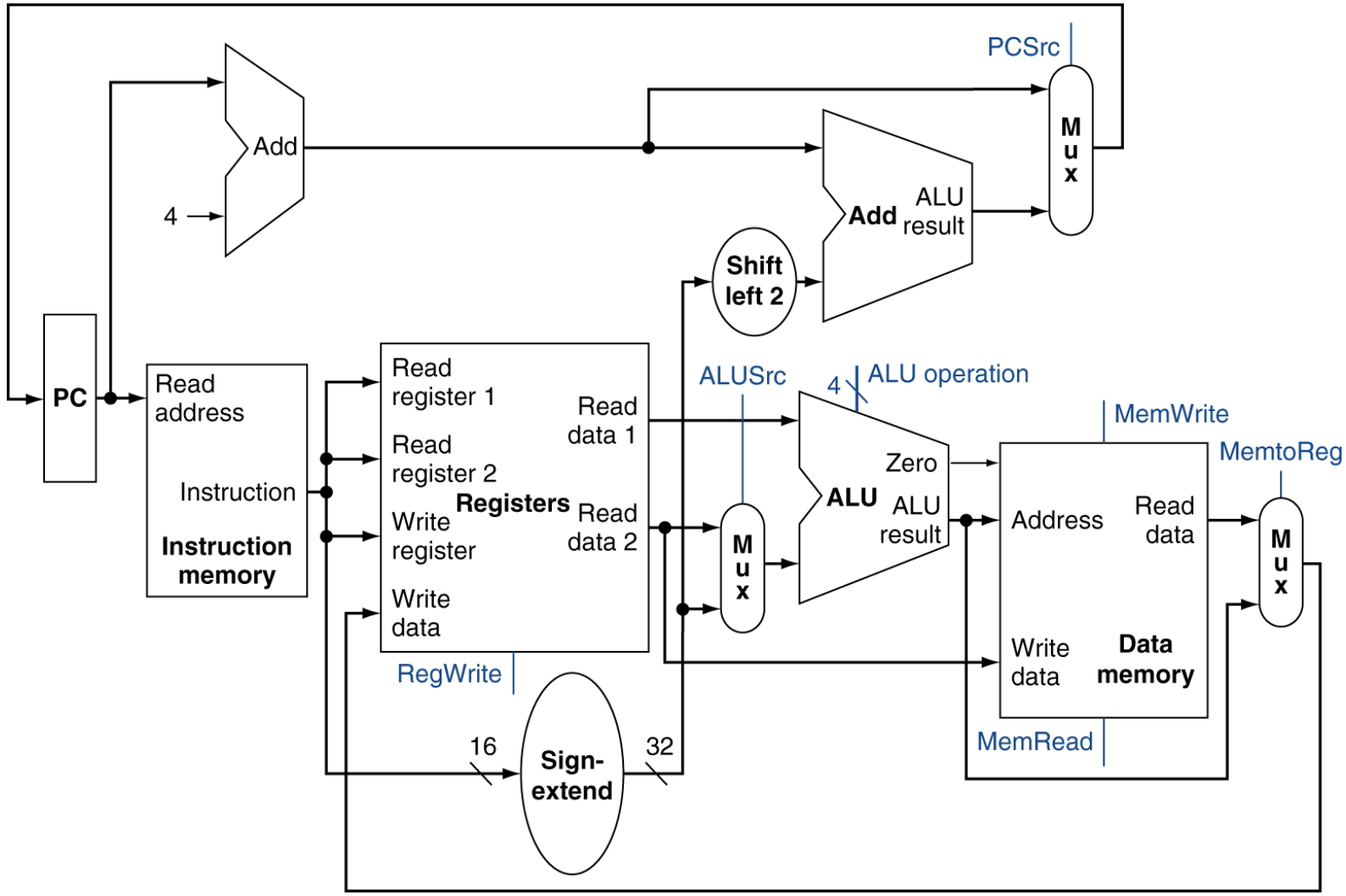
CSCI 210: Computer Architecture
Lecture 26: Control Path

Stephen Checkoway
Oberlin College
Apr. 27, 2022
Slides from Cynthia Taylor

Announcements

- Problem Set 8 due Friday
- Lab 7 due Sunday
- Office Hours Friday 13:30–14:30

Full Datapath So Far

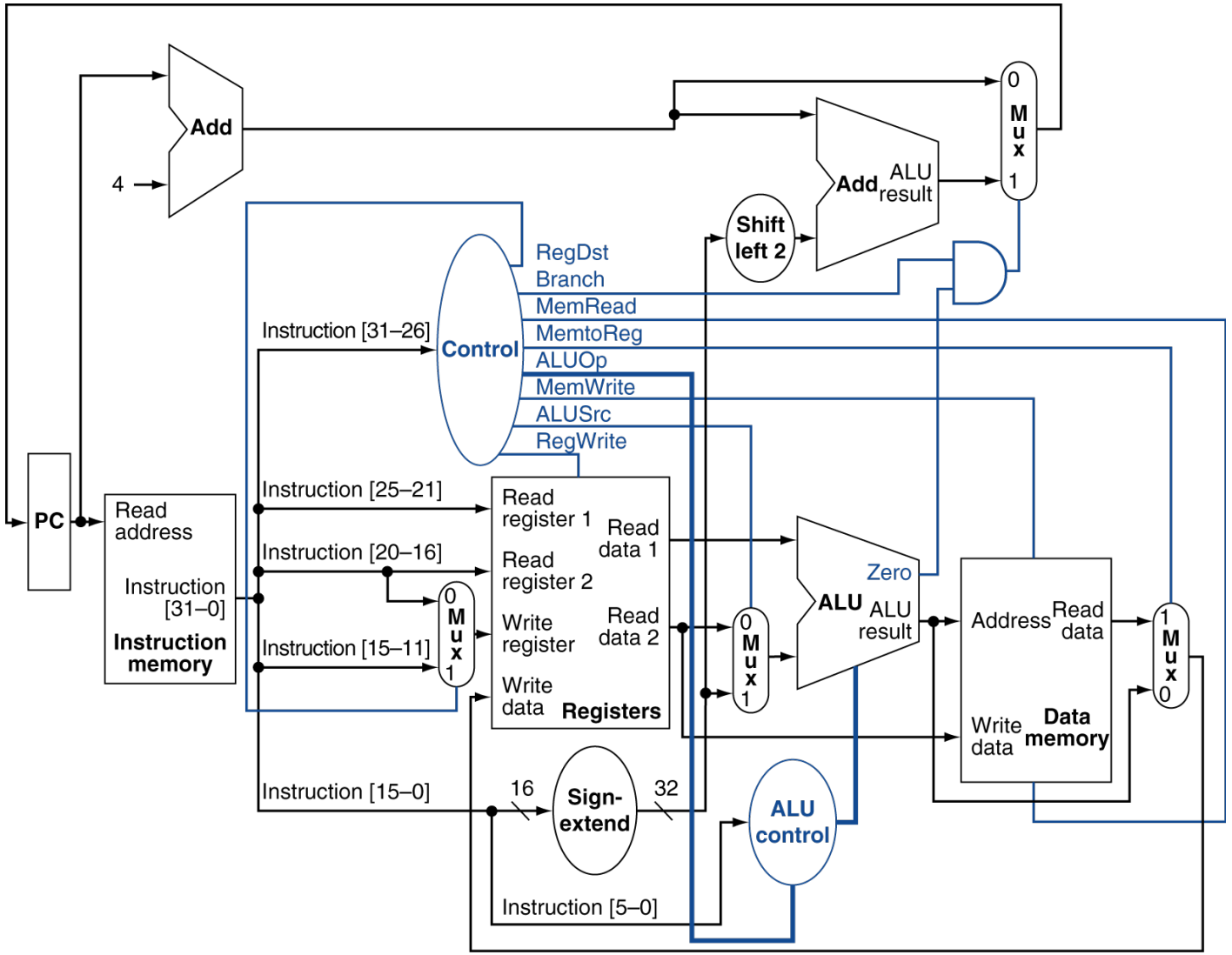


Control Path

- Our datapath is complicated, and we don’t use each element every time

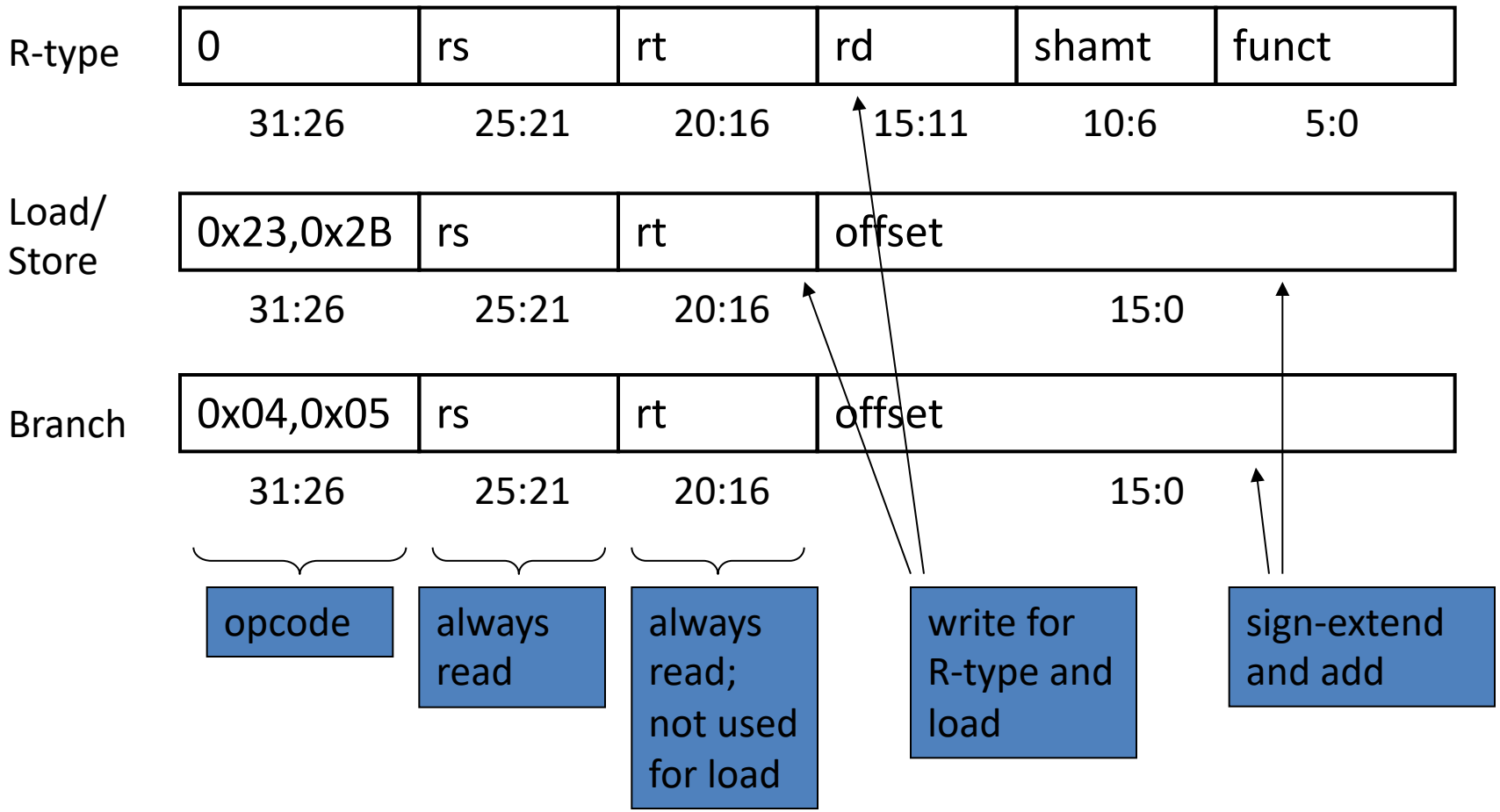
- How do we know which elements to use?

Datapath With Control



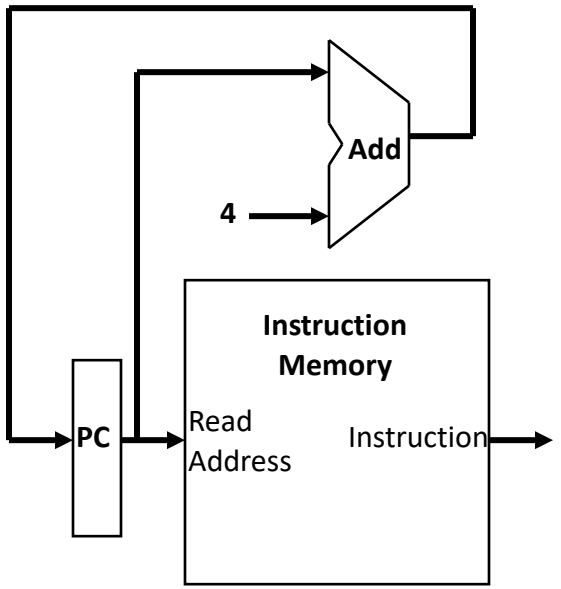
The Main Control Unit

Control signals derived from instruction

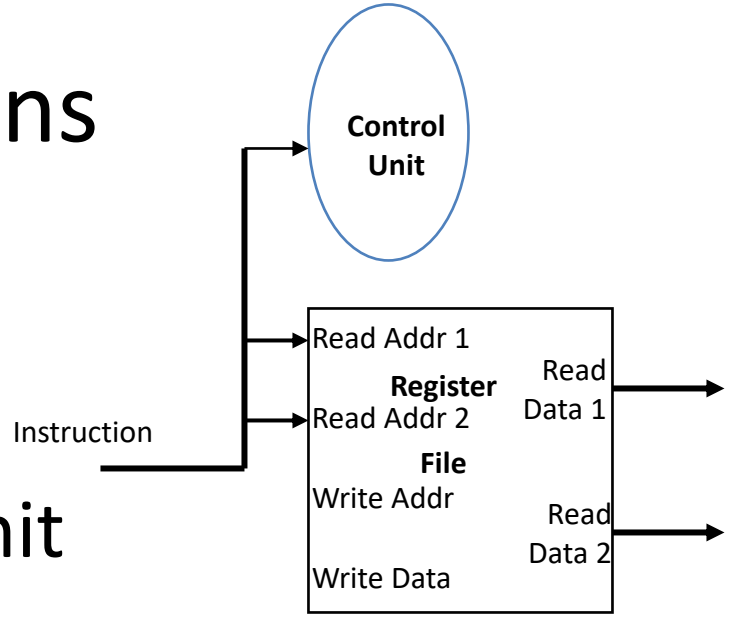


Fetching Instructions

- Read instruction from Instruction Memory
- Updating PC value to address of next (sequential) instruction
- PC is updated every clock cycle, so it does not need an explicit write control signal just a clock signal
- Read from memory each time, so we don’t need an explicit control signal



Decoding Instructions



- Send fetched instruction's opcode to the main control unit

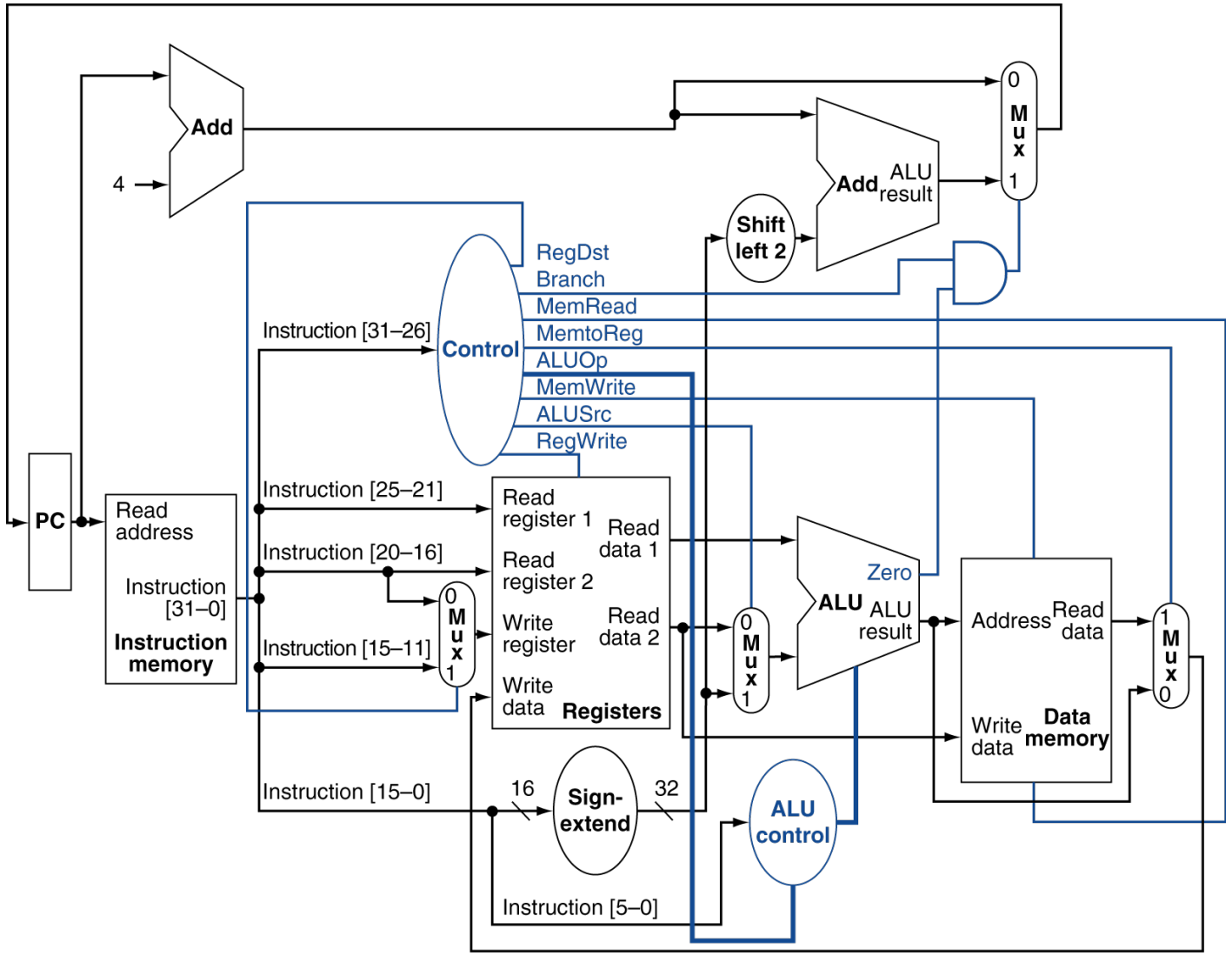
R-type	0	rs	rt	rd	shamt	funct
	31:26	25:21	20:16	15:11	10:6	5:0
Load/ Store	35 or 43	rs	rt	address		
	31:26	25:21	20:16	15:0		

- Read two values from the Register File
- Register File addresses are contained in the instruction

After decode

After reading opcode

- Produce most control signals
- Includes the ALUOp control signal—which goes to the ALU control unit—and the ALUSrc control signal which selects the ALU's second operand



ALU Control Unit

- Combinational logic derives 2-bit ALUOp signal from opcode
- ALU Control Unit takes ALUOp and instruction funct field as inputs and derives a 4-bit ALU control signal

opcode	ALUOp	Operation	ALU function
lw	00	load word	add
sw	00	store word	add
beq	01	branch equal	subtract
R-type	10	arithmetic/logic	depends on funct

For load/store, our ALU operation will be

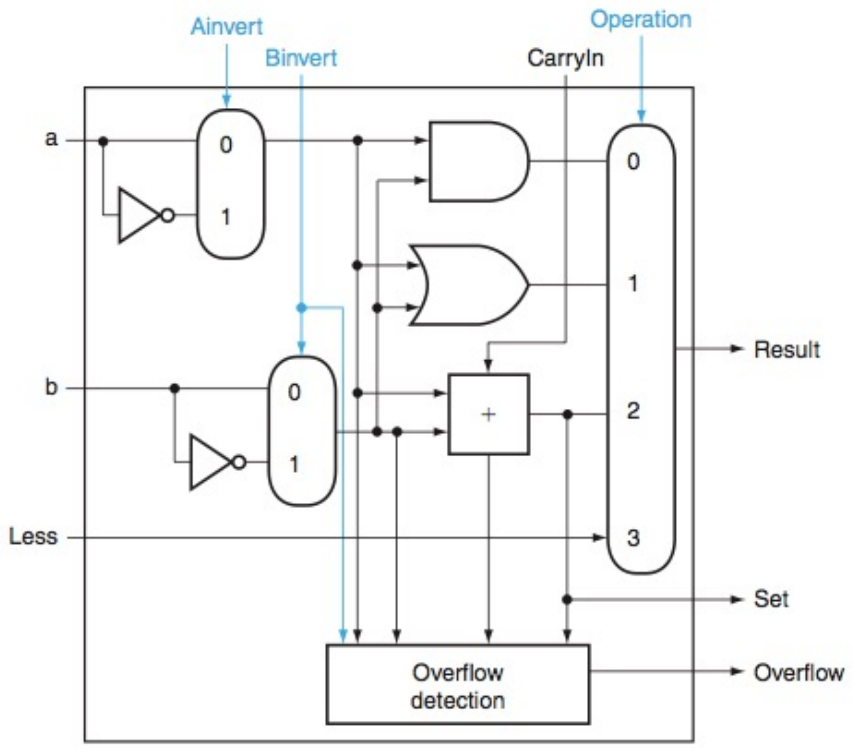
A. Add

B. And

C. Set less than

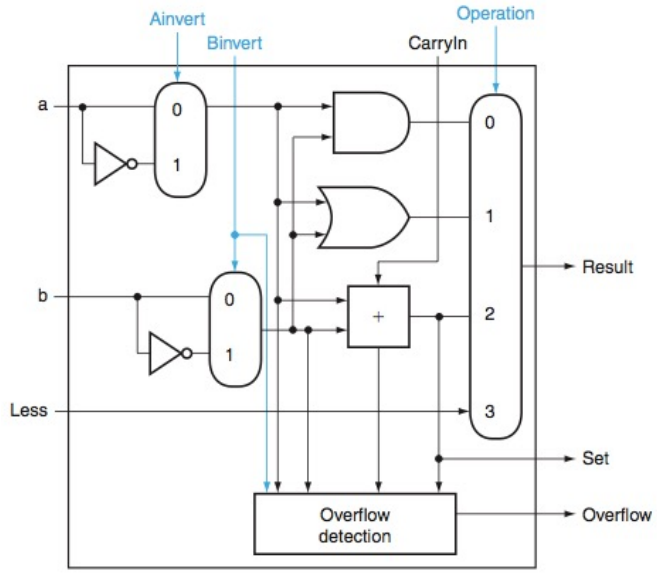
D. Subtract

E. None of the above



ALU Control

- ALU used for
  - Load/Store: op = add
  - Branch: op = subtract
  - R-type: op depends on funct field



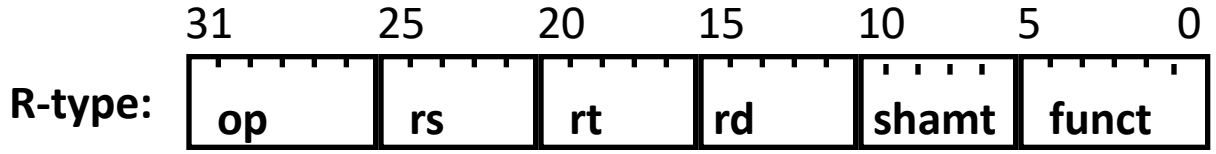
ALU control	Function	Ainvert	Binvert/CarryIn0	Operation
0000	AND	0	0	00
0001	OR	0	0	01
0010	add	0	0	10
0110	subtract	0	1	10
0111	set-on-less-than	0	1	11
1100	NOR	1	1	00

ALU Control

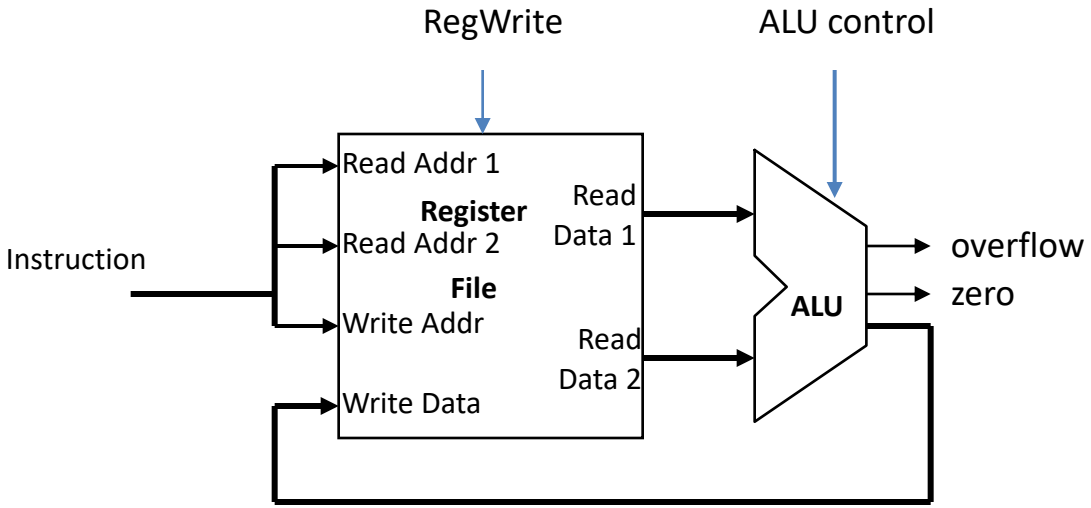
opcode	Instruction	ALUOp	funct	ALU function	ALU control
lw	load word	00	XXXXXX	add	0010
sw	store word	00	XXXXXX	add	0010
beq	branch equal	01	XXXXXX	subtract	0110
R-type	add	10	100000	add	0010
	subtract		100010	subtract	0110
	AND		100100	AND	0000
	OR		100101	OR	0001
	set-on-less-than		101010	set-on-less-than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

Executing R Format Operations

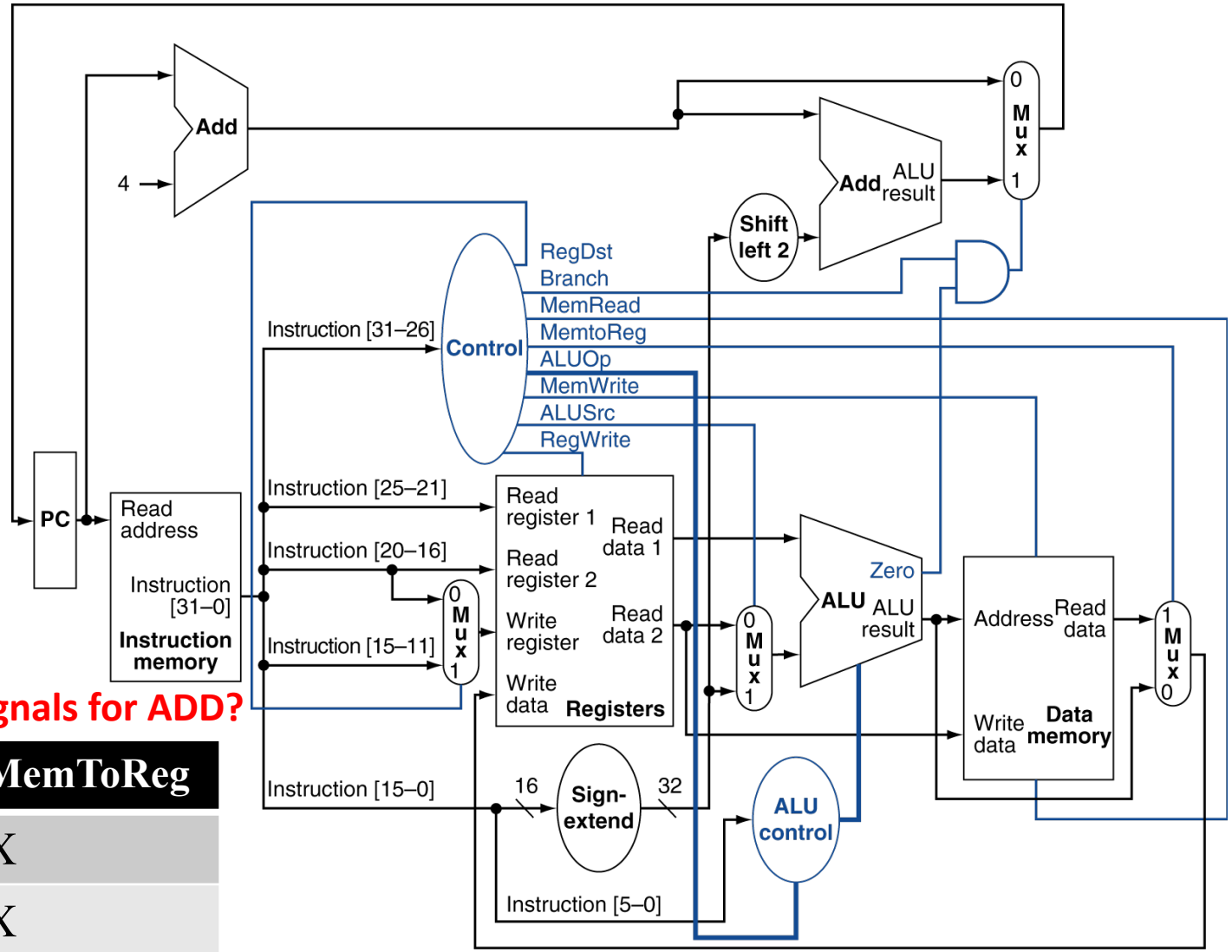
- R format operations (`add, sub, slt, and, or`)



  - perform operation (specified by funct) on values in rs and rt
  - store the result back into the Register File (into location rd)



Note that Register File is not written every cycle (e.g., **`sw`**), so we need an explicit write control signal for the Register File



instruction control signals for ADD?

Select	RegDst	MemToReg
A	0	X
B	1	X
C	0	1
D	1	0
E	None of the above	

Reading

- Next lecture: Pipelining
  - Section 5.6

- Problem Set 8 due Friday

- Lab 7 due Sunday