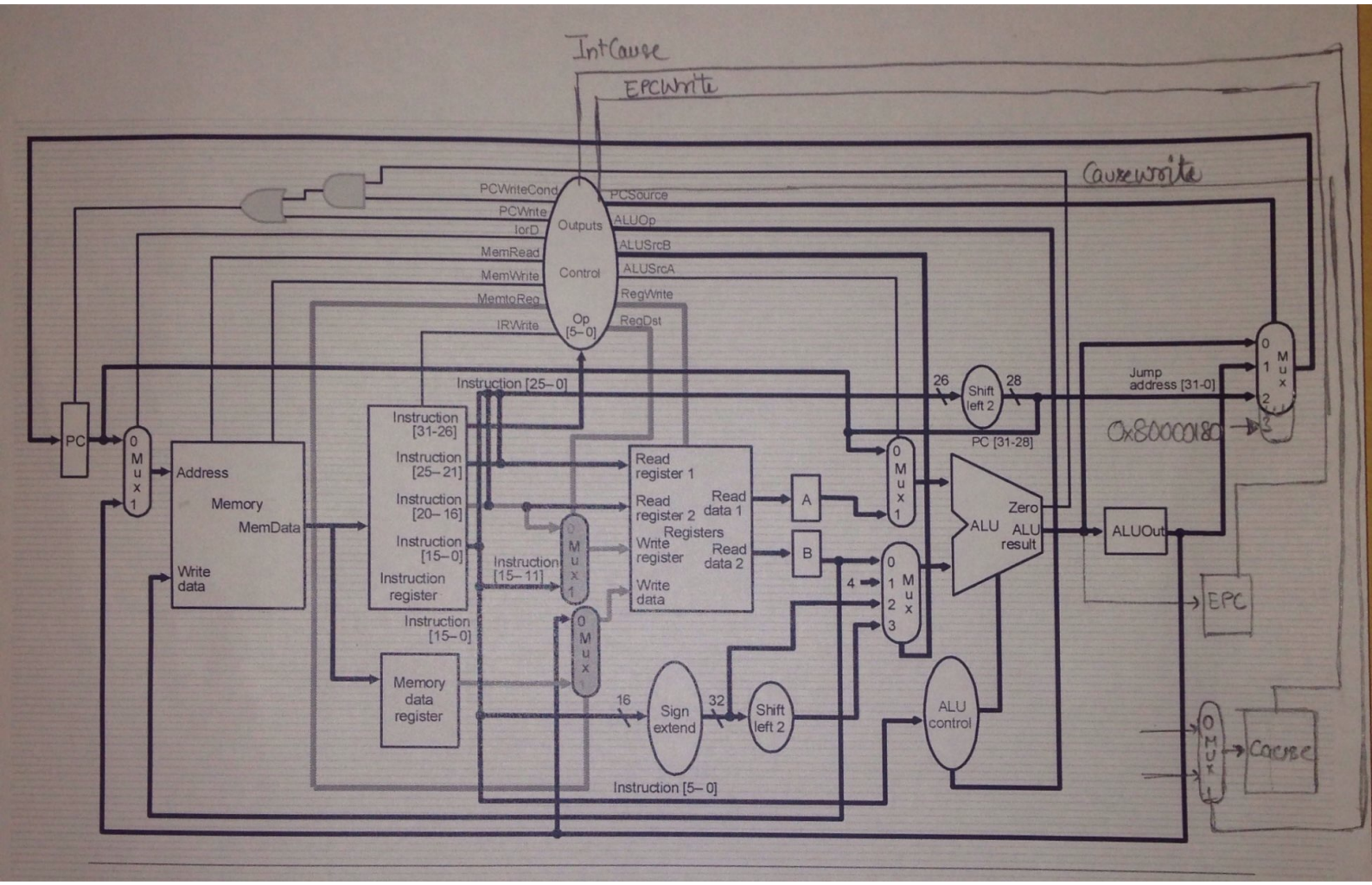
(2)

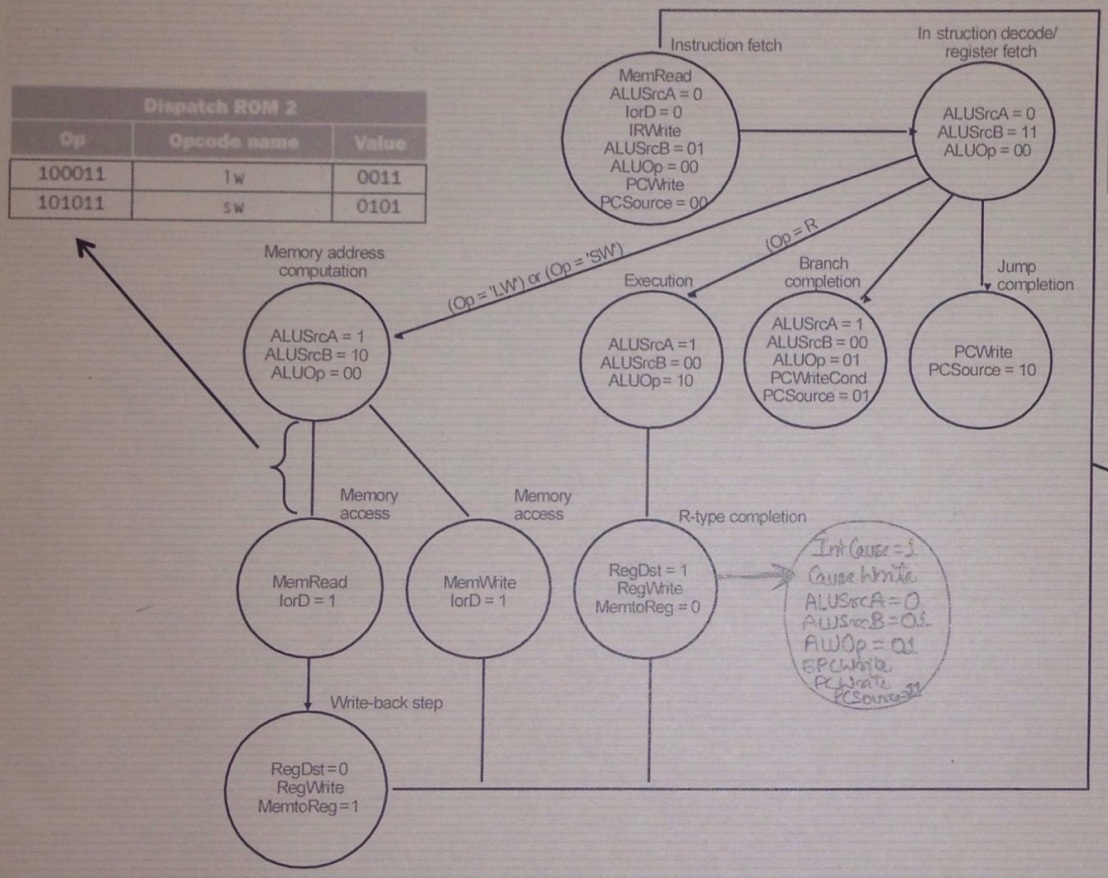
Cycle	ALUSrB	ALUOp	RegDst	Instruction	PCWriteCond
12	01	00	0	0x8D0D0000	0
19	10	00	0	0xAD2D0000	0

(c) This is a model answer your implementation may vary.

↳ On detecting an Overflow exception we will jump to a special address + offset. This offset will also be an indicator of "type" of exception. For example we can consider an "Overflow" as a arthematic exception. There could be other class of exceptions like PC not at word boundary exception. All these types of exception will have an offset associated with it as defined in ISA of the architecture. The memory location at address+offset will have "handler" written for the necessary action that needs to be taken for this certain type of exception. It will flush the instructions in pipeline and start executing the code present at handler.



2b (I)



Dispatch ROM 2

Op	Opcode name	Value
100011	lw	0011
101011	sw	0101

Sequencing Options

AddrCtl value	Action
0	Set state to 0
1	Dispatch with ROM 1
2	Dispatch with ROM 2
3	Use the incremented state

Dispatch ROM 1

Op	Opcode name	Value
000000	R-format	0110
000010	jmp	1001
000100	beq	1000
100011	lw	0010
101011	sw	0010
11111	Overflow	1010

2b(II)