Computer Systems II

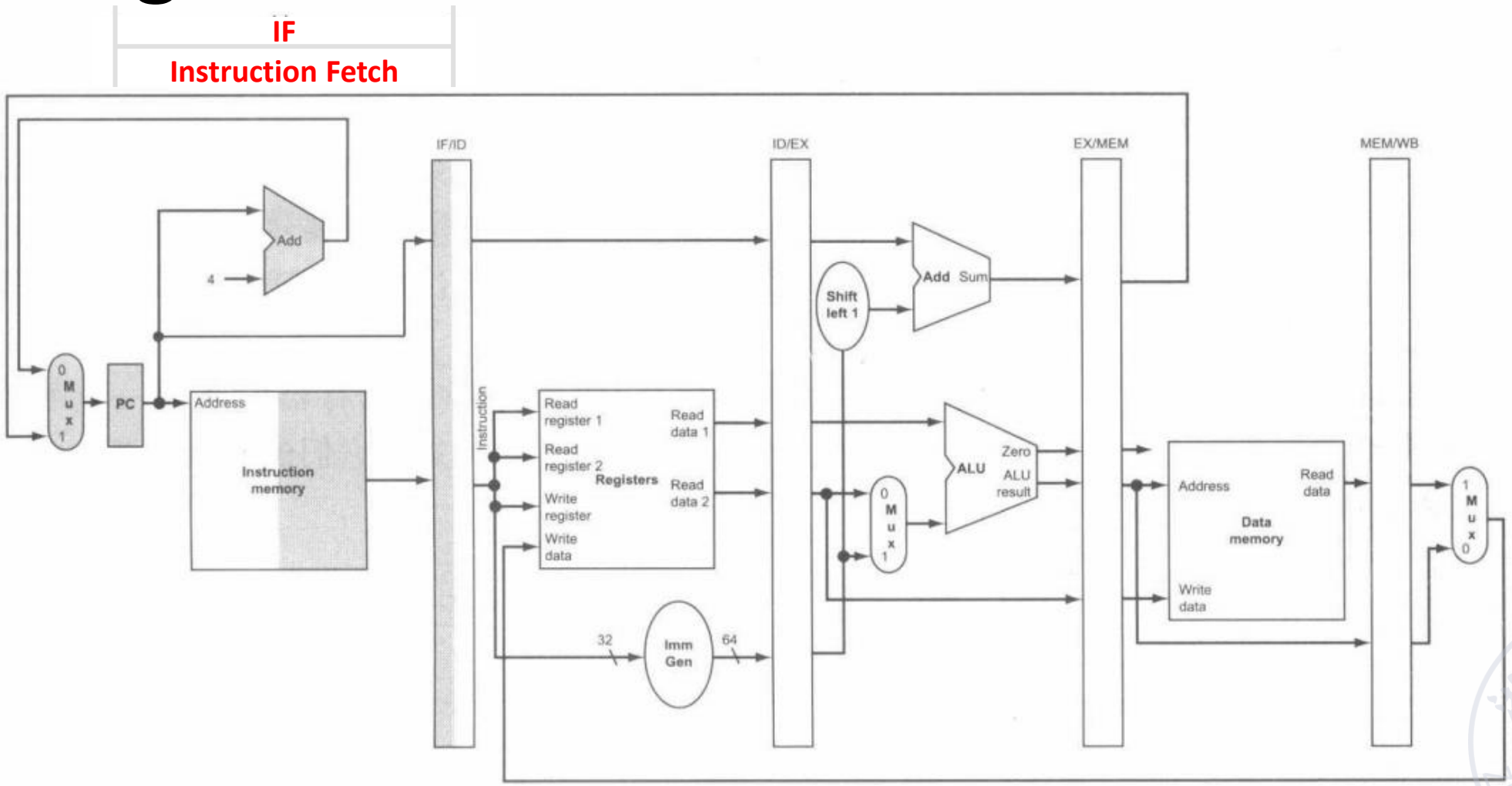
Li Lu
Room 605, CaoGuangbiao Building
li.lu@zju.edu.cn
https://person.zju.edu.cn/lynnluli

Understand How Pipelined Datapath Work

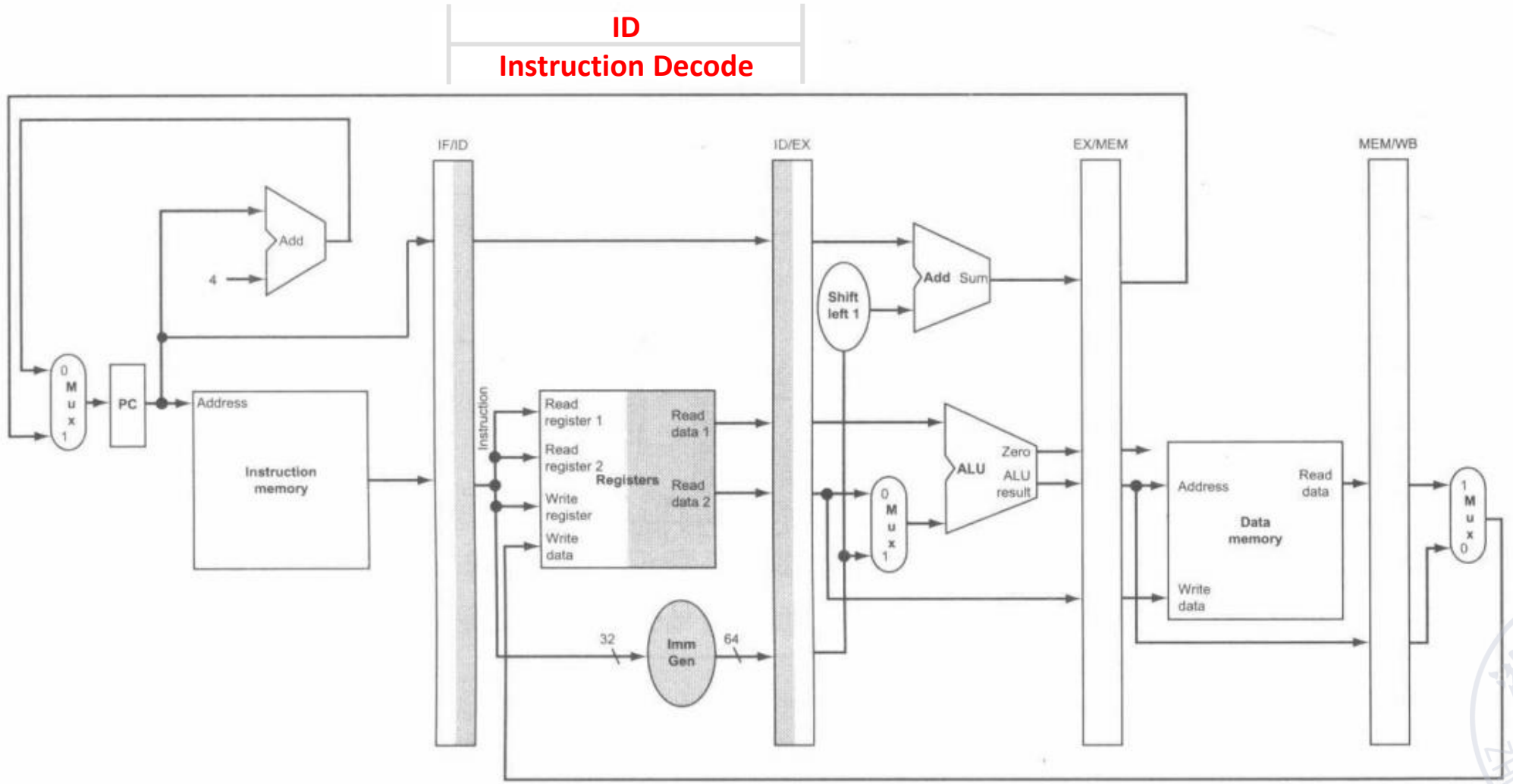
Through Load Instruction and Store Instruction

Load Instruction

IF Stage of Load Instruction

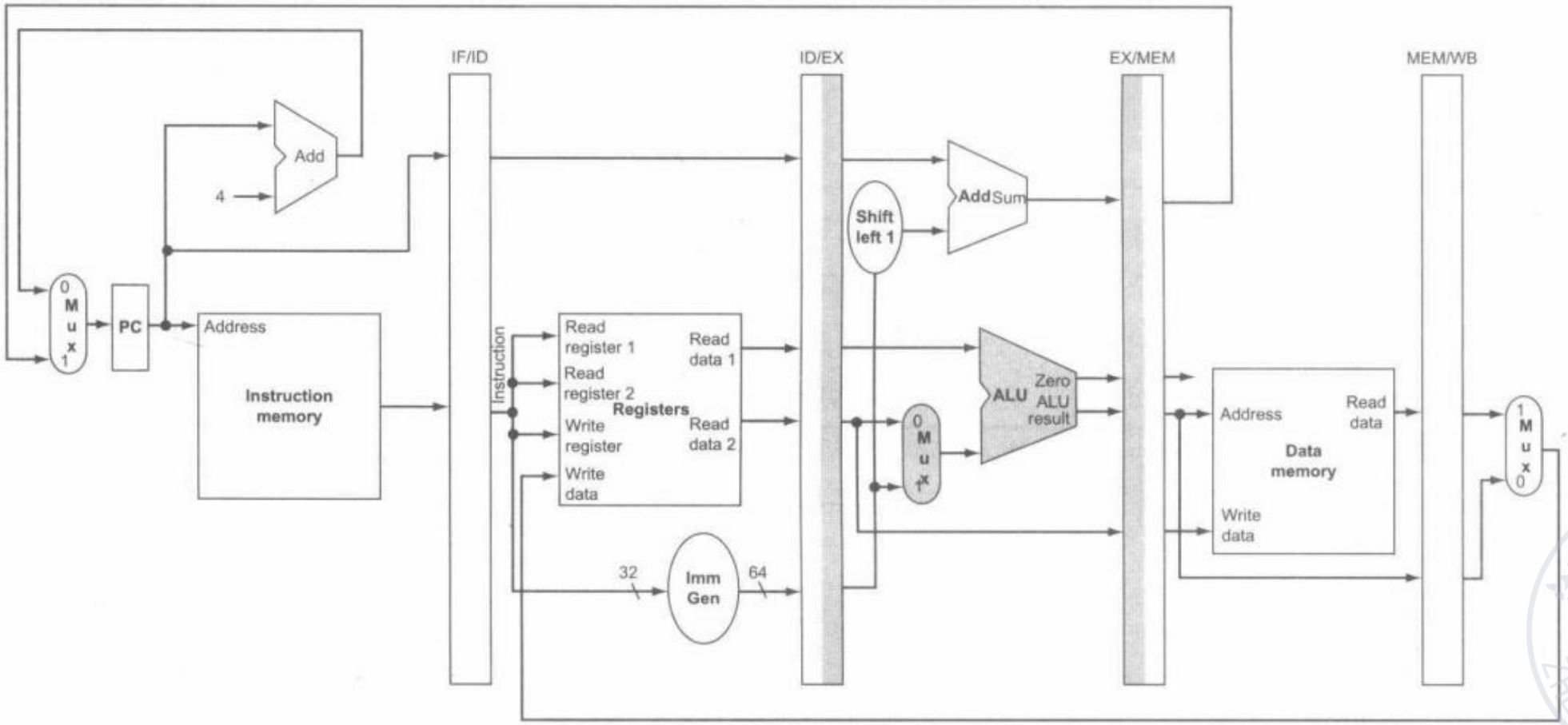


ID Stage of Load Instruction



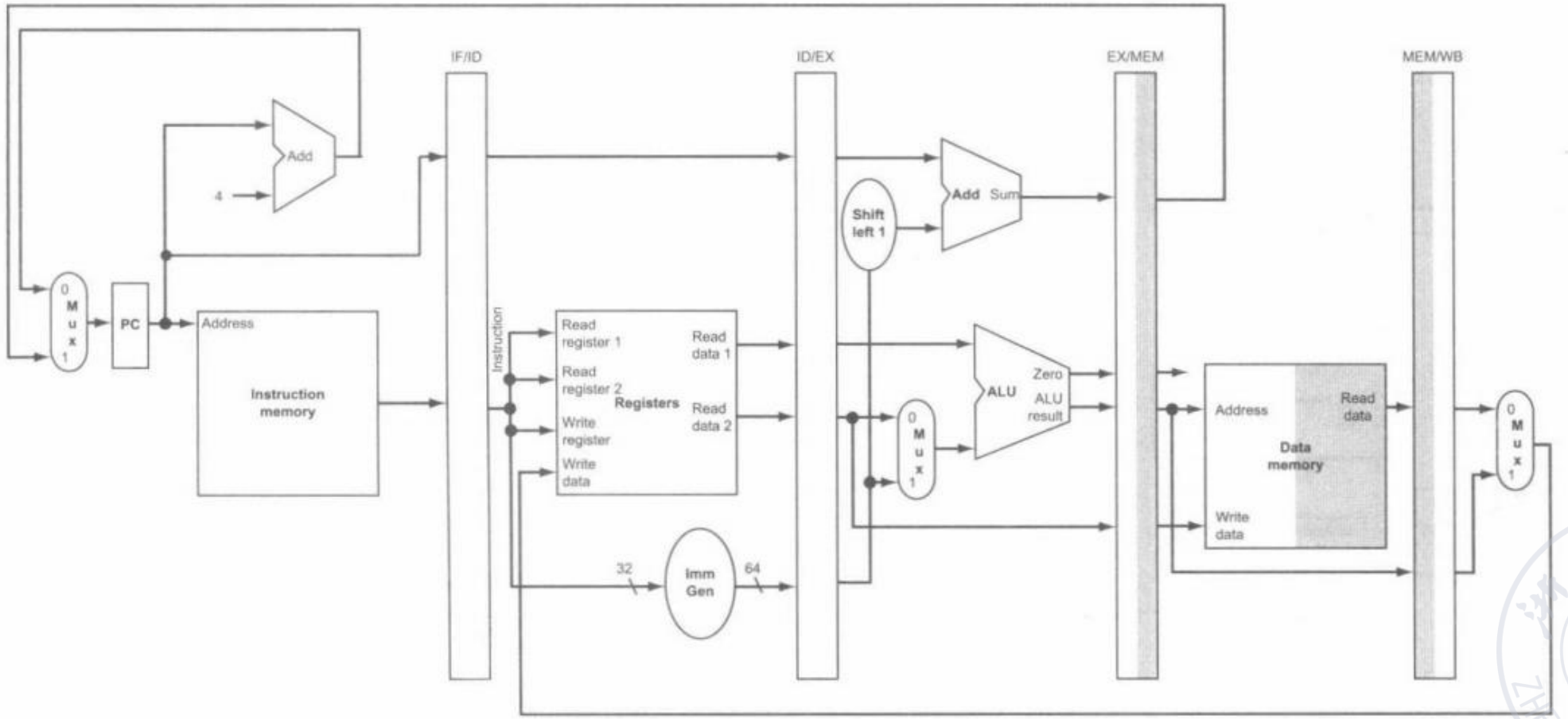
EX Stage of Load Instruction

EX
Execution

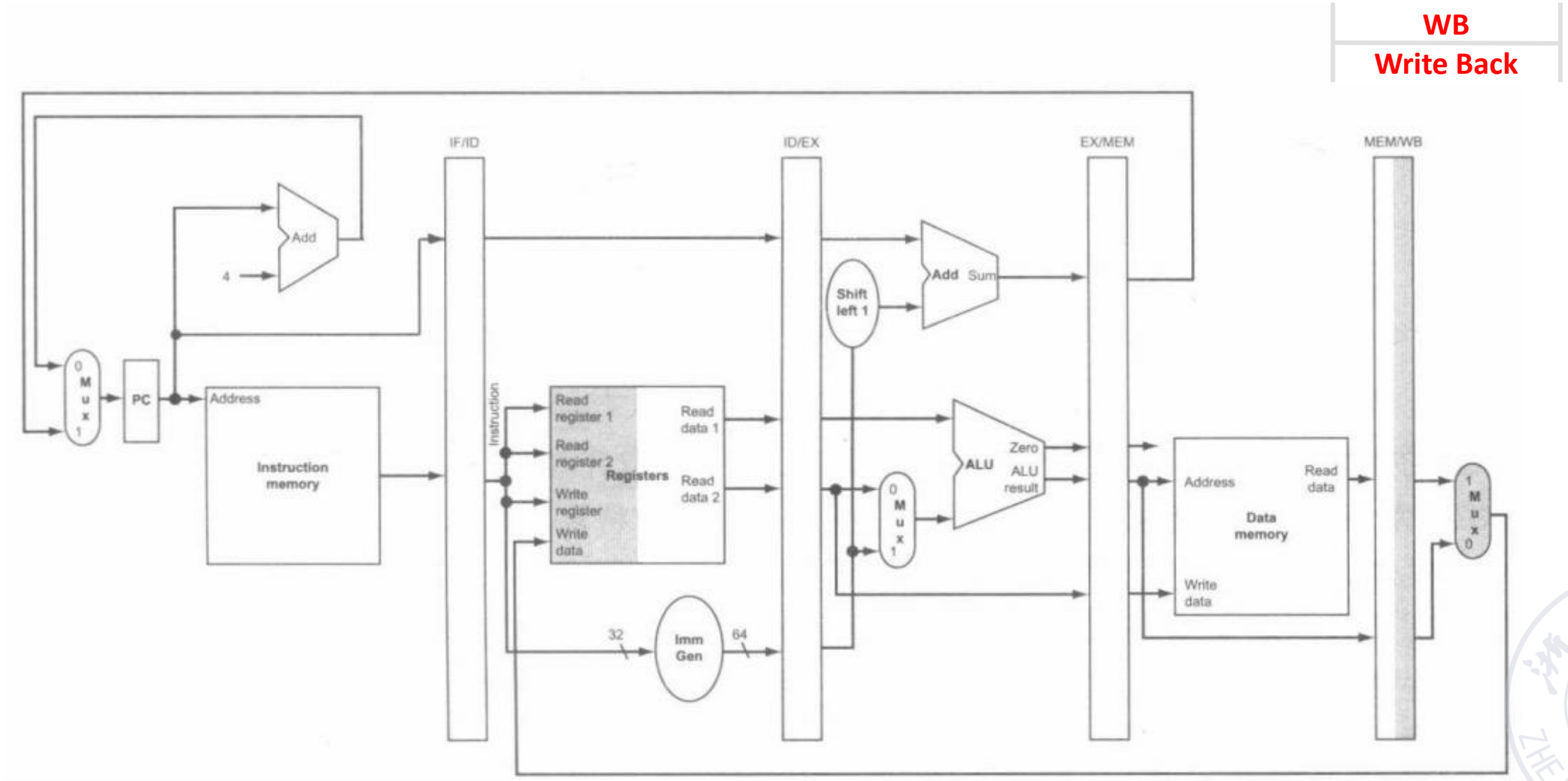


MEM Stage of Load Instruction

MEM
Memory Access

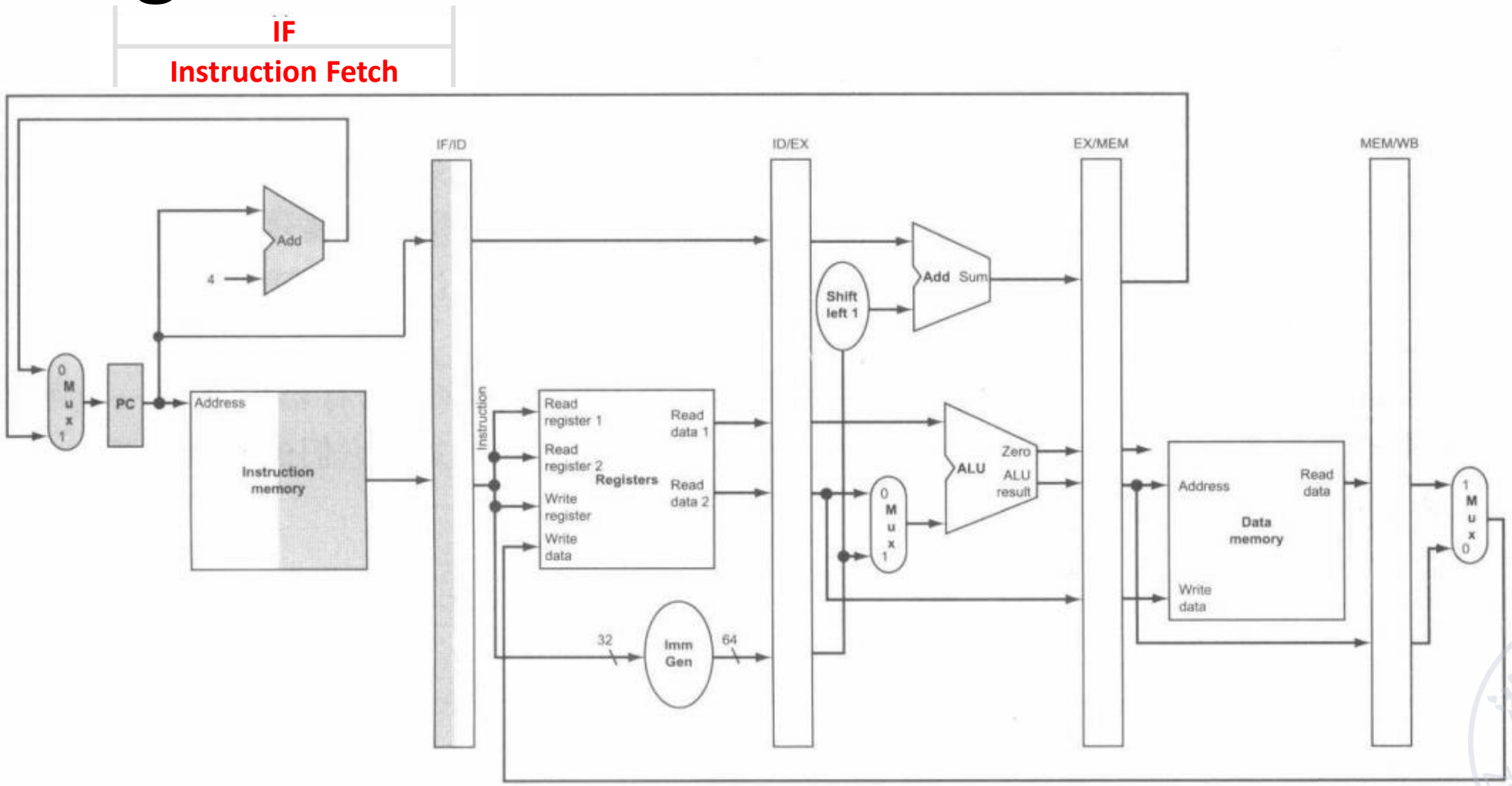


WB Stage of Load Instruction

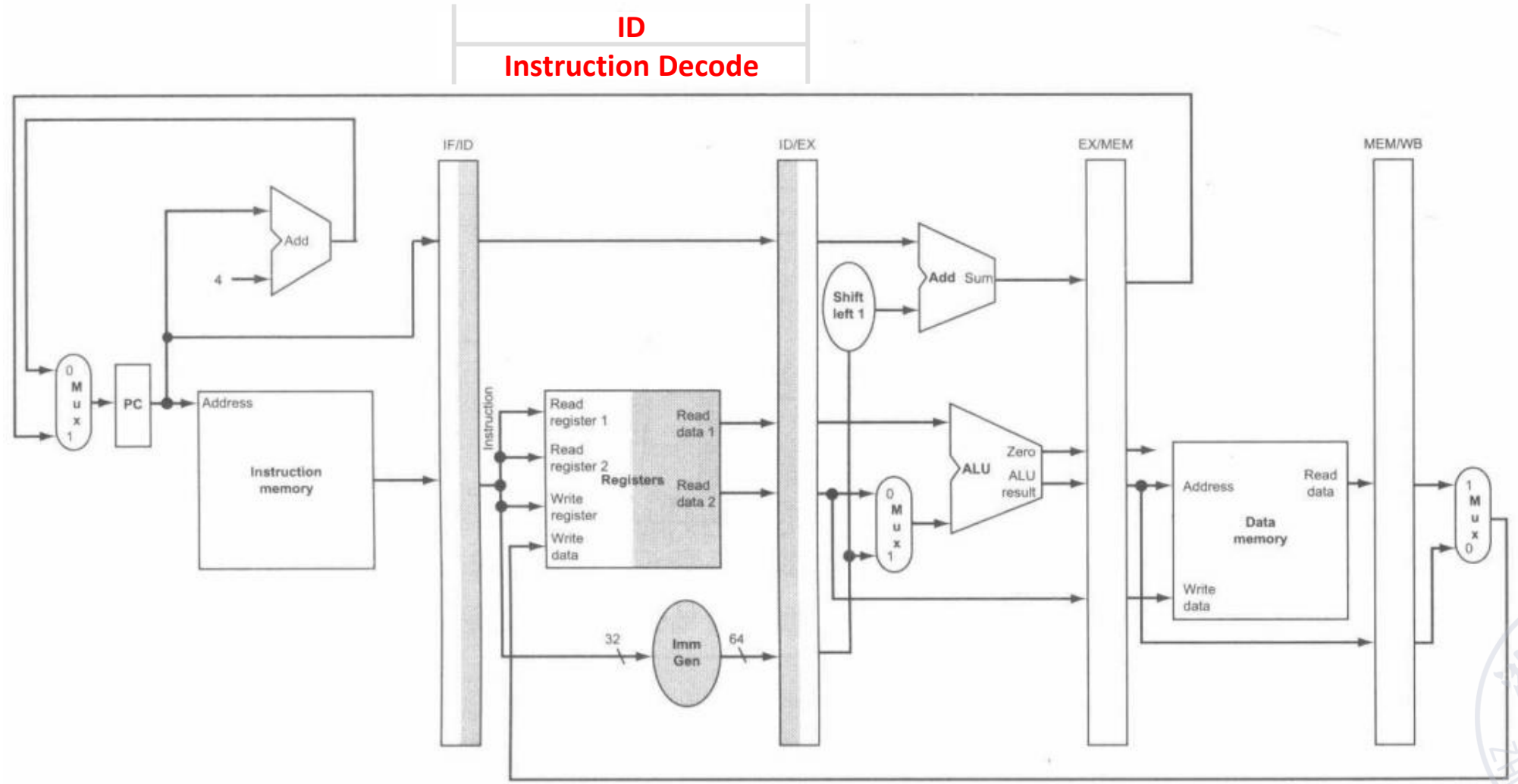


Store Instruction

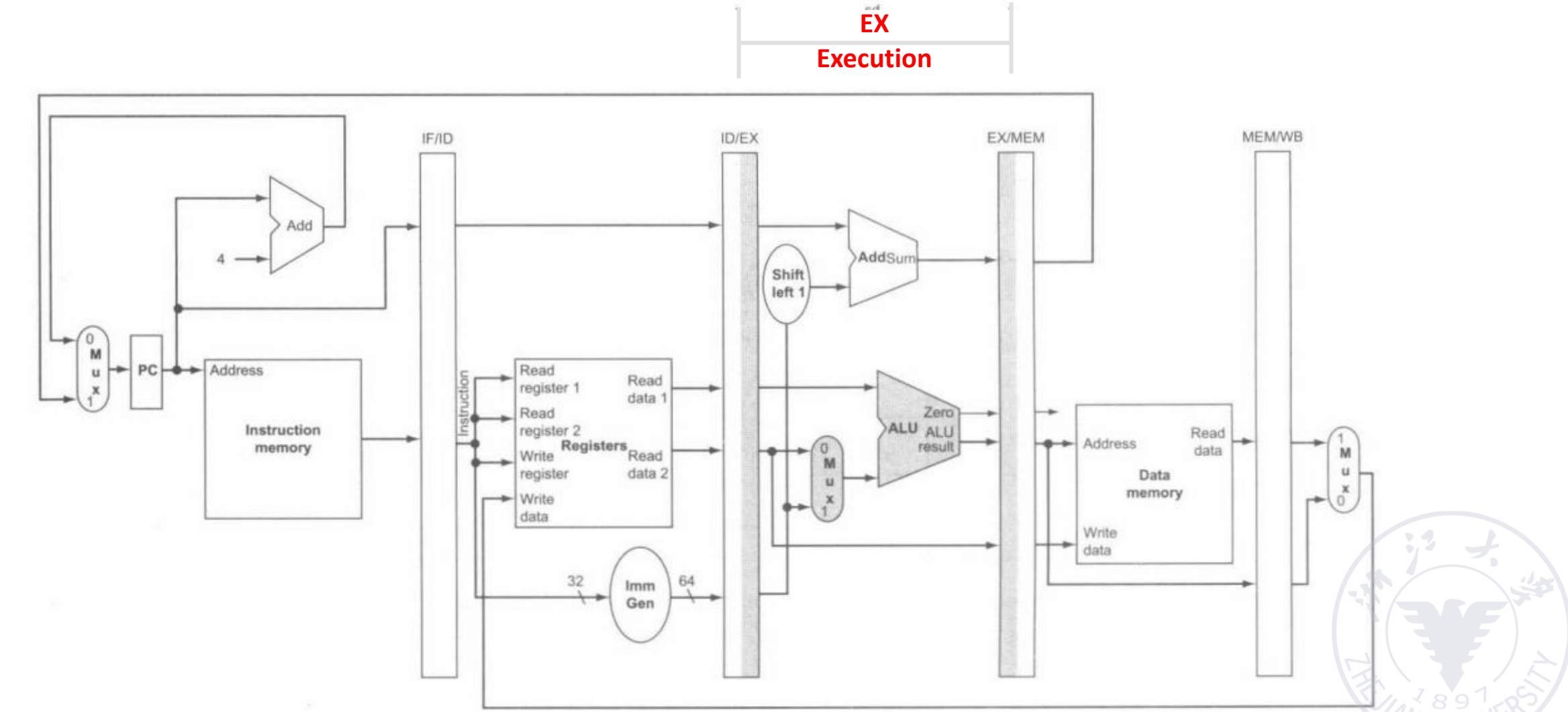
IF Stage of Store Instruction



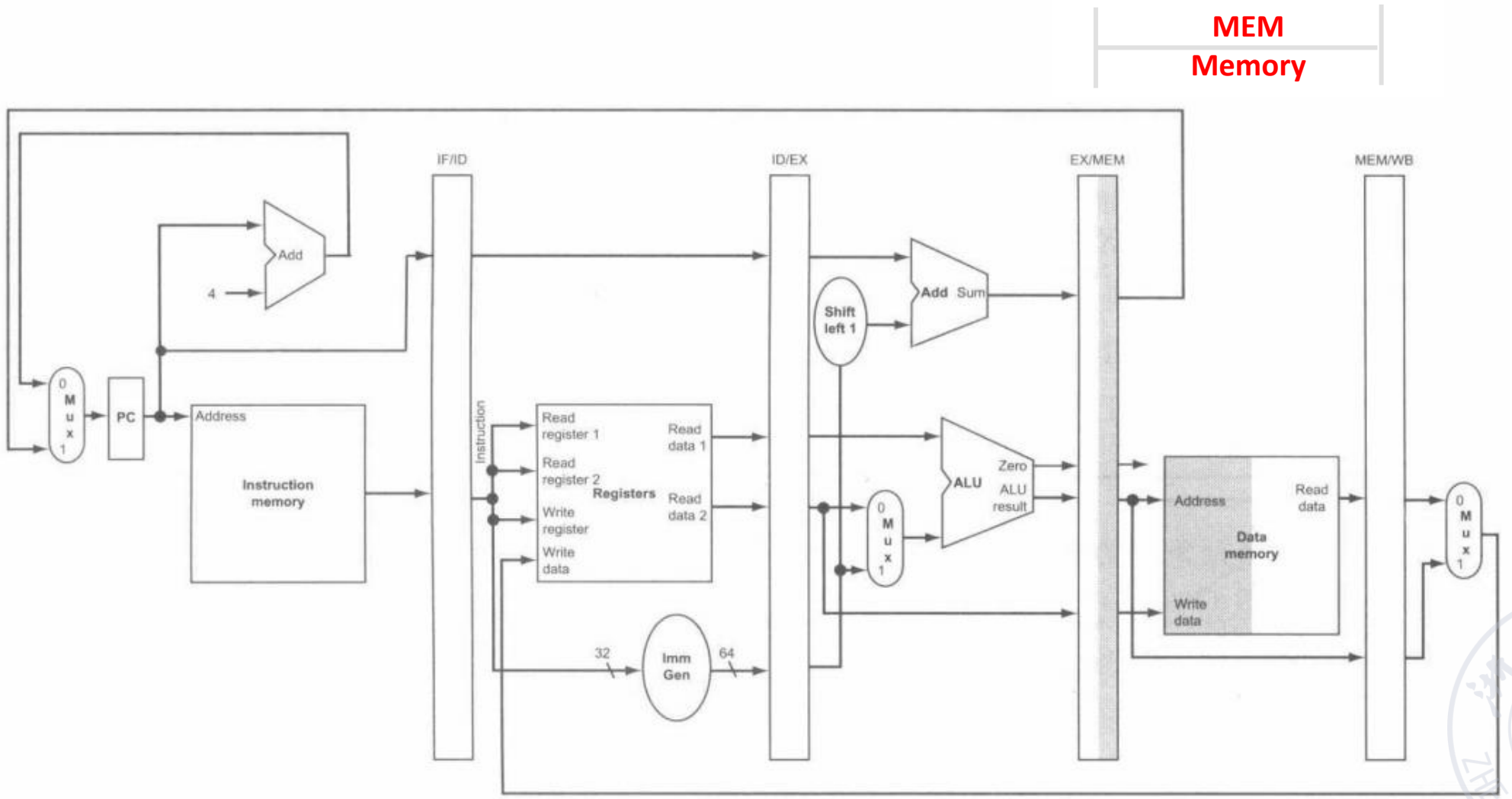
ID Stage of Store Instruction



EX Stage of Store Instruction

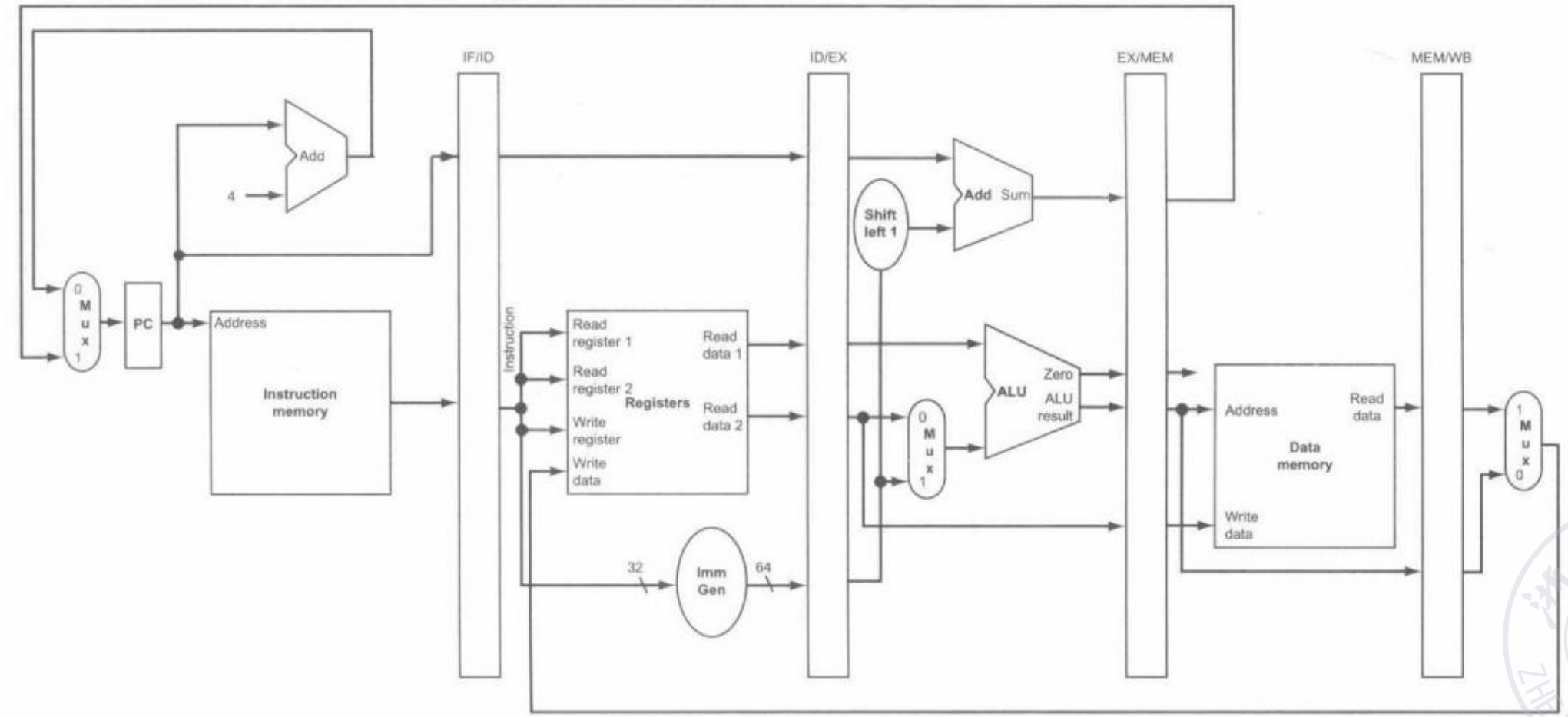


MEM Stage of Store Instruction



WB Stage of Store Instruction

WB
Write Back

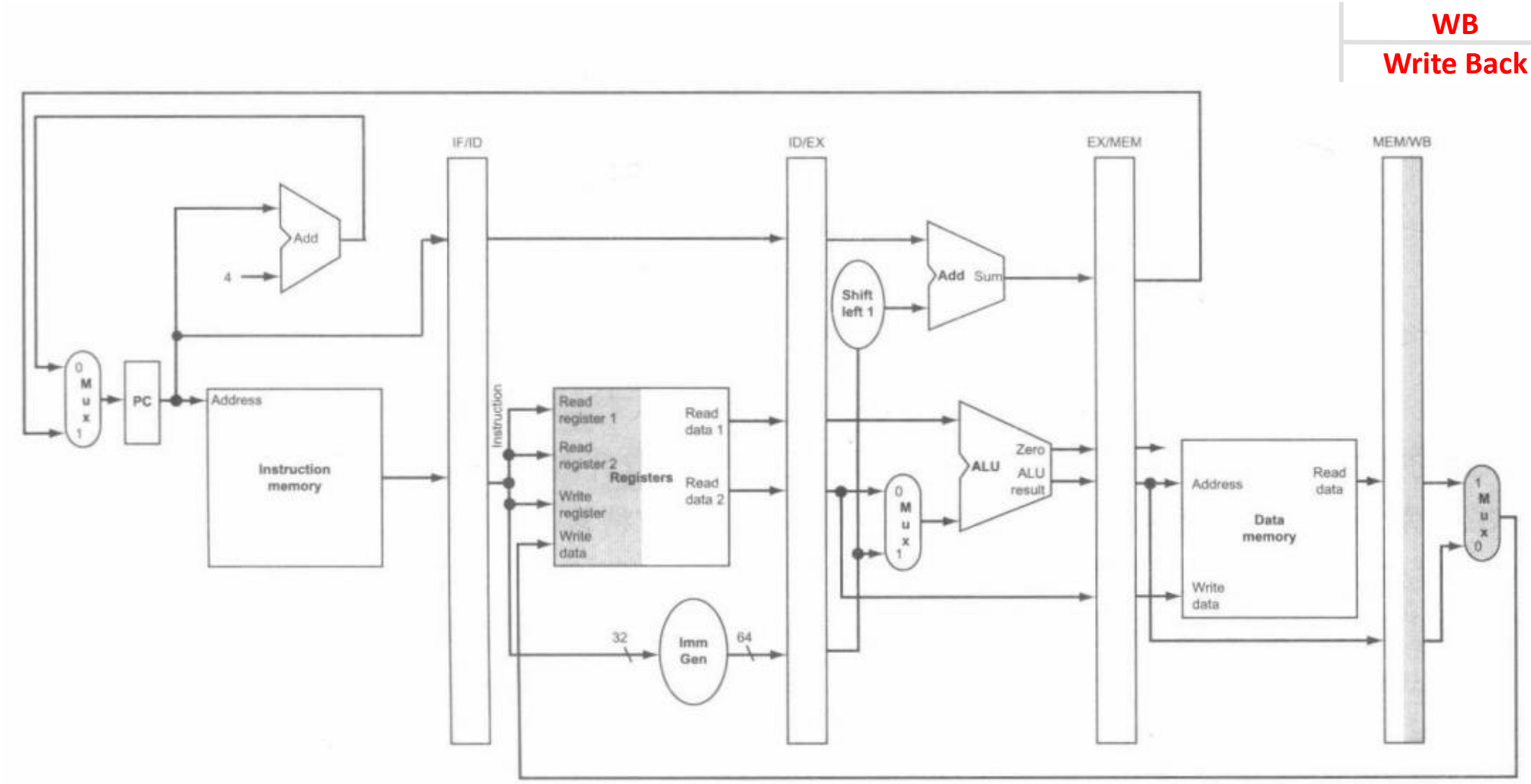


Review of Store Instruction

To pass something from an early pipe stage to a later pipe stage, the information must be placed in a pipeline register.

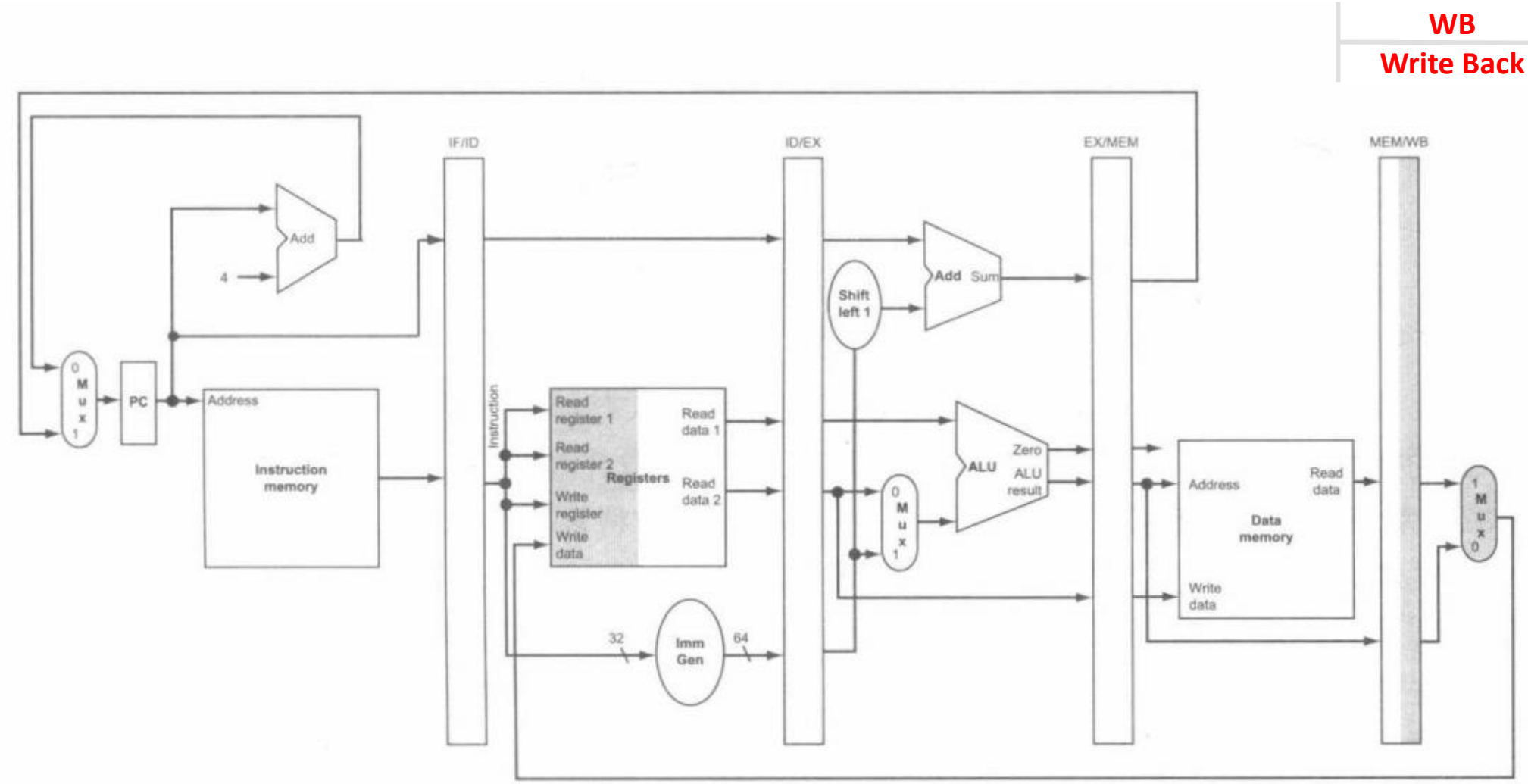
Review of Load Instruction in WB Stage

WB
Write Back



Review of Load Instruction in WB Stage

WB
Write Back



Let's uncover a bug in the design of the load instruction

Review the Load Instruction in WB Stage

Write Back to the Register

- **MEM/WB** — Supplies data
- **ID/EX Register** — Supplies the write register number

Review the Load Instruction in WB Stage

Write Back to the Register

- **MEM/WB** — Supplies data
- **ID/EX Register** — Supplies the write register number

Question: Is it the right write register number?

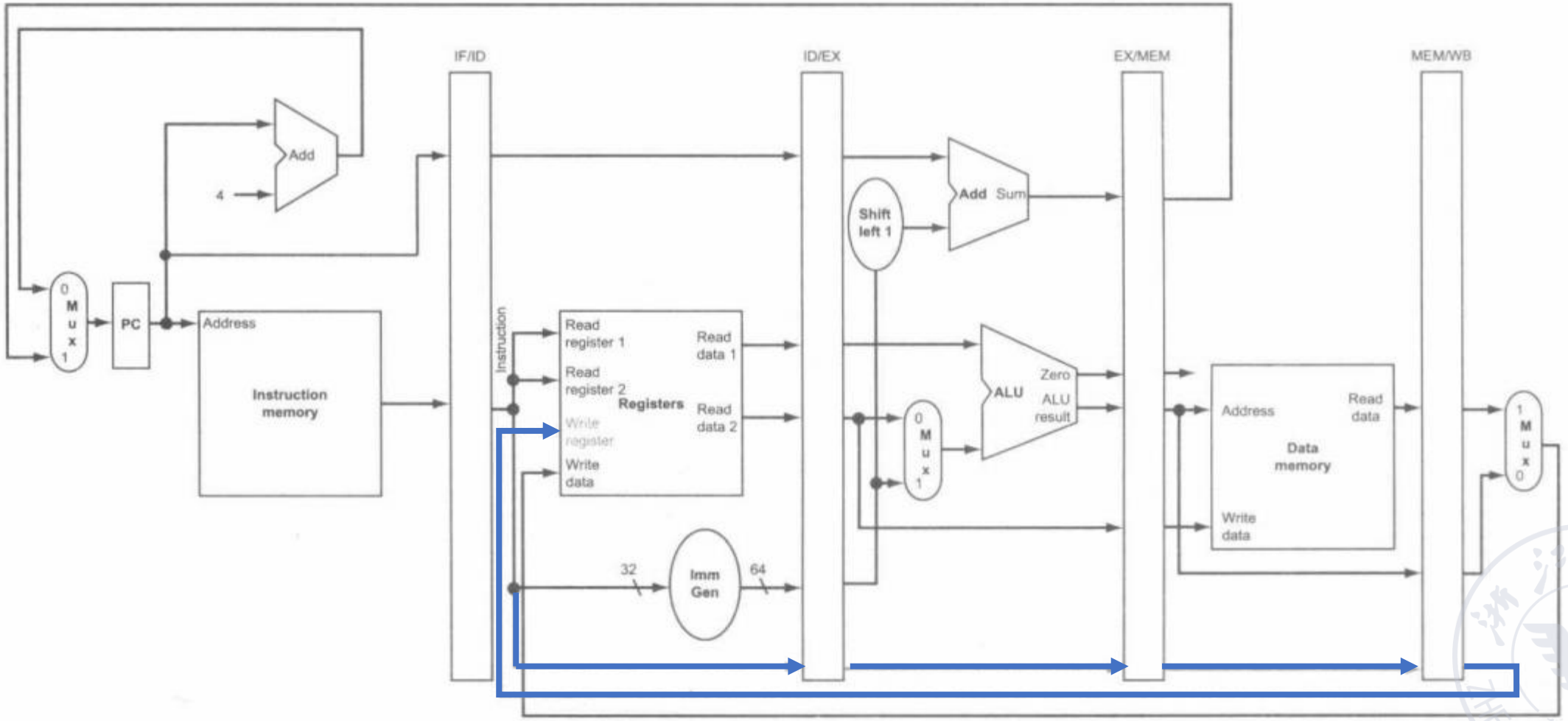
Review the Load Instruction in WB Stage

Write Back to the Register

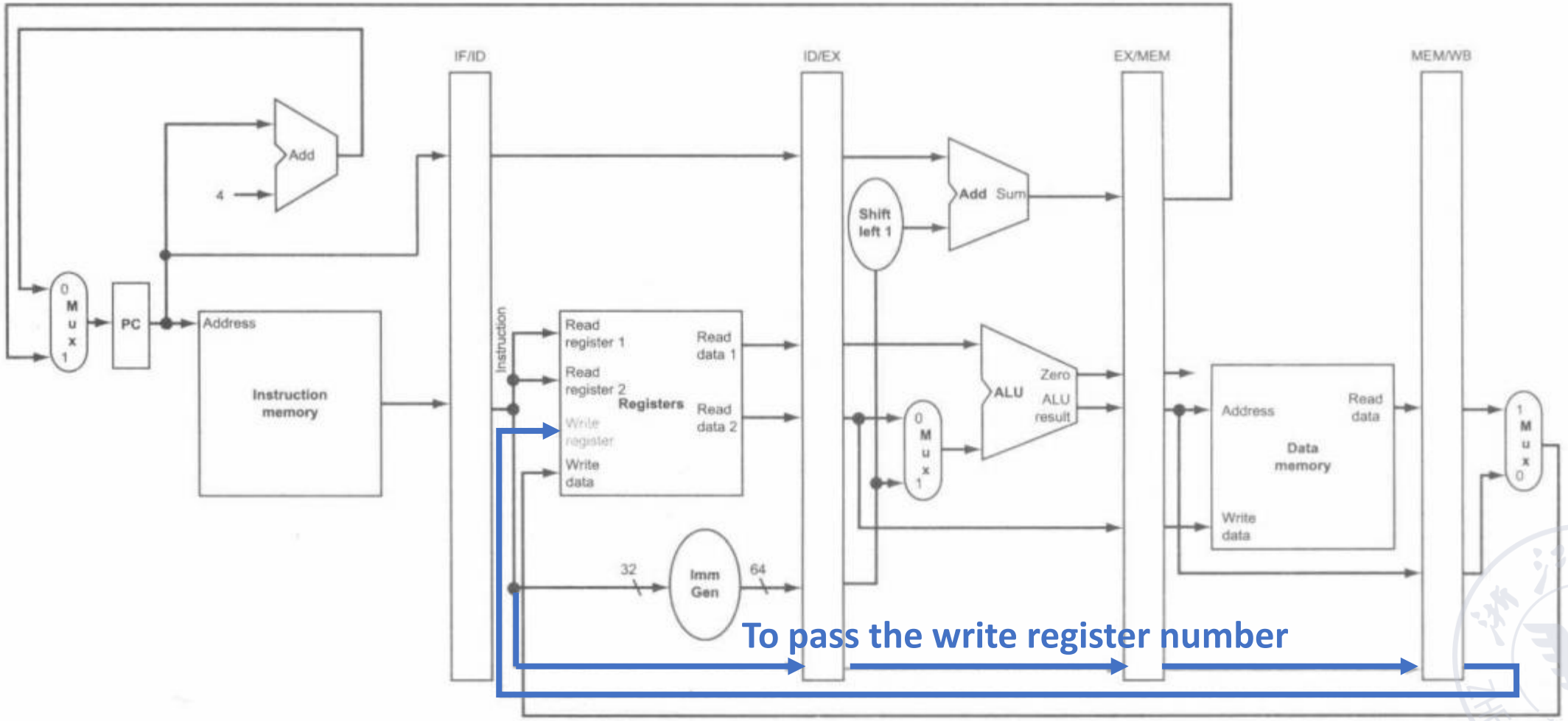
- **MEM/WB** — Supplies data
- **ID/EX Register** — Supplies the [illegible]

ID/EX Register has been modified by another instruction

The Corrected Pipelined Datapath to Handle the Load Instruction Properly

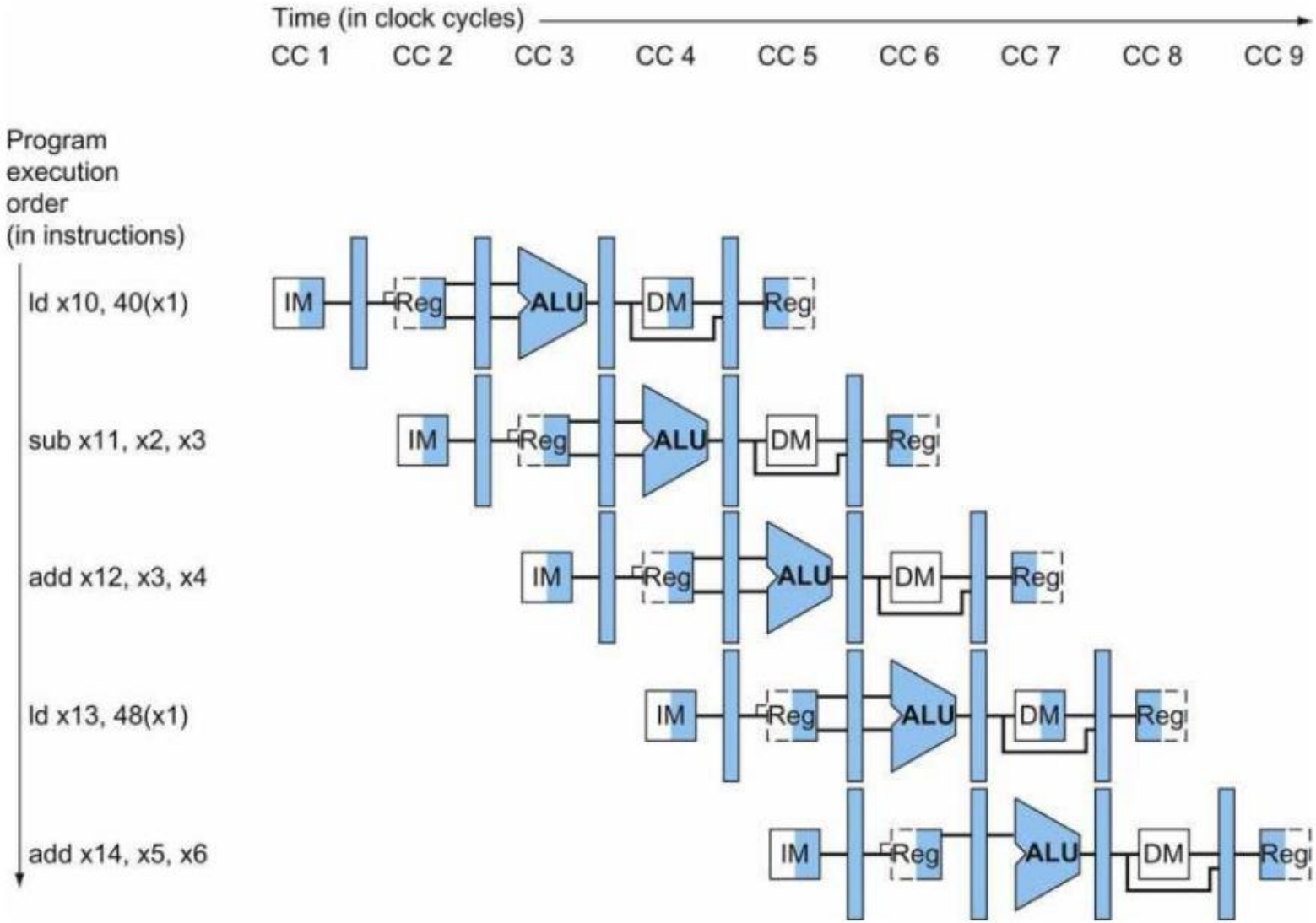


The Corrected Pipelined Datapath to Handle the Load Instruction Properly

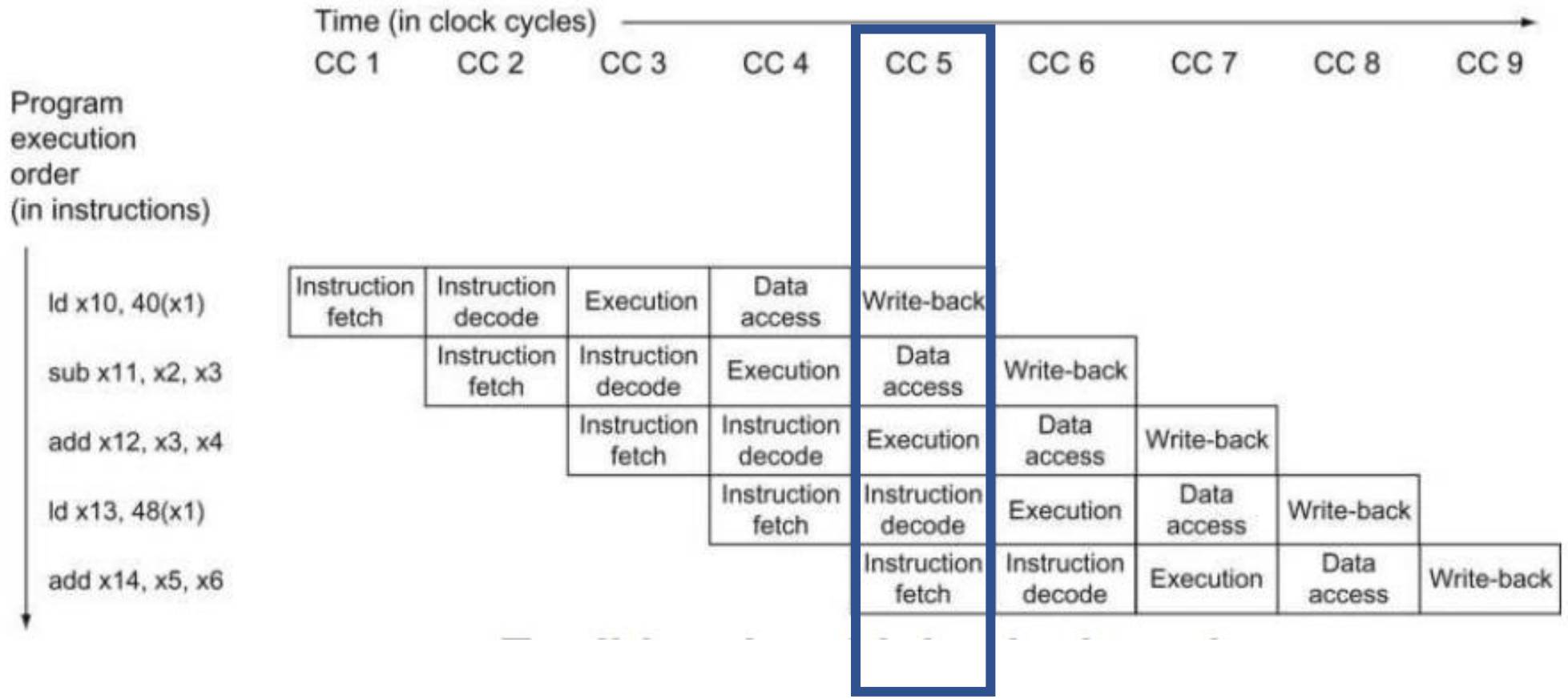


Graphically Representing Pipelines

Multiple-Clock-Cycle Pipeline Diagram of Five Instructions

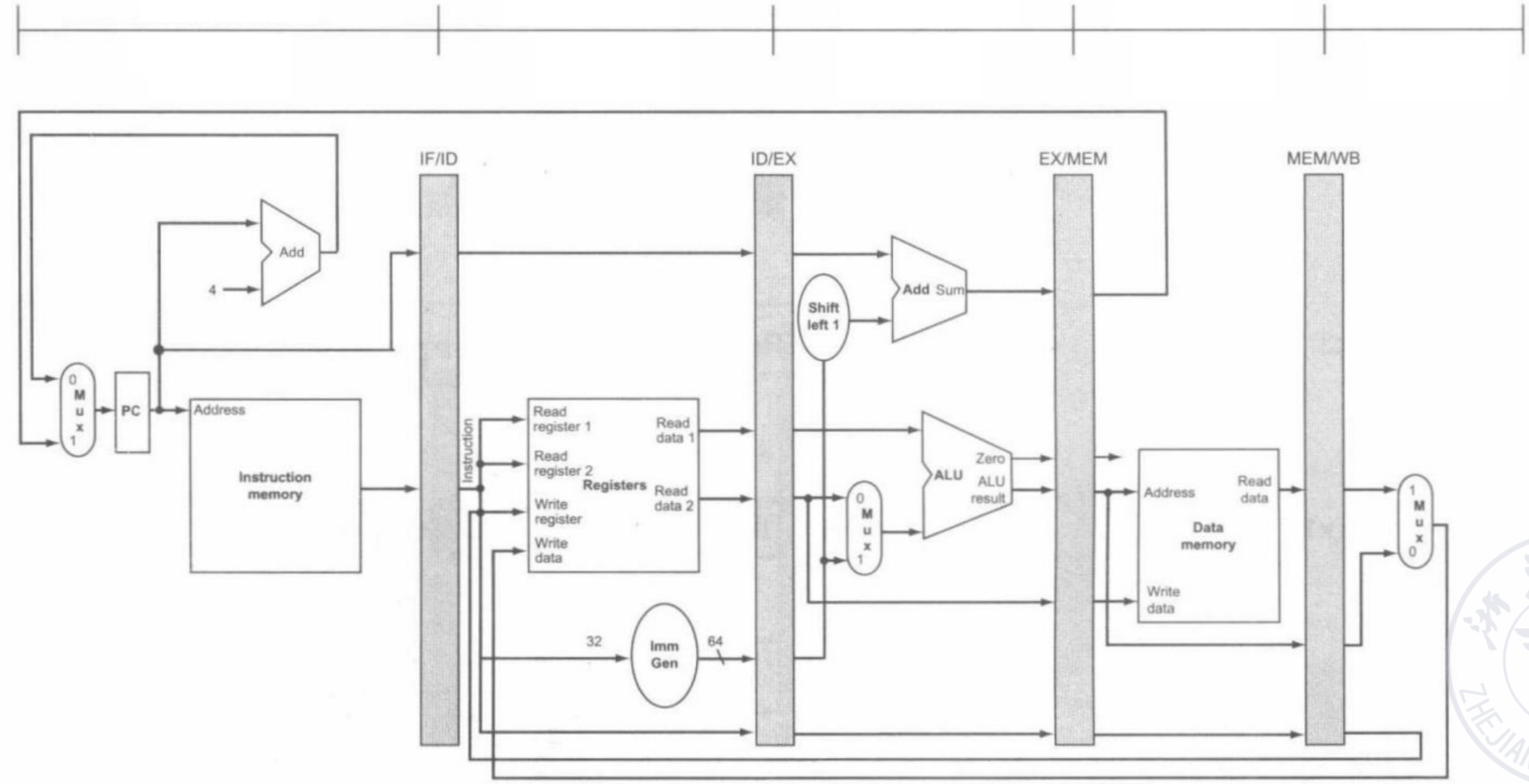


Traditional Multiple-Clock-Cycle Pipeline Diagram



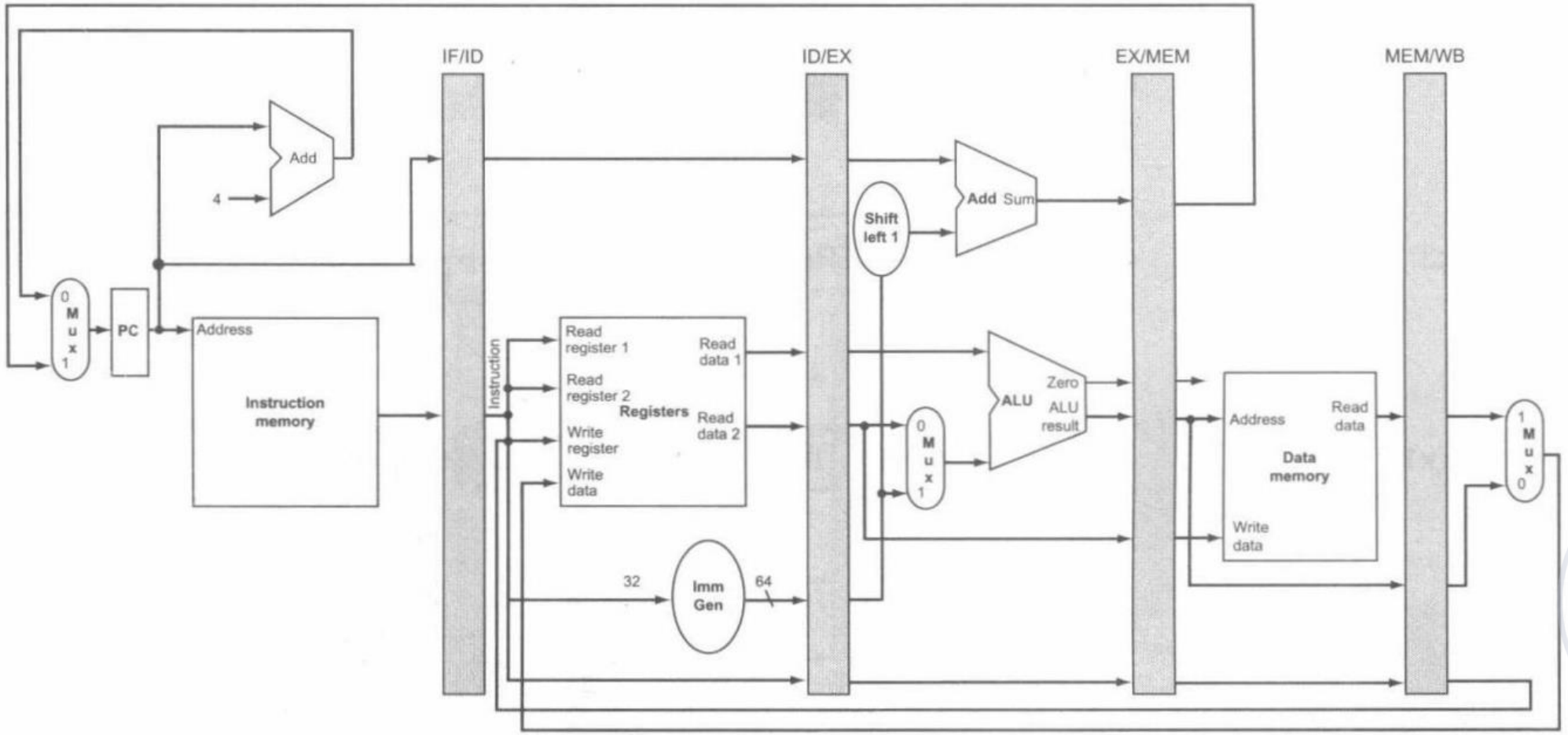
Now draw the Single-clock-cycle pipeline diagram at CC5

Single-Clock-Cycle Pipeline Diagram at CC5



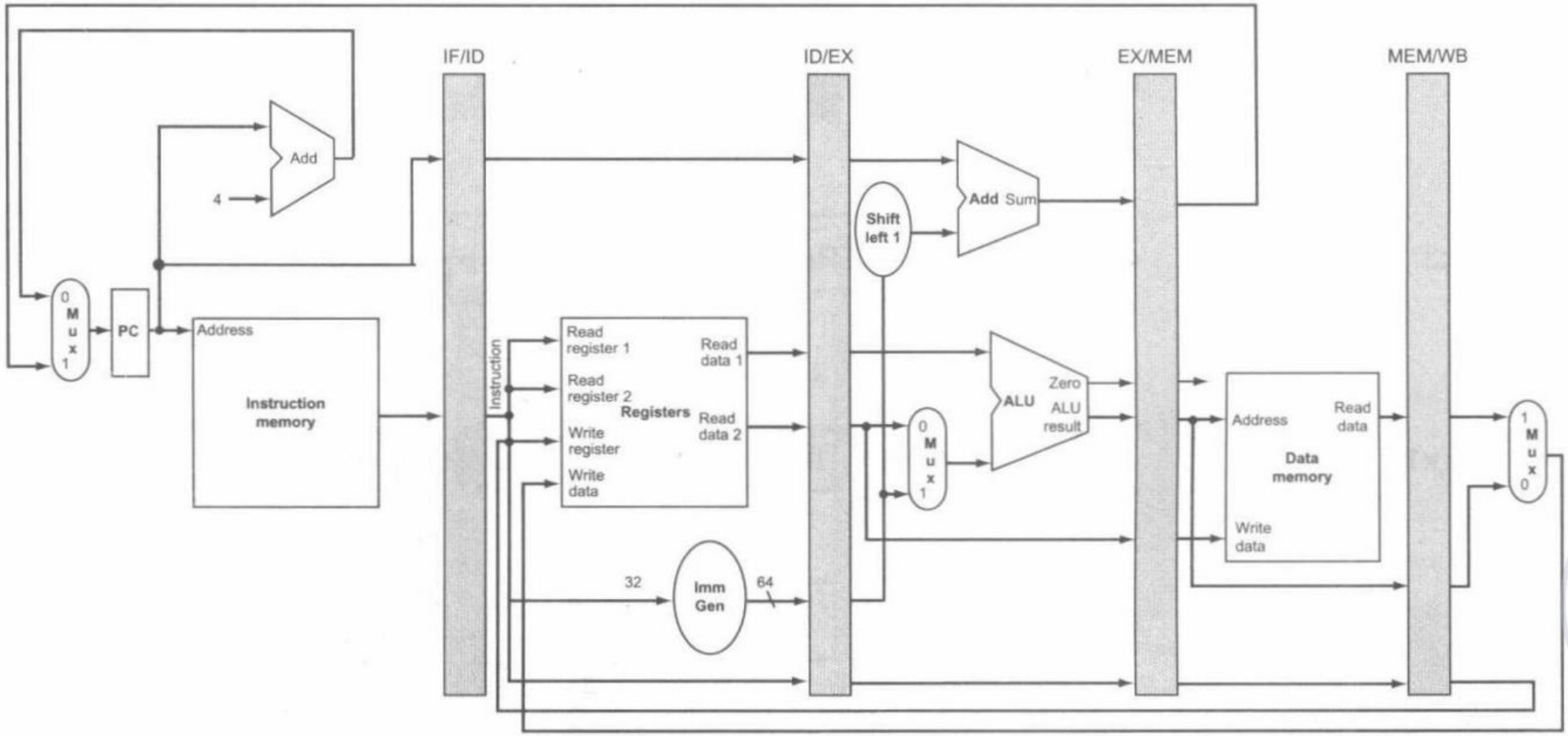
Single-Clock-Cycle Pipeline Diagram at CC5

Question: Write name and stage of instructions



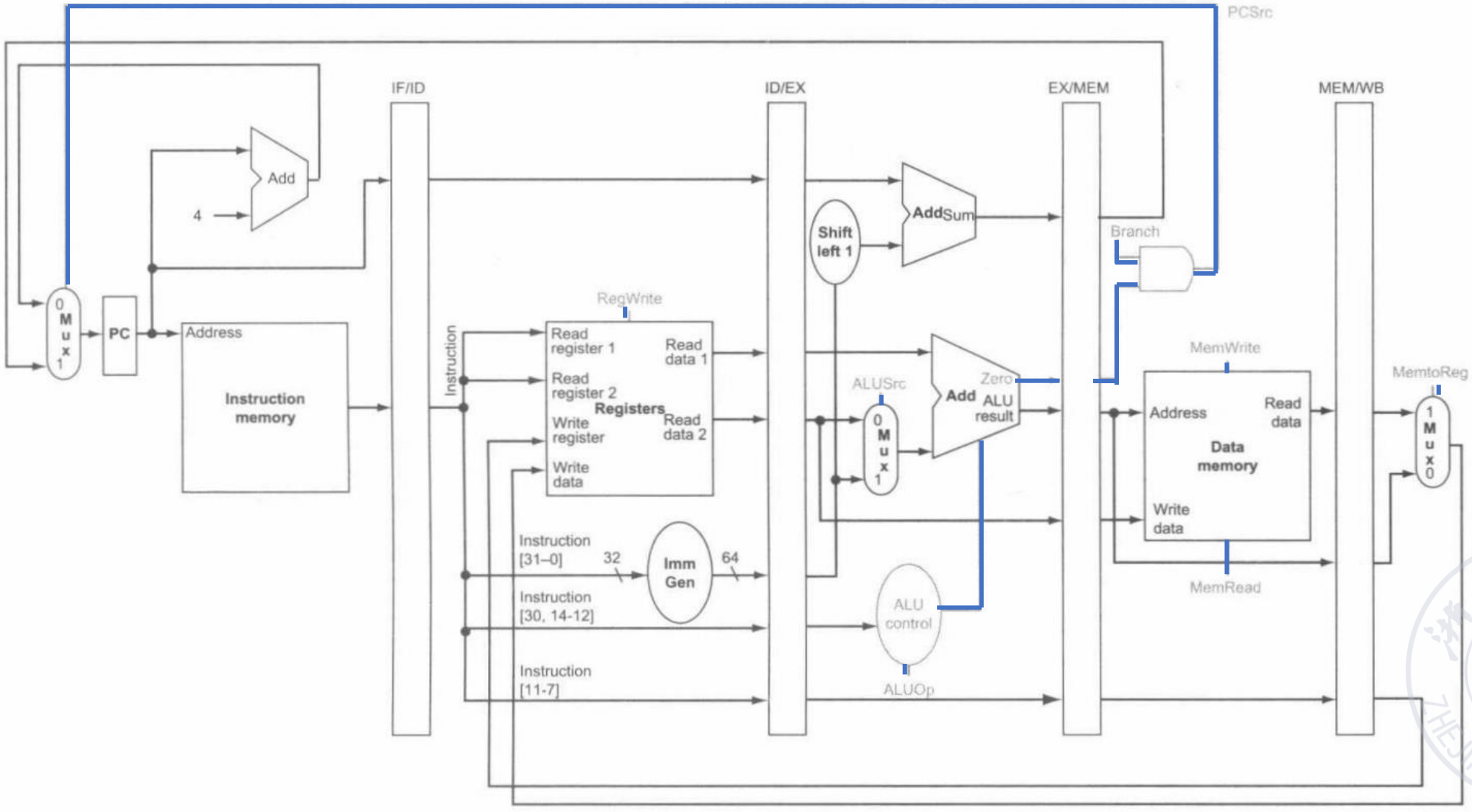
Single-Clock-Cycle Pipeline Diagram at CC5

add x14,x5,x6	ld x13,48(x1)	add x12,x3,x4	sub x11,x2,x3	ld x10,40(x1)
Instruction Fetch	Instruction Decode	Execution	Memory	Write Back

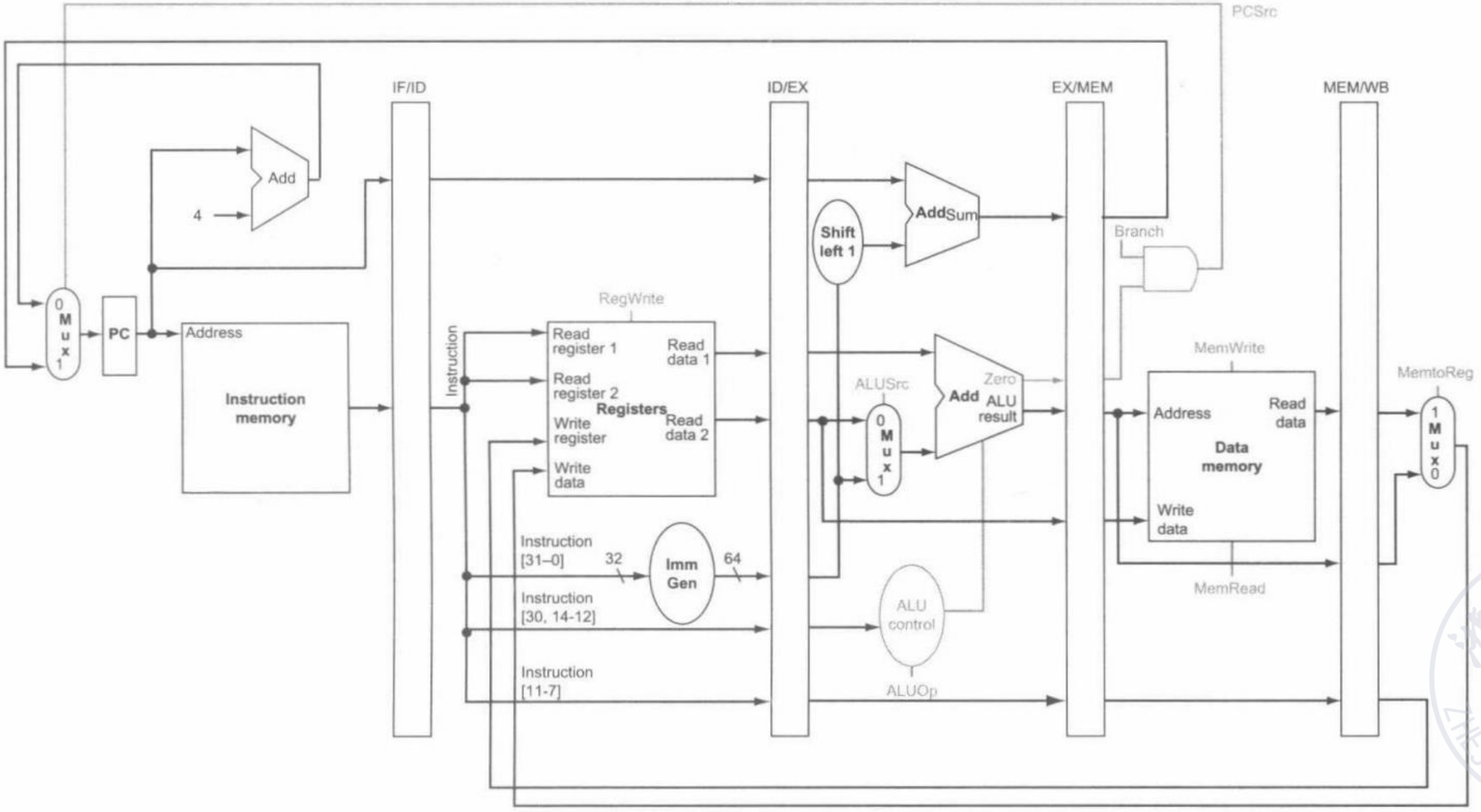


Pipelined Control

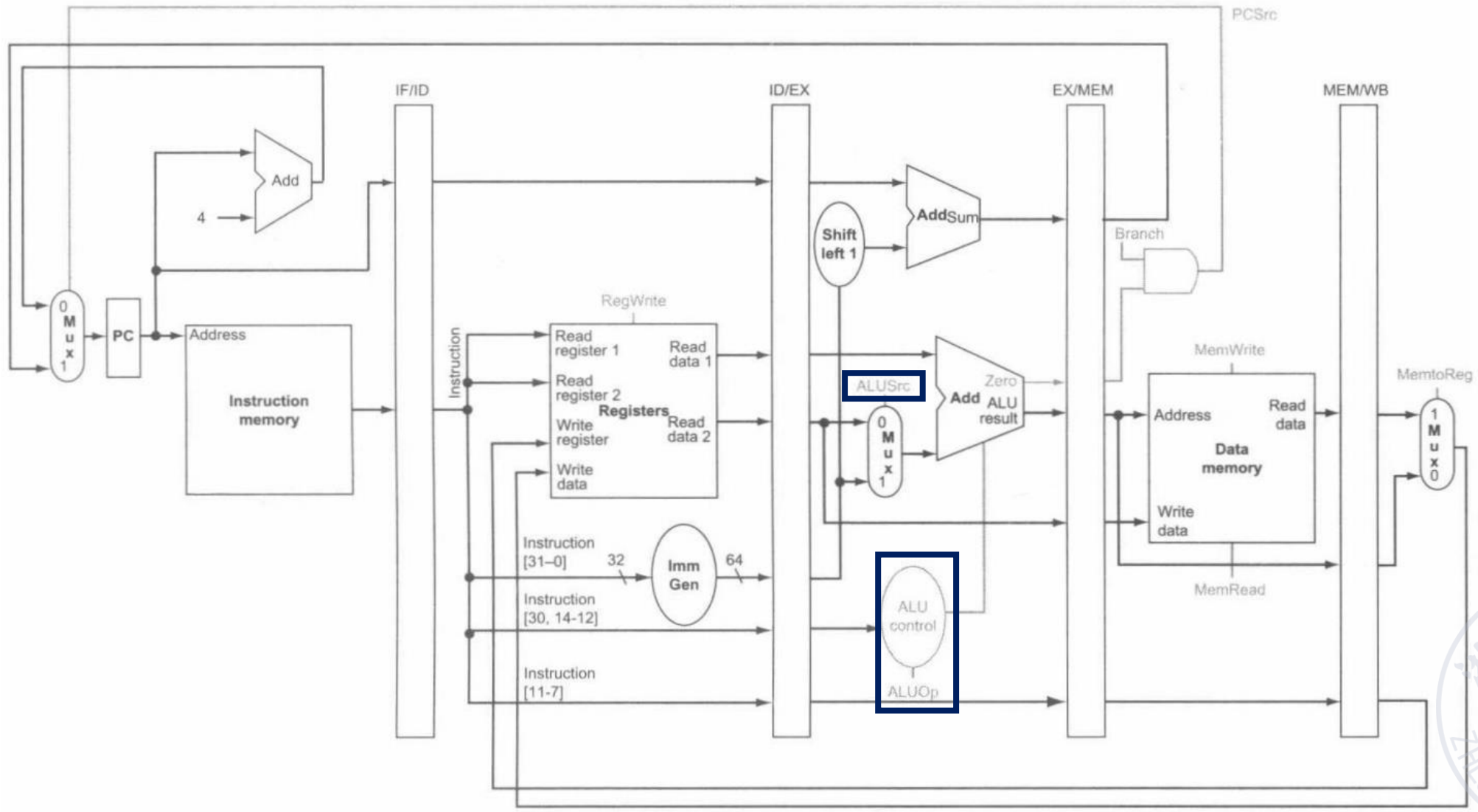
Pipelined Datapath with Control Signals



No Extra Design at Instruction Fetch and Instruction Decode



Control Signals at Execution Stage



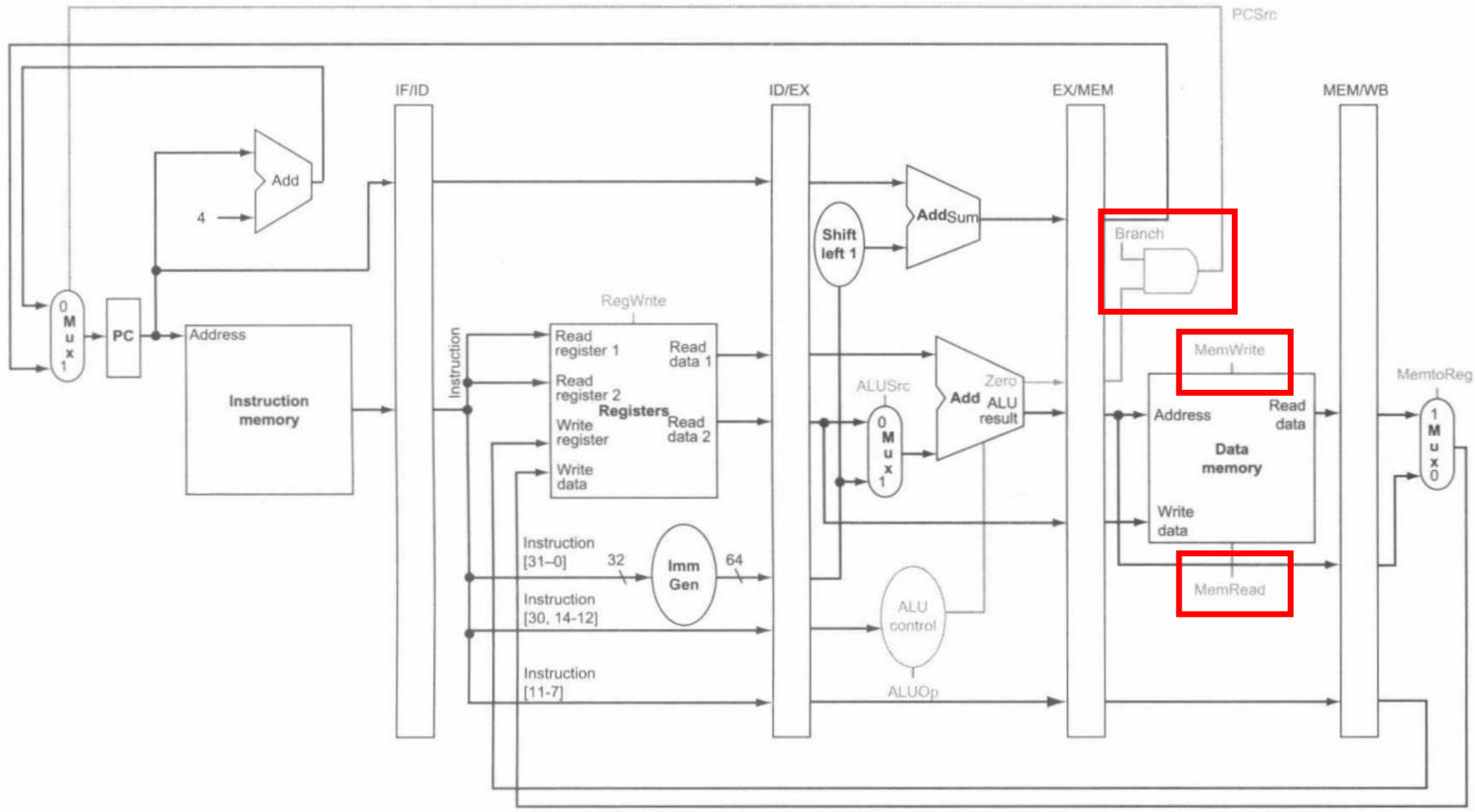
Signals Related to ALU Operations

Instruction	ALUOp	operation	Funct7 field	Funct3 field	Desired ALU action	ALU control input
ld	00	load doubleword	XXXXXXX	XXX	add	0010
sd	00	store doubleword	XXXXXXX	XXX	add	0010
beq	01	branch if equal	XXXXXXX	XXX	subtract	0110
R-type	10	add	0000000	000	add	0010
R-type	10	sub	0100000	000	subtract	0110
R-type	10	and	0000000	111	AND	0000
R-type	10	or	0000000	110	OR	0001

Signal name	Effect when deasserted	Effect when asserted
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, 12 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

These signals select the ALU operations

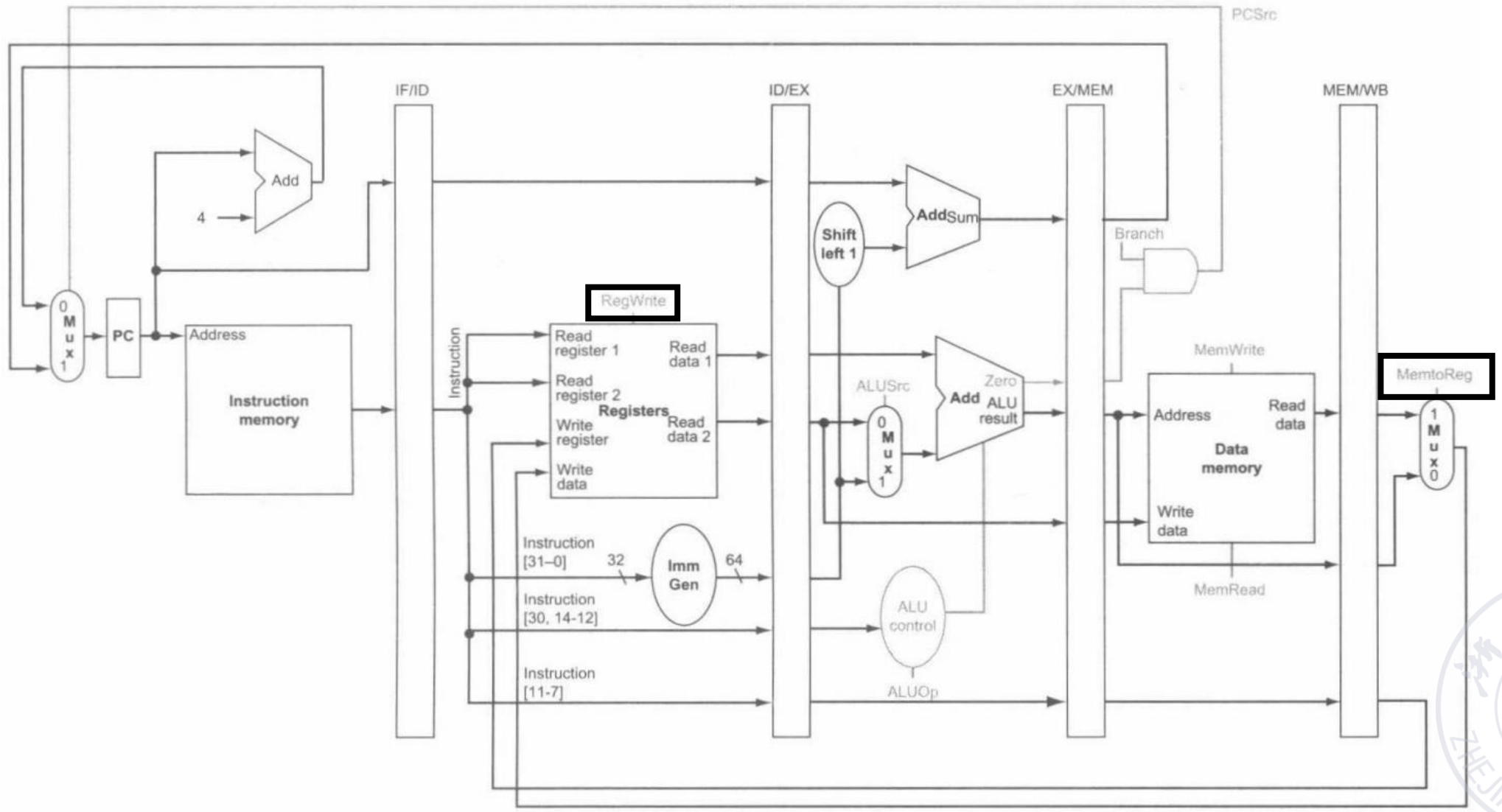
Control Signals at Memory Access Stage



Signals Related to Branch If Equal, Load, and Store Instructions

Signal name	Effect when deasserted	Effect when asserted
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, 12 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

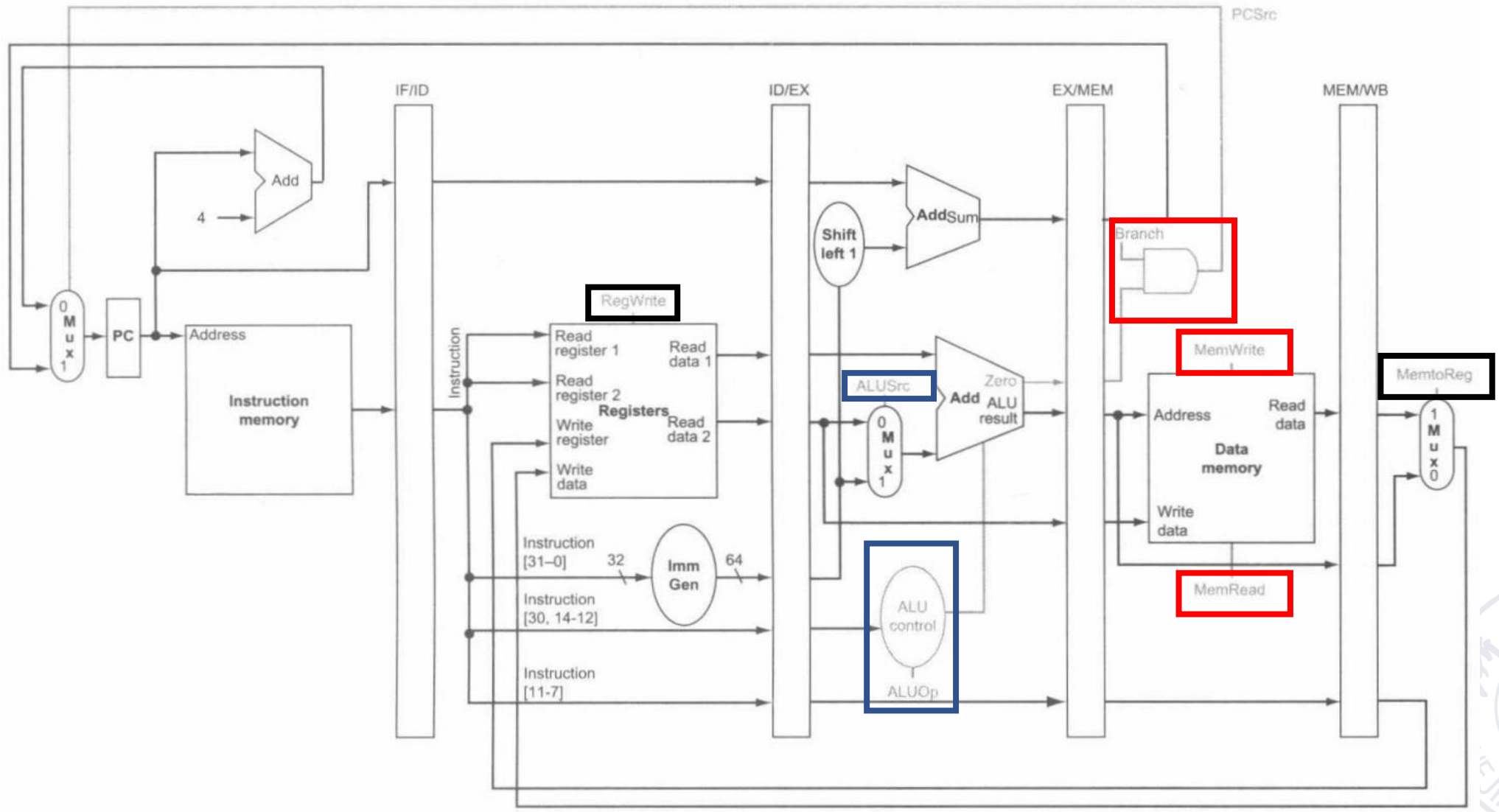
Control Signals at Write Back Stage



Signals Related to the Write Operation

Signal name	Effect when deasserted	Effect when asserted
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, 12 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

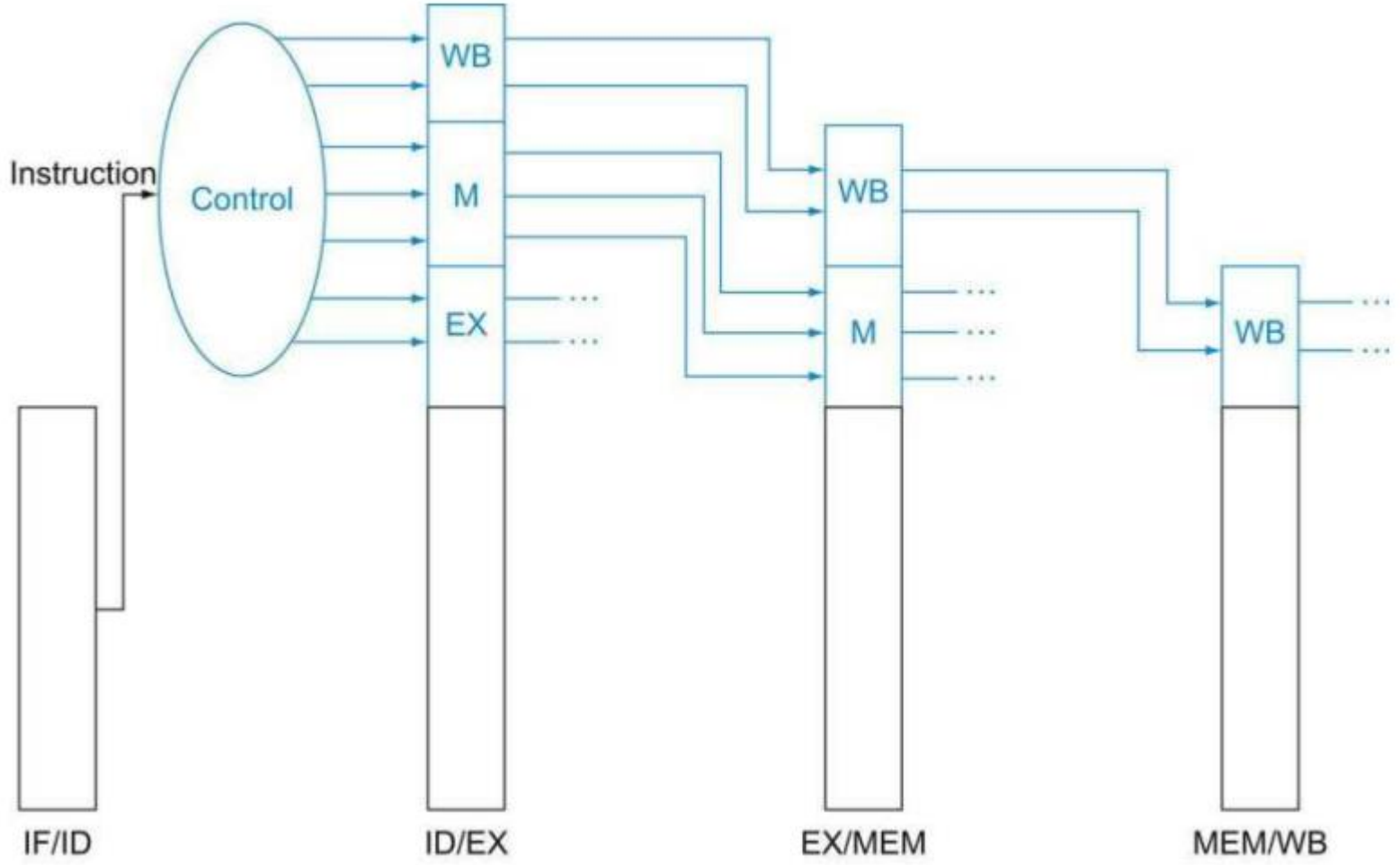
Review of 7 Control Lines



Seven Control Lines in Last Three Stages

Instruction	Execution/address calculation stage control lines		Memory access stage control lines			Write-back stage control lines	
	ALUOp	ALUSrc	Branch	Mem-Read	Mem-Write	Reg-Write	Memto-Reg
R-format	10	0	0	0	0	1	0
ld	00	1	0	1	0	1	1
sd	00	1	0	0	1	0	X
beq	01	0	1	0	0	0	X

Extend Pipeline Registers to Include Control Information



Pipelined Datapath with the Control Signals