CPT_S 260 Intro to Computer Architecture
Lecture 28

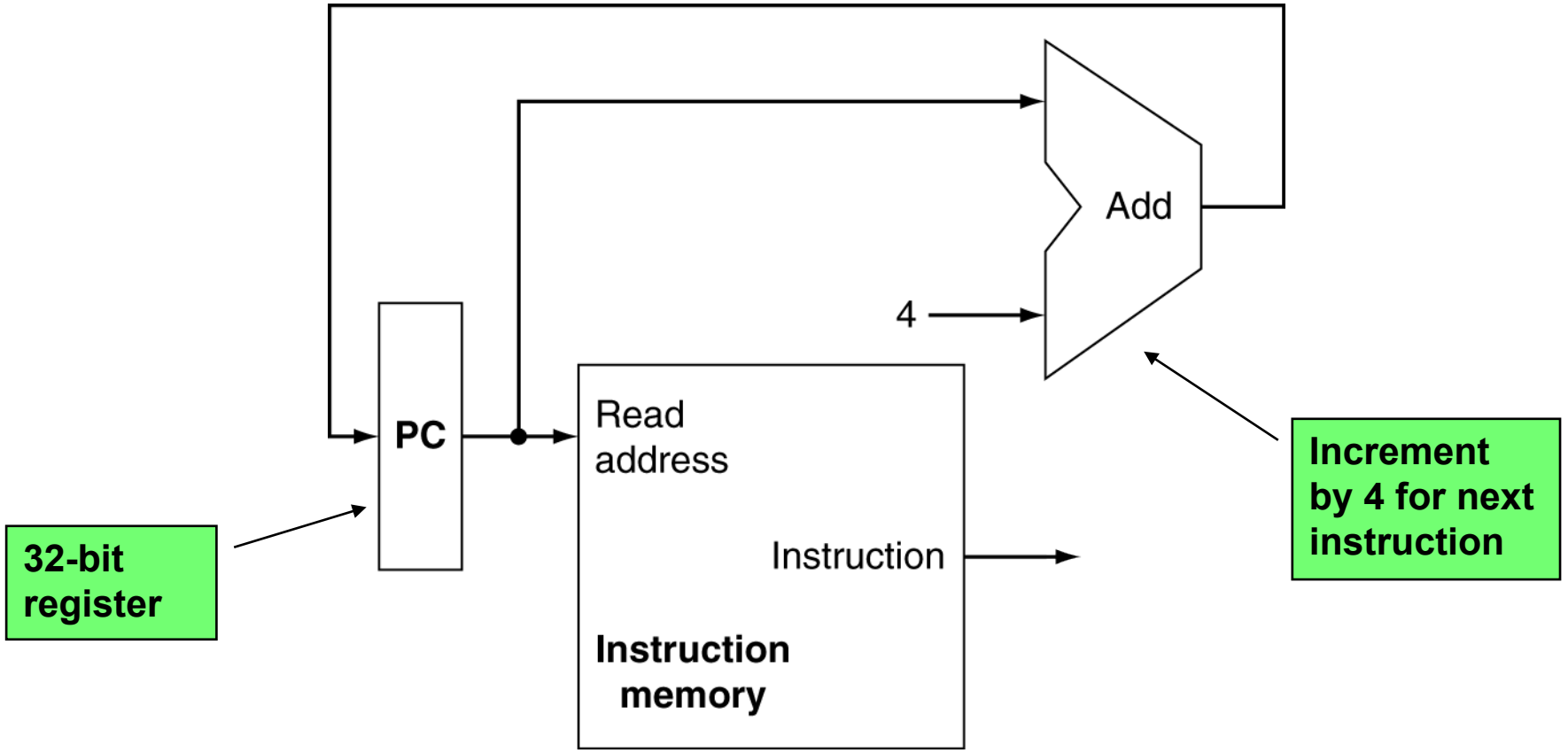
Single Cycle MIPS Control
March 25, 2022

Ganapati Bhat
School of Electrical Engineering and Computer Science
Washington State University

Announcements

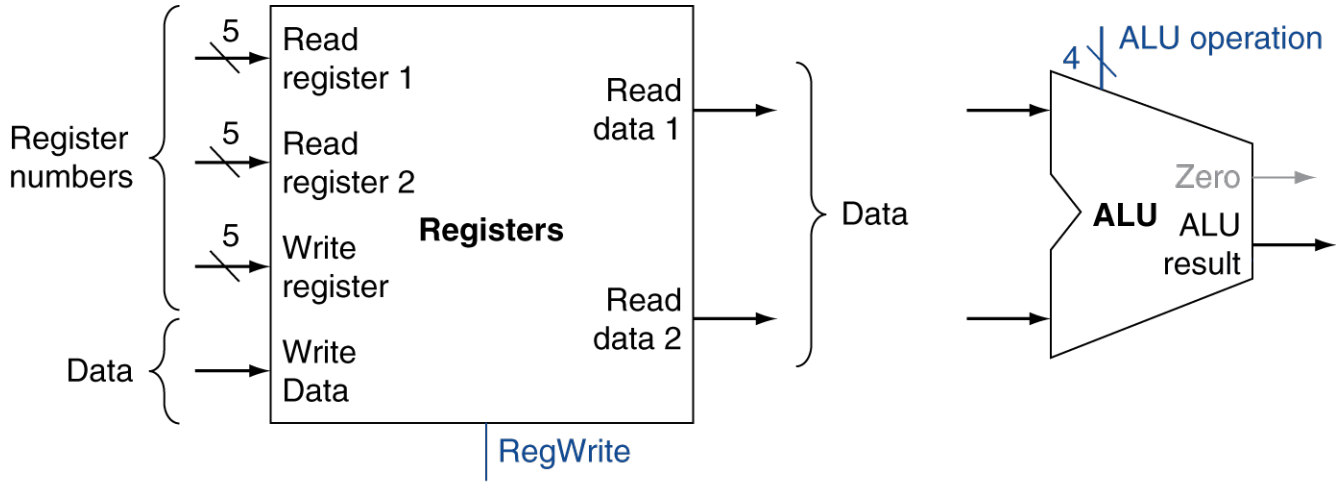
- **Mid term exam 2**
  - Friday, April 1, 2022
  - Format will be similar to exam 1
  - Tentative topics: MIPS, digital logic, single cycle MIPS implementation
- **Final exam**
  - May 4, 2022
  - 7:30 pm to 9:30 pm
  - If you're taking Calculus 3, please reach out to me

Recap: Instruction Fetch



Recap: R-Format Instructions

- Read two register operands
- Perform arithmetic/logical operation
- Write register result

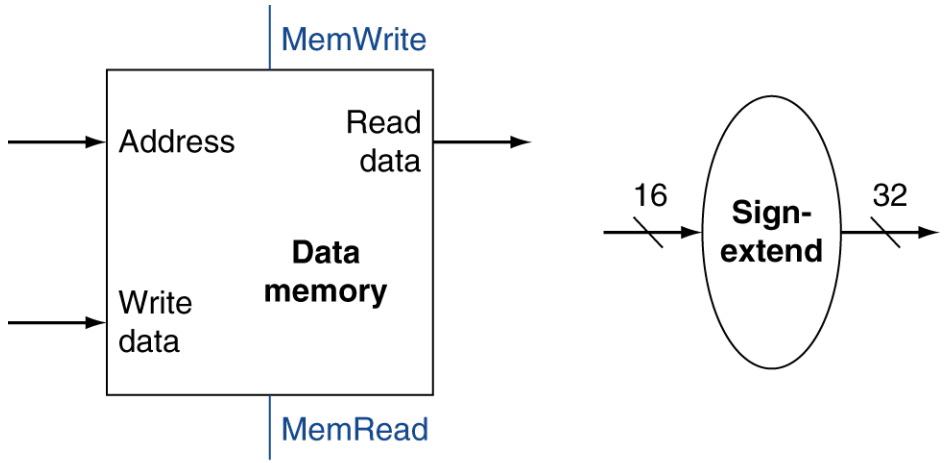


a. Registers

b. ALU

Recap: Load/Store Instructions

- **Read register operands**
- **Calculate address using 16-bit offset**
  - Use ALU, but sign-extend offset
- **Load: Read memory and update register**
- **Store: Write register value to memory**



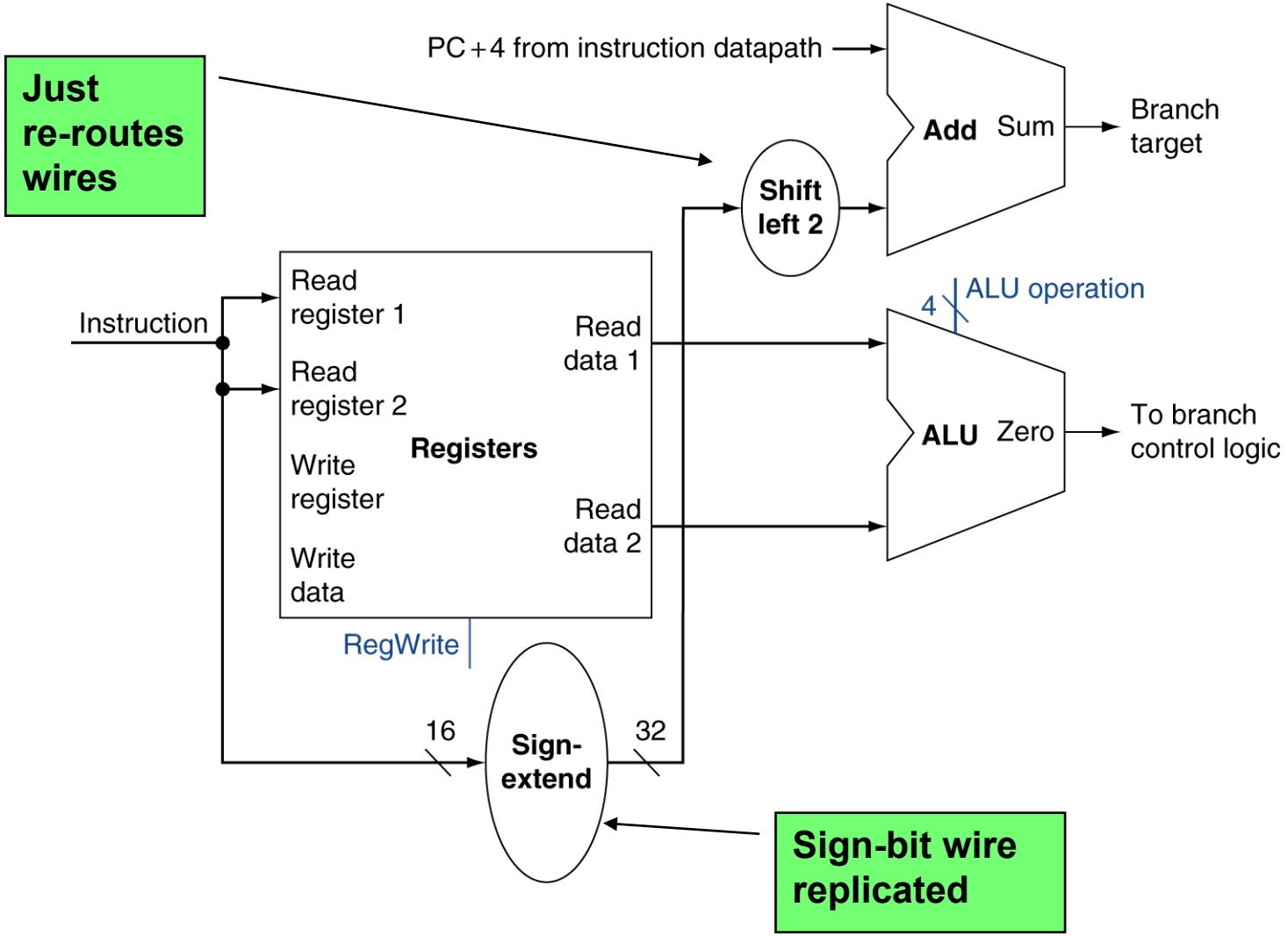
a. Data memory unit

b. Sign extension unit

Branch Instructions

- **Read register operands**
- **Compare operands**
  - Use ALU, subtract and check Zero output
- **Calculate target address**
  - Sign-extend displacement
  - Shift left 2 places (word displacement)
  - Add to PC + 4
    - » Already calculated by instruction fetch

Branch Instructions



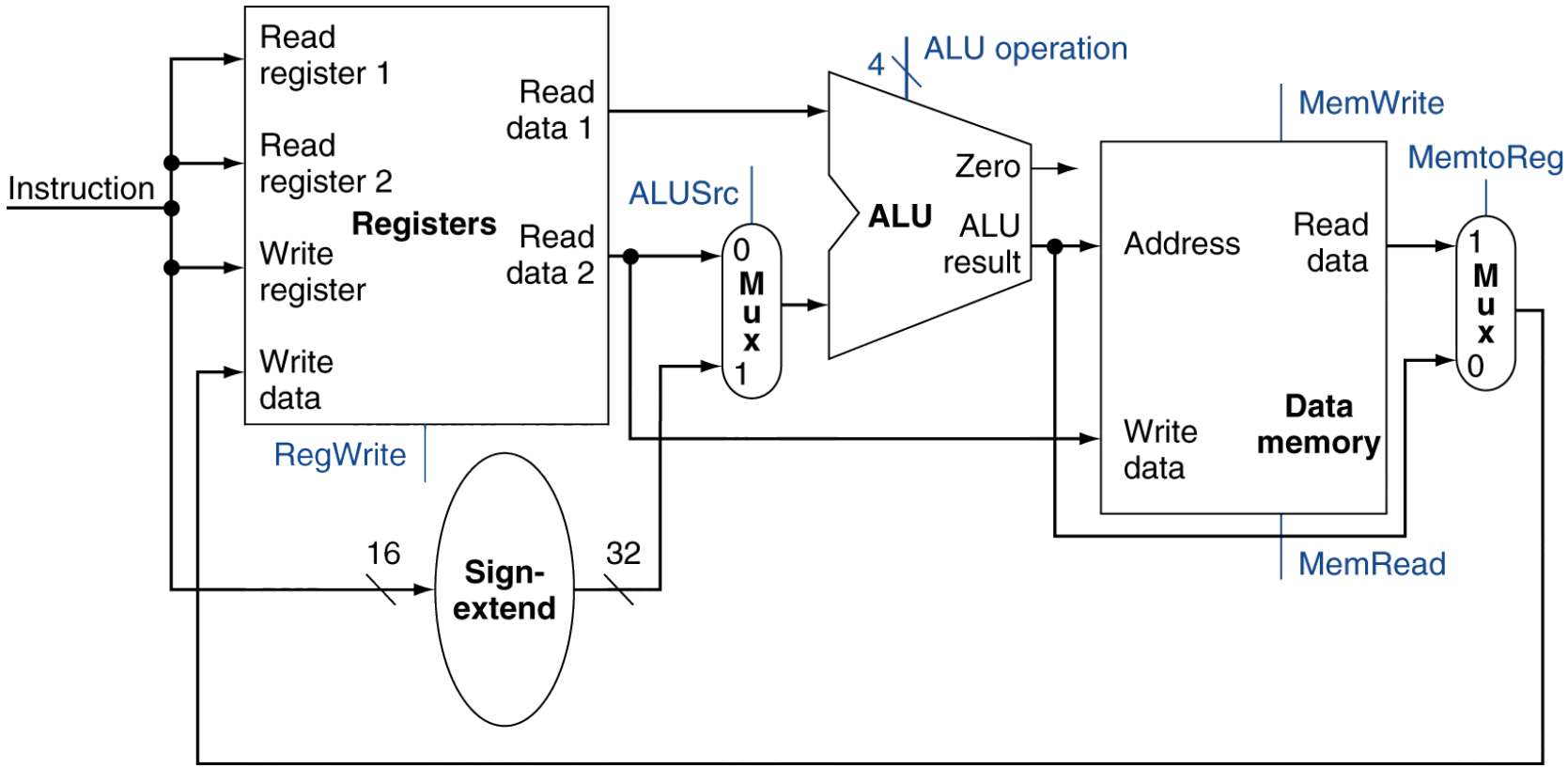
Composing the Elements

- **First-cut data path does an instruction in one clock cycle**
  - Each datapath element can only do one function at a time
  - Hence, we need separate instruction and data memories
- **Use multiplexers where alternate data sources are used for different instructions**

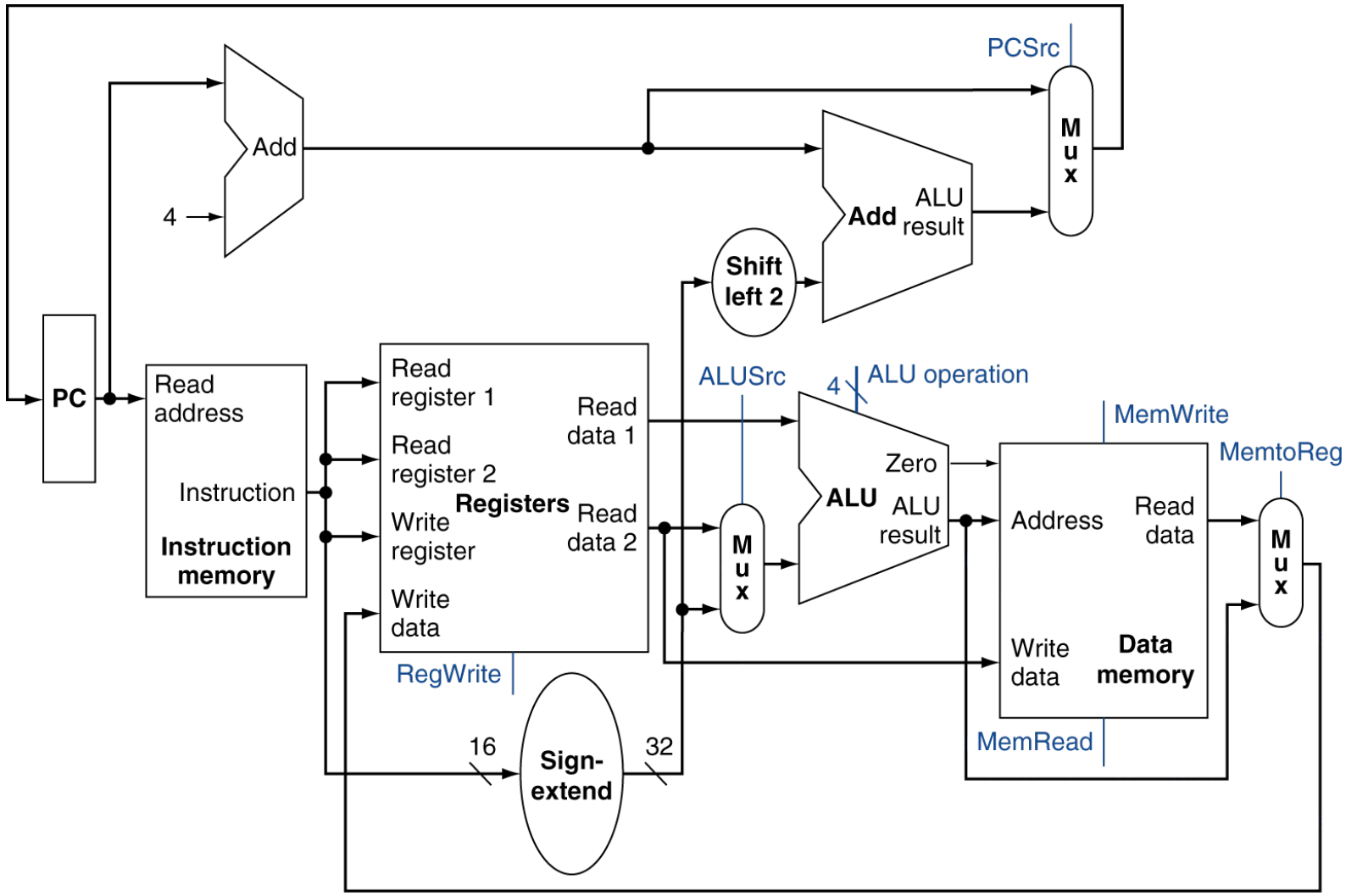
R-type/ Load-Store Datapath

- **Operations of arithmetic-logical (or R-type) instructions and the memory instructions are quite similar**
- **The arithmetic-logical instructions use the ALU, with the inputs coming from the two registers**
- **The memory instructions can also use the ALU to do the address calculation**
  - Second input is the sign-extended 16-bit off set field from the instruction
- **The value stored into a destination register comes from the ALU (for an R-type instruction) or the memory (for a load).**

R-Type/Load/Store Datapath

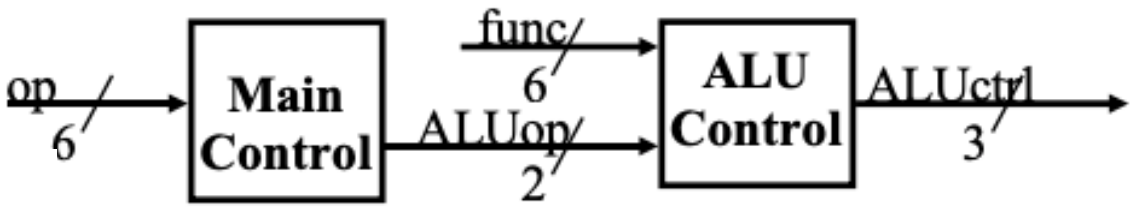


Full Datapath



ALU Control

- **ALU used for**
  - Load/Store: F = add
  - Branch (e.g., beq): F = subtract
  - R-type: F depends on funct field



ALU control	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set-on-less-than
1100	NOR

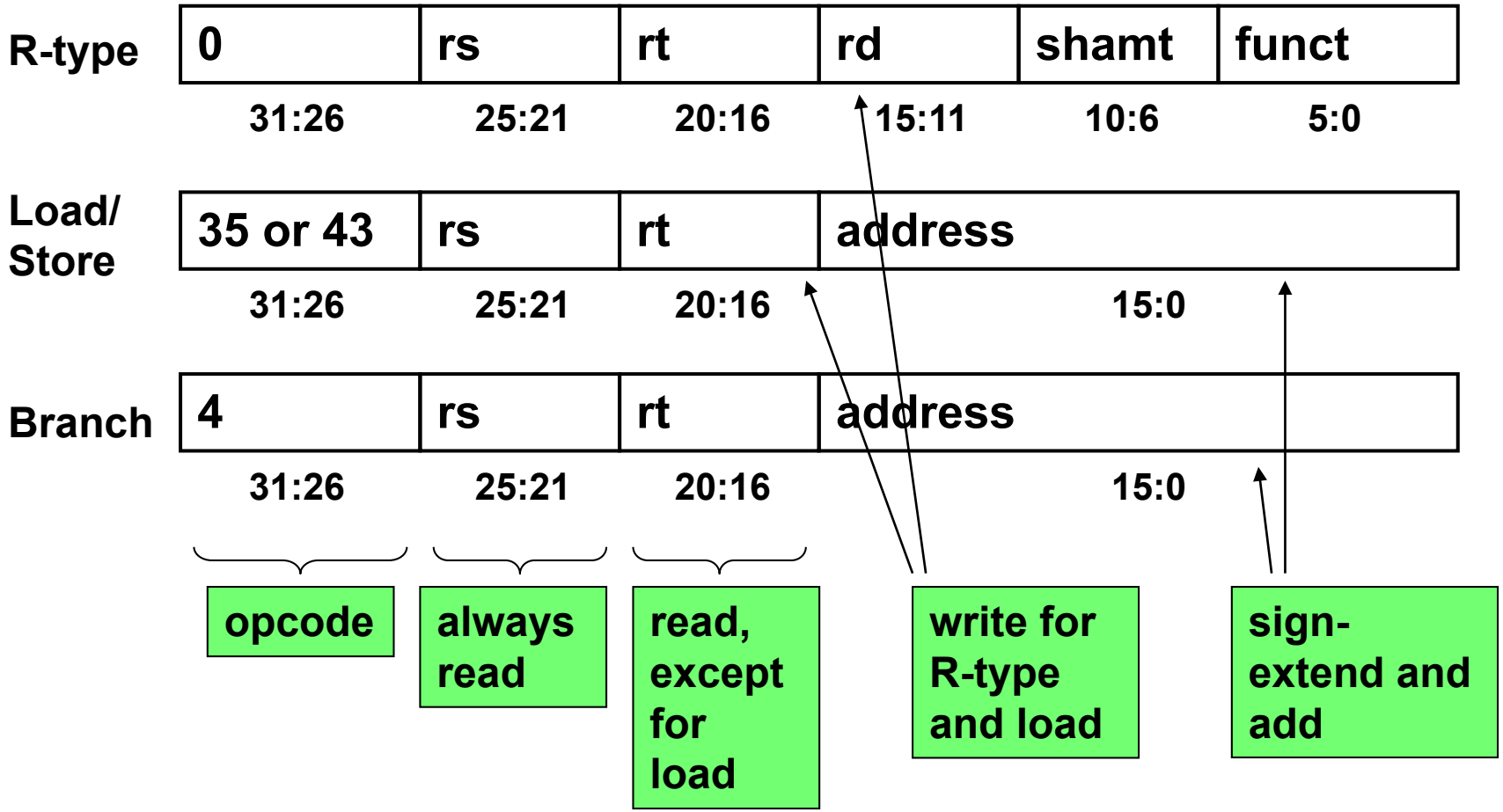
ALU Control

- **Assume 2-bit ALUOp derived from opcode**
  - Combinational logic derives ALU control

opcode	ALUOp	Operation	funct	ALU function	ALU control
lw	00	load word	XXXXXX	add	0010
sw	00	store word	XXXXXX	add	0010
beq	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
		subtract	100010	subtract	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		set-on-less-than	101010	set-on-less-than	0111

The Main Control Unit

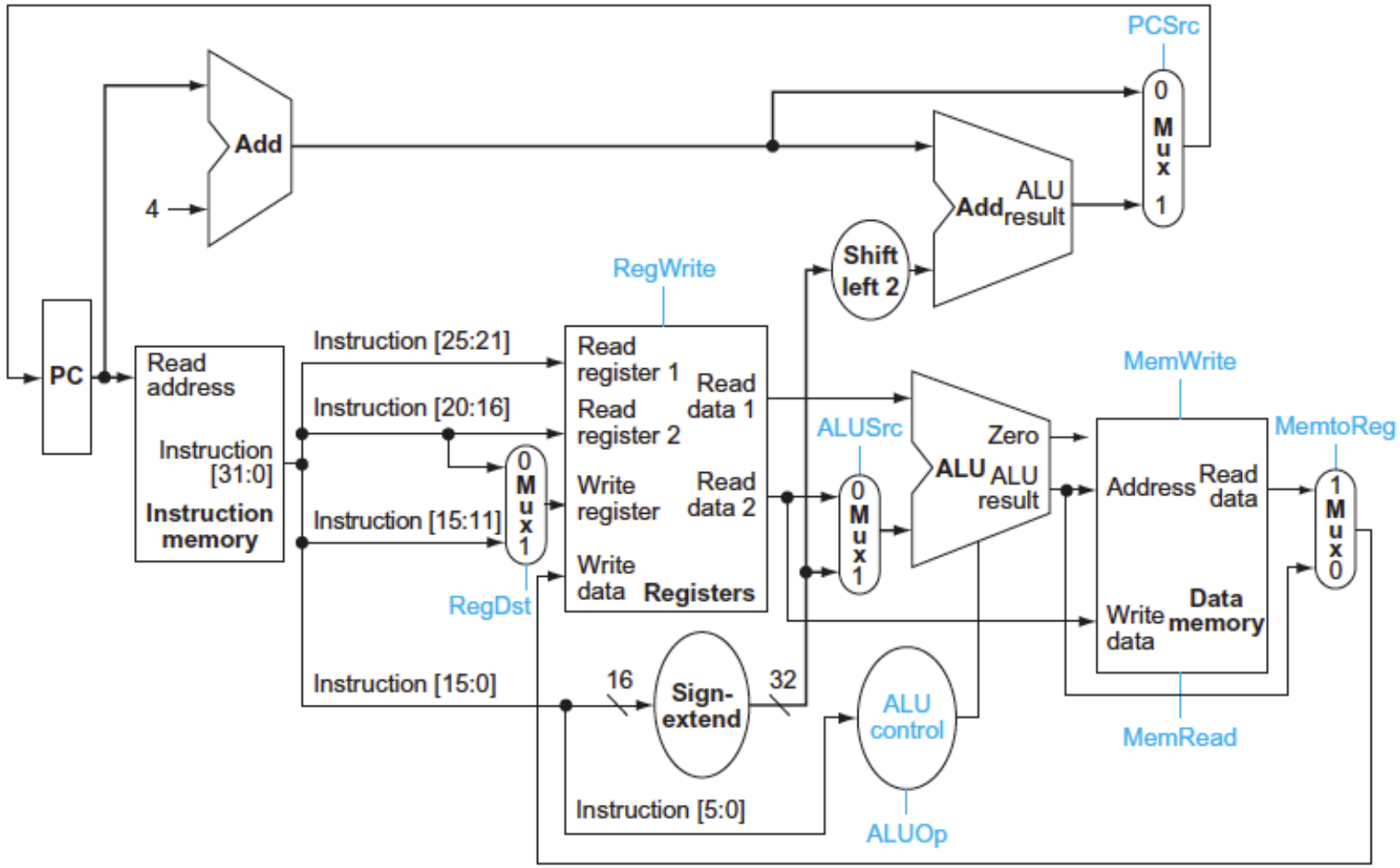
- Control signals derived from instruction



Observations

- The op field, which as we saw in Chapter 2 is called the opcode, is always contained in bits 31:26. We will refer to this field as Op[5:0]
- The two registers to be read are always specified by the rs and rt fields, at positions 25:21 and 20:16. This is true for the R-type instructions, branch equal, and store
- The base register for load and store instructions is always in bit positions 25:21 (rs)
- The 16-bit off set for branch equal, load, and store is always in positions 15:0
- The destination register is in one of two places. For a load it is in bit positions 20:16 (rt), while for an R-type instruction it is in bit positions 15:11 (rd). Th us, we will need to add a multiplexor to select which fi eld of the instruction is used to indicate the register number to be written

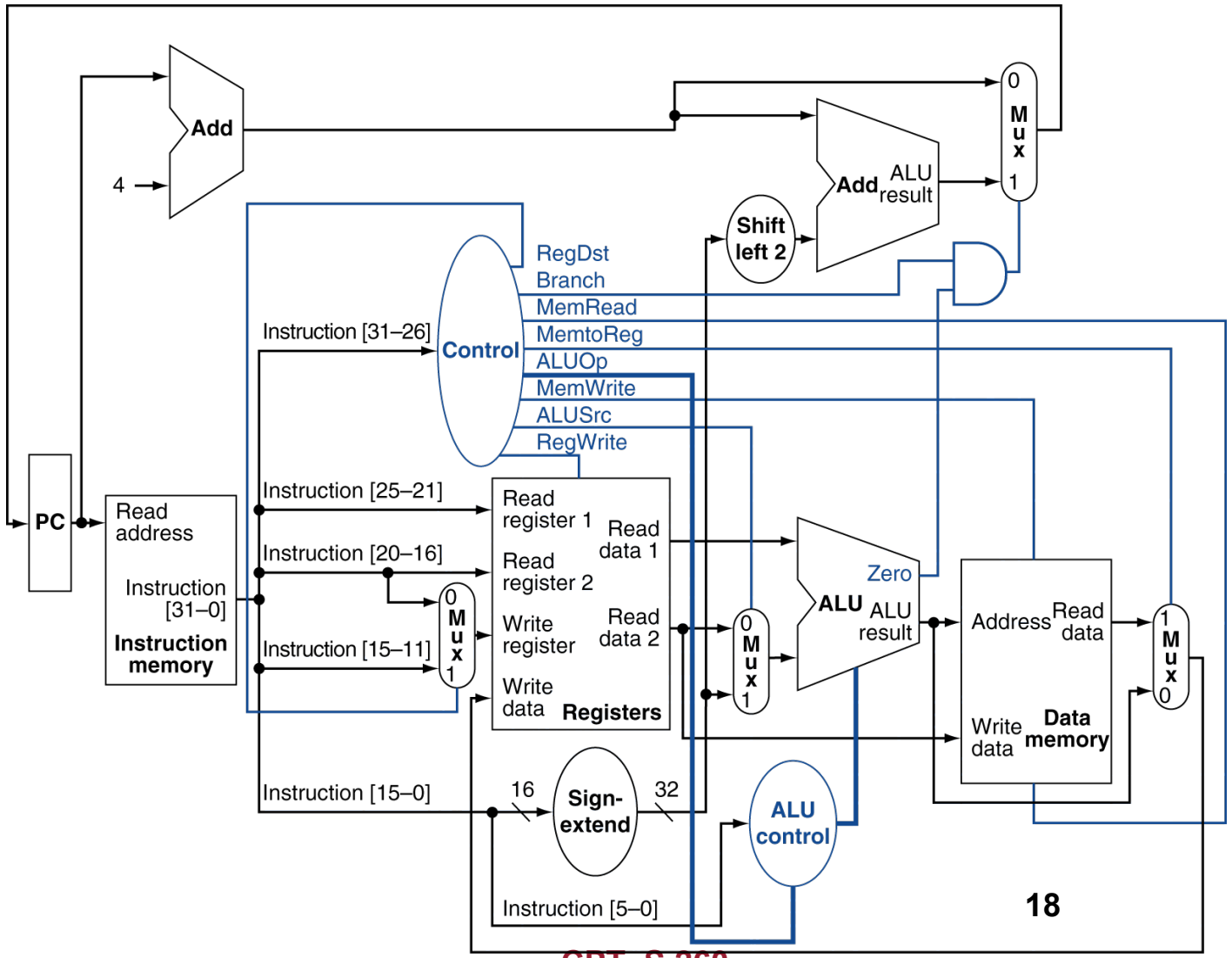
Control Signals



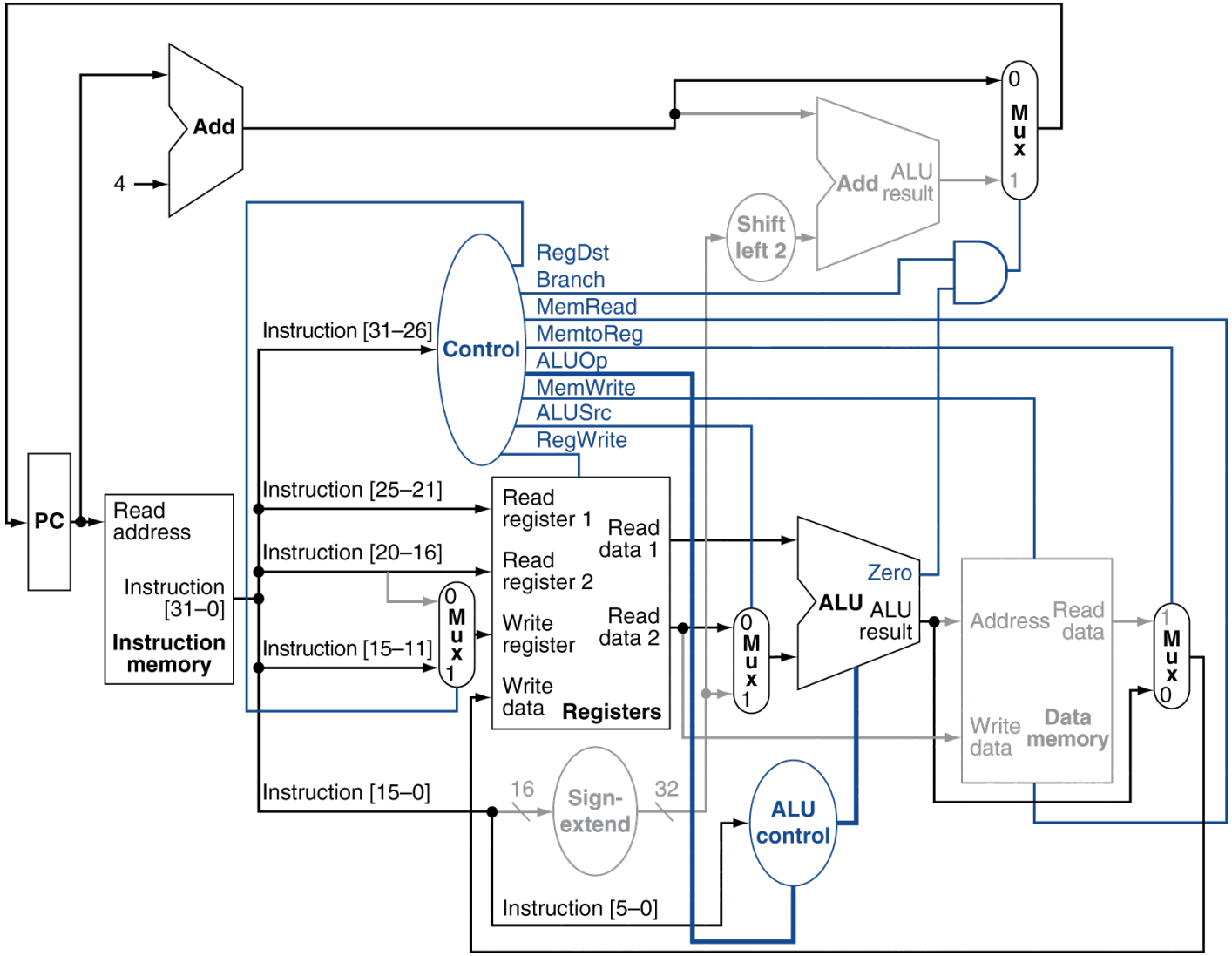
Effect of each Control Signal

Signal name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the Write register comes from the rt field (bits 20:16).	The register destination number for the Write register comes from the rd field (bits 15:11).
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, lower 16 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.

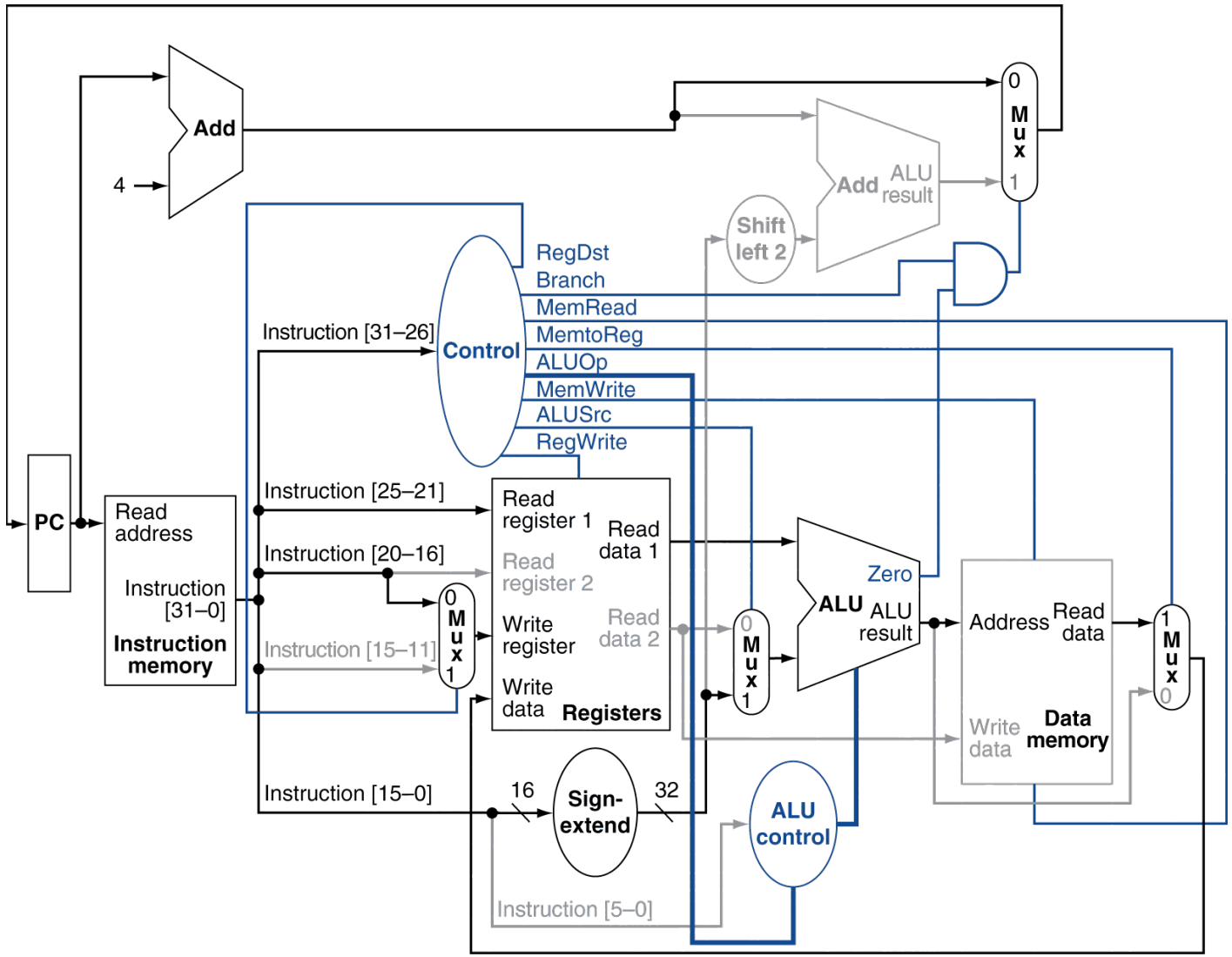
Datapath with Control



R-Type Instruction



Load Instruction



BEQ Instruction