CSM151B HW4
Michael Zhou
804 663 317

4.7
1.
Opcode = (101011)2 = (43)10
This is a load/store function

Rs = (00011)2 = 3
Rt = (00010)2 = 2
Address = (0000000000010100)2 = 20

Thus, the instruction is sw $s2, 20($s3).

The input of sign-extended is:
0000 0000 0000 0000 0000 0000 0001 0100

The output of the shift left is:
(XXXX 1000 1000 0000 0000 1010 0000)2, where XXXX is the most significant 4 bits of PC+4

2.
ALU control input is:
Function field of instruction: (010100)2 = (43)10
It's a SW instruction
ALUop is 00

Input1 of ALU is (00011)2
Input2 of ALU is (00000000000000000000000000010100)2

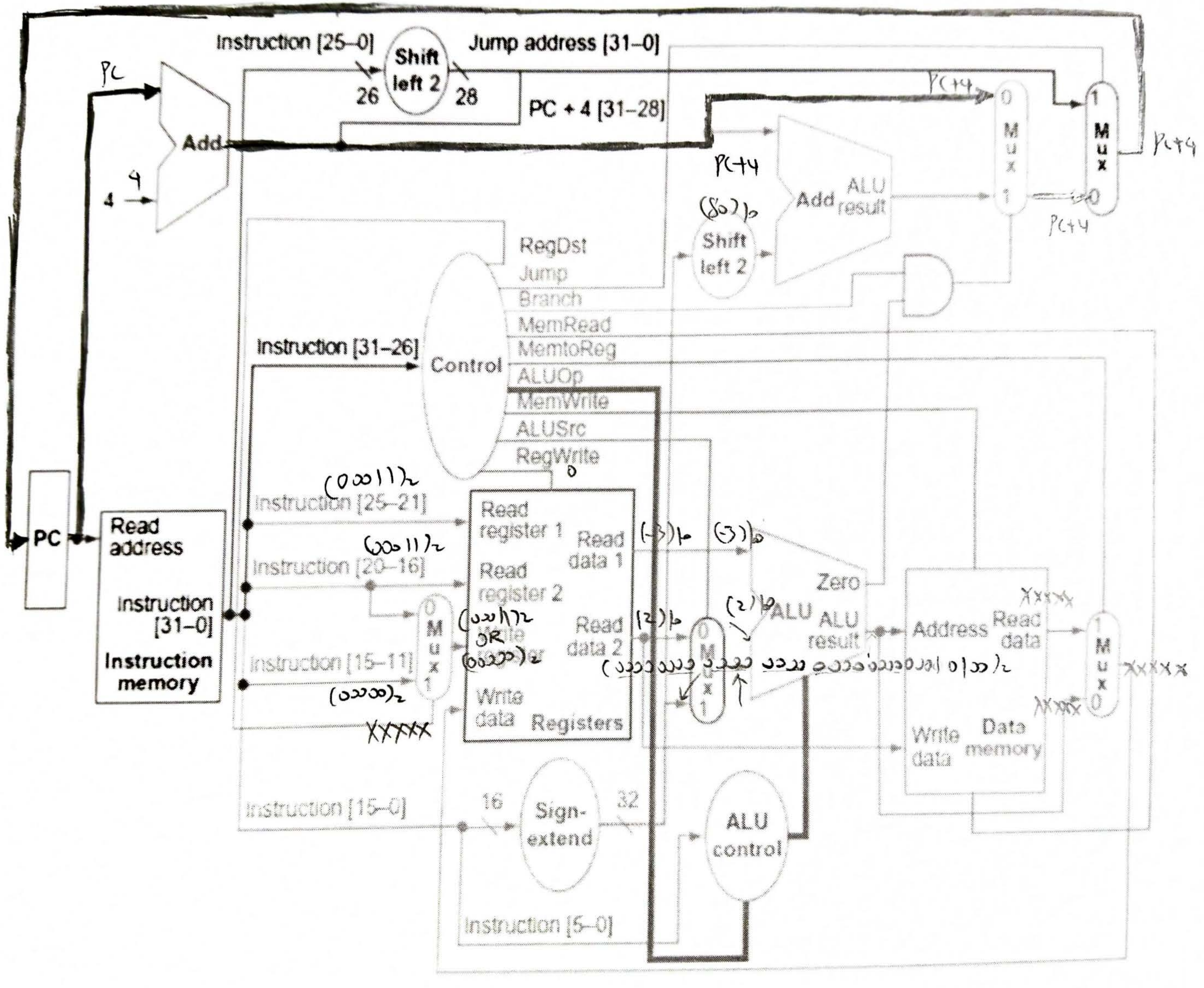
3.
The new PC is PC+4 The data path is shown in the diagram attached below.

4.
Two muxes at branch: PC+4
Mux before the registers: Can be either 00010 or 00000
Mux before the ALU: 0000 0000 0000 0000 0000 0000 0001 0100
Mux after data memory: Random bits from "read data" port

5.
ALU input 1: (-3)10
ALU input 2: (20)10
Top left adder: current PC and 4
add(ALU result): pc + 4 and (80)10

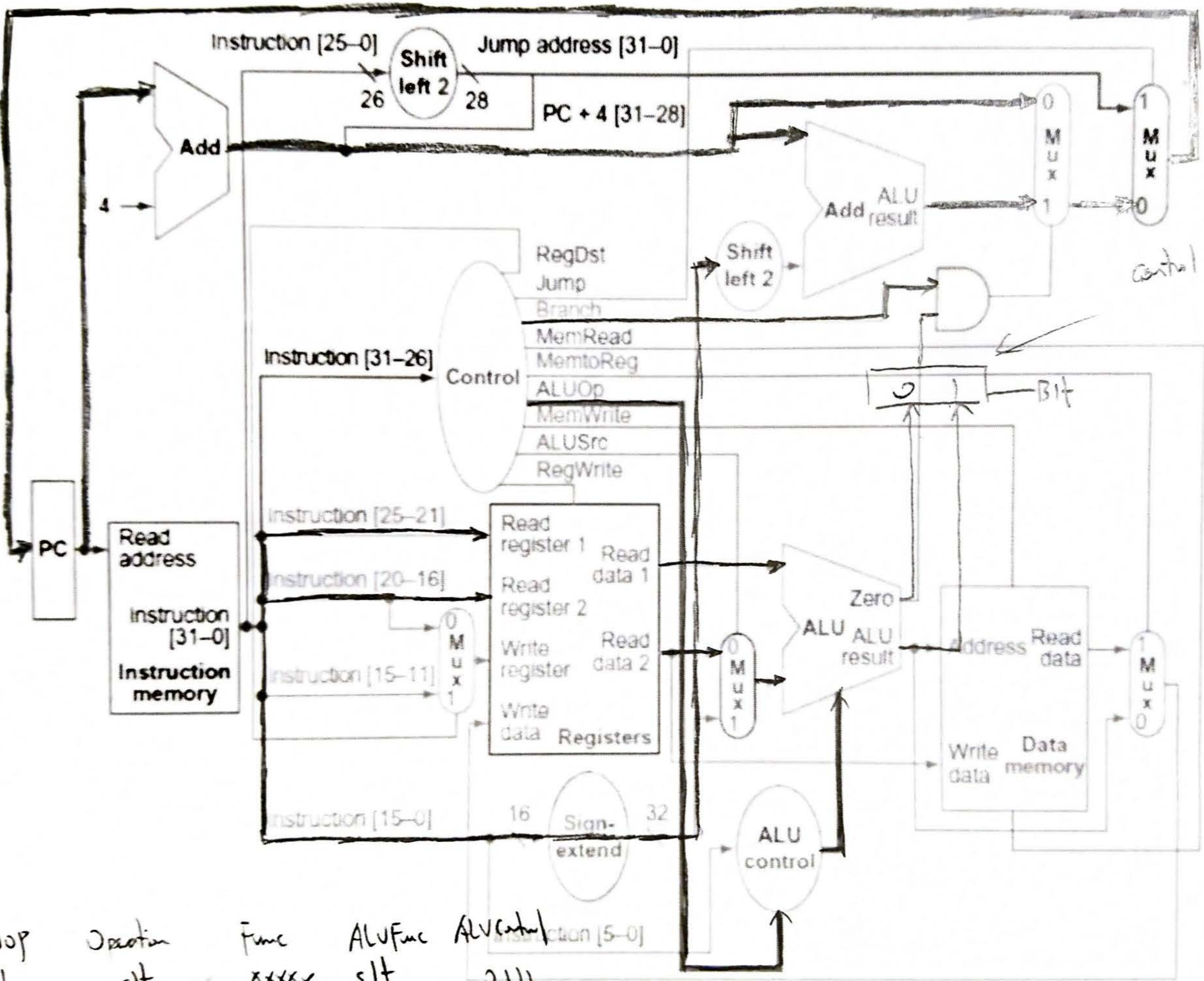
6.
Read register 1: 00011
Read register 2: 00010
Write register: 00010 or 00000 (don't care)

Write data: don't care
RegWrite:0


Instruction [25–0]
Shift left 2
26
28
Jump address [31–0]
PC + 4 [31–28]
PC
Add
4
4
P(+4
P(+4
P(+4
P(+4
(80)₁₀
Add
ALU result
Shift left 2
M u x
0
1
M u x
1
0
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
0
Control
Instruction [31–26]
PC
Read address
Instruction [31–0]
Instruction memory
(00011)₂
Instruction [25–21]
(00011)₂
Instruction [20–16]
Instruction [15–11]
(00000)₂
XXXXX
M u x
0
1
(00011)₂
OR
(00000)₂
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
(-3)₁₀
(-3)₁₀
(2)₁₀
(2)₁₀
(00000000000000000000000001010000)₂
M u x
0
1
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
XXXXX
XXXXX
XXXXX
M u x
1
0
Instruction [15–0]
16
32
Sign-extend
ALU control
Instruction [5–0]

```
if(R[rs]<R[rt])
  PC=Pc+4 + SE(I)
else
  PC=PC+4
```

blt=



control flag for blt.

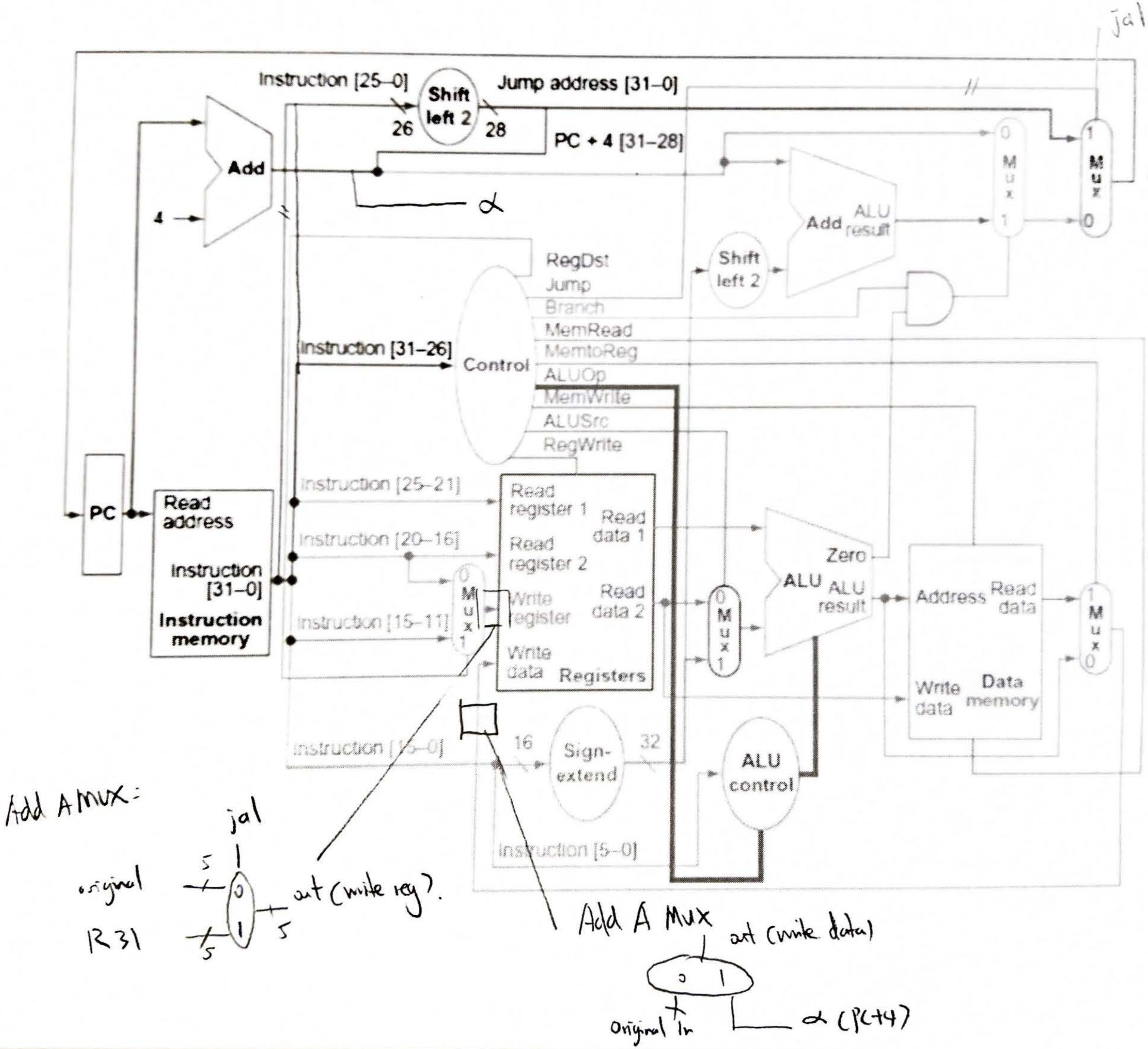
Control for BLT

Regdst : x
Branch : x
Memread : 0
MemtoReg : x
ALUOP : 01
MemtoWrite : 0
ALUSRC : 0
RegWrite : 0
BLT : 1

ALU Control:

Opcode.	ALUOP	Operation	Func	ALUFunc	ALUControl
Blt	11	slt	xxxxx	slt	0111

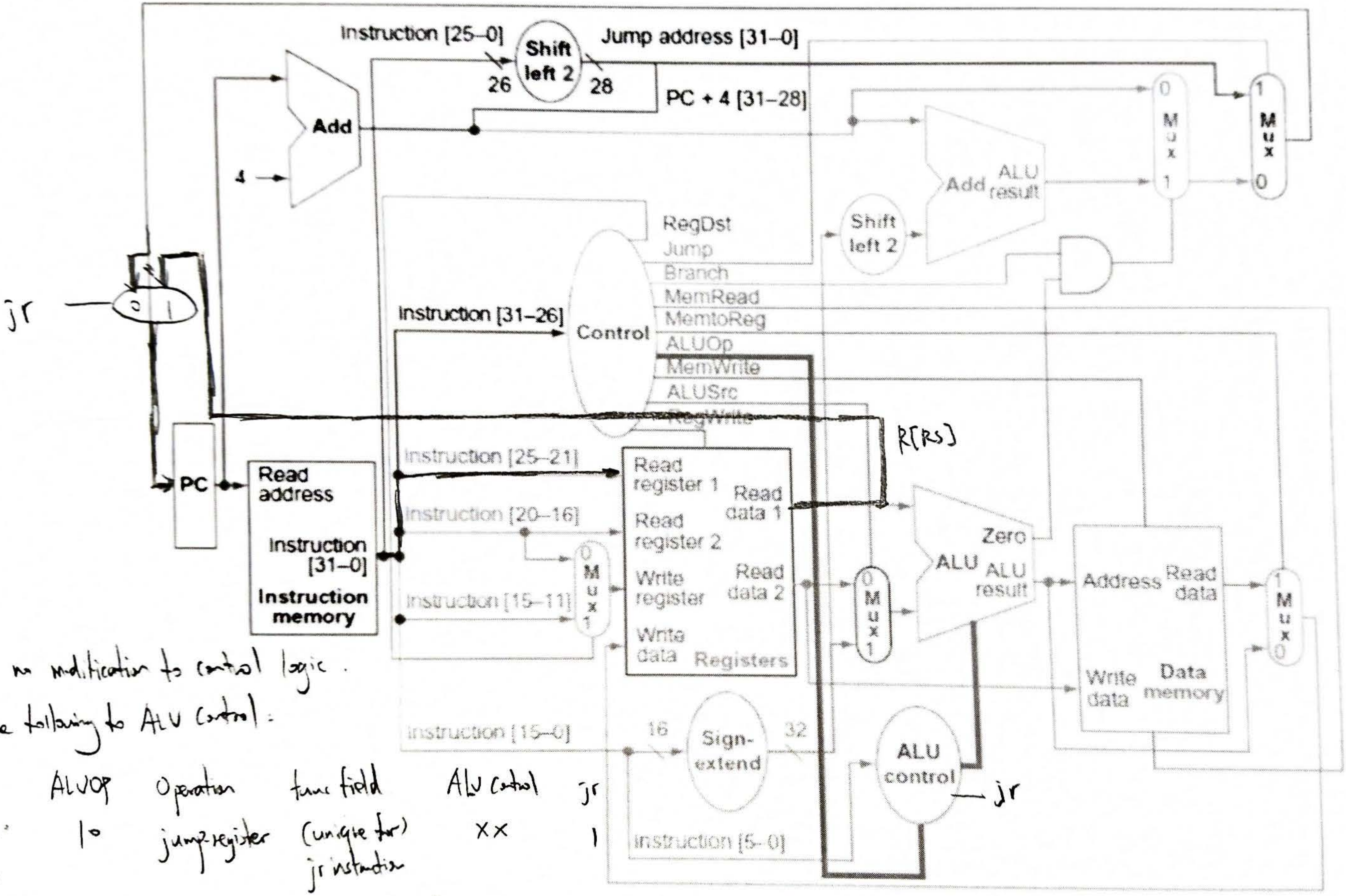
jal : R[$r31] = PC+4
PC = [31 ... 28](PC+4) | [27 ... 0](1<<2)



Control logic for JAL

Branch :	X
ALUOP	XX
Reg Dst	X
Mem to Reg :	X
ALU SRC	X
REG WRITE	1
MEM READ	∅
MEM WRITE	∅
JUMP	1
JAL	1

jr:
PC = R[rs]



There's no modification to control logic.

Add the following to ALU control:

Opcode	ALUOP	Operation	func field	ALU control	jr
R-type. 000000	10	jump-register	(unique for) jr instruction	XX	1

ALU control outputs jr=0 except for jr instruction.