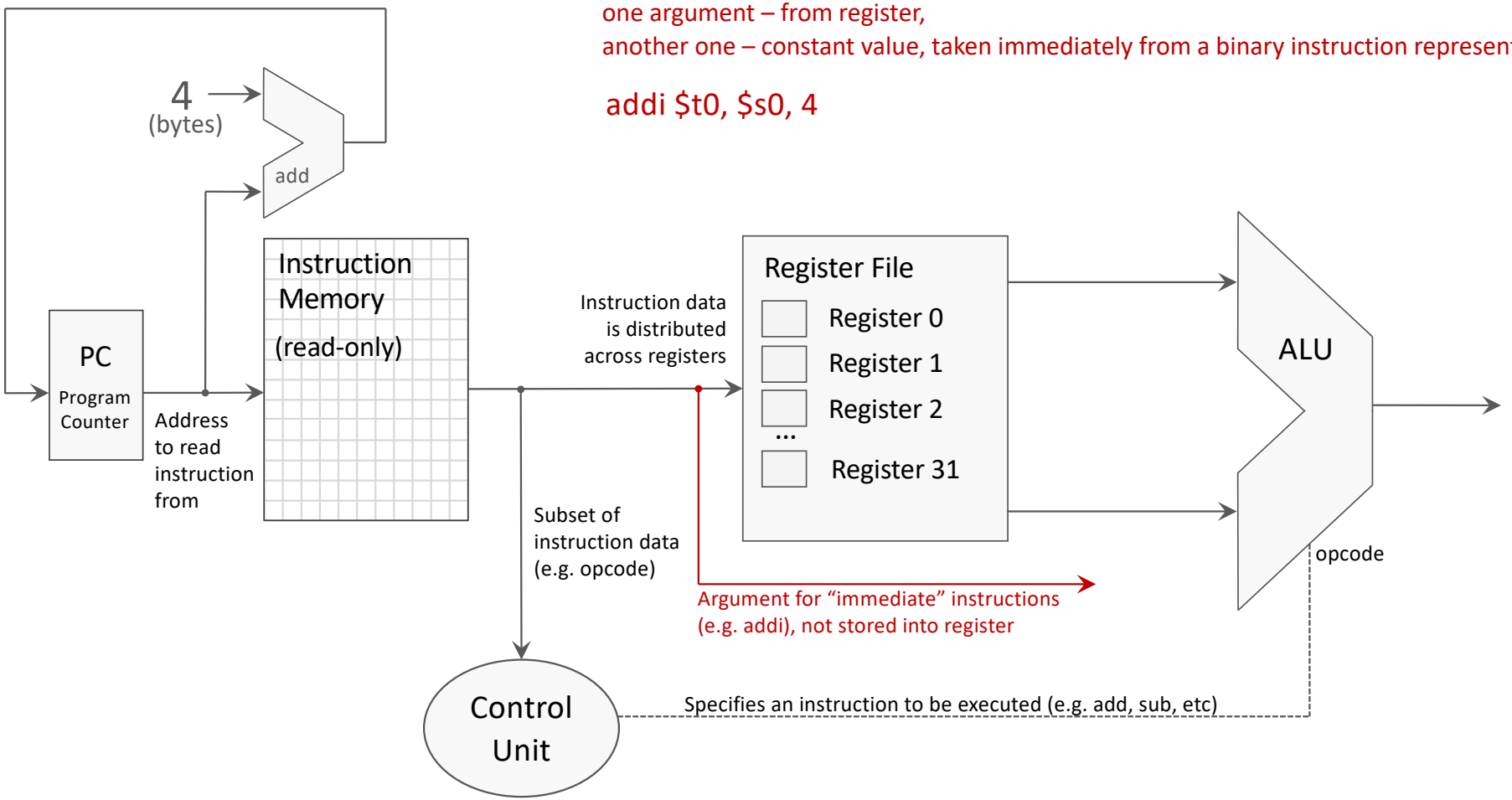
Execution of R-type Instructions



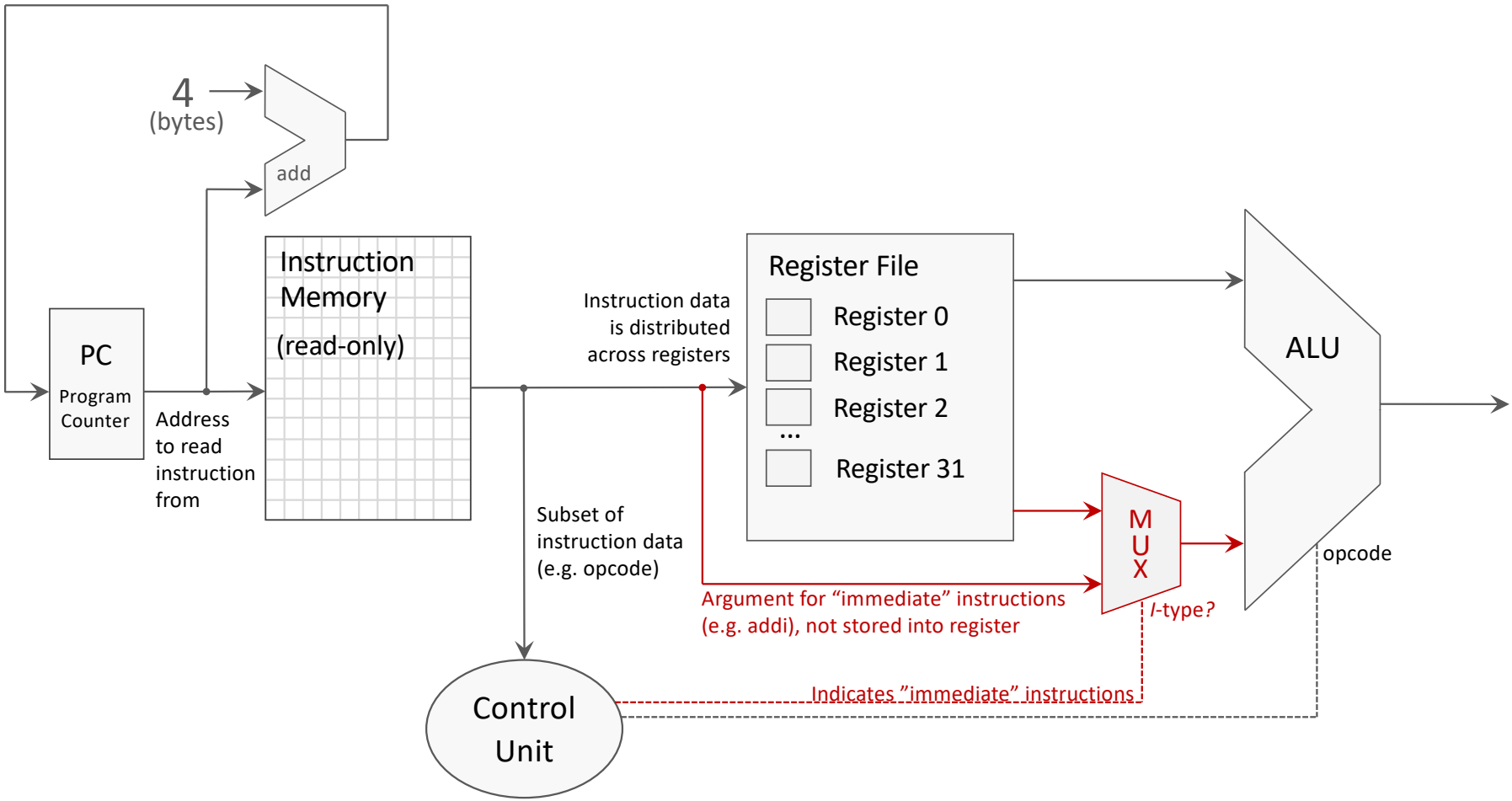
Execution of I-type Instructions

I-type instruction:
one argument – from register,
another one – constant value, taken immediately from a binary instruction representation

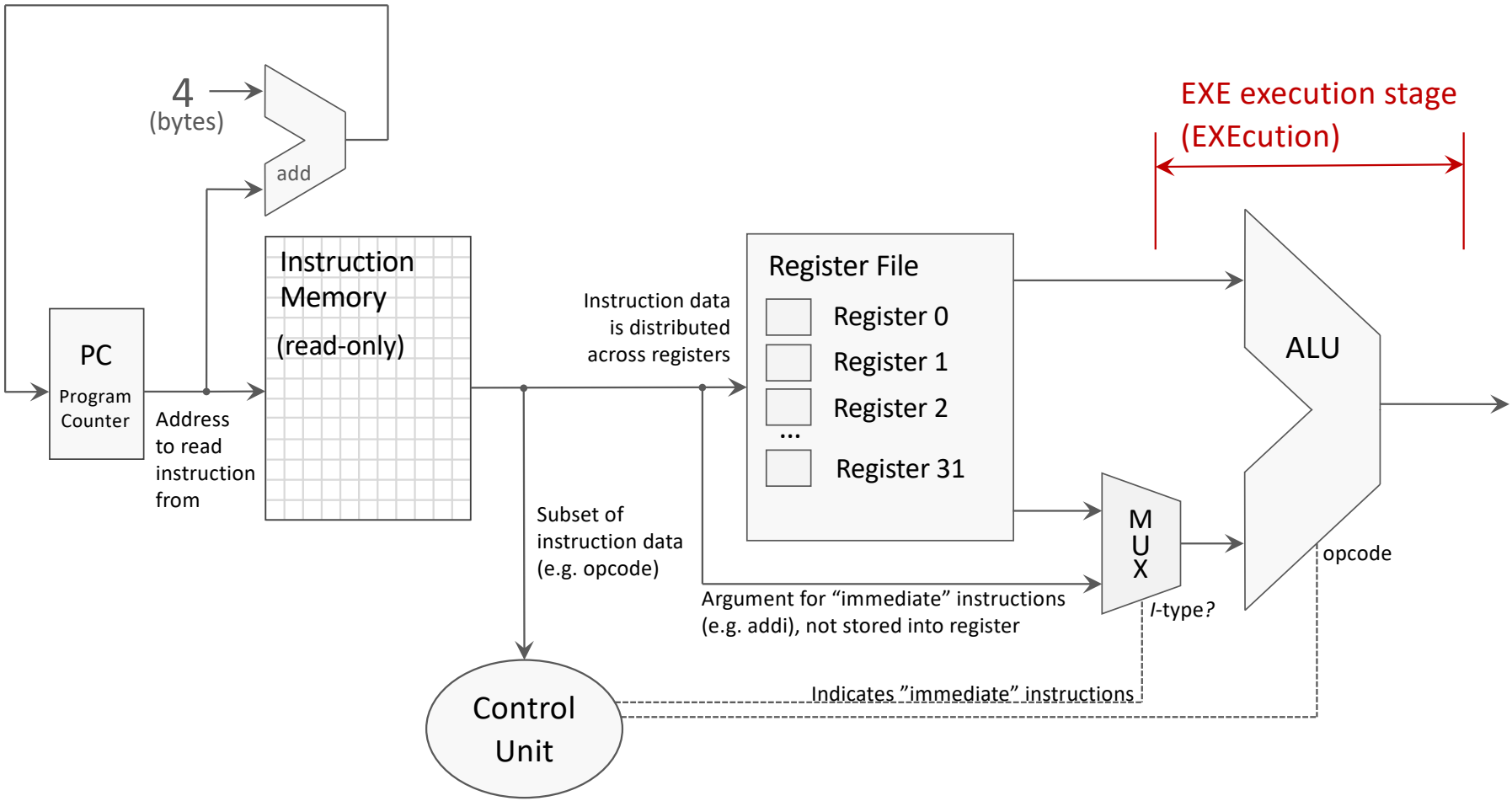
addi $t0, $s0, 4



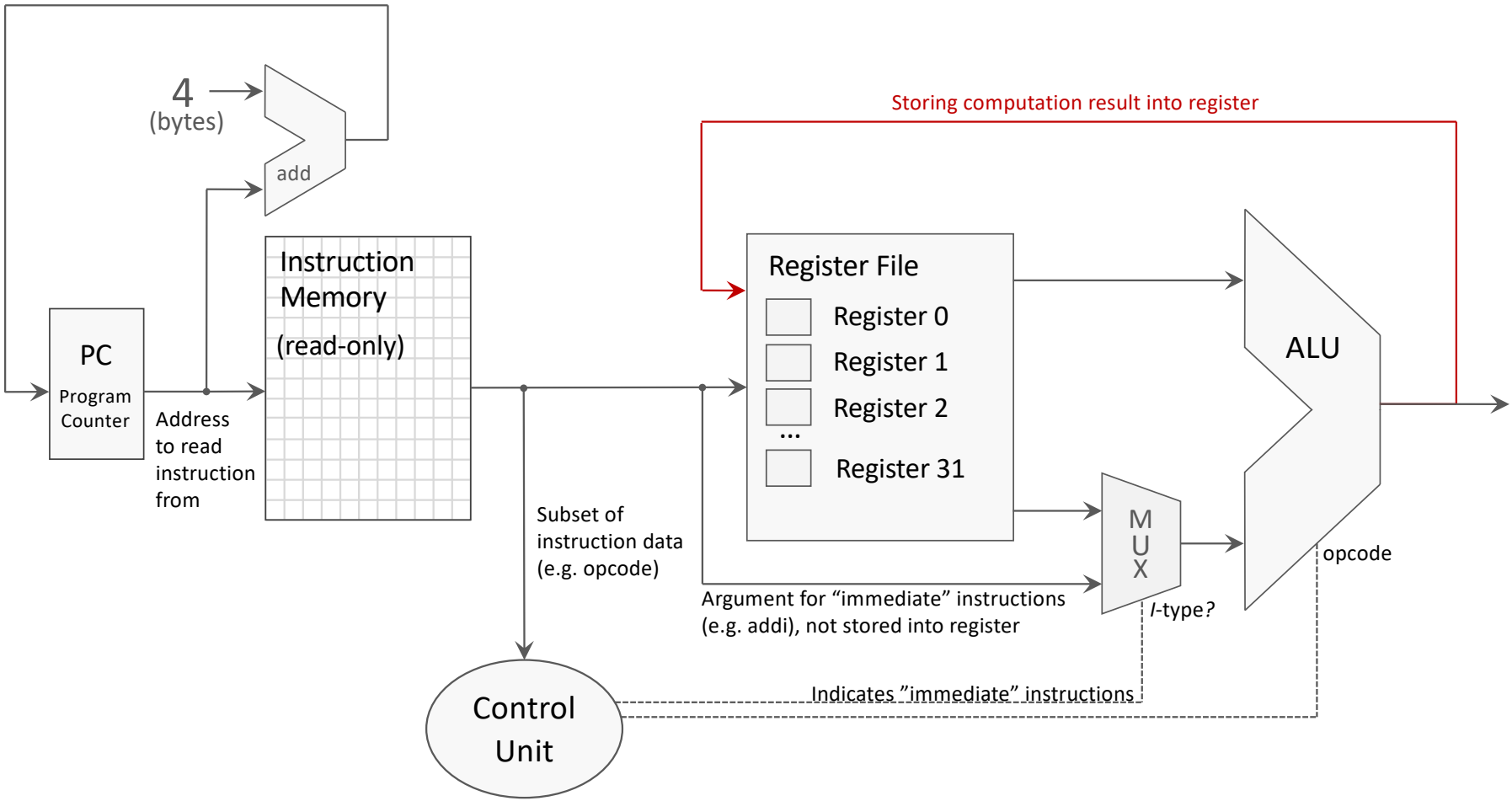
Execution of I-type Instructions



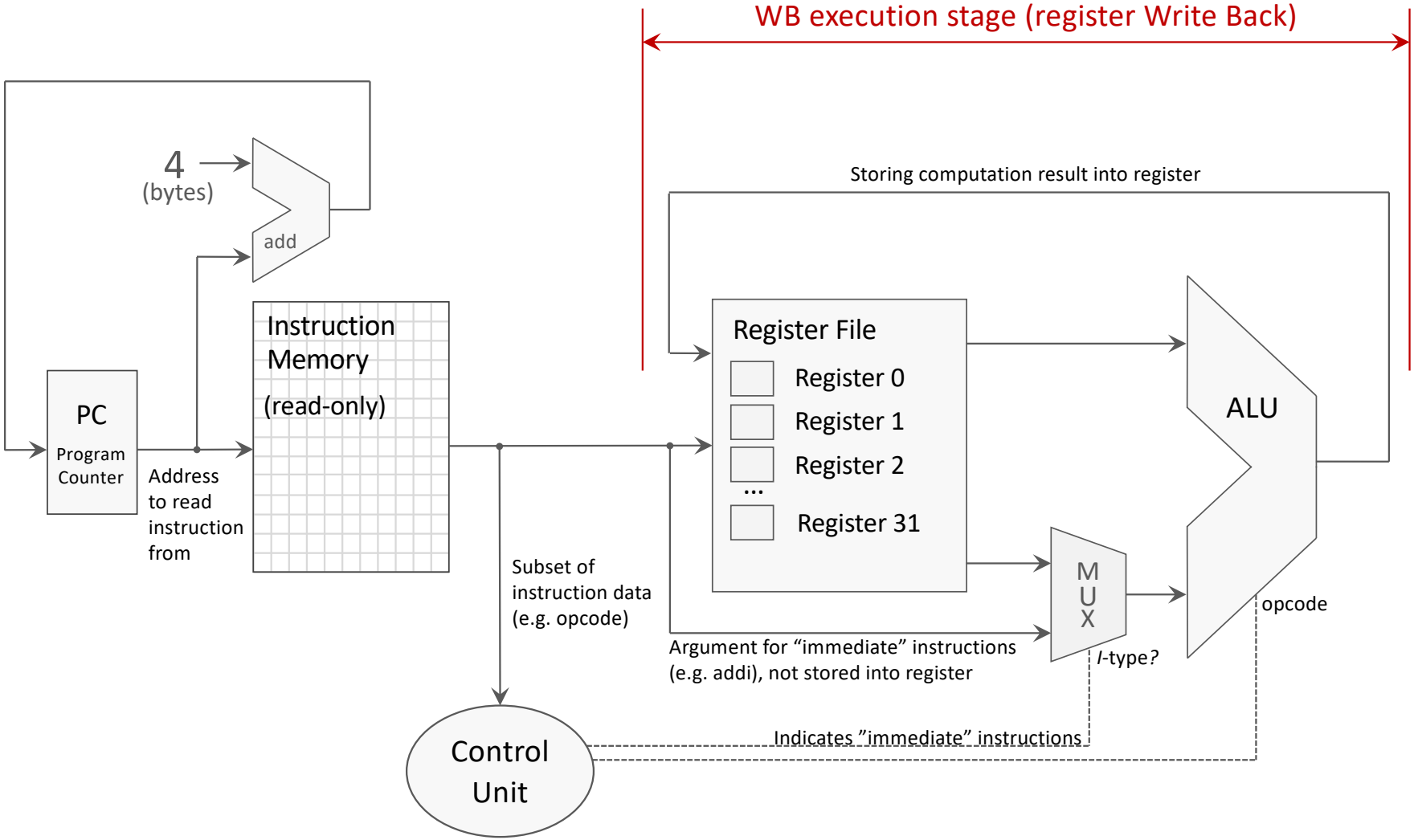
Execution of ALU Instructions



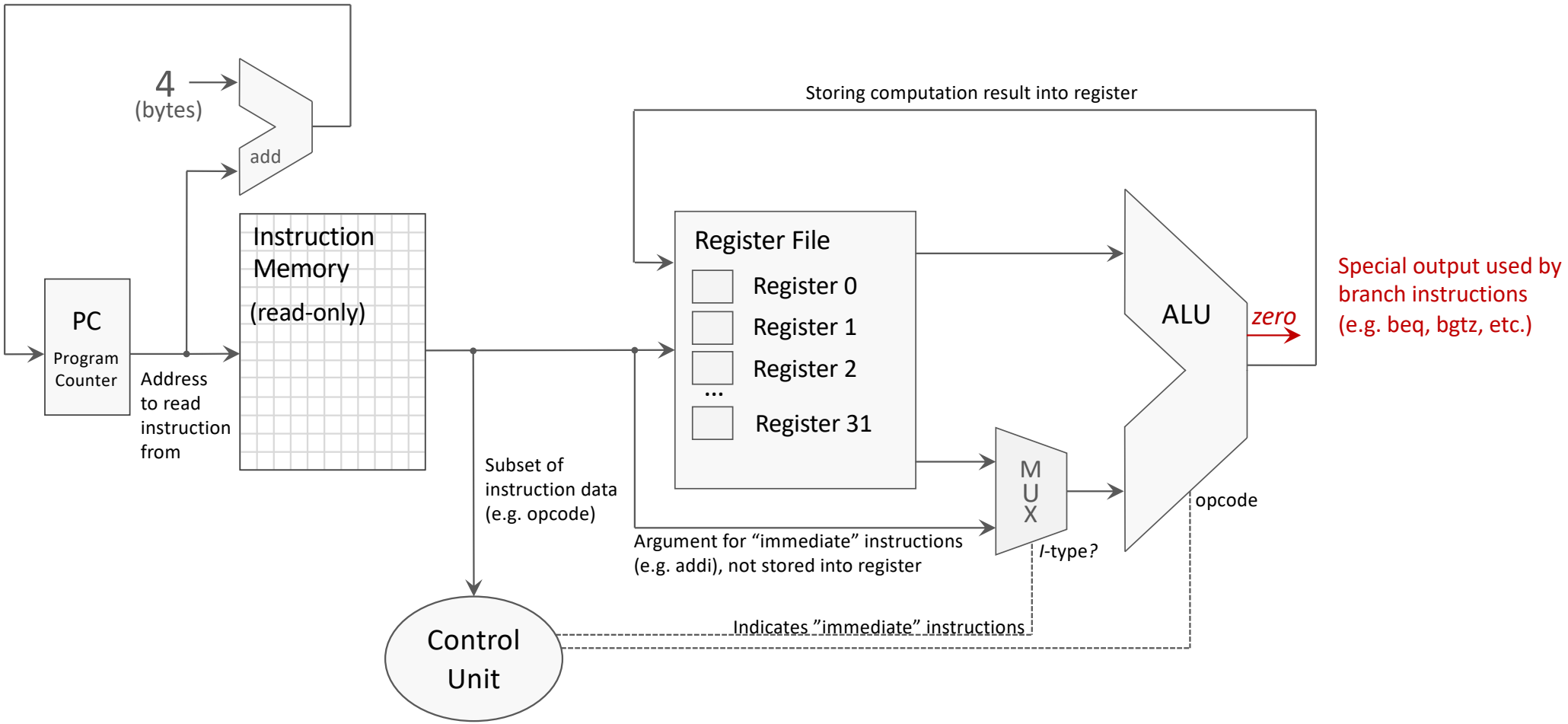
Storing the Result of Executed ALU Instruction



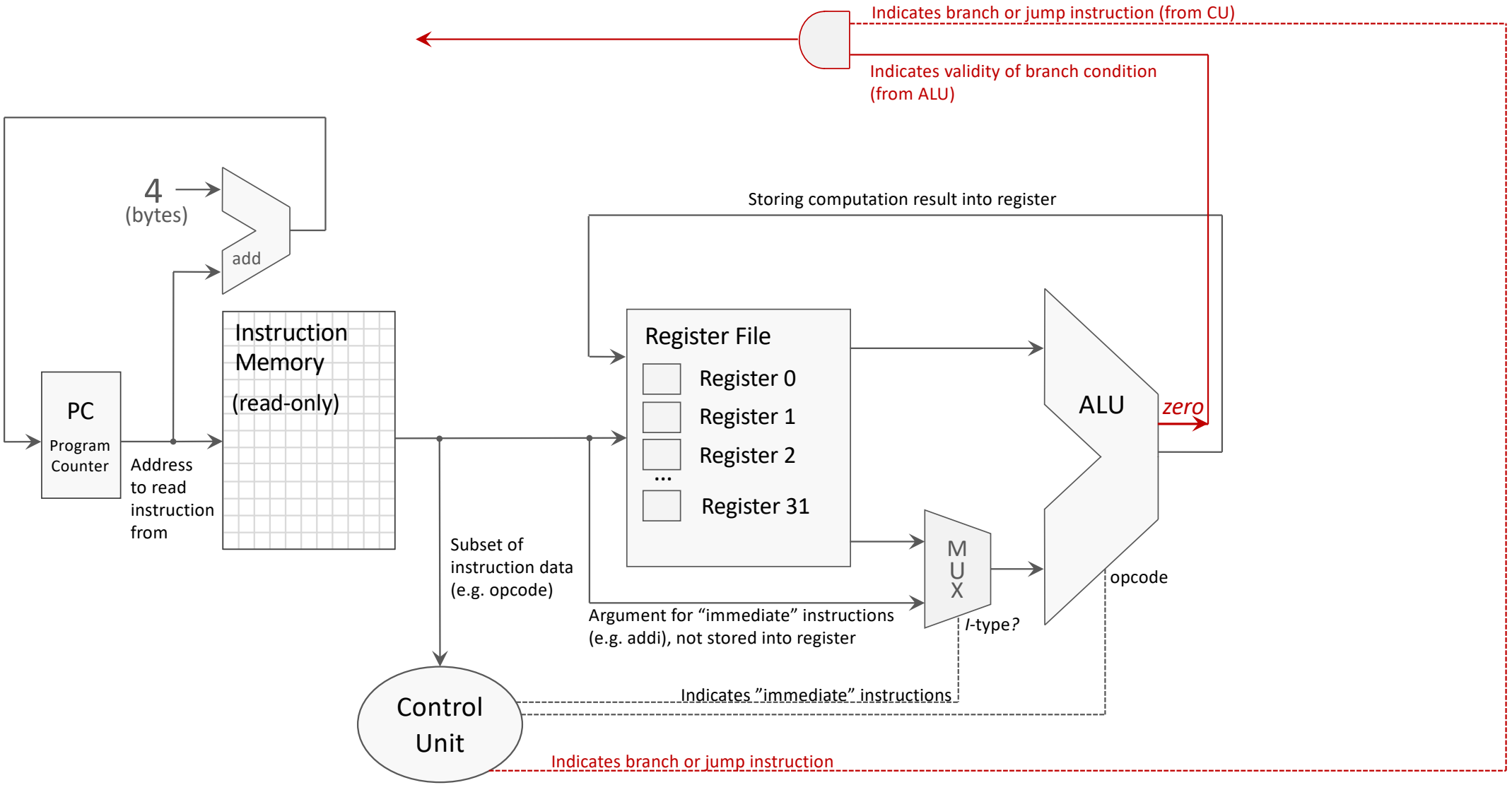
Storing the Result of Executed ALU Instruction



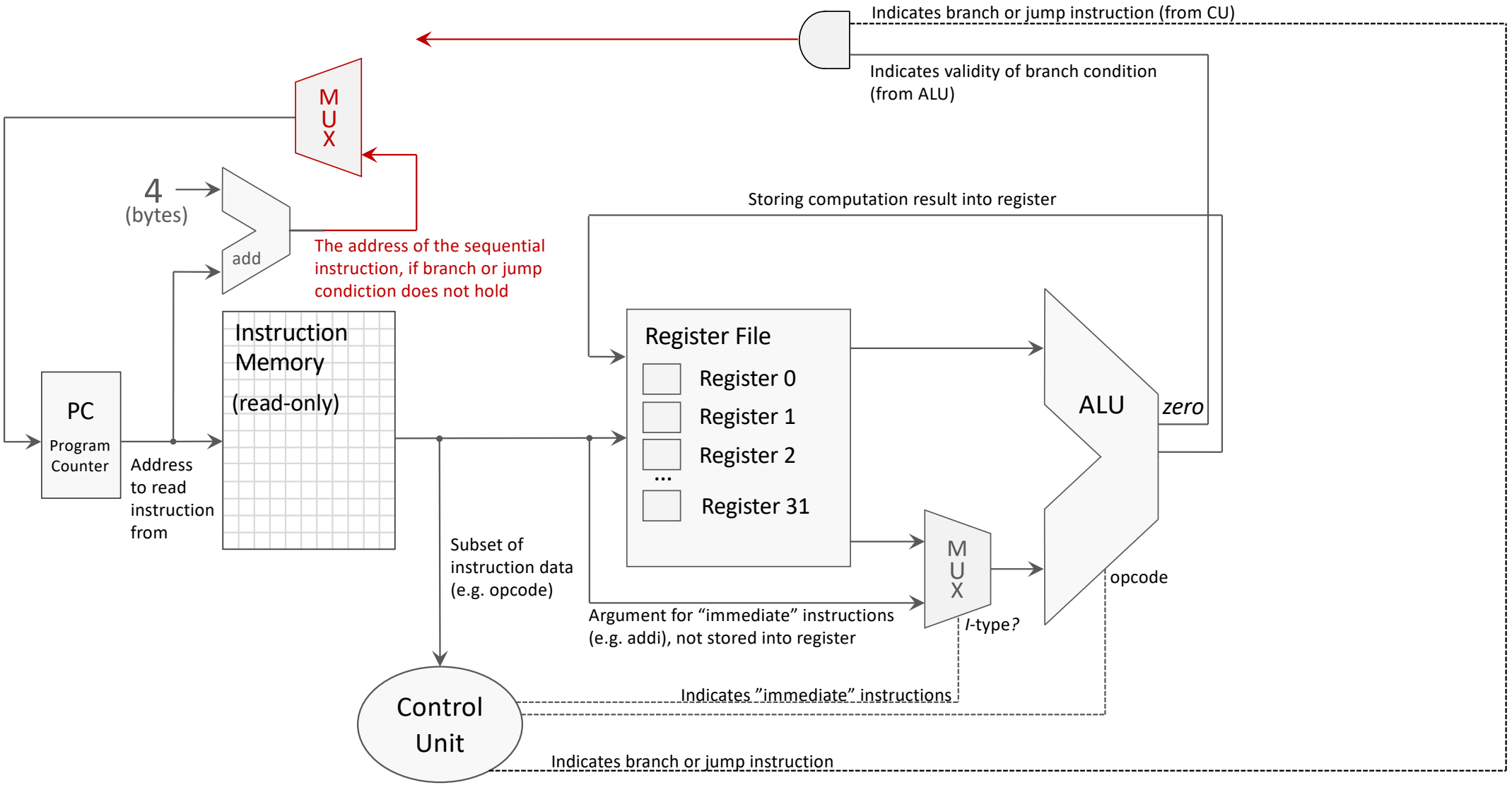
“Zero”: Special-Purpose Output of ALU



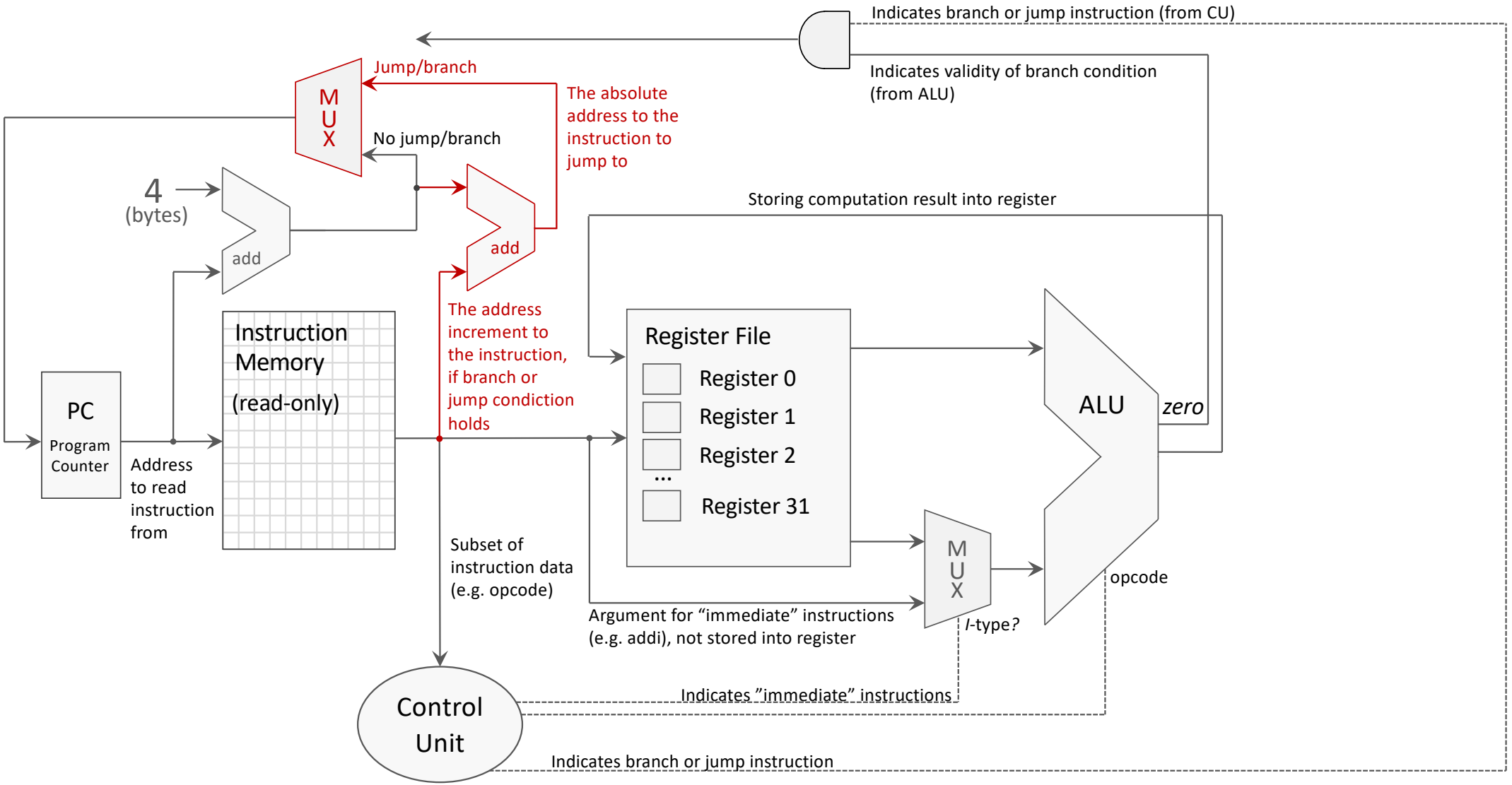
Branch and Jump Instructions



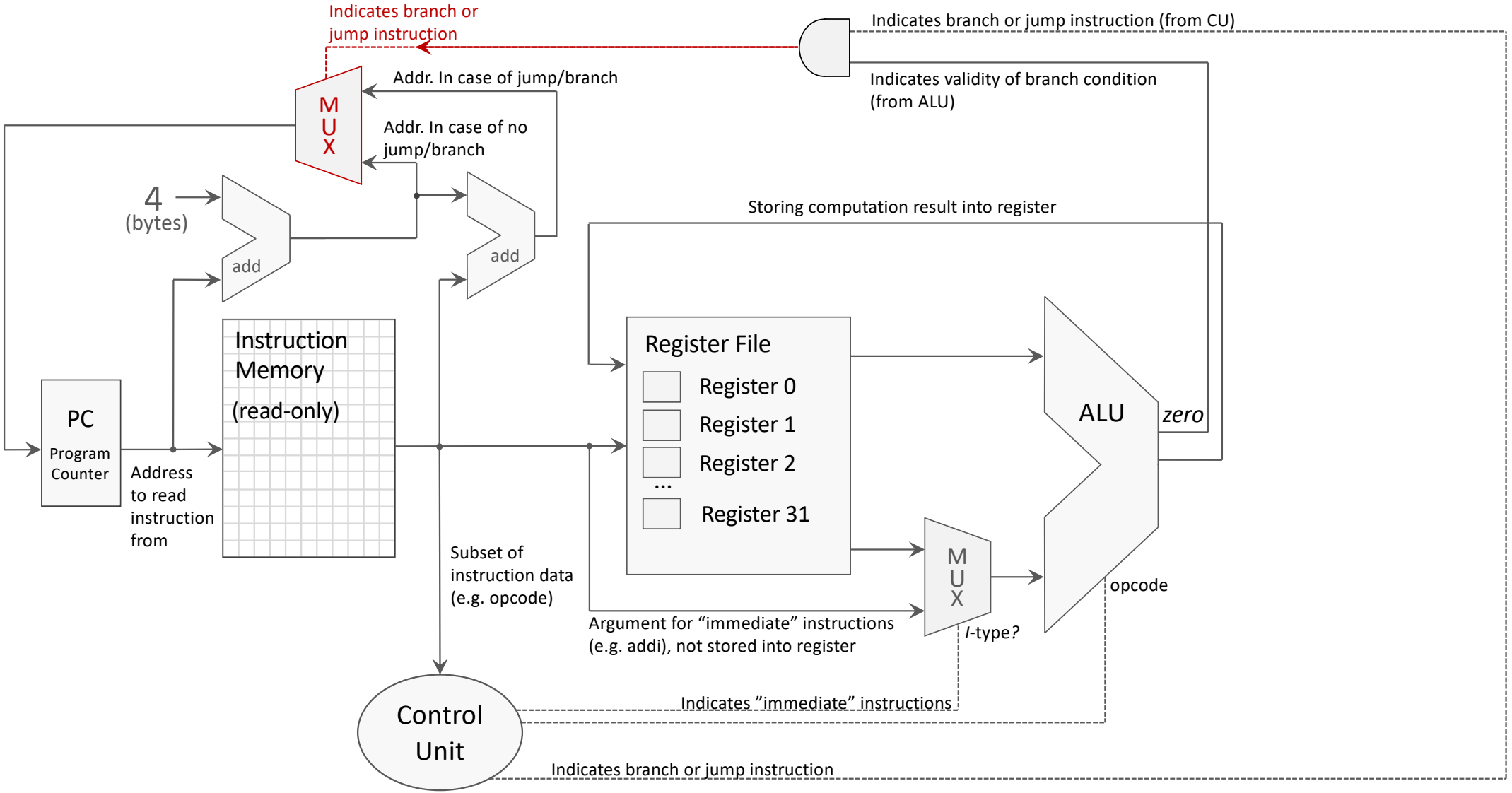
Branch and Jump Instructions



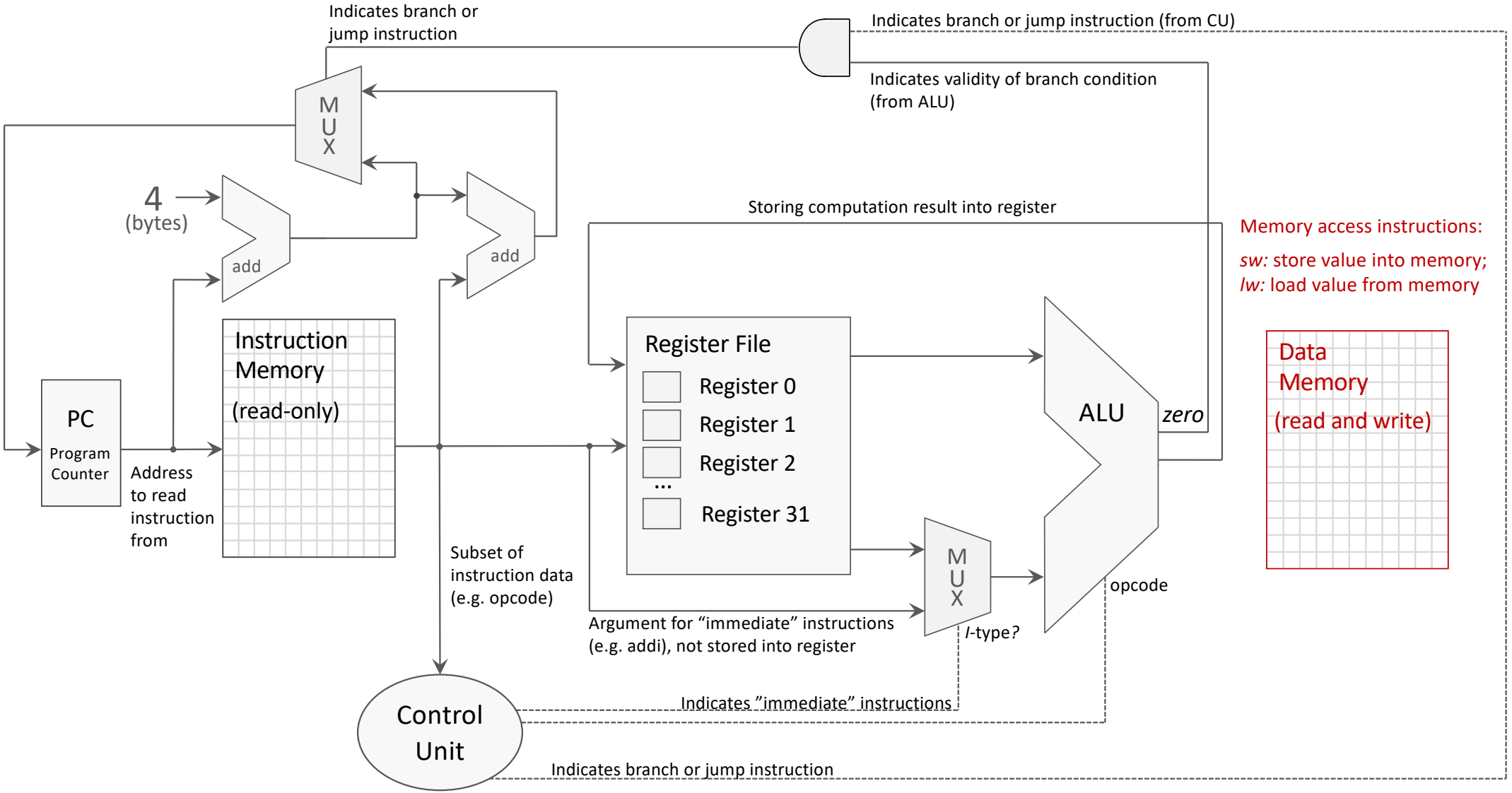
Branch and Jump Instructions



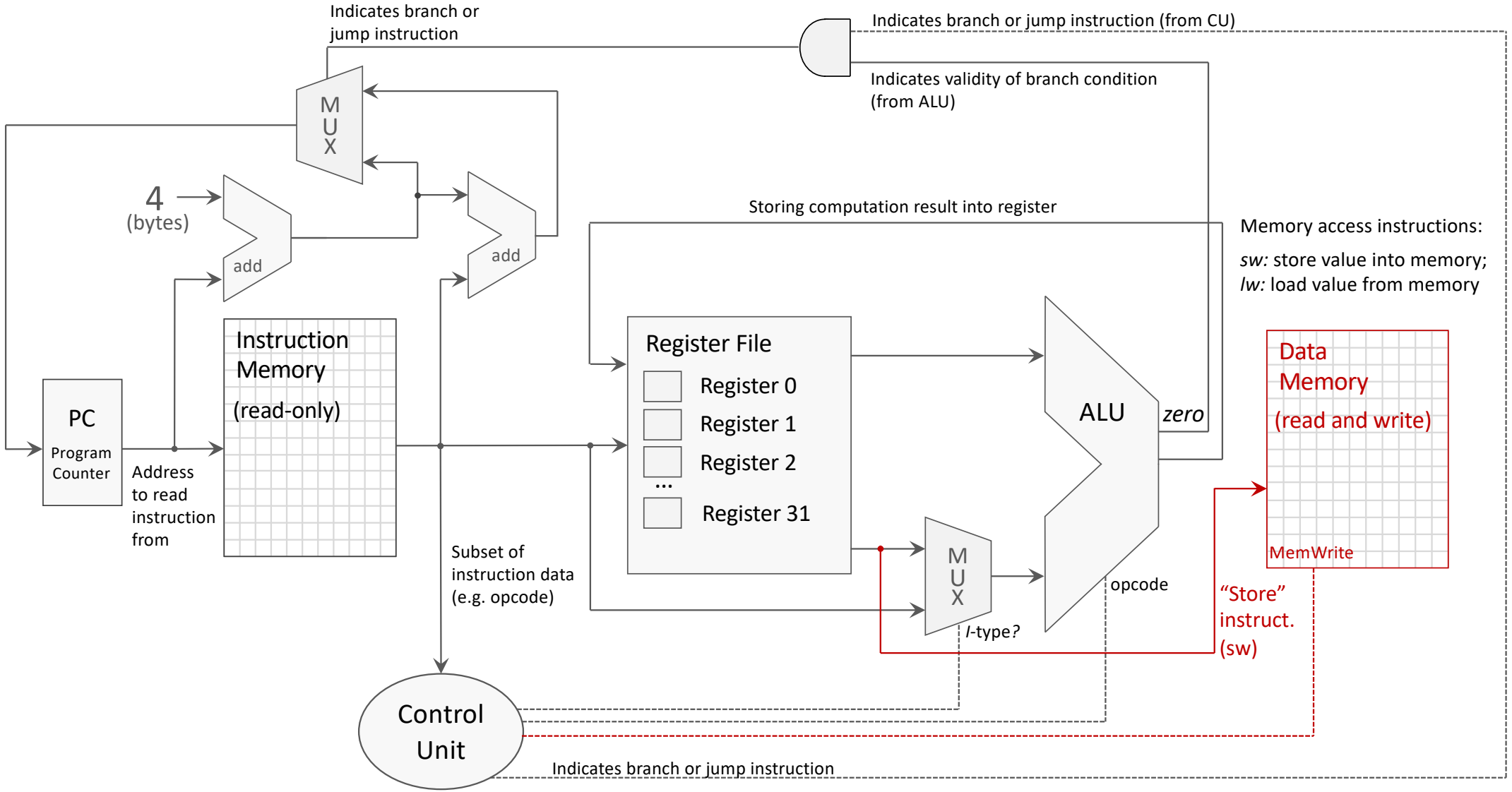
Branch and Jump Instructions



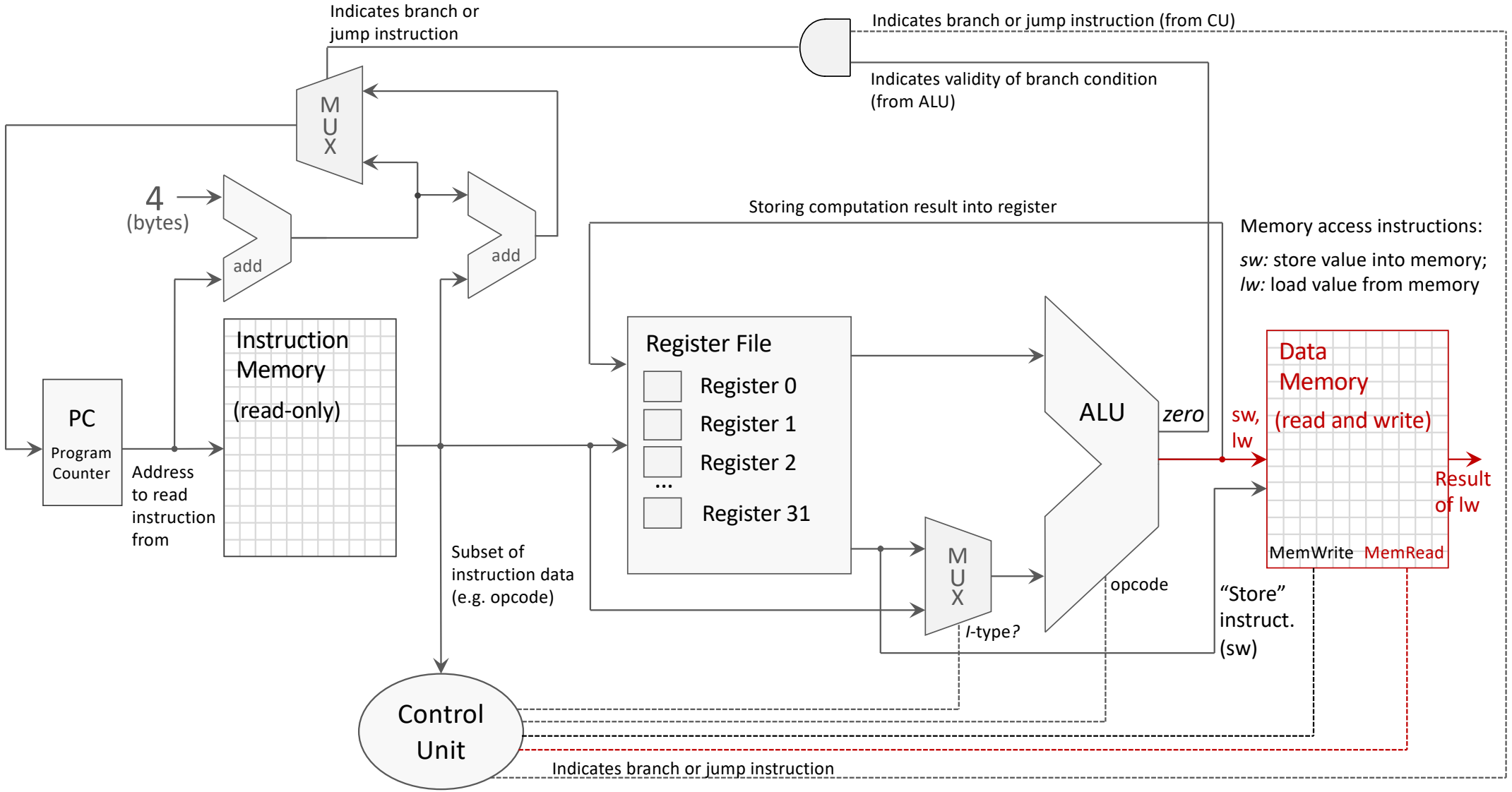
Memory Access Instructions: *lw* and *sw*



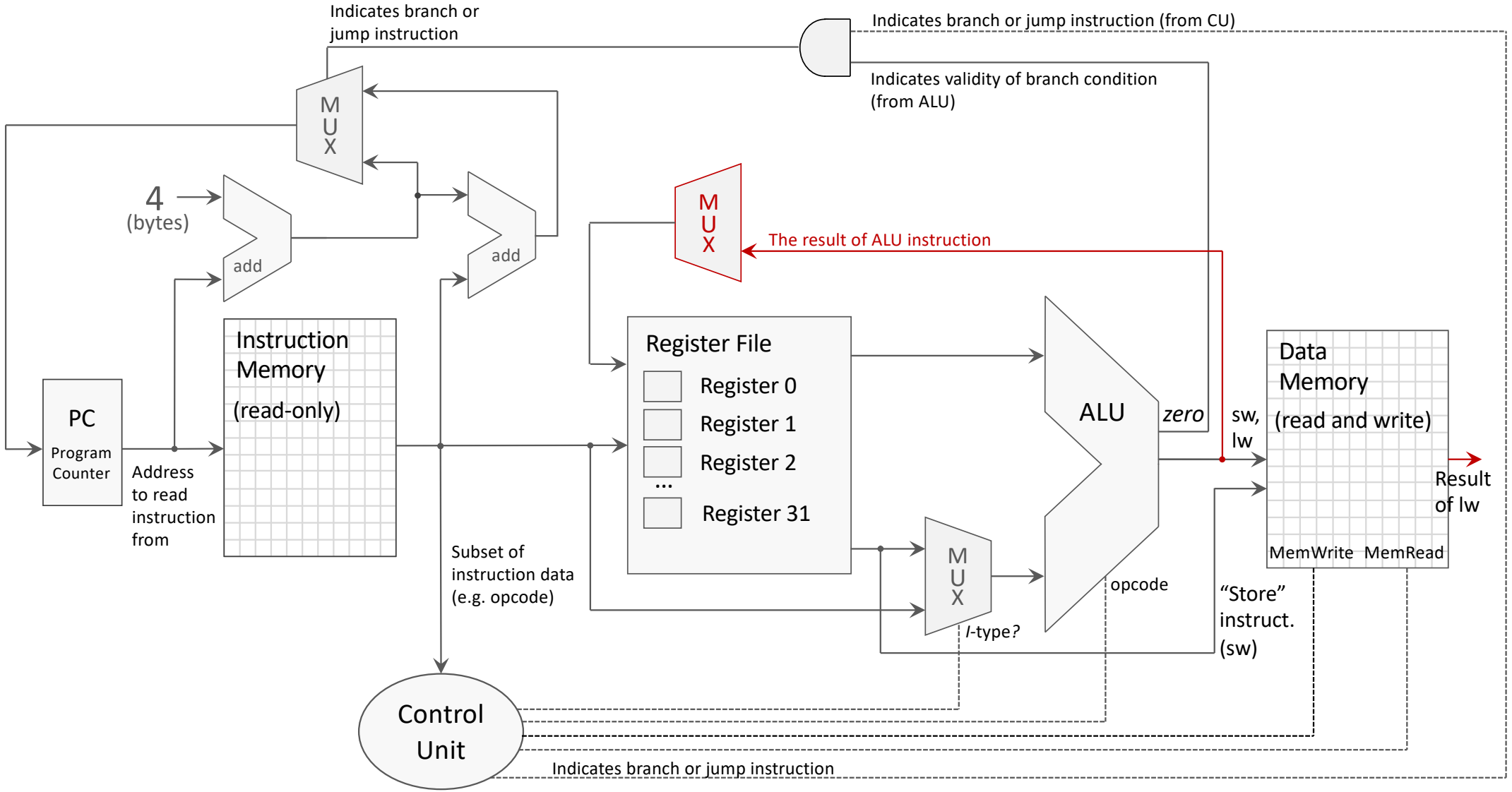
“Store” Memory Instruction



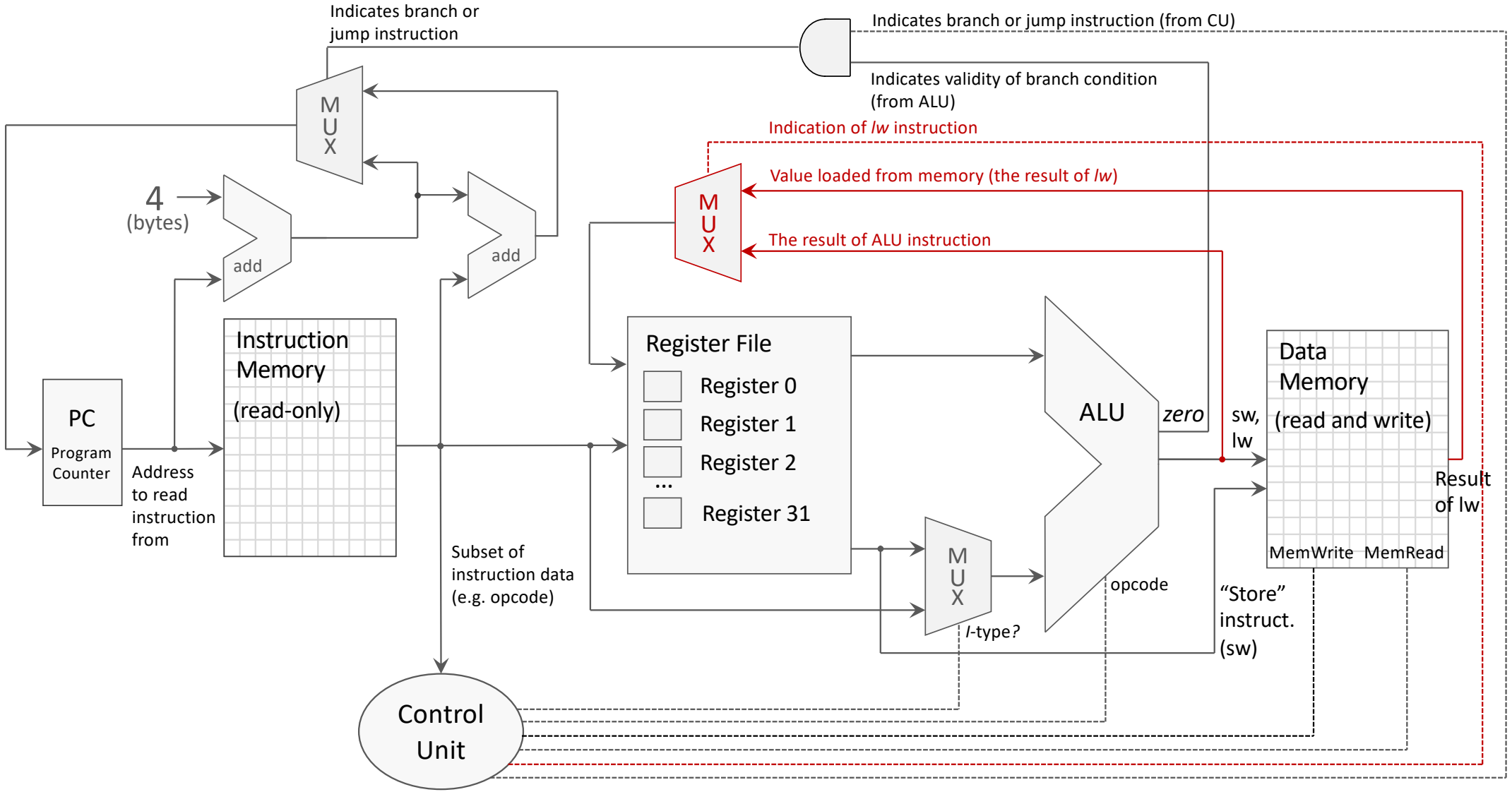
“Load” Memory Instruction



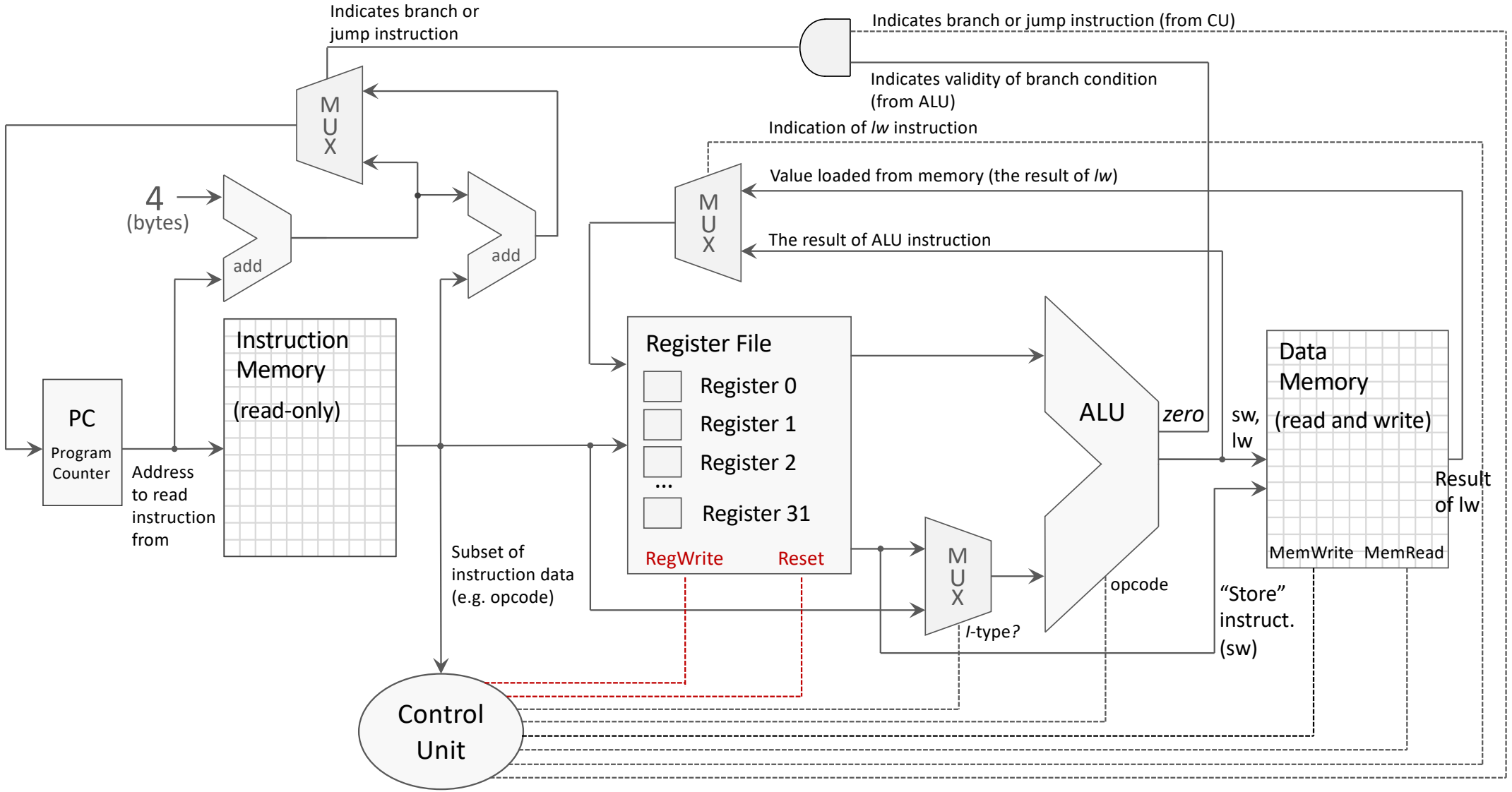
“Load” Memory Instruction



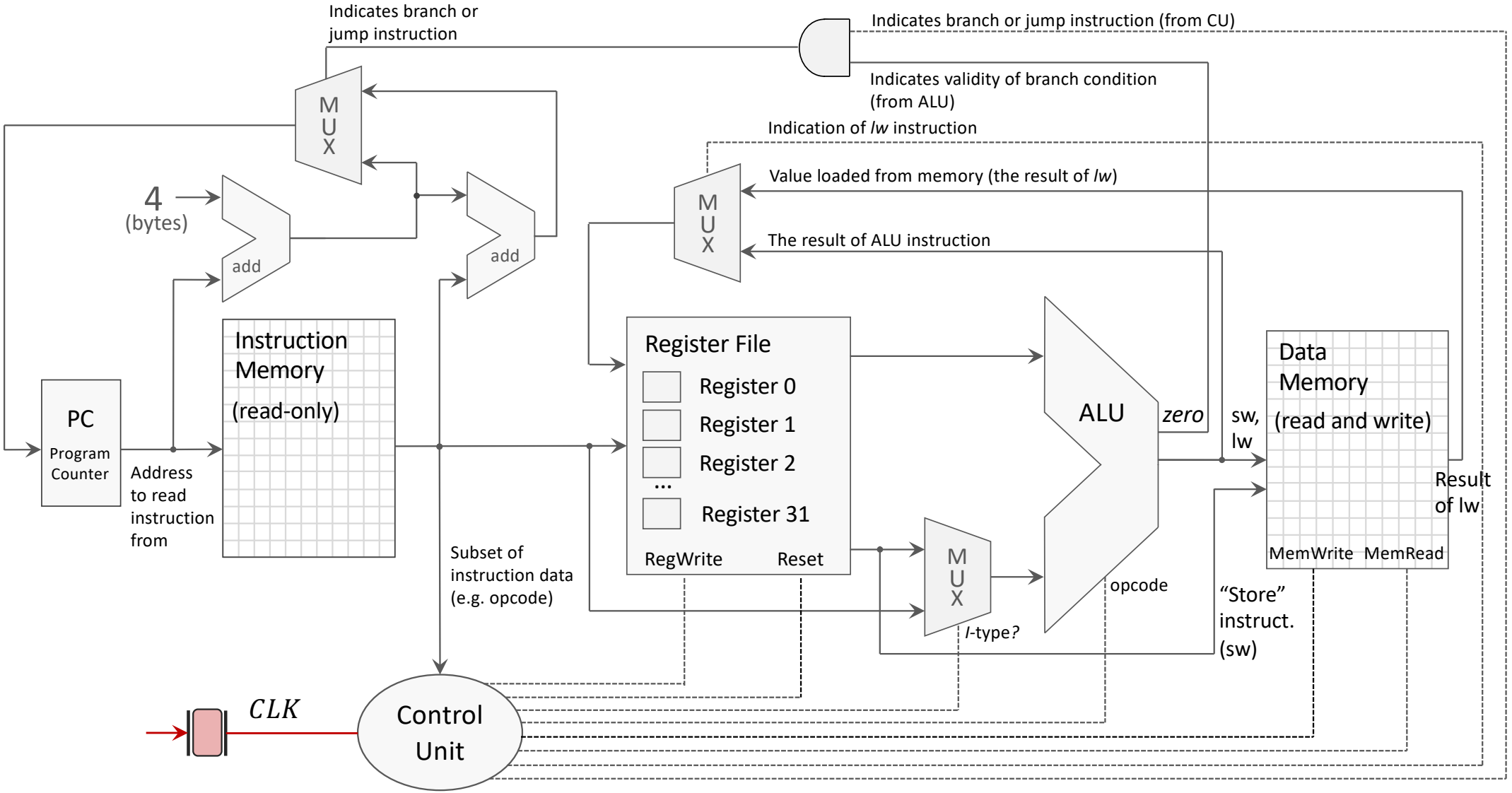
“Load” Memory Instruction



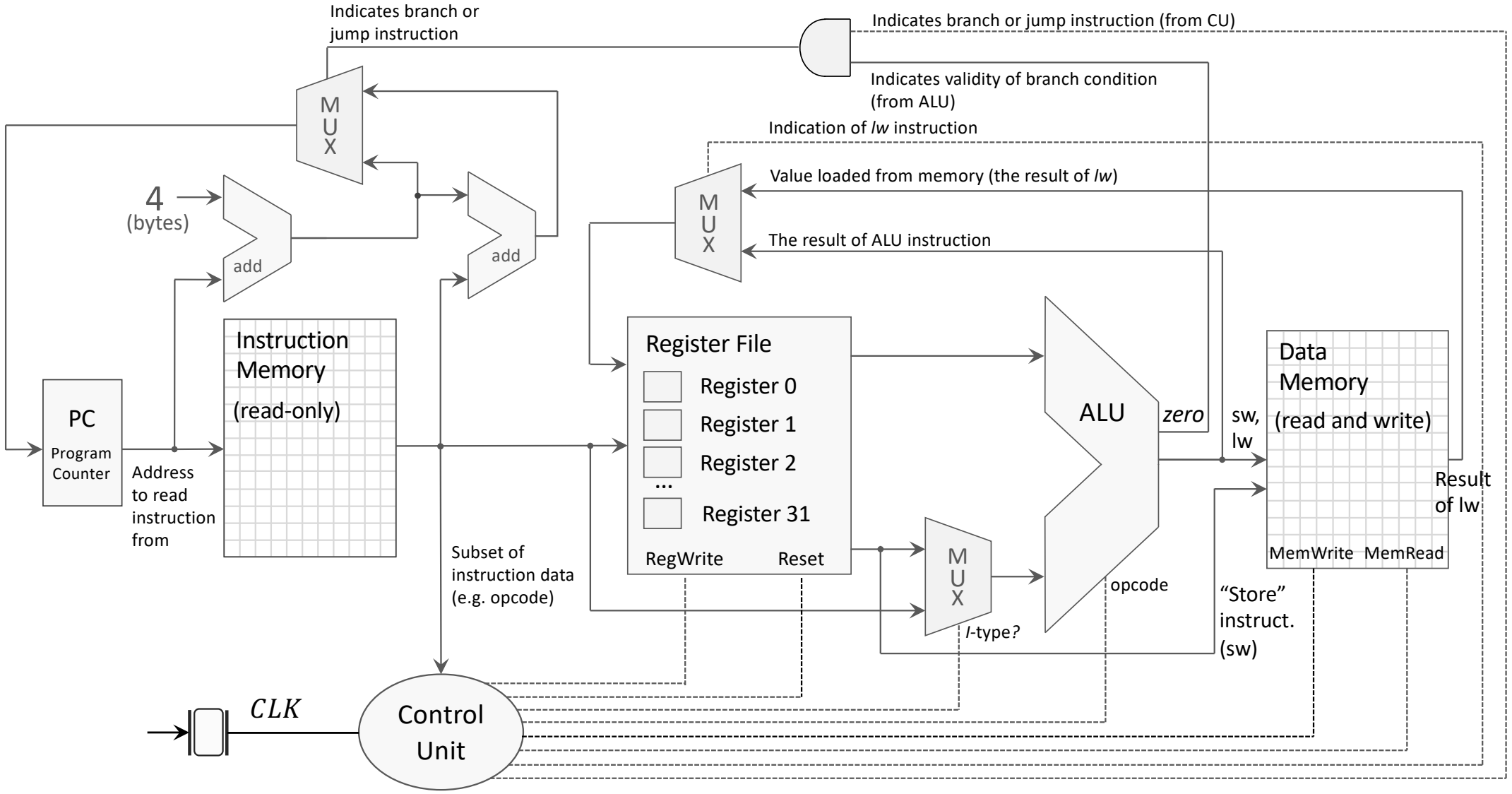
The Control of Register Read/Write and Reset



MIPS Processor: Clocking and Synchronization



MIPS Processor: The Basic Implementation Scheme (Single Clock Cycle)



MIPS Processor: Mapping of Digital Subcircuits to Execution Stages: IF, ID, EXE, MA, WB