

Register field in instructions

- First register (Read Register 1): Bits [25-21]
 - Used by all instructions to read the first operand
- Second register (Read Register 2): Bits [20-16]
 - Used by:
 - R-format instructions (eg. `add`, `sub`)
 - store (`sw`) to get the value to write memory
 - branch (`beq`)

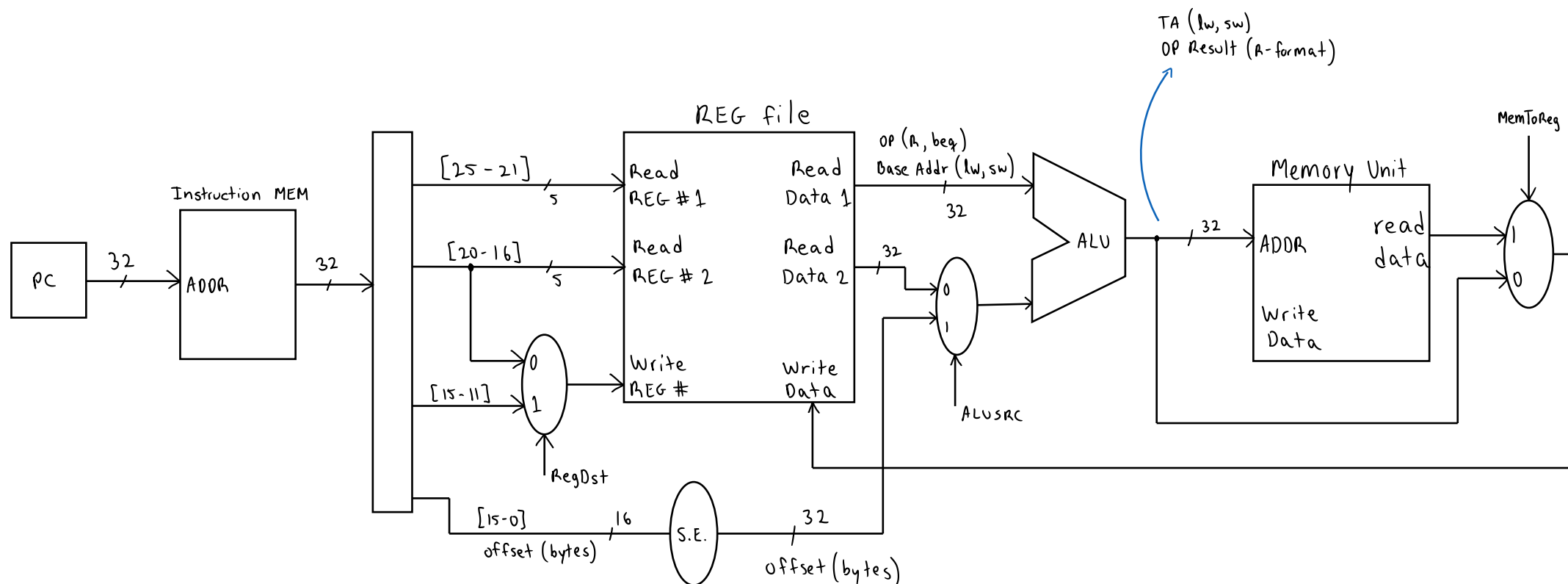
Write Registers

- Load instruction (`lw`)
 - Destination Register in bits [20-16]
- R-format
 - Destination Register in bits [15-11]

Notes for Exam:

- Be able to draw the datapath for any individual instruction
- Know what each multiplexer does, inputs/outputs, and control line behavior
- For individual datapaths, include only relevant parts, no unnecessary control lines
- Be able to trace PC updates through the nested multiplexers using control line values

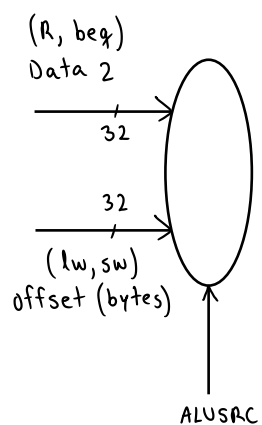
Full Datapath with control lines to accomodate different instructions



ALUSRC

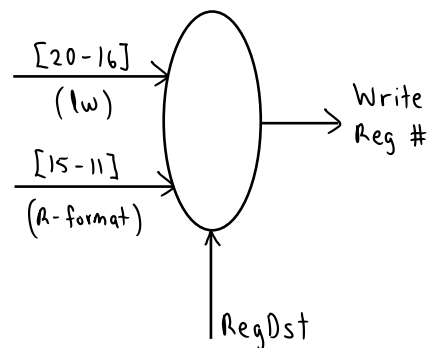
R-format : Data 2
beq : with Reg # [20-26]

lw : 32 bits
sw : offset (bytes)

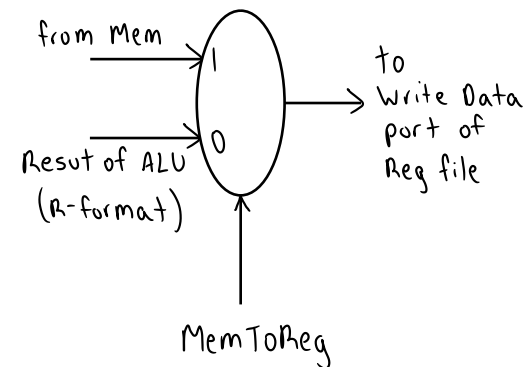


Write Reg

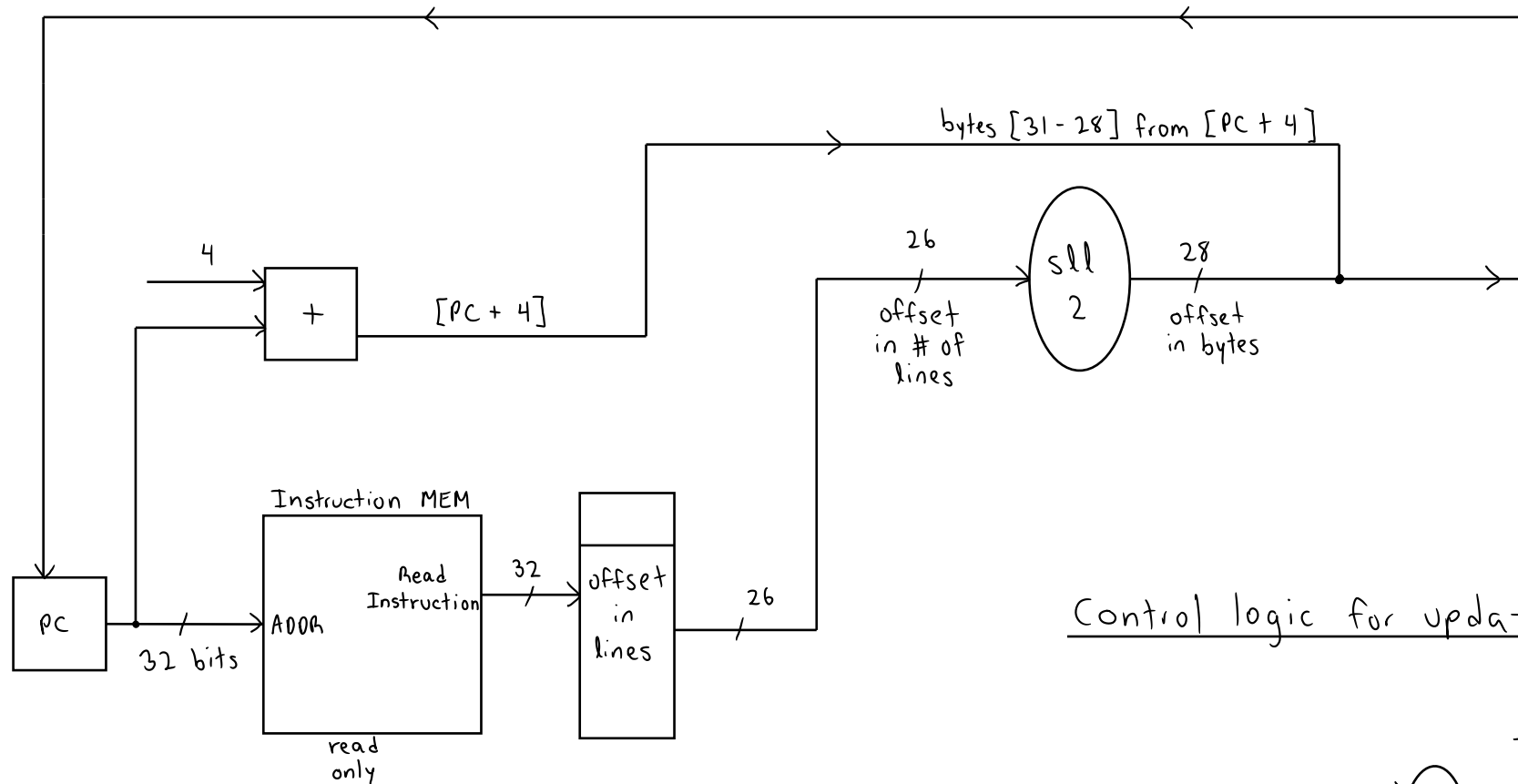
lw : [20-16]
R-format : [15-11]



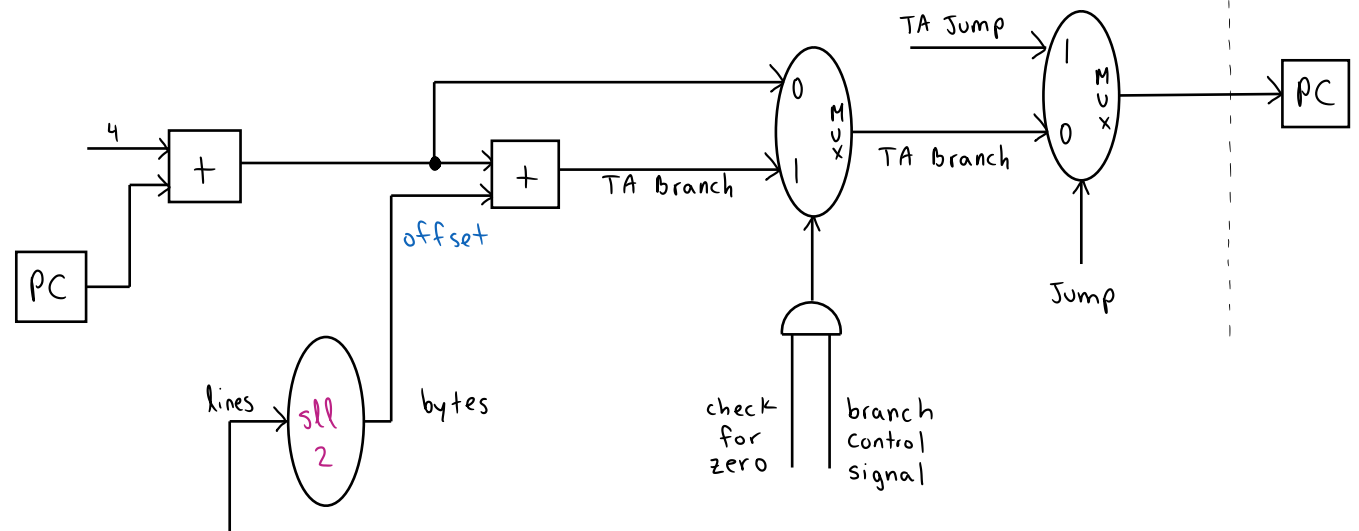
Write Data



Jump



Control logic for updating PC Register



lw
sw
R-format } → PC + 4

beq → if Cond is not satisfied → PC + 4
→ if Cond is satisfied → TA Branch

J → TA jump