4.1) 4.1.1:

Instruction	Interpretation
AND Rd, Rs, Rt	Reg[rd] = Reg[Rs] AND Reg[Rt]

Opcode	ExtSel	BSrc	OpSel	MemW	RegW	WBSrc	RegDst	PCSrc
ALU	*	Reg	Func	no	yes	ALU	rd	pc+4

ALU operation is AND. Opsel tells the ALU to perform AND. BSrc looks into the register. WBSrc is set to use the output of ALU. RegDst is rd. PCSrc is PC+4 because of lack of jumps.

4.1.2: 1) Get instruction address
2) Get operands from memory
3) ALU performs AND
4) Write to register

4.1.3: Data Memory doesn't produce an output. Branch produces an output, but is never used.

Scanned with CamScanner

4.7) 4.7.1: Sign extended: 00000000000000000000000010100
jumpShift left 2: 00001000100000000000101000

4.7.2: ALUop [1-0]: 00
Instruction [5-0] = 20 (00101000)

4.7.3: New PC: PC+4

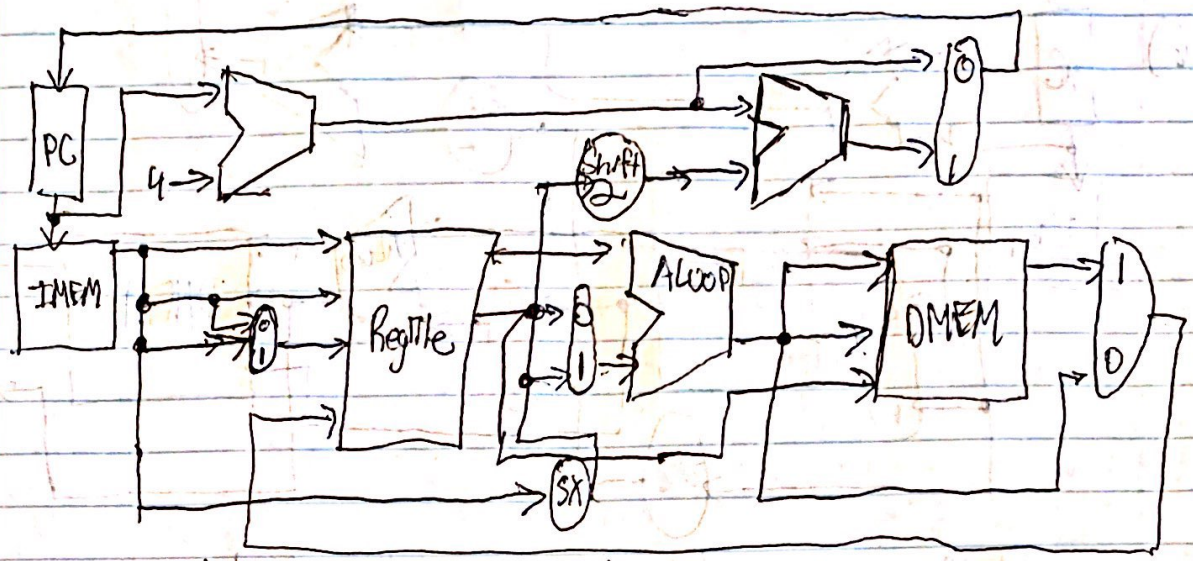
4.7.4: WriReg Mux: 2
ALU Mux: 20 (00010100)
Mem|ALUMux; x
Branch Mux: PC+4

4.7.5: ALU: -3, 00010100
Add(PC+4): PC, 4
Add(Branch): PC+4, 01010000 (80)

4.7.6: Read Register 1: (00011) = 3
Read Register 2: 2 (00010)
Write Register: x|?
Write Data: x|?
Reg Write: 0

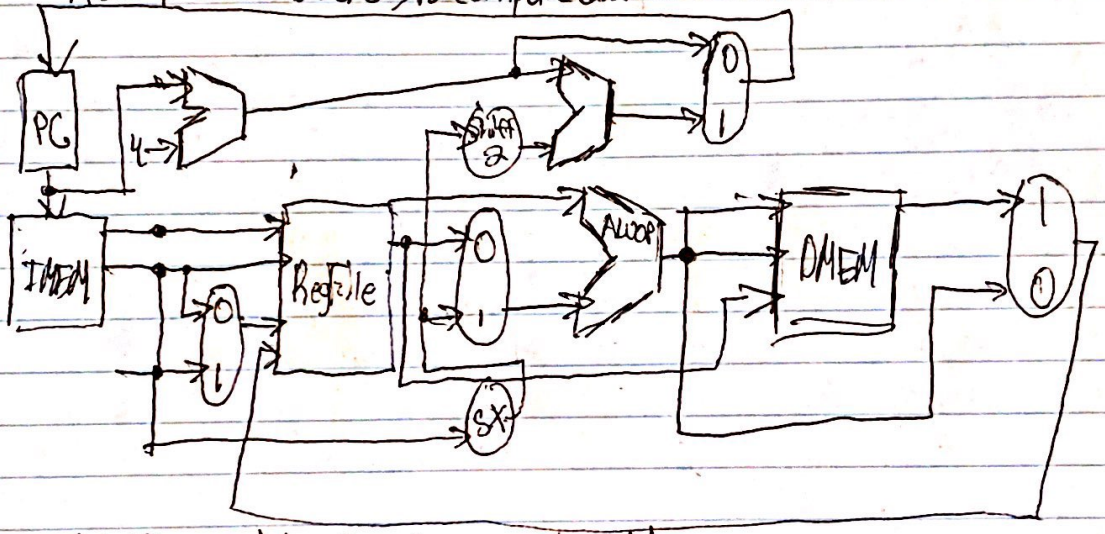
Scanned with CamScanner

3)



ALUOp Must be 010, to compute address

4)

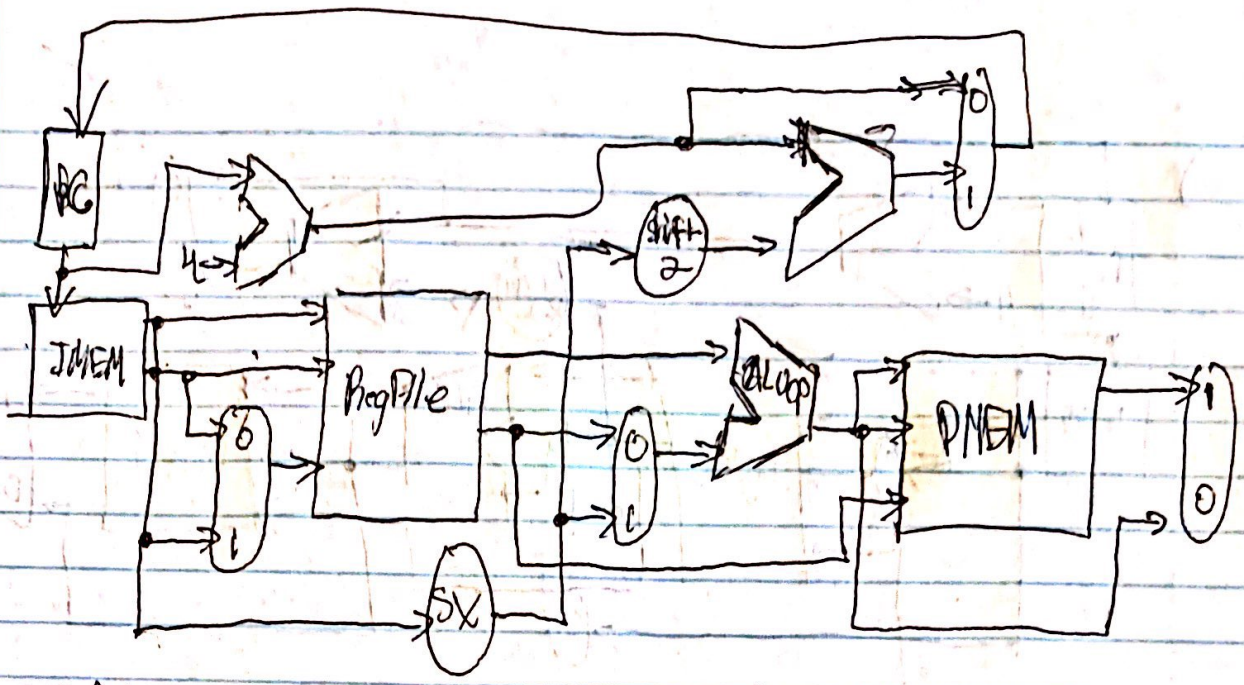


ALUOp must be 010, to compute address

5)

Scanned with CamScanner

6)



ALUOp is determined by the instruction's funct field

Scanned with CamScanner