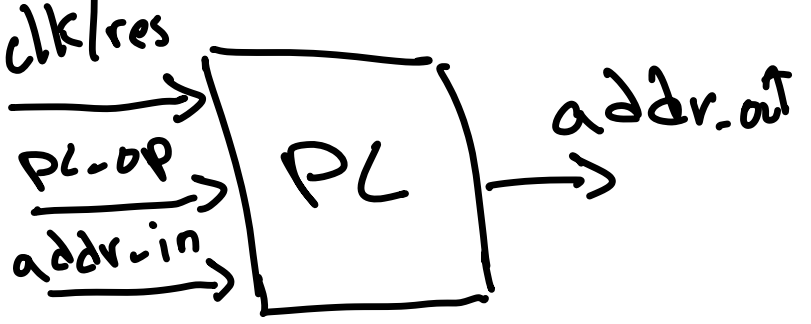
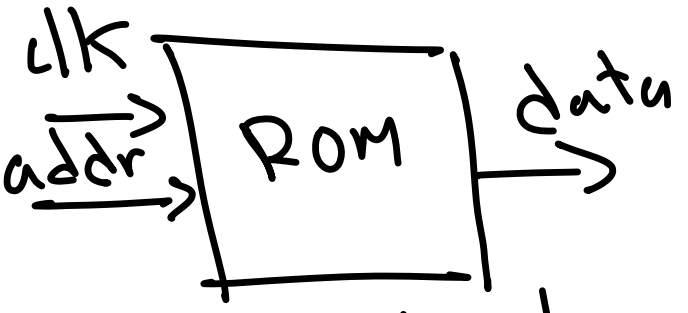
PC



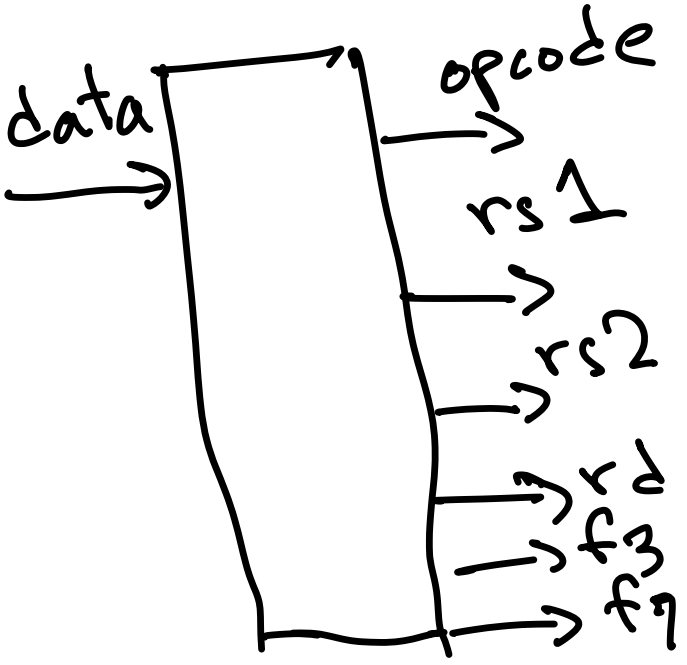
Based on pc-op, either increments addr by 32 bits or changes addr_out to addr-in (jump).

ROM

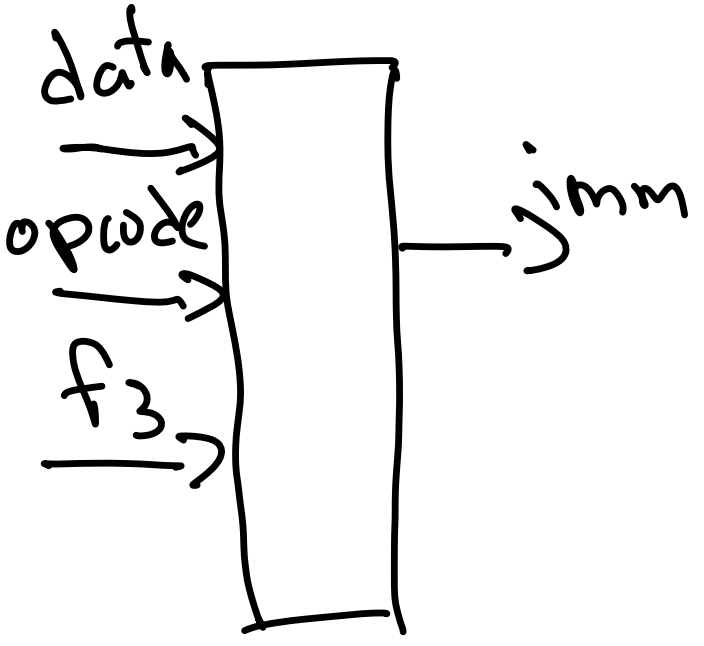


holds instructions

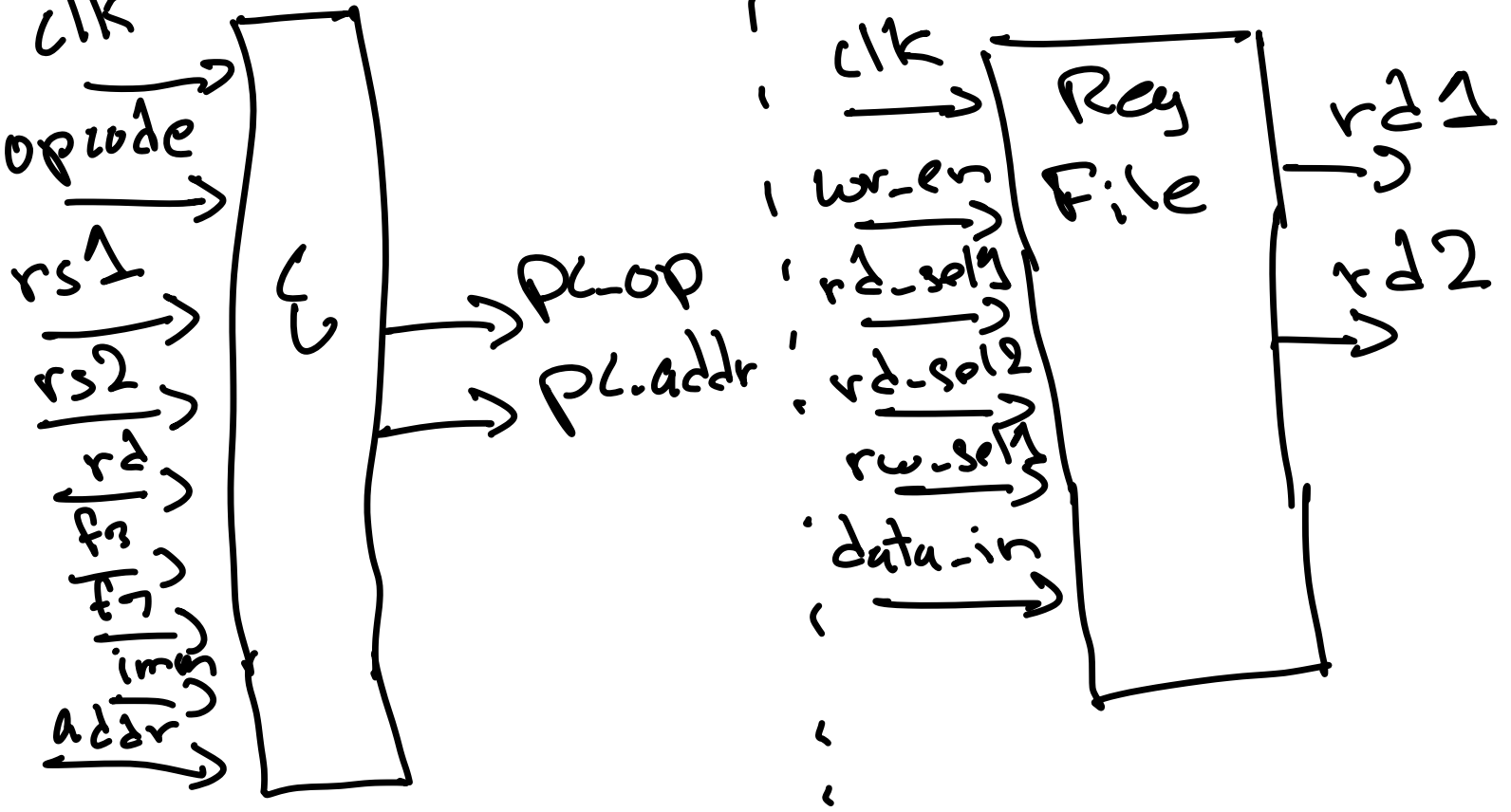
Decoder



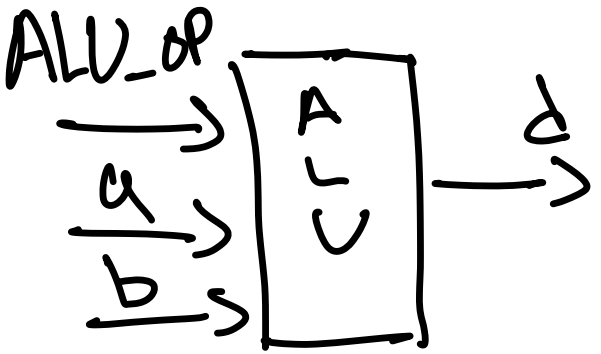
Imm gen



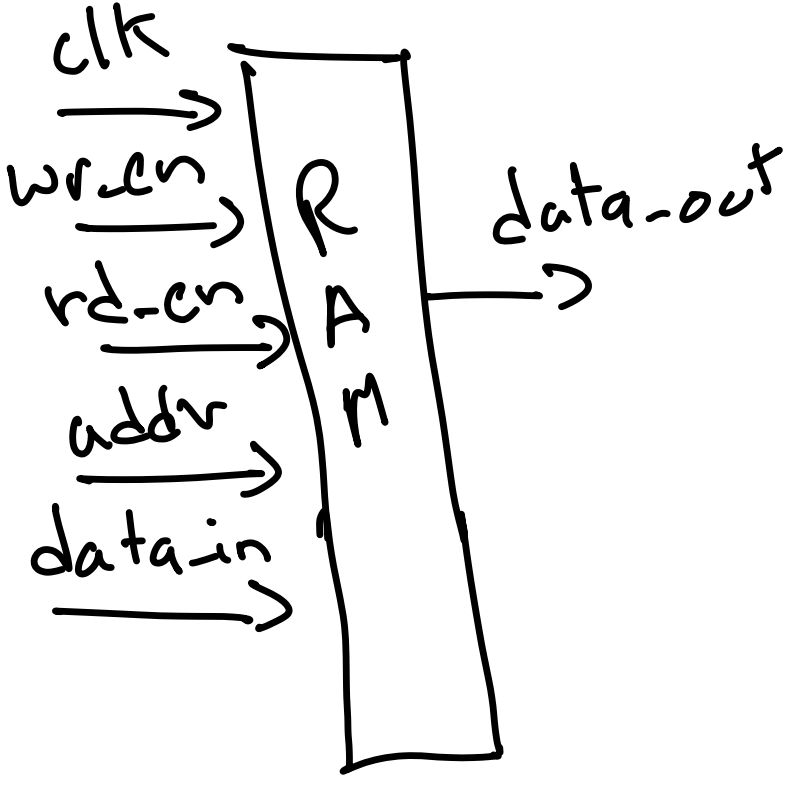
Control Unit | Reg File



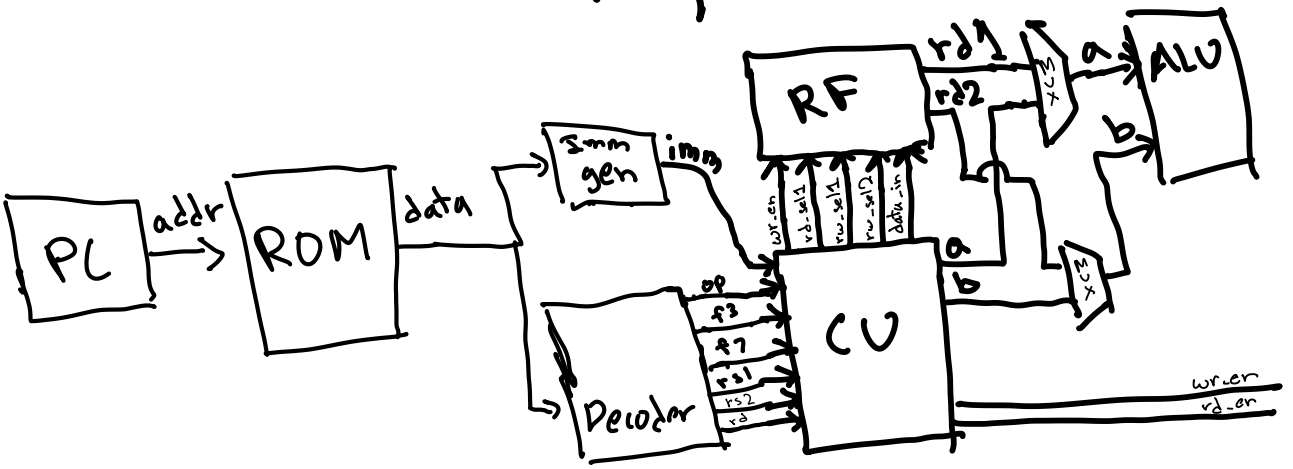
ALU



RAM

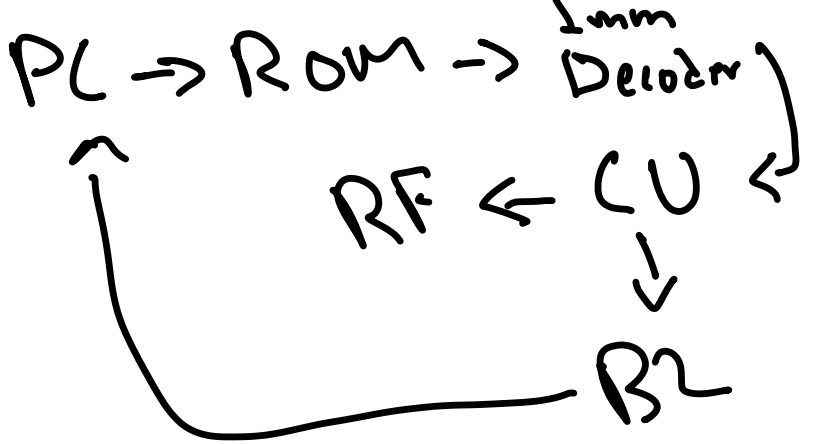


Try 2


PC
addr
ROM
data
Imm gen
imm
RF
rd1
rd2
MUX
a
ALU
b
wr_sel1
rd_sel1
rd_sel2
data_in
wr_en
op
f3
f7
rs1
rs2
rd
Decoder
CU
a
b
MUX
wr_en
rd_en

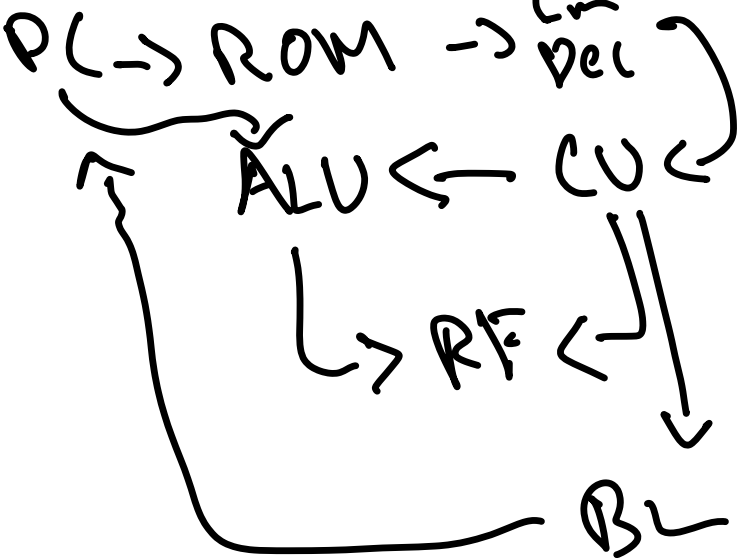
LUI rd, imm

loads a 20-bit imm into upper 20 bits of rd



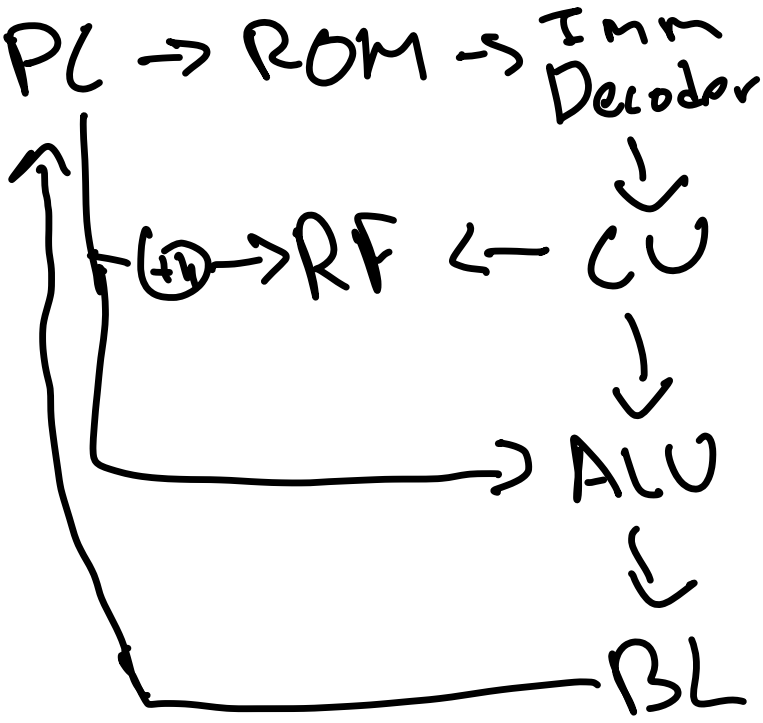
AUIPC rd, imm

adds PC + imm<<12 and stores in rd



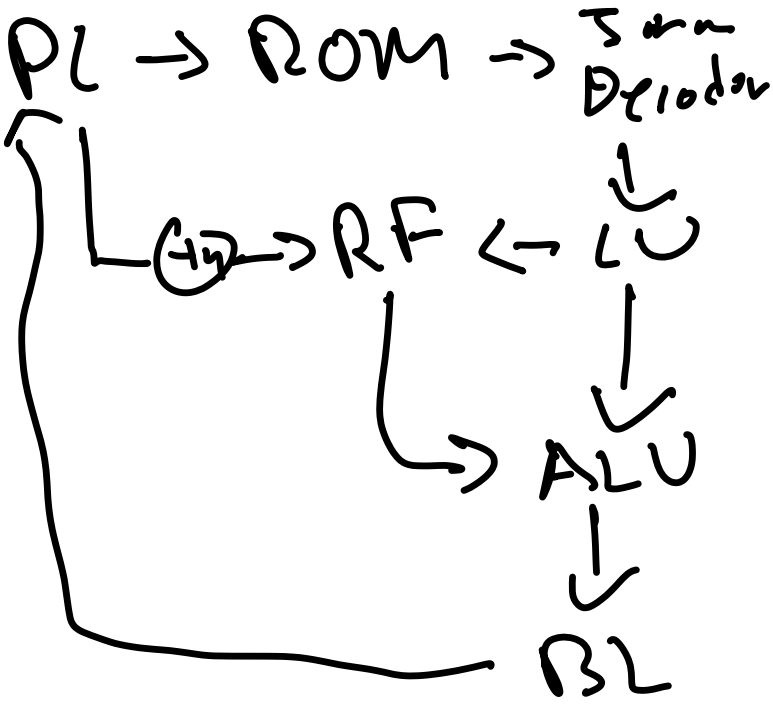
JAL rd, imm

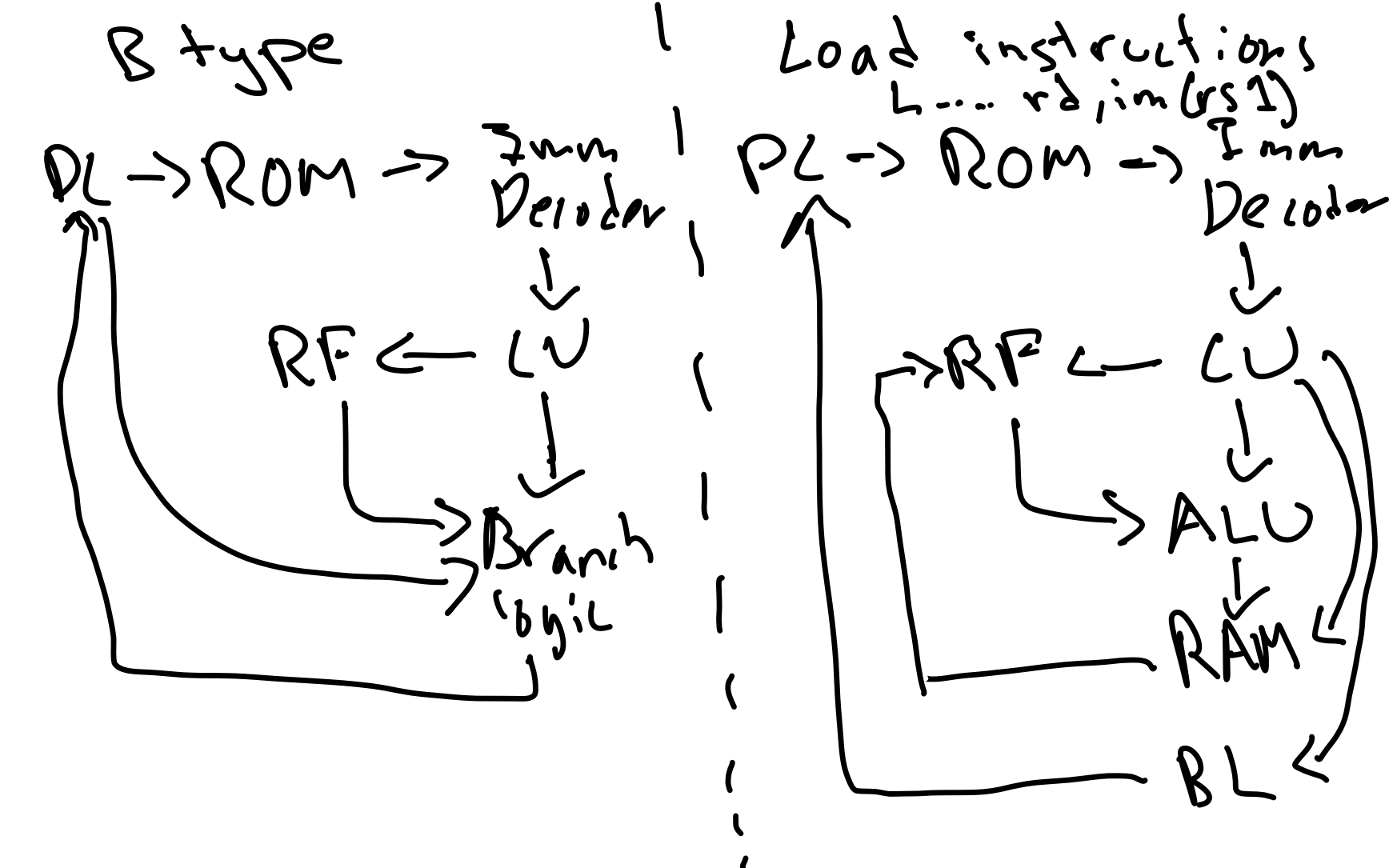
jumps to PC+imm, writes (PC+4) to rd



JALR rd, imm(rs1)

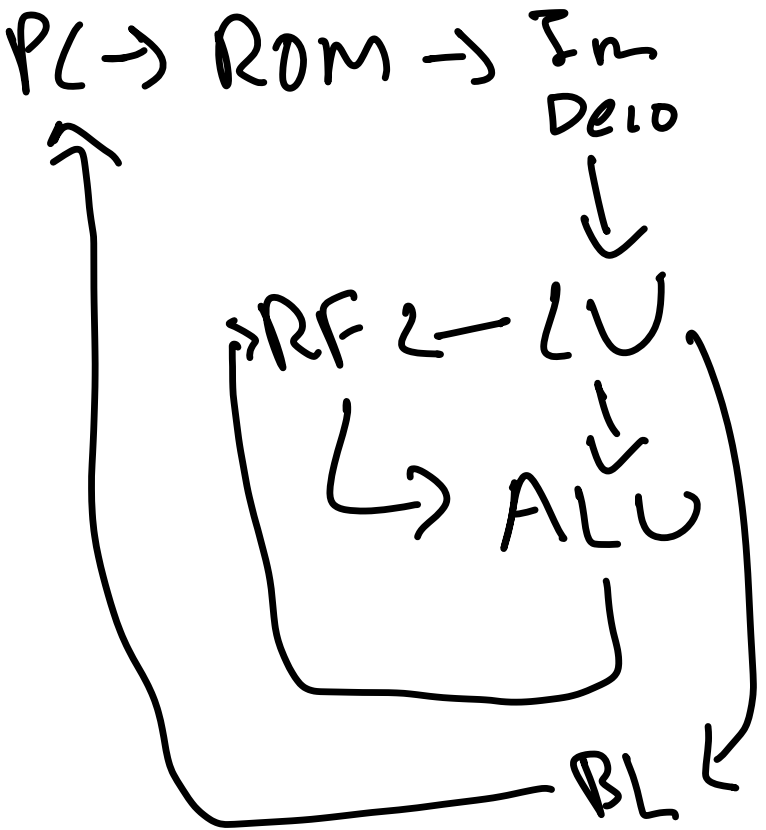
jumps to rs1+imm, writes (PC+4) to rd




B type
PL -> ROM -> Imm Decoder
RF
CU
Branch logic
Load instructions
L... rd, im(rs1)
PL -> ROM -> Imm Decoder
RF
CU
ALU
RAM
BL

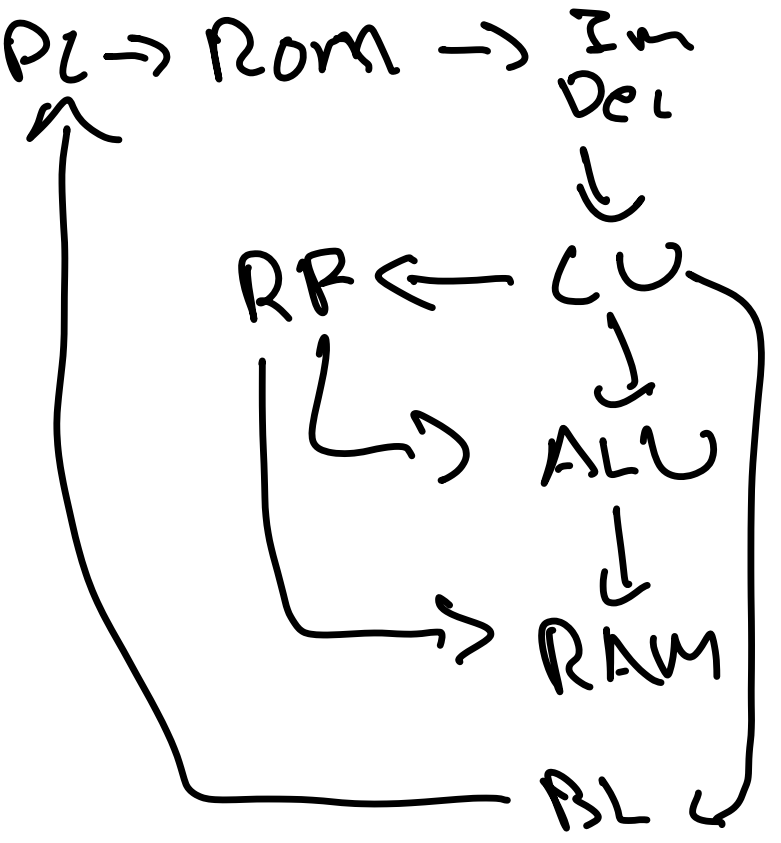
Register-imm instructions

...I rd, rs1, imm



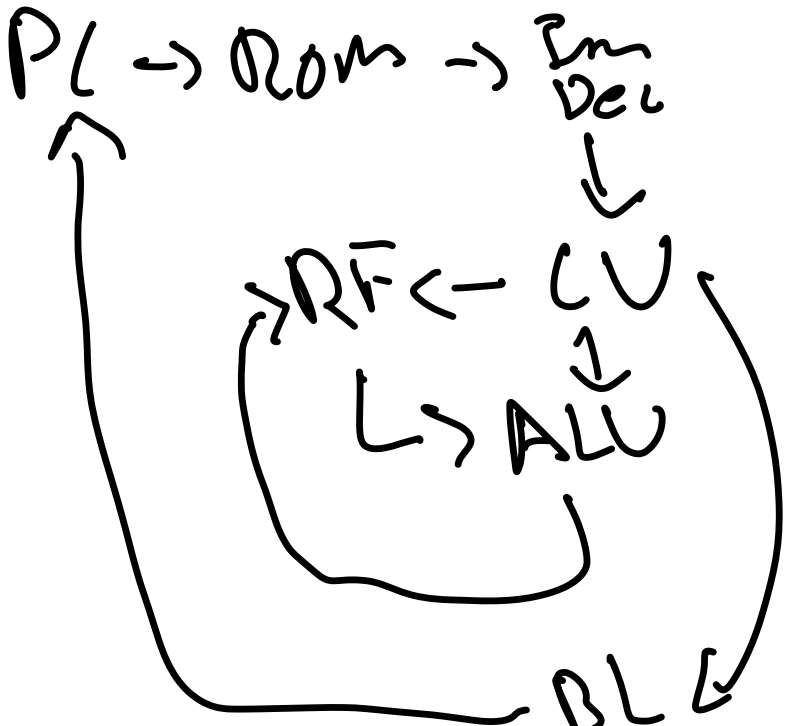
Mem-store instruction

S.... rs2, im(rs1)



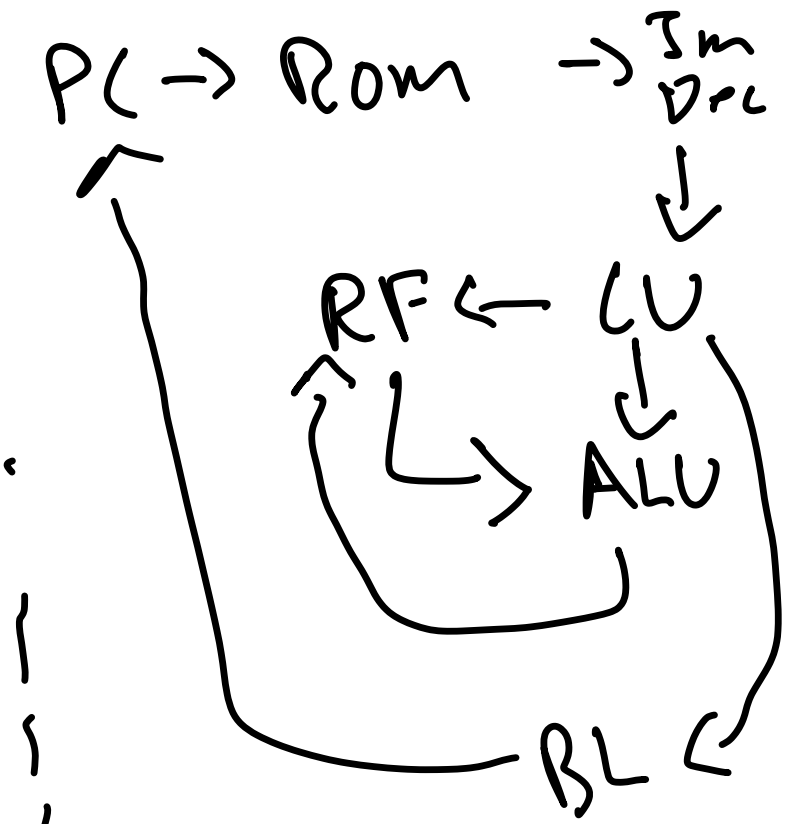
Shift instructions

s— rd, rs1, shamt



R type

--- rd, rs1, rs2



Acc, Imm gen can be connected to ALU instead of through CU.