Chapter 7: Microarchitecture

Multicycle RISC-V Processor

Single- vs. Multicycle Processor

- **Single-cycle:**
  - + simple
  - - cycle time limited by longest instruction (`lw`)
  - - separate memories for instruction and data
  - - 3 adders/ALUs
- **Multicycle processor** addresses these issues by breaking instruction into **shorter steps**
  - o shorter instructions take fewer steps
  - o can re-use hardware
  - o cycle time is faster

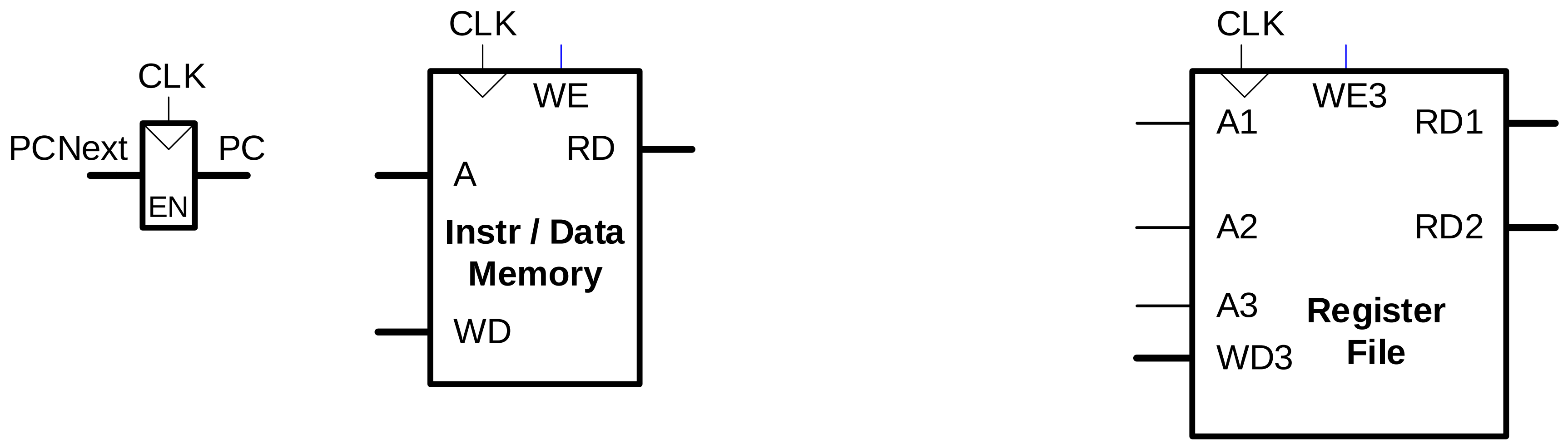
Single- vs. Multicycle Processor

- **Single-cycle:**
  - + simple
  - - cycle time limited by longest instruction (`lw`)
  - - separate memories for instruction and data
  - - 3 adders/ALUs
- **Multicycle:**
  - + higher clock speed
  - + simpler instructions run faster
  - + reuse expensive hardware on multiple cycles
  - - sequencing overhead paid many times

Same design steps as single-cycle:
- **first datapath**
- **then control**

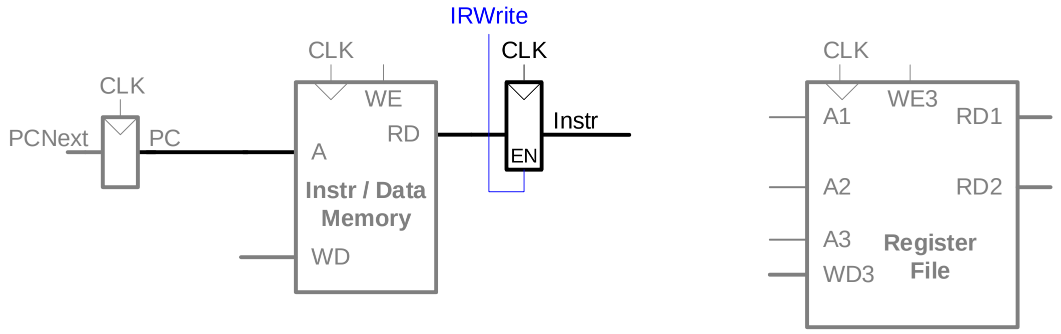
Multicycle State Elements

Replace separate Instruction and Data memories with a single unified memory – more realistic



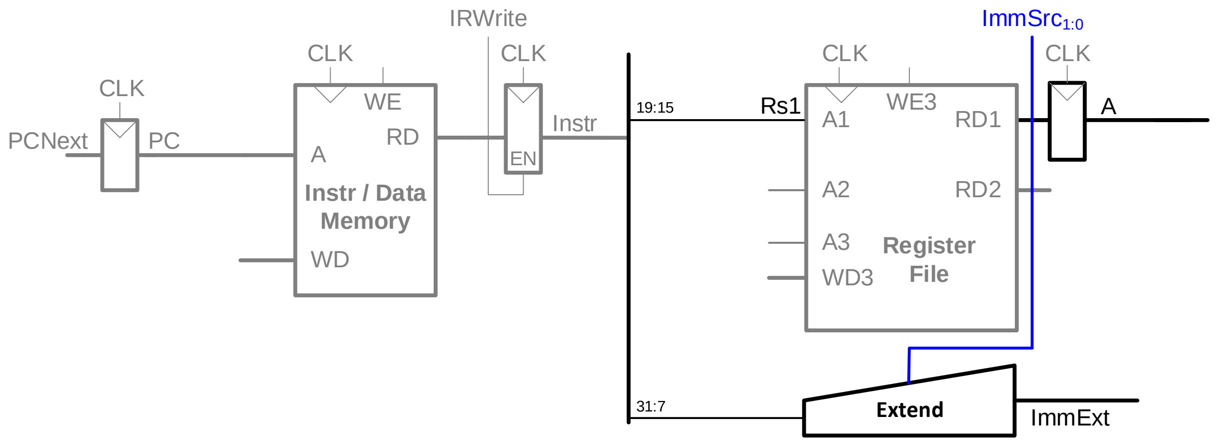
Multicycle Datapath: Instruction Fetch

STEP 1: Fetch instruction



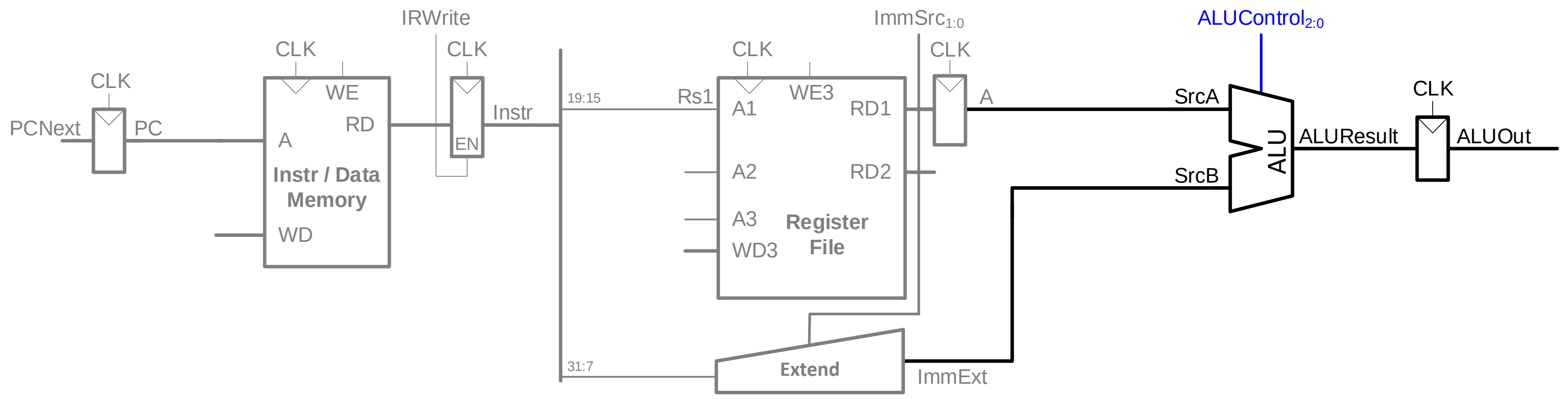
Multicycle Datapath: `lw` Get Sources

STEP 2: Read source operand from RF and extend immediate



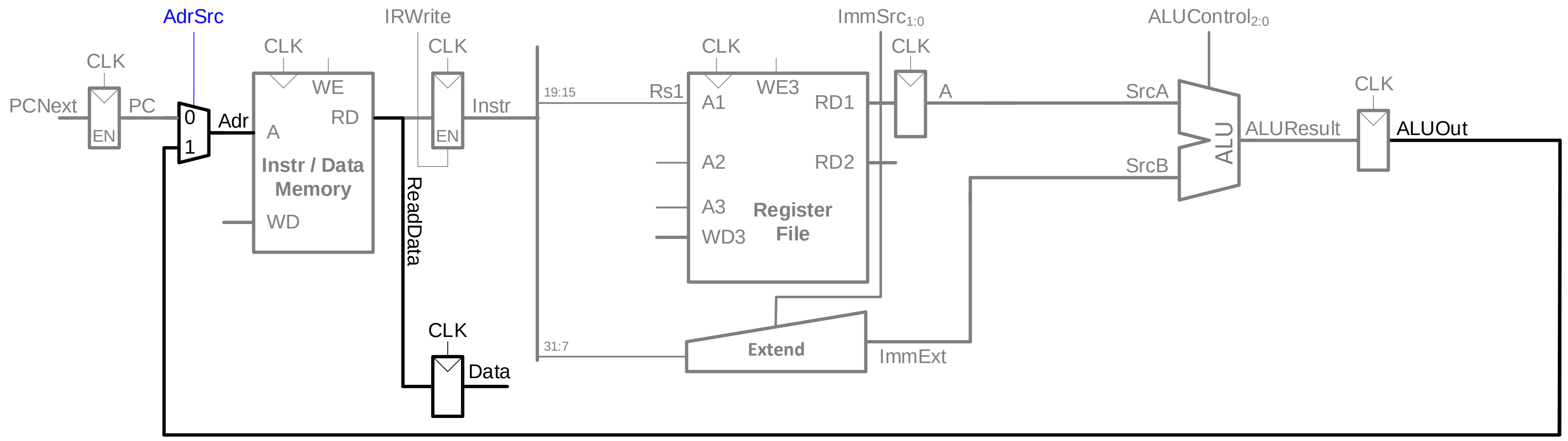
Multicycle Datapath: `lw` Address

STEP 3: Compute the memory address



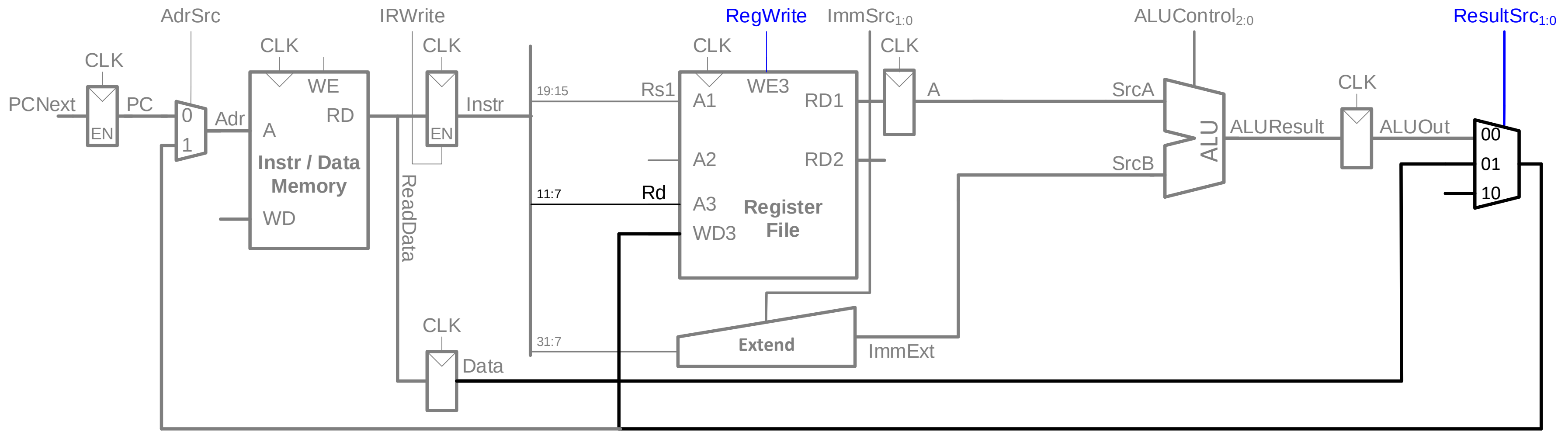
Multicycle Datapath: `lw` Memory Read

STEP 4: Read data from memory



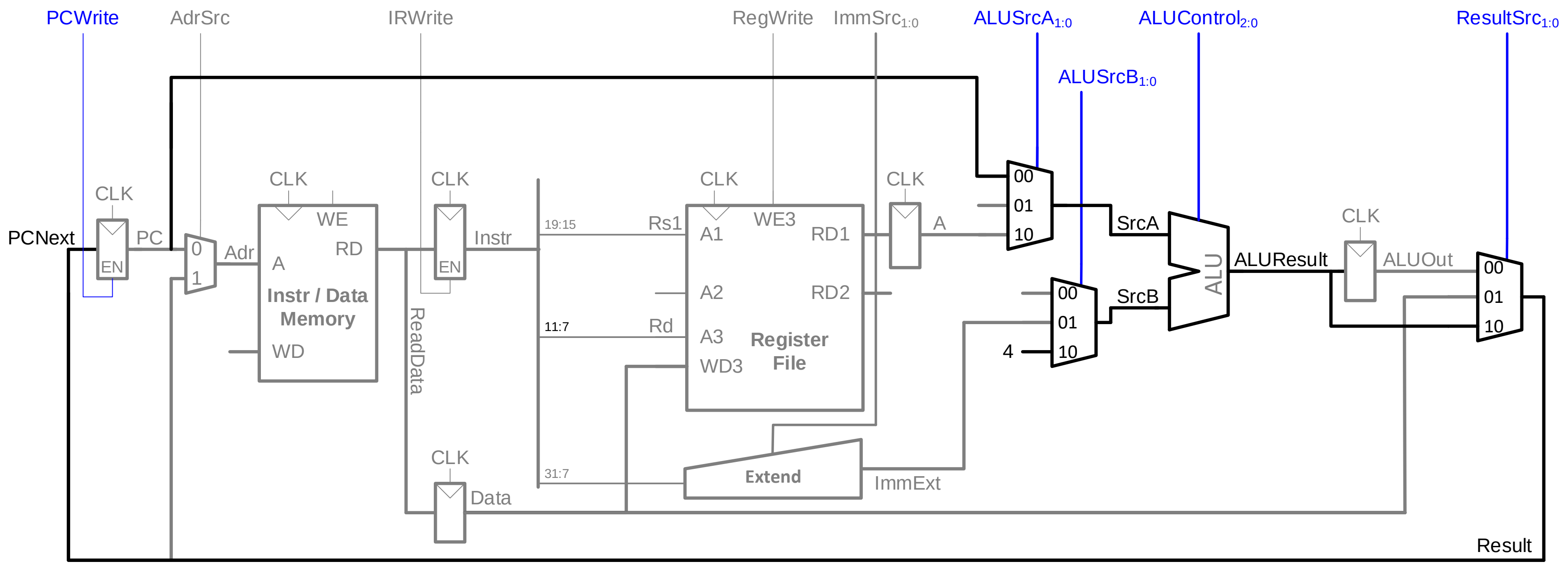
Multicycle Datapath: `lw` Write Register

STEP 5: Write data back to register file



Multicycle Datapath: Increment PC

STEP 6: Increment PC: PC = PC+4

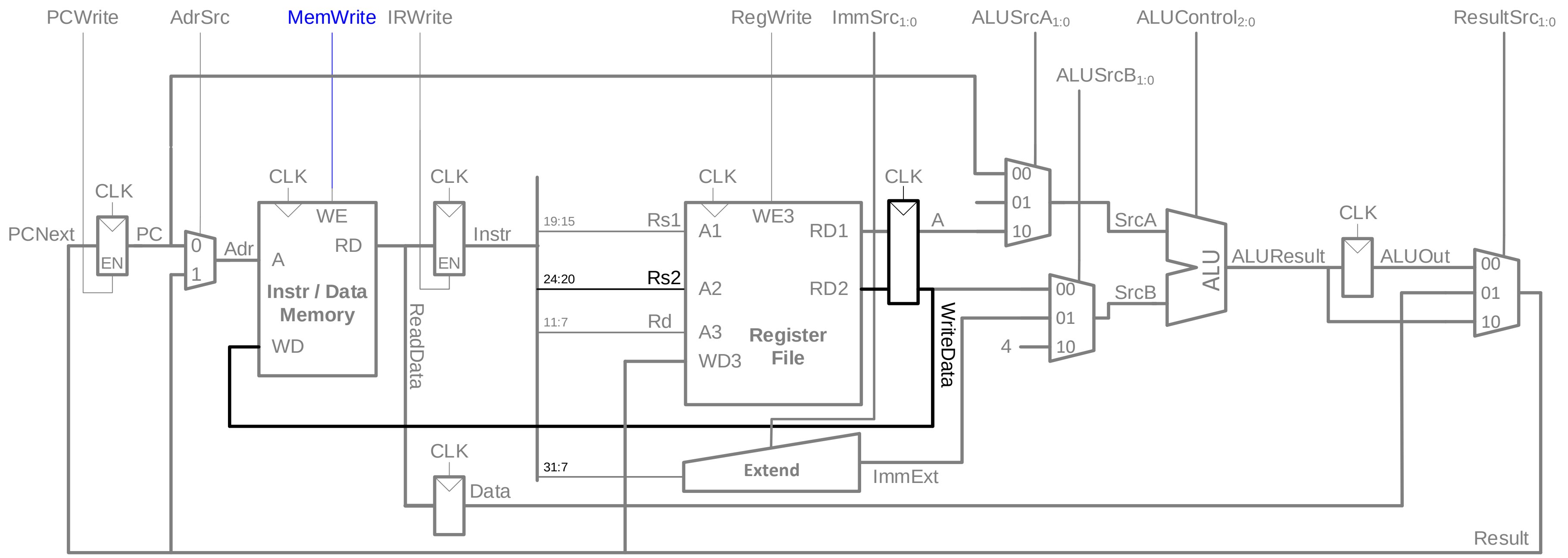


Chapter 7: Microarchitecture

Multicycle Datapath: Other Instructions

Multicycle Datapath: `sw`

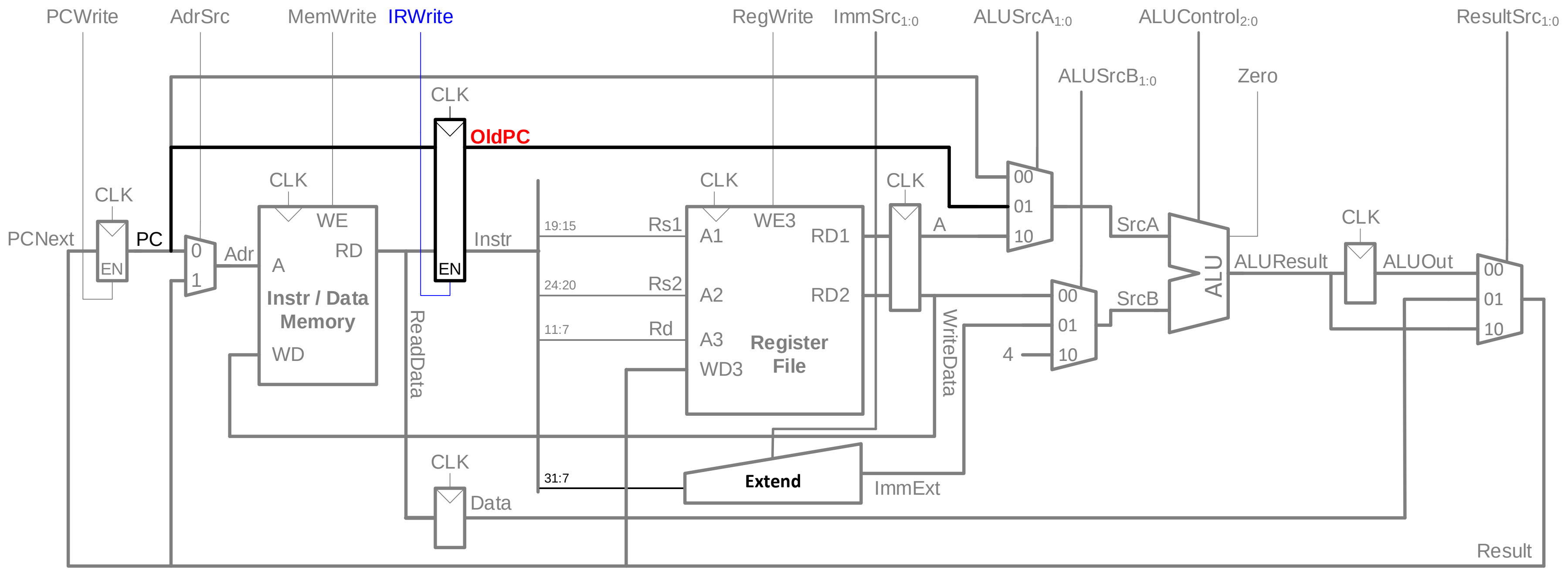
Write data in **rs2** to memory



Multicycle Datapath: beq

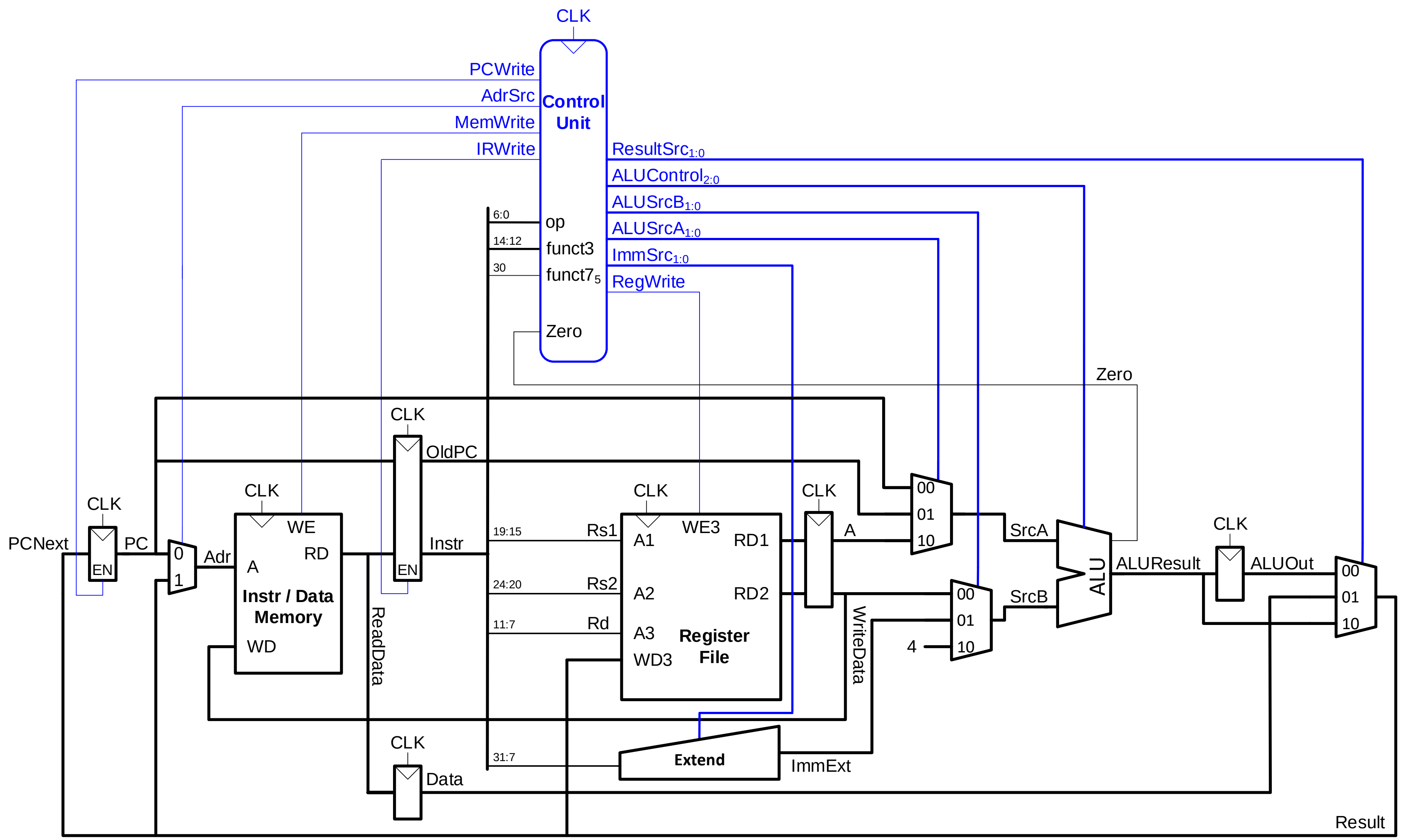
Calculate branch target address:

BTA = PC + imm



PC is updated in Fetch stage, so need to save **old (current) PC**

Multicycle RISC-V Processor



CLK
Control Unit
PCWrite
AdrSrc
MemWrite
IRWrite
ResultSrc1:0
ALUControl2:0
ALUSrcB1:0
ALUSrcA1:0
ImmSrc1:0
RegWrite
6:0
op
14:12
funct3
30
funct75
Zero
Zero
CLK
OldPC
PCNext
CLK
EN
PC
0
1
Adr
CLK
WE
RD
A
Instr / Data Memory
WD
ReadData
EN
Instr
19:15
Rs1
24:20
Rs2
11:7
Rd
CLK
WE3
A1
RD1
A2
RD2
A3
WD3
Register File
CLK
A
WriteData
00
01
10
SrcA
00
01
10
4
SrcB
ALU
ALUResult
CLK
ALUOut
00
01
10
CLK
Data
31:7
Extend
ImmExt
Result