CSE 112: Computer Organization

Instructor: Sujay Deb

Lecture 11

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

AMS Lab @IIITD

Single-Cycle
RISC-V Processor

Example Program

- Design datapath
- View example program executing

Example Program:

Address	Instruction	Type	Fields						Machine Language
			imm$_{11:0}$		rs1	f3	rd	op	
0x1000	L7: lw x6, -4(x9)	I	111111111100		01001	010	00110	0000011	FFC4A303
			imm$_{11:5}$	rs2	rs1	f3	imm$_{4:0}$	op	
0x1004	sw x6, 8(x9)	S	0000000	00110	01001	010	01000	0100011	0064A423
			funct7	rs2	rs1	f3	rd	op	
0x1008	or x4, x5, x6	R	0000000	00110	00101	110	00100	0110011	0062E233
			imm$_{12,10:5}$	rs2	rs1	f3	imm$_{4:1,11}$	op	
0x100C	beq x4, x4, L7	B	1111111	00100	00100	000	10101	1100011	FE420AE3

Single-Cycle RISC-V Processor

- **Datapath:** start with `lw` instruction
- **Example:** `lw x6, -4(x9)`

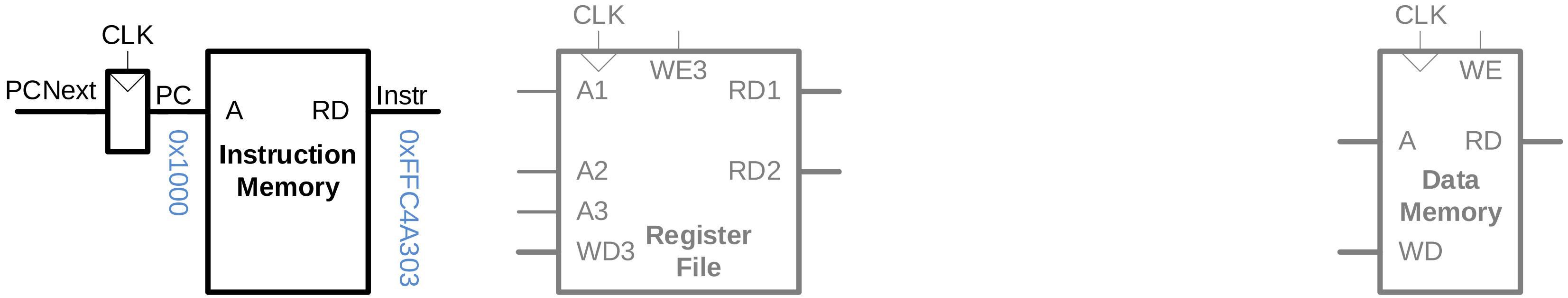
  `lw rd, imm(rs1)`

I-Type

31:20	19:15	14:12	11:7	6:0
$\text{imm}_{11:0}$	rs1	funct3	rd	op
12 bits	5 bits	3 bits	5 bits	7 bits

Single-Cycle Datapath: `lw` fetch

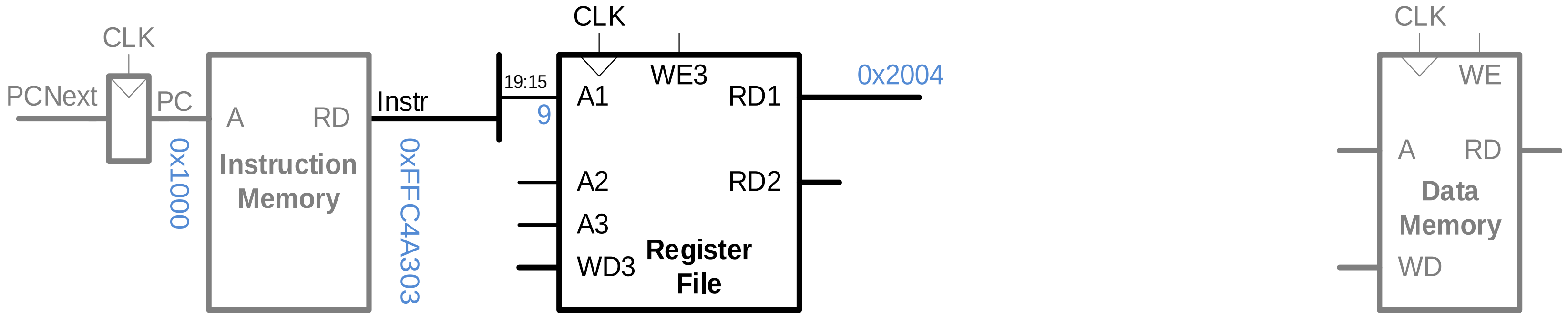
STEP 1: Fetch instruction



Address	Instruction	Type	Fields				Machine Language	
			imm$_{11:0}$	rs1	f3	rd	op	
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011	FFC4A303

Single-Cycle Datapath: `lw` Reg Read

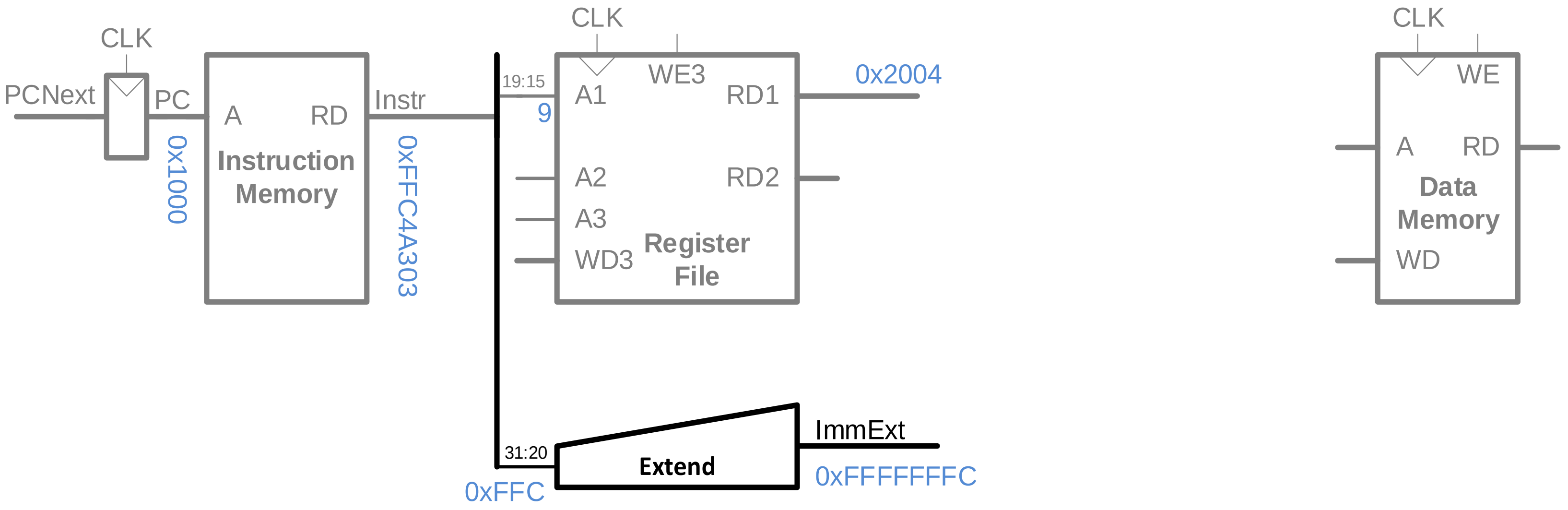
STEP 2: Read source operand (**rs1**) from RF



Address	Instruction	Type	Fields				Machine Language	
			$\text{imm}_{11:0}$	rs1	f3	rd	op	
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011	FFC4A303

Single-Cycle Datapath: `lw` Immediate

STEP 3: Extend the immediate

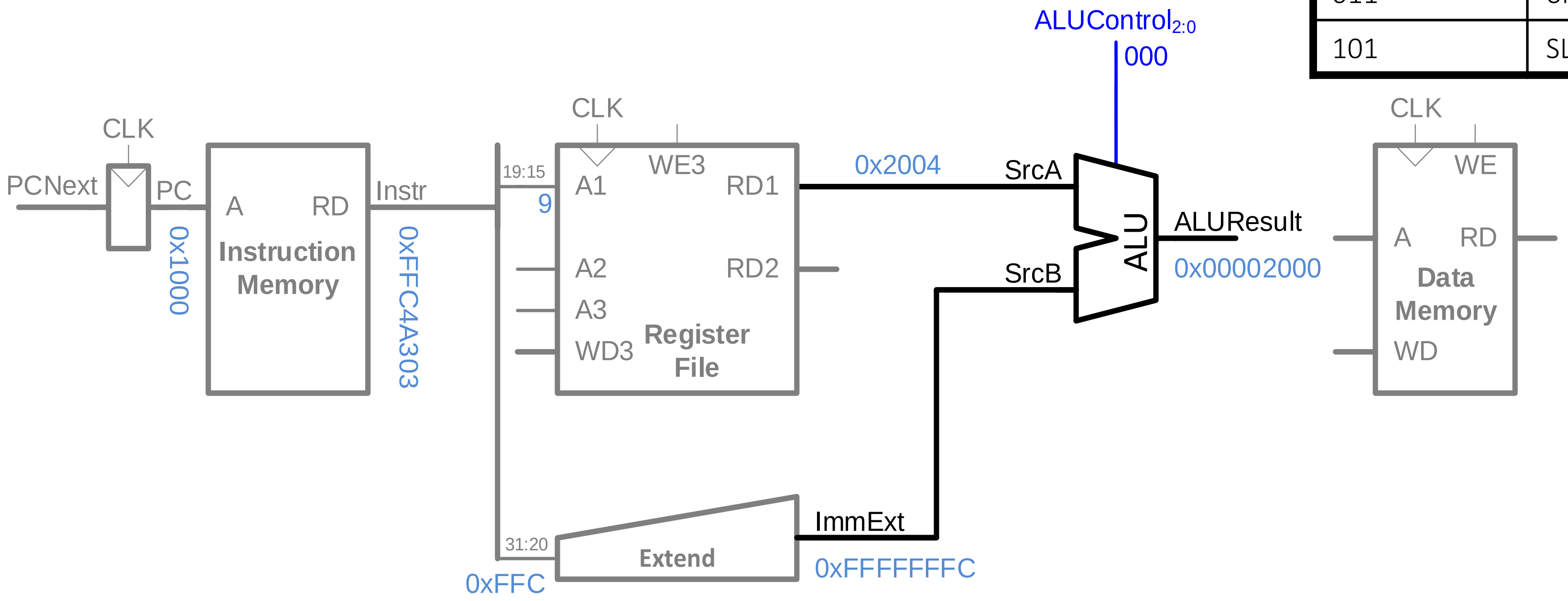


Address	Instruction	Type	Fields					Machine Language
			$imm_{11:0}$	rs1	f3	rd	op	
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011	FFC4A303

Single-Cycle Datapath: `lw` Address

STEP 4: Compute the memory address

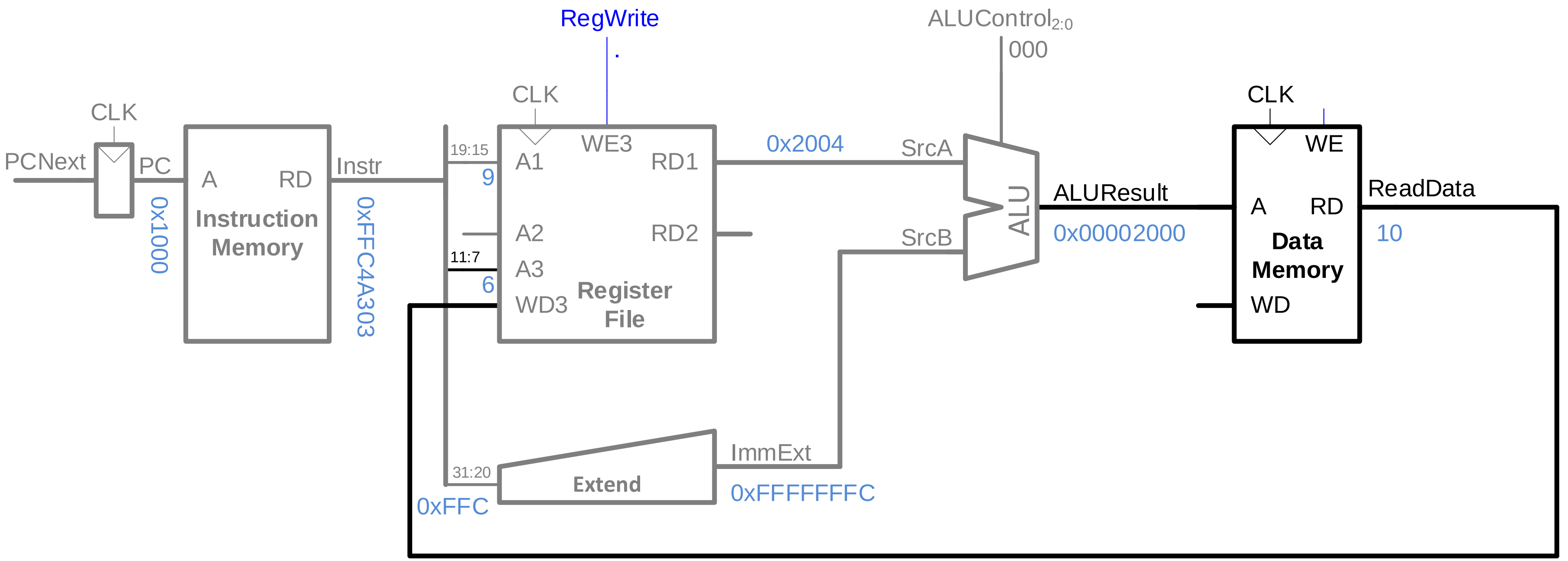
ALUControl$_{2:0}$	Function
000	add
001	subtract
010	and
011	or
101	SLT



Address	Instruction	Type	Fields					Machine Language
			imm$_{11:0}$	rs1	f3	rd	op	
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011	FFC4A303

Single-Cycle Datapath: `lw` Mem Read

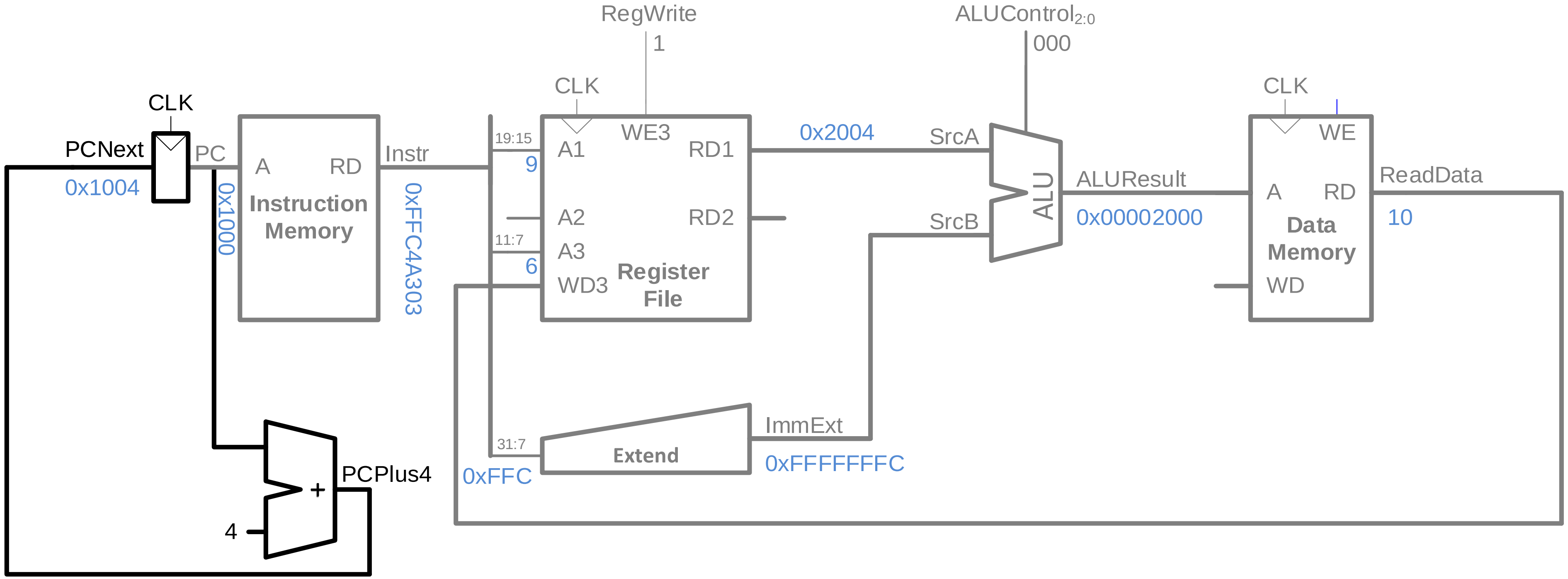
STEP 5: Read data from memory and write it back to register file



Address	Instruction	Type	Fields					Machine Language
			imm$_{11:0}$	**rs1**	**f3**	**rd**	**op**	
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011	FFC4A303

Single-Cycle Datapath: PC Increment

STEP 6: Determine address of next instruction

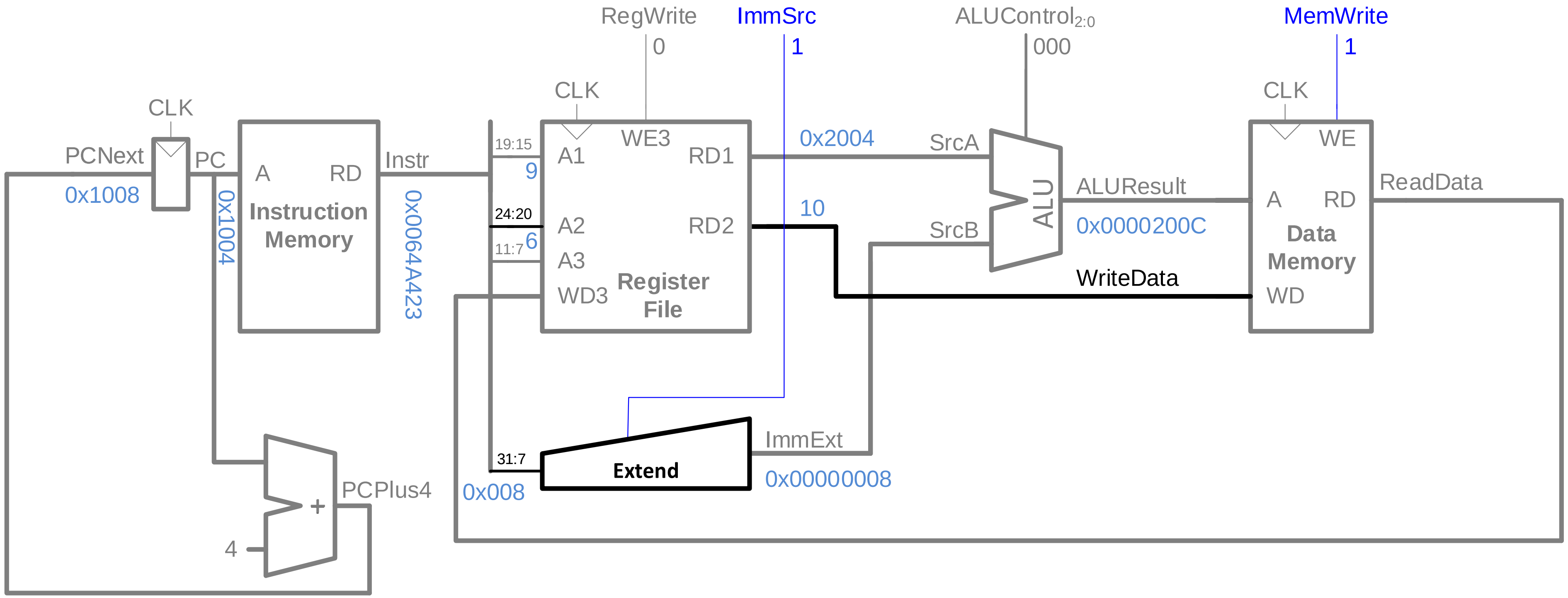


Address	Instruction	Type	Fields					Machine Language
			imm$_{11:0}$	**rs1**	**f3**	**rd**	**op**	
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011	FFC4A303

Single-Cycle Datapath: Other Instructions

Single-Cycle Datapath: `sw`

- **Immediate:** now in {instr[31:25], instr[11:7]}
- **Add control signals:** ImmSrc, MemWrite



Address	Instruction	Type	Fields						Machine Language
			imm$_{11:5}$	rs2	rs1	f3	imm$_{4:0}$	op	
0x1004	sw x6, 8(x9)	S	0000000	00110	01001	010	01000	0100011	0064A423

Single-Cycle Datapath: Immediate

ImmSrc	ImmExt	Instruction Type
0	{{20{instr[31]}}, **instr[31:20]**}	I-Type
1	{{20{instr[31]}}, **instr[31:25], instr[11:7]**}	S-Type

I-Type

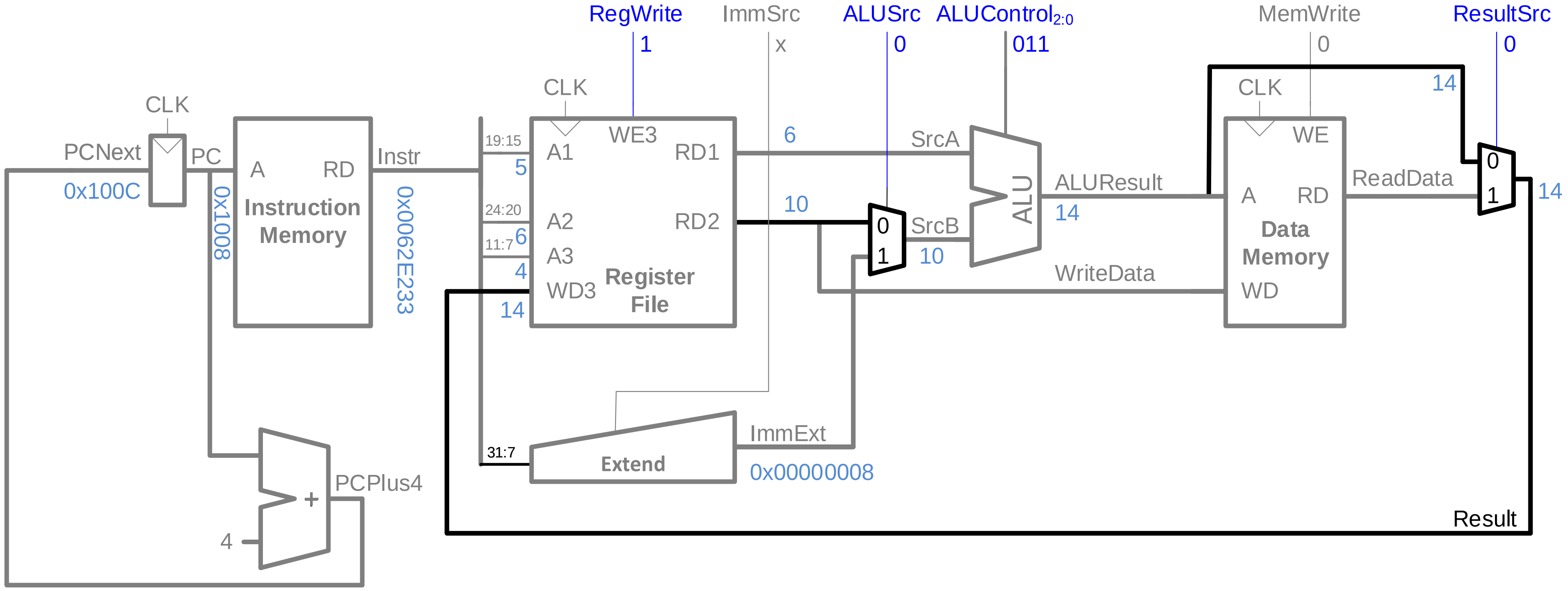
31:20	19:15	14:12	11:7	6:0
imm$_{11:0}$	rs1	funct3	rd	op
12 bits	5 bits	3 bits	5 bits	7 bits

S-Type

31:25	24:20	19:15	14:12	11:7	6:0
imm$_{11:5}$	rs2	rs1	funct3	**imm$_{4:0}$**	op
7 bits	5 bits	5 bits	3 bits	5 bits	7 bits

Single-Cycle Datapath: R-type

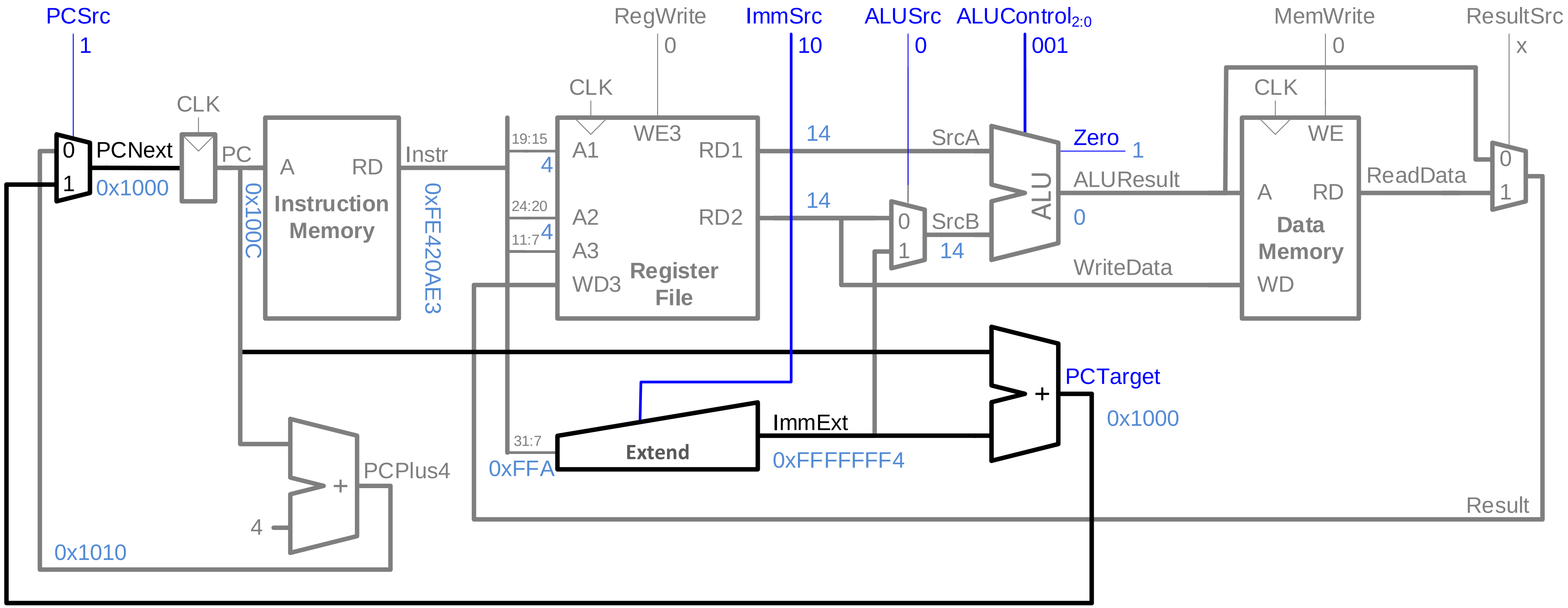
- Read from **rs1** and **rs2** (instead of **imm**)
- Write *ALUResult* to **rd**



Address	Instruction	Type	Fields						Machine Language
			funct7	rs2	rs1	f3	rd	op	
0x1008	or x4, x5, x6	R	0000000	00110	00101	110	00100	0110011	0062E233

Single-Cycle Datapath: beq

Calculate **target address:** PCTarget = PC + imm



Address	Instruction	Type	Fields						Machine Language
			imm$_{12,10:5}$	rs2	rs1	f3	imm$_{4:1,11}$	op	
0x100C	beq x4, x4, L7	B	1111111	00100	00100	000	10101	1100011	FE420AE3

Single-Cycle Datapath: ImmExt

$\text{ImmSrc}_{1:0}$	ImmExt	Instruction Type
00	{{20{instr[31]}}, **instr[31:20]**}	I-Type
01	{{20{instr[31]}}, **instr[31:25], instr[11:7]**}	S-Type
10	{{19{instr[31]}}, **instr[31], instr[7],** **instr[30:25], instr[11:8], 1’b0**}	B-Type

I-Type

31:20	19:15	14:12	11:7	6:0
$\text{imm}_{11:0}$	rs1	funct3	rd	op
12 bits	5 bits	3 bits	5 bits	7 bits

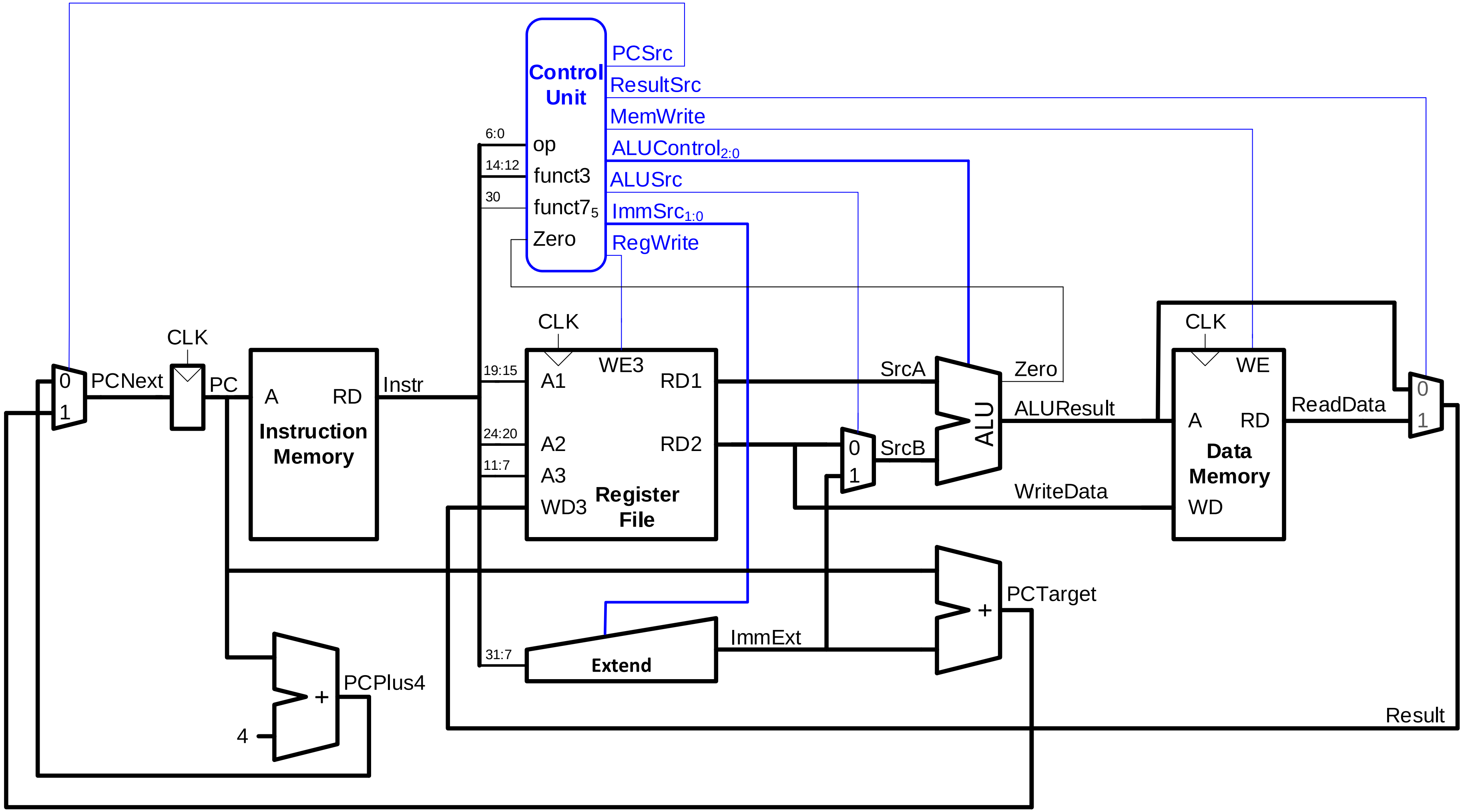
S-Type

31:25	24:20	19:15	14:12	11:7	6:0
$\text{imm}_{11:5}$	rs2	rs1	funct3	**$\text{imm}_{4:0}$**	op
7 bits	5 bits	5 bits	3 bits	5 bits	7 bits

B-Type

31:25	24:20	19:15	14:12	11:7	6:0
$\text{imm}_{12,10:5}$	rs2	rs1	funct3	**$\text{imm}_{4:1,11}$**	op
7 bits	5 bits	5 bits	3 bits	5 bits	7 bits

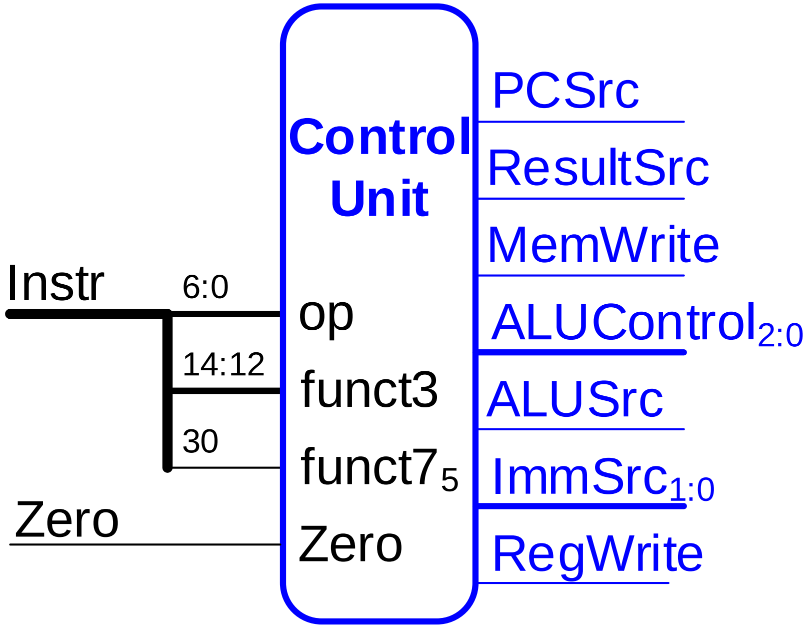
Single-Cycle RISC-V Processor



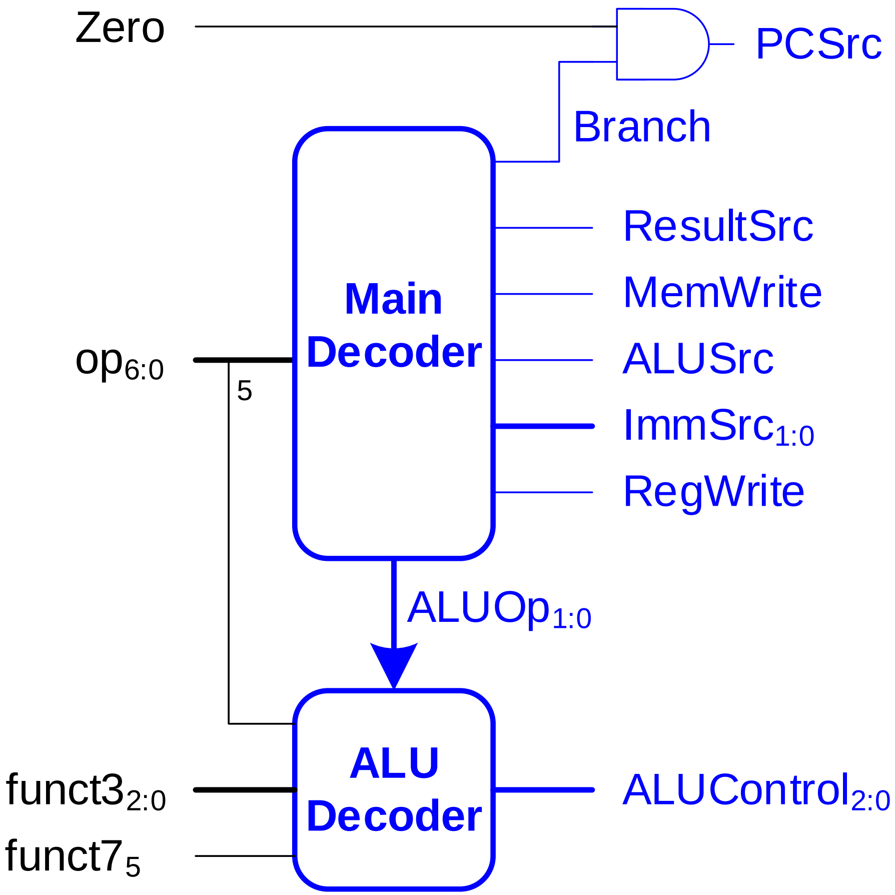
Single-Cycle Control

Single-Cycle Control

High-Level View

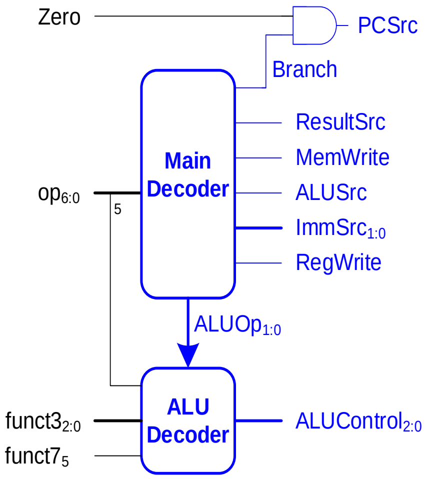


Low-Level View



Single-Cycle Control: Main Decoder

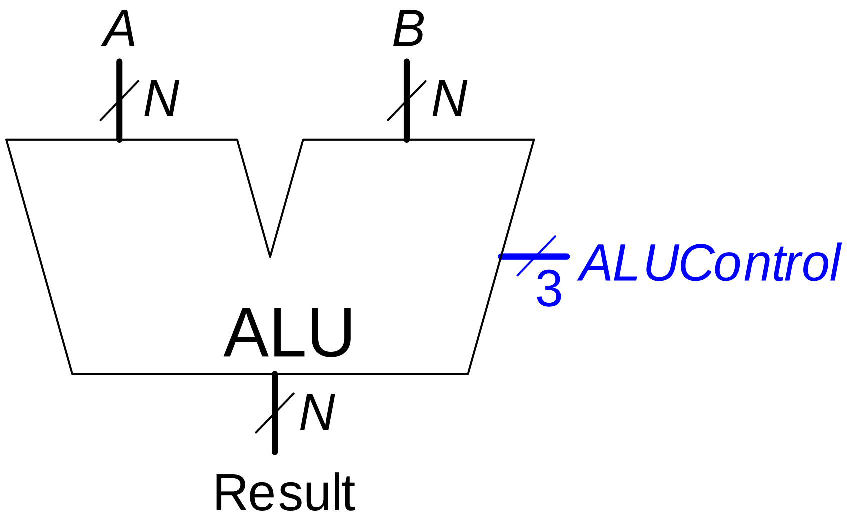
op	Instr.	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp
3	**lw**							
35	**sw**							
51	R-type							
99	**beq**							



Class Interaction # 13

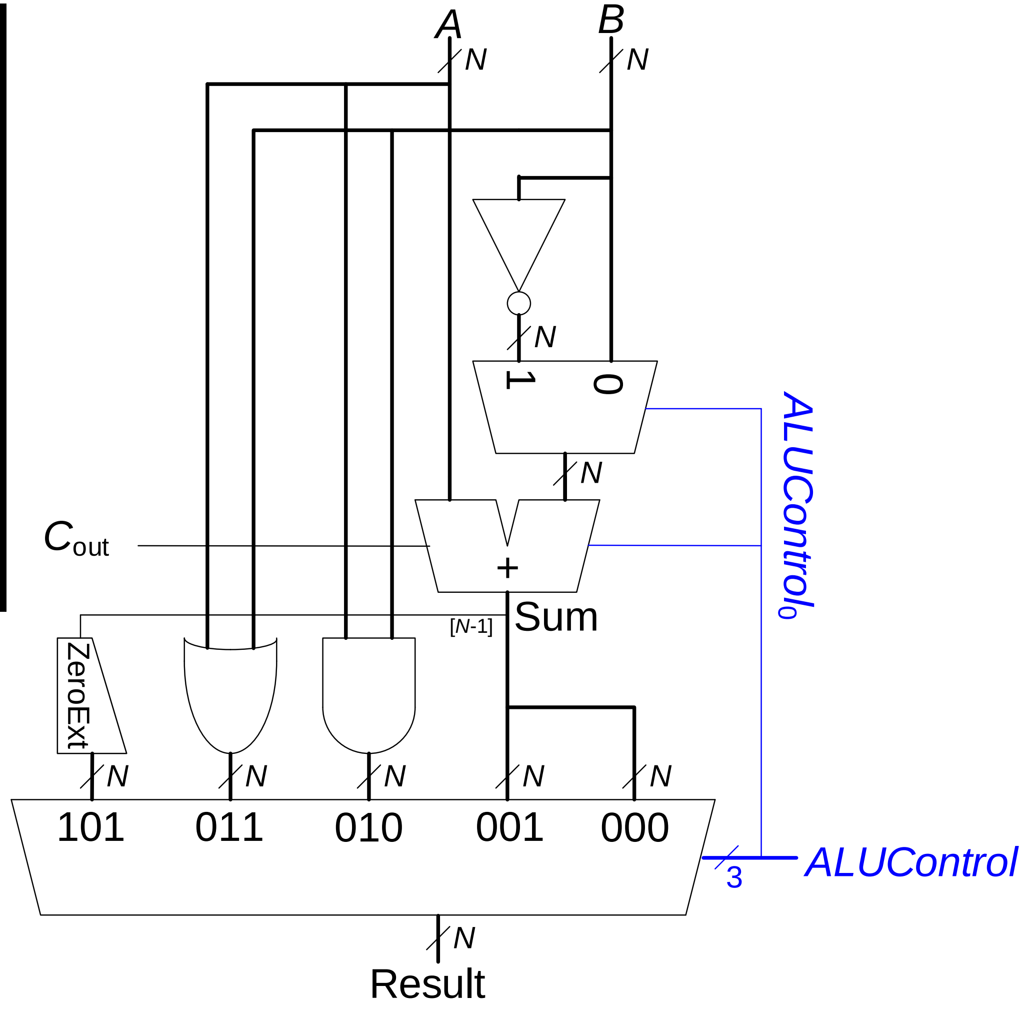
Review: ALU

$\text{ALUControl}_{2:0}$	Function
000	add
001	subtract
010	and
011	or
101	SLT



Review: ALU

$ALUControl_{2:0}$	Function
000	add
001	subtract
010	and
011	or
101	SLT


A
N
B
N
N
1
0
ALUControl₀
N
Cout
+
[N-1]
Sum
ZeroExt
N
N
N
N
N
101
011
010
001
000
3
ALUControl
N
Result