CENG3420 Computer Organization & Design
Lecture 06 Review: Datapath

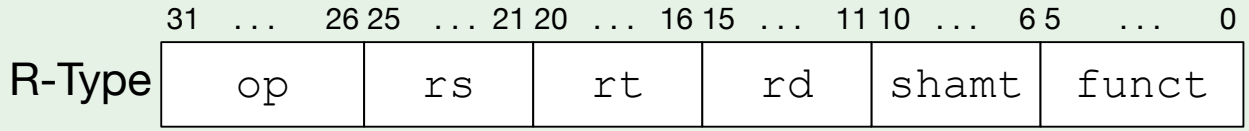
Bei Yu

Spring 2016

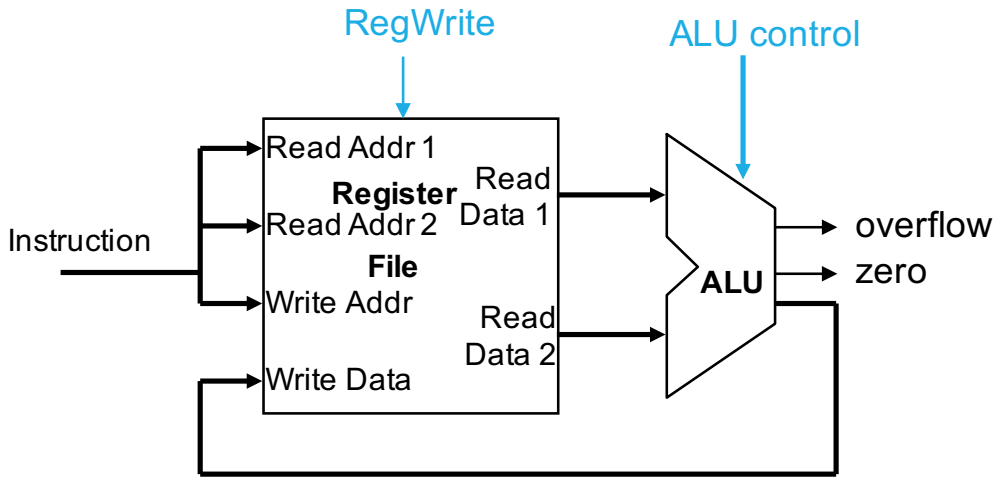
byu@cse.cuhk.edu.hk

香港中文大學
The Chinese University of Hong Kong

Executing R-Type Operations (e.g. `add`)

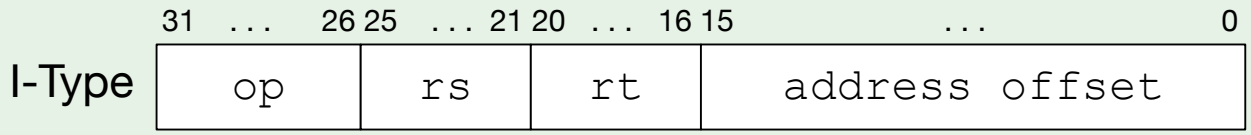


- Perform operation (`op` and `funct`) on values in `rs` and `rt`
- Store the result back into the Register File (into location `rd`)

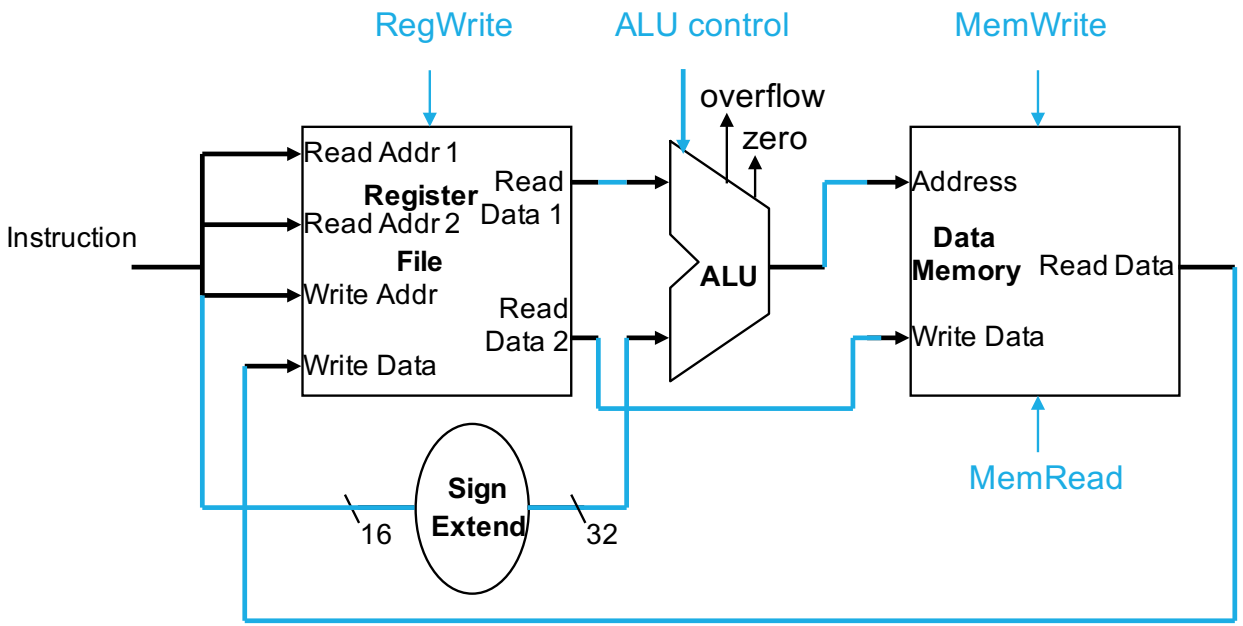


- Note that Register File is not written every cycle (e.g. `sw`), so we need an explicit write control signal for the Register File

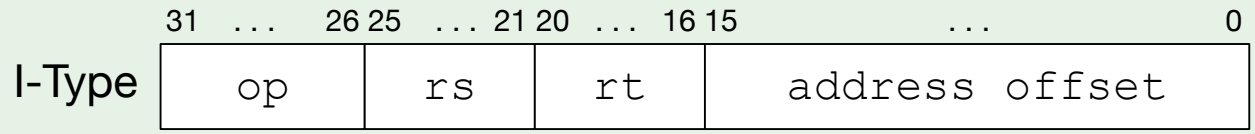
Executing Load & Store Operations (`lw`,`sw`)

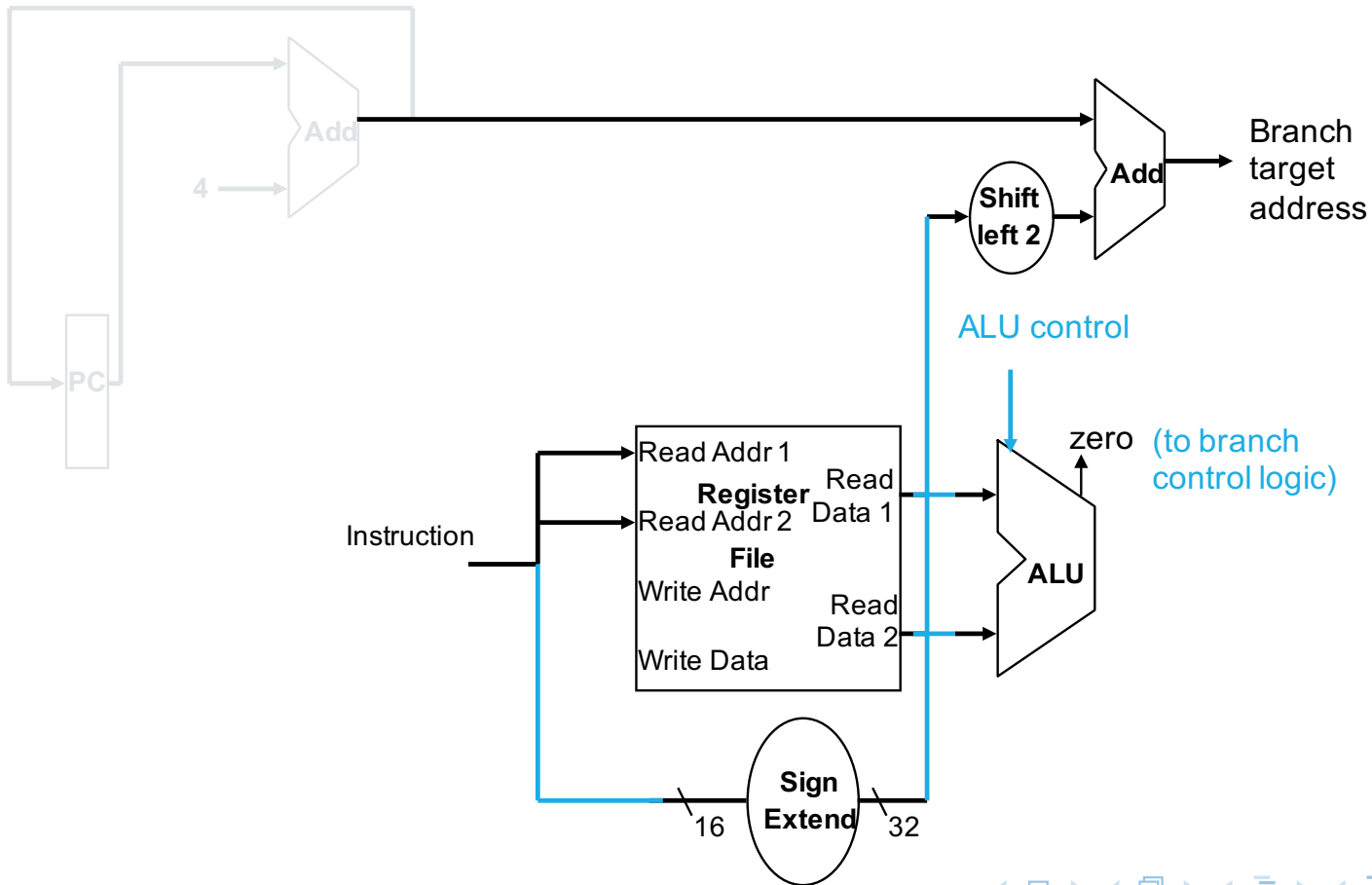


- Compute a memory address by `rs` and `offset`
- Original data is in register `rt`

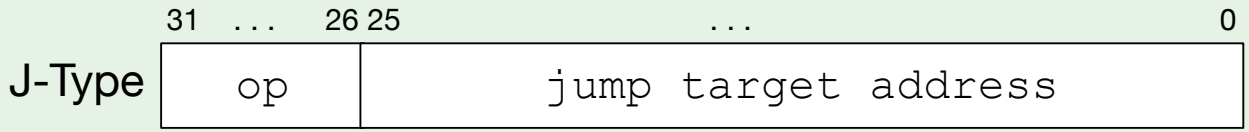


Executing Branch Operations (`beq`)

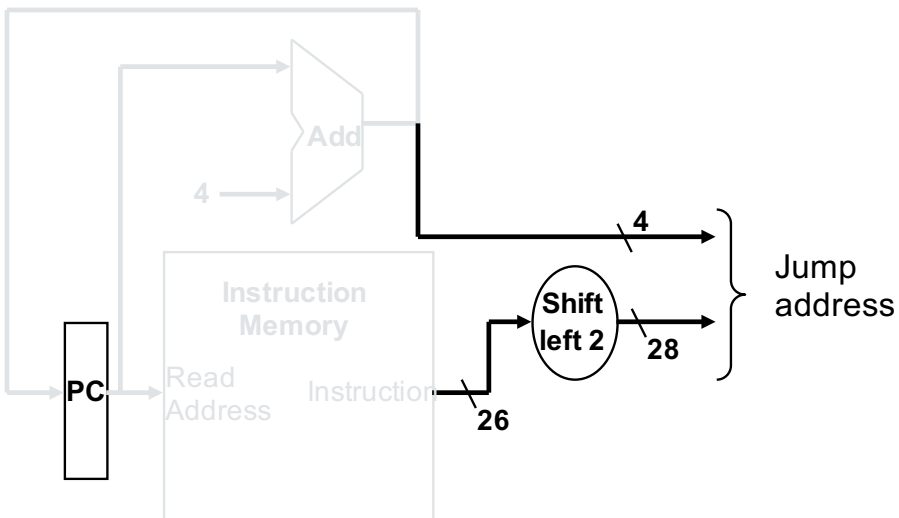




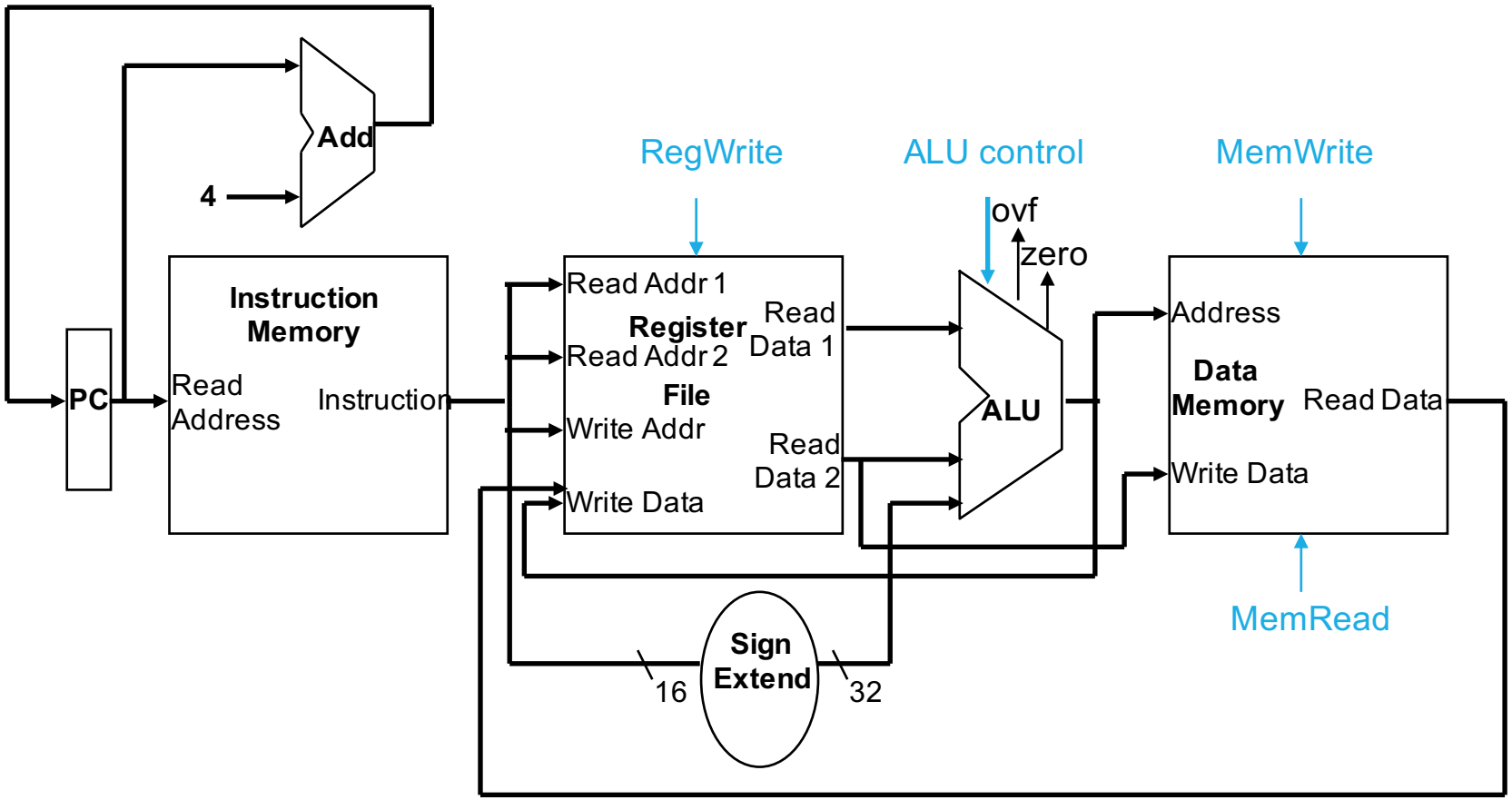
Executing Jump Operations (`j`)



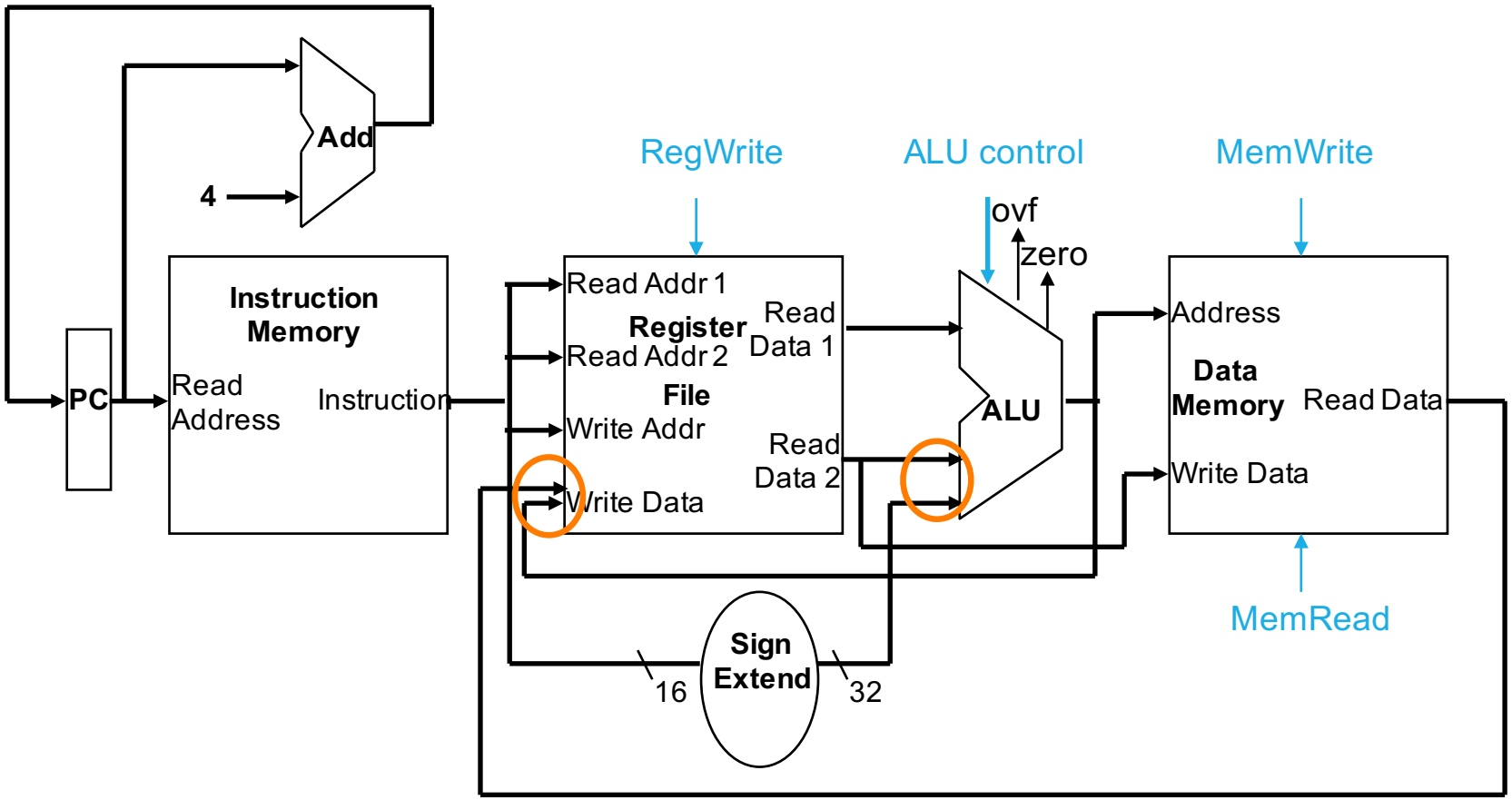
- Replace the lower 28 bits of the PC with the lower 26 bits of the fetched instruction shifted left by 2 bits



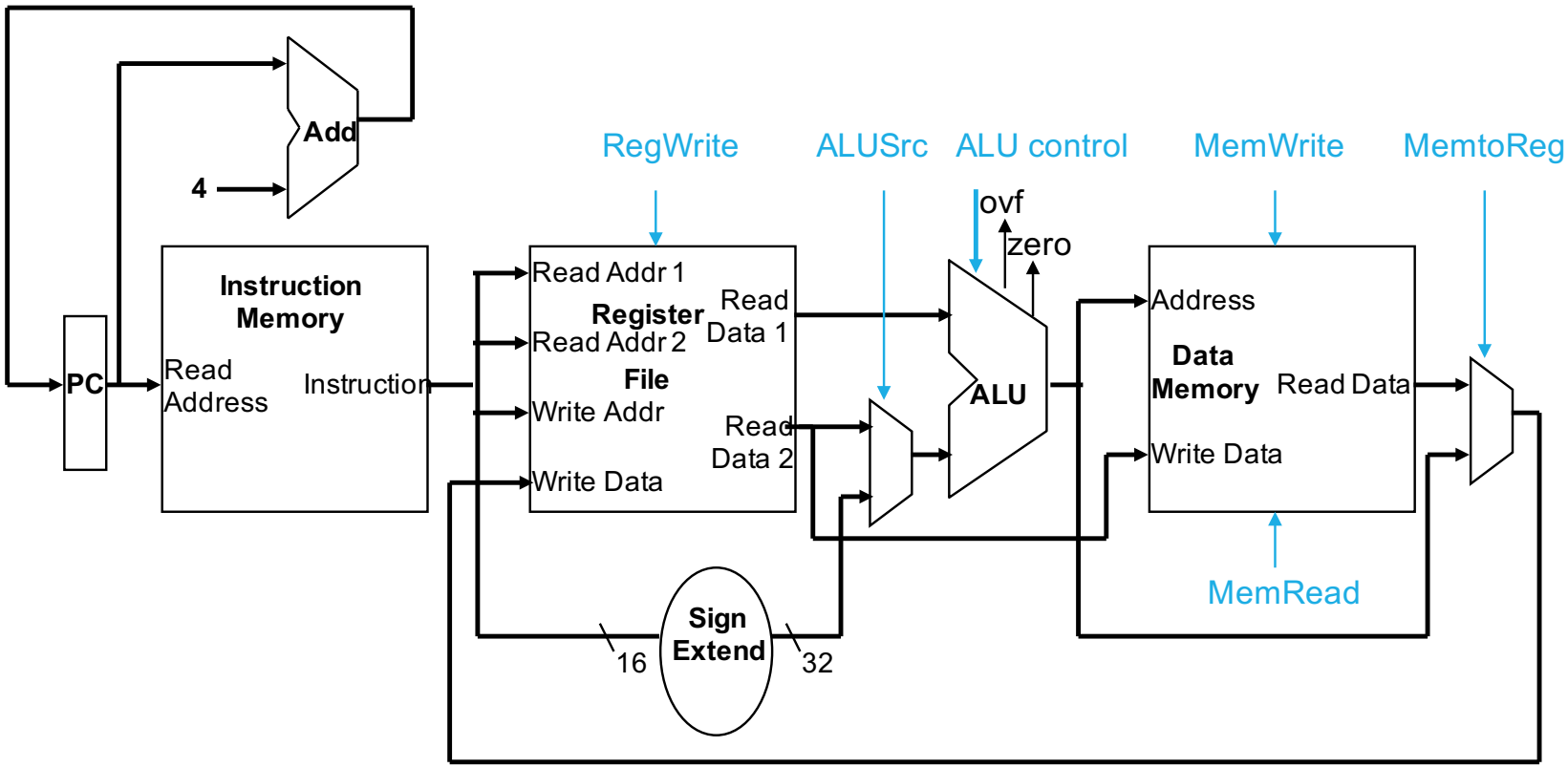
Fetch, R, and Memory Access Portions



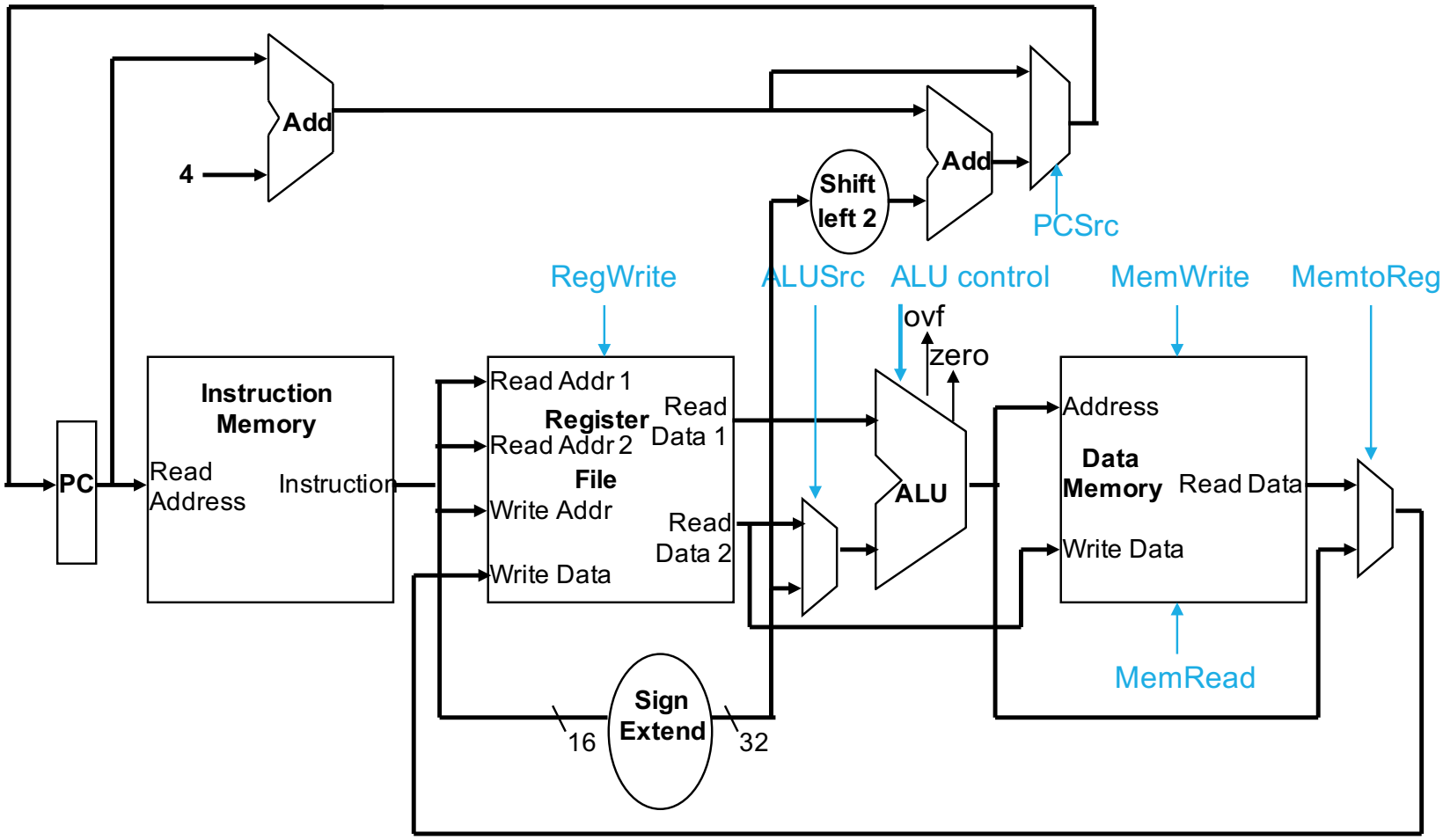
Fetch, R, and Memory Access Portions



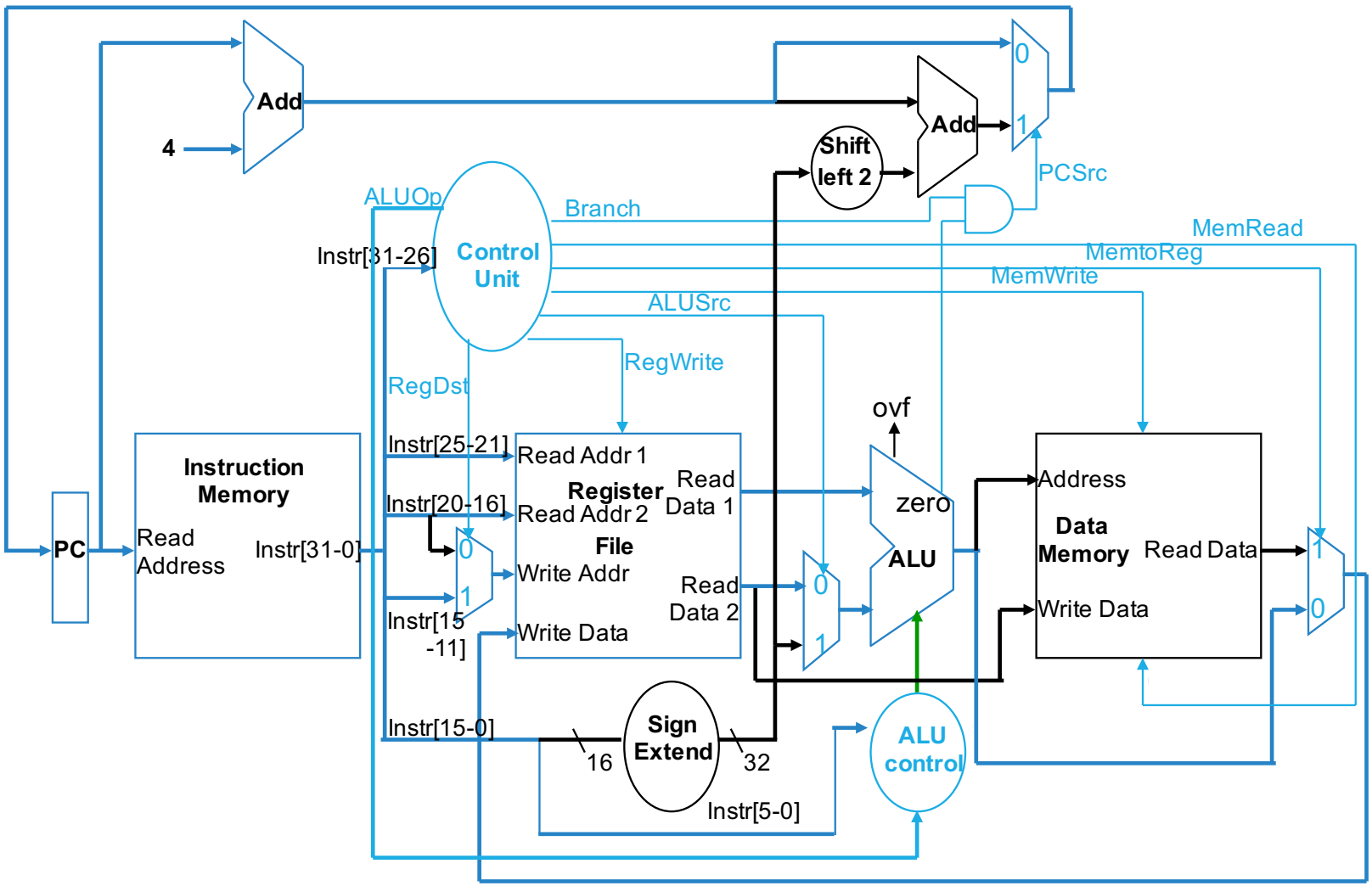
Multiplexor Insertion



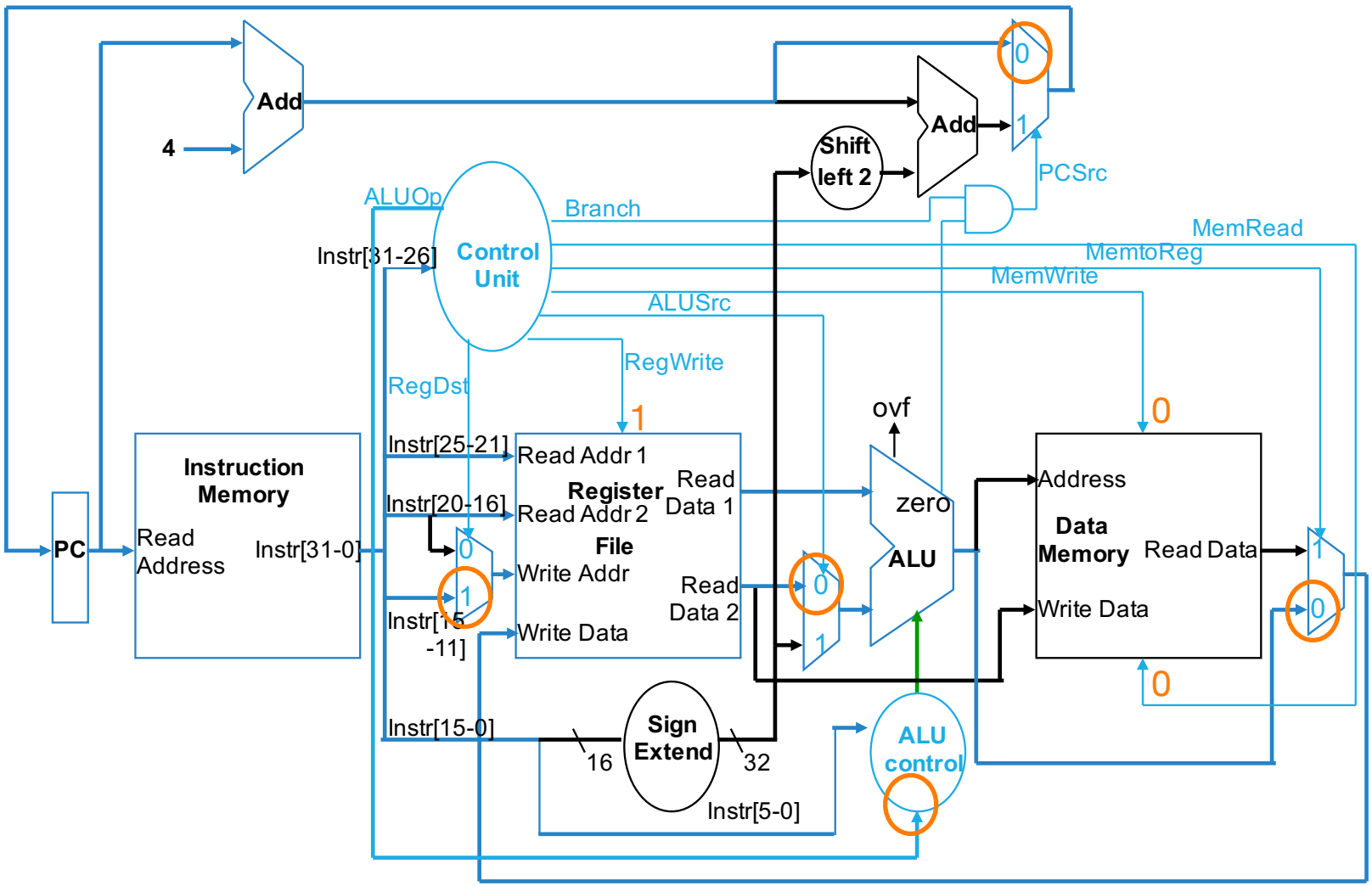
Adding the Branch Portion



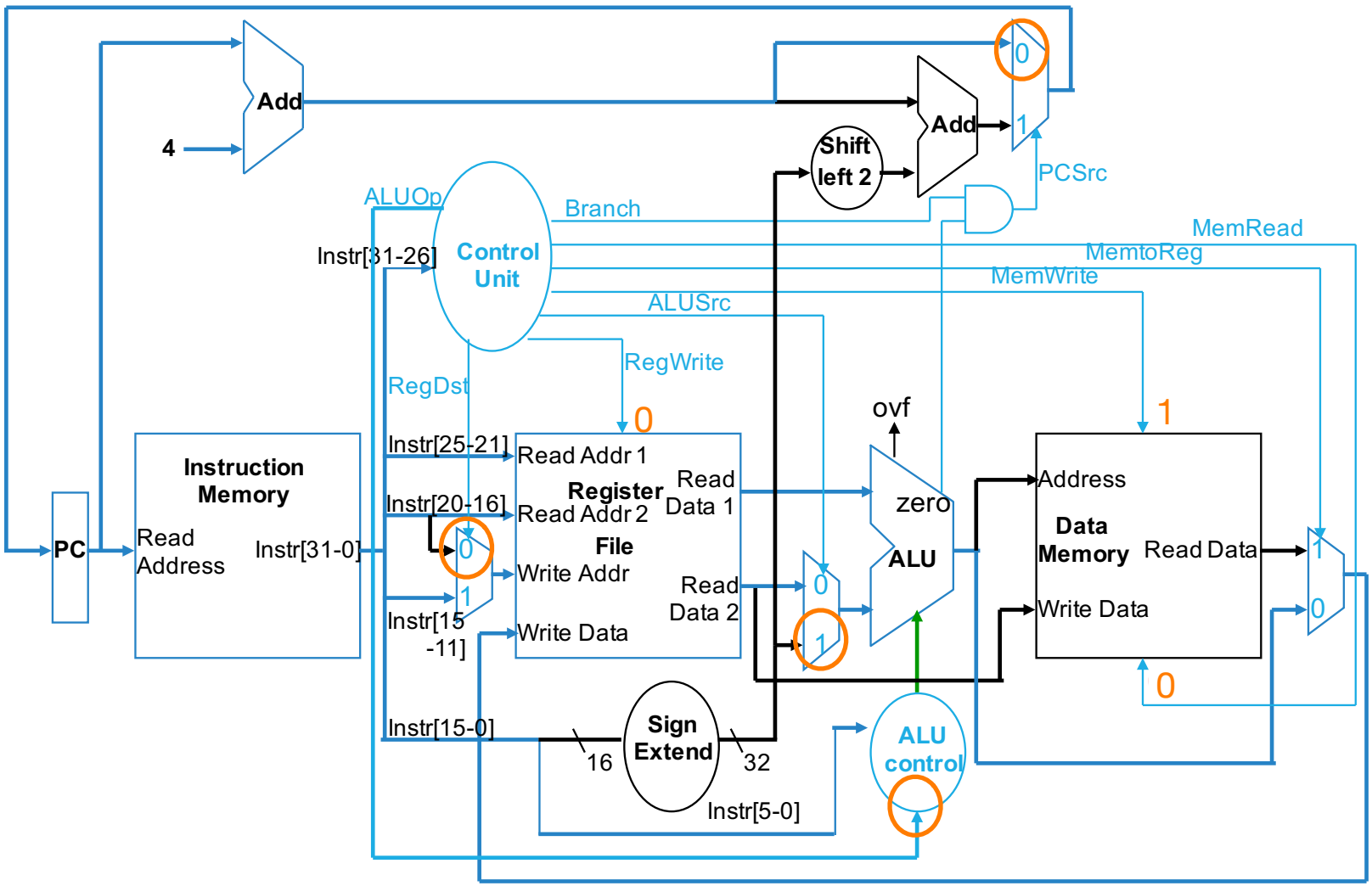
R-type Instruction Data/Control Flow



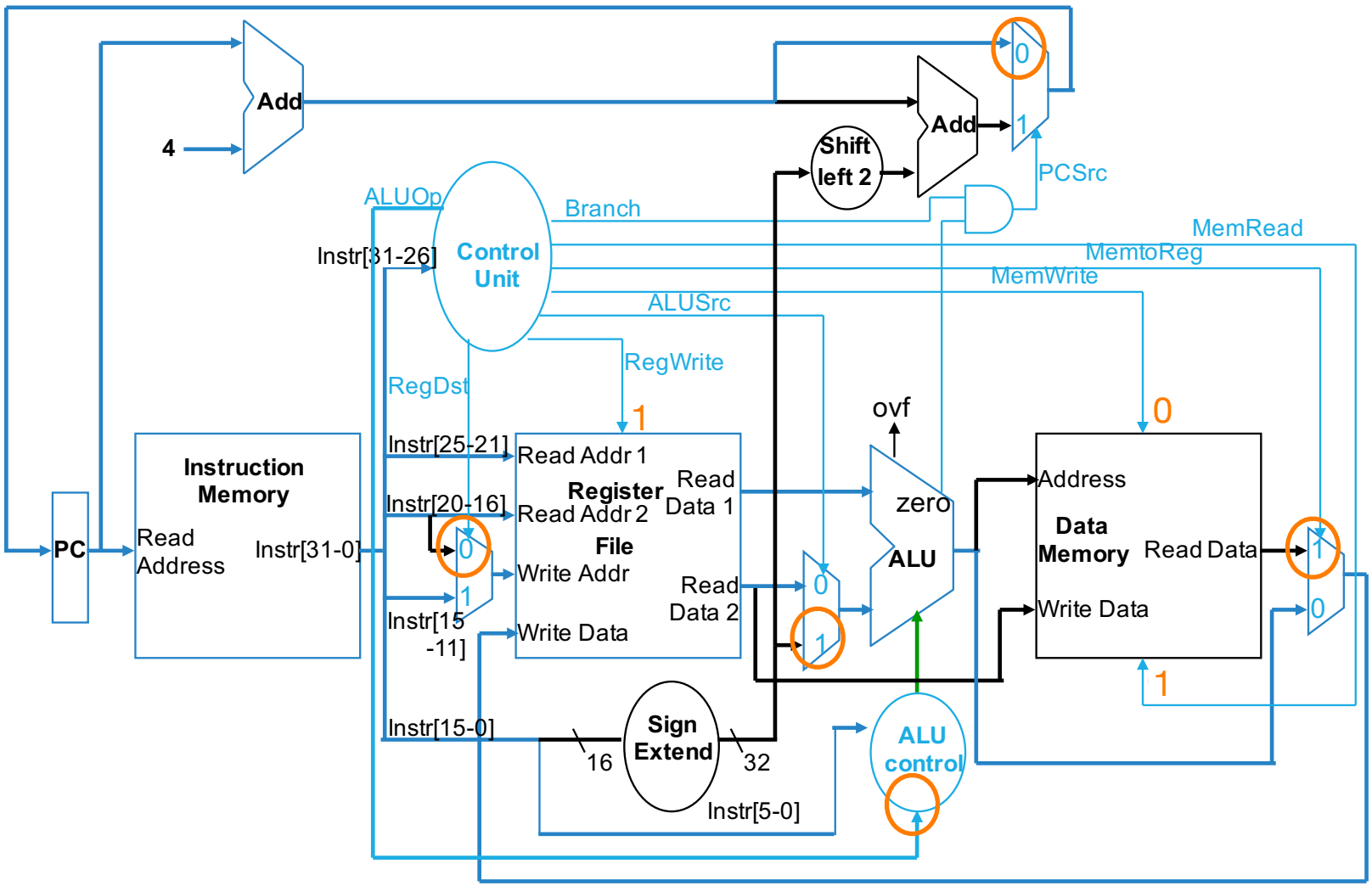
R-type Instruction Data/Control Flow



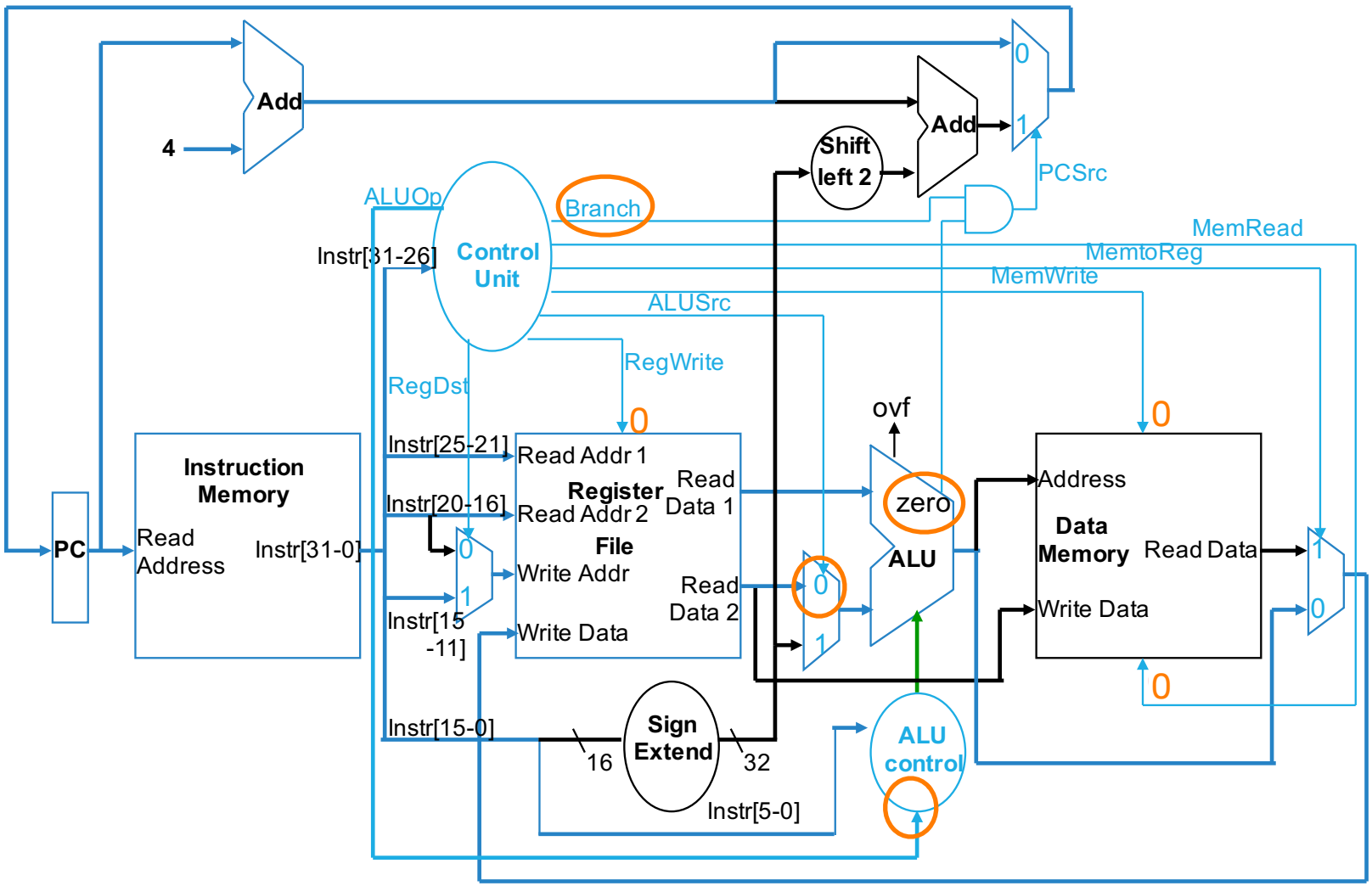
Store Word Instruction Data/Control Flow



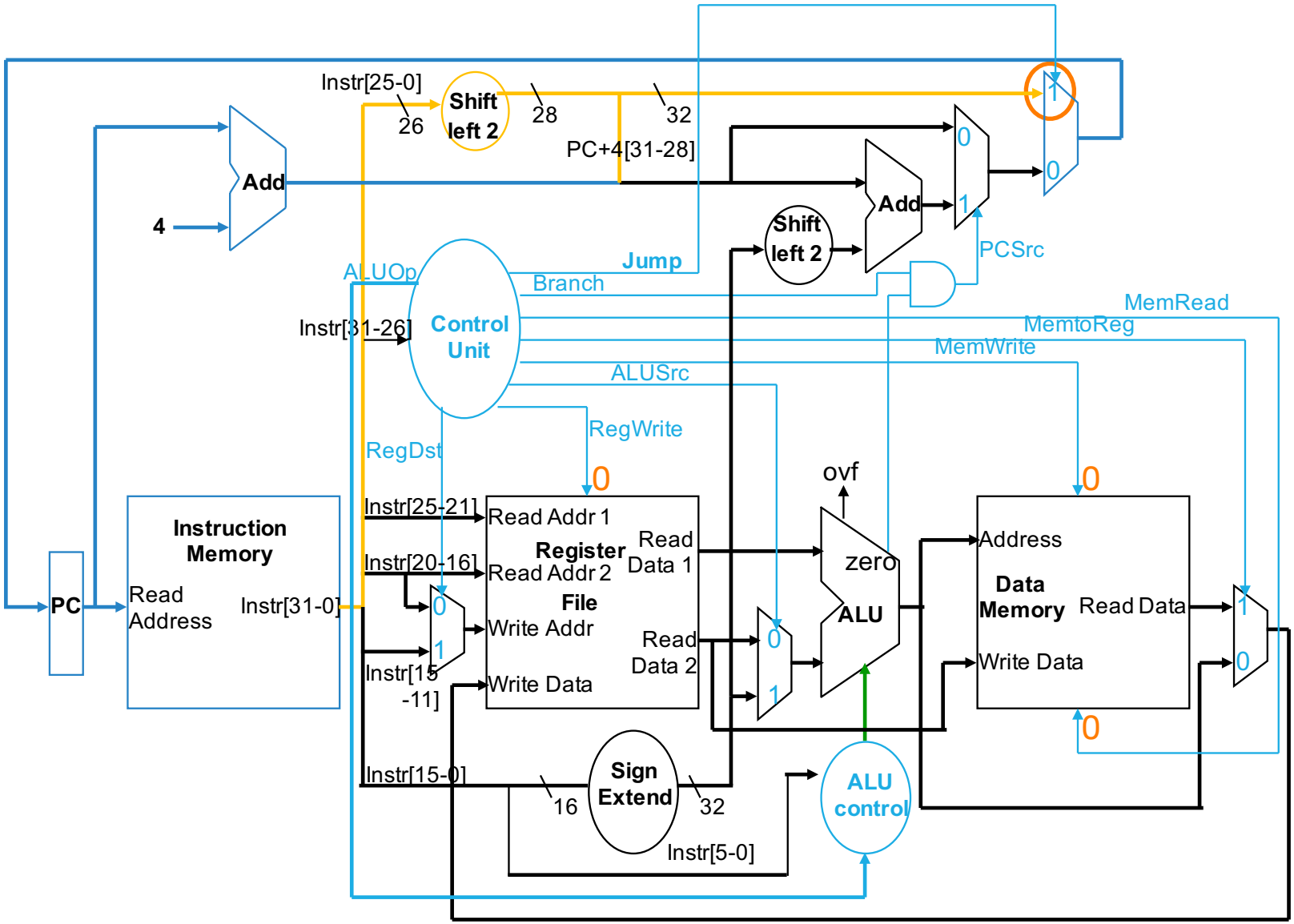
Load Word Instruction Data/Control Flow



Branch Instruction Data/Control Flow



Adding the Jump Operation



Main Control Unit

	RegDst	ALUSrc	MemReg	RegWr	MemRd	MemWr	Branch	ALUOp	Jump
R-Type	1	0	0	1	0	0	0	10	0
`lw`	0	1	1	1	1	0	0	00	0
`sw`	X	1	X	0	0	1	0	00	0
`beq`	X	0	X	0	0	0	1	01	0
`j`	X	X	X	0	0	0	X	XX	1