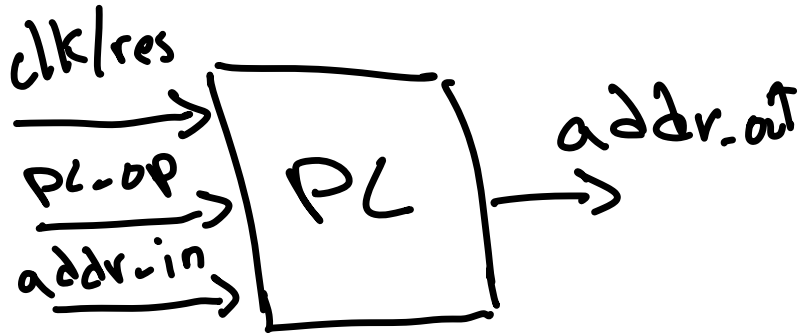
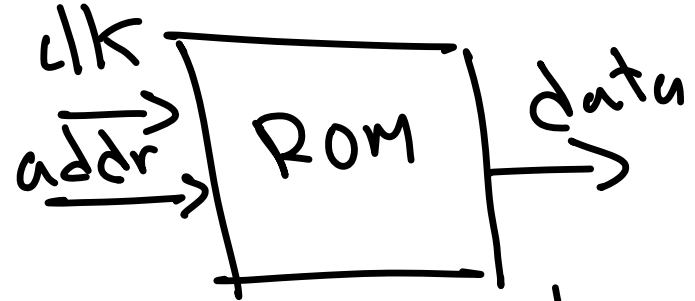


# PC



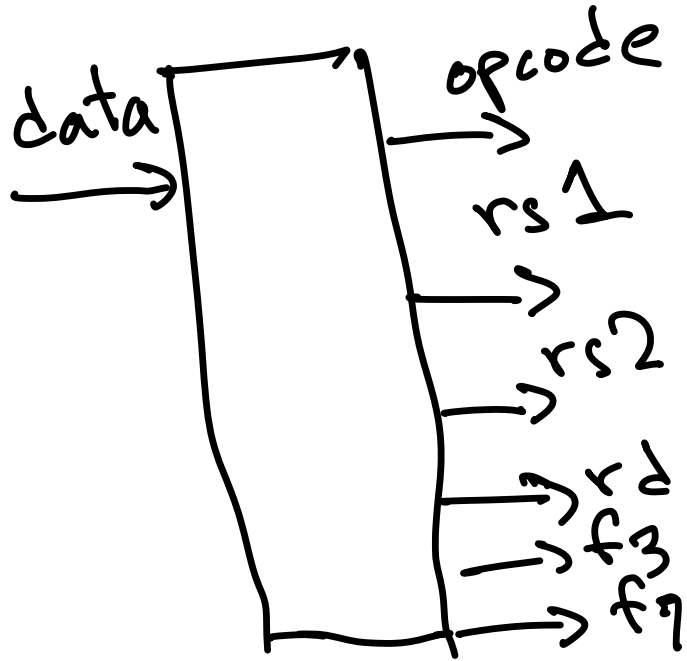
Based on PC-op, either increments addr by 32 bits or changes addr-out to addr-in (jump).

# ROM

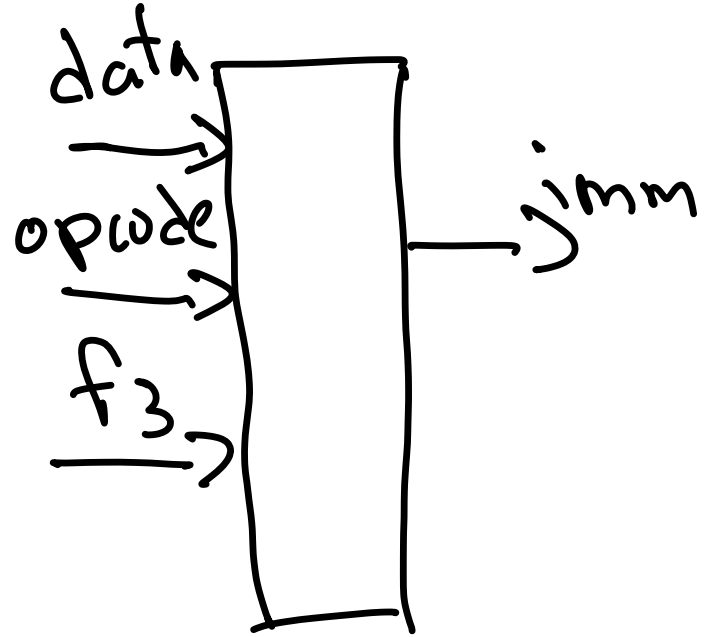


holds instructions

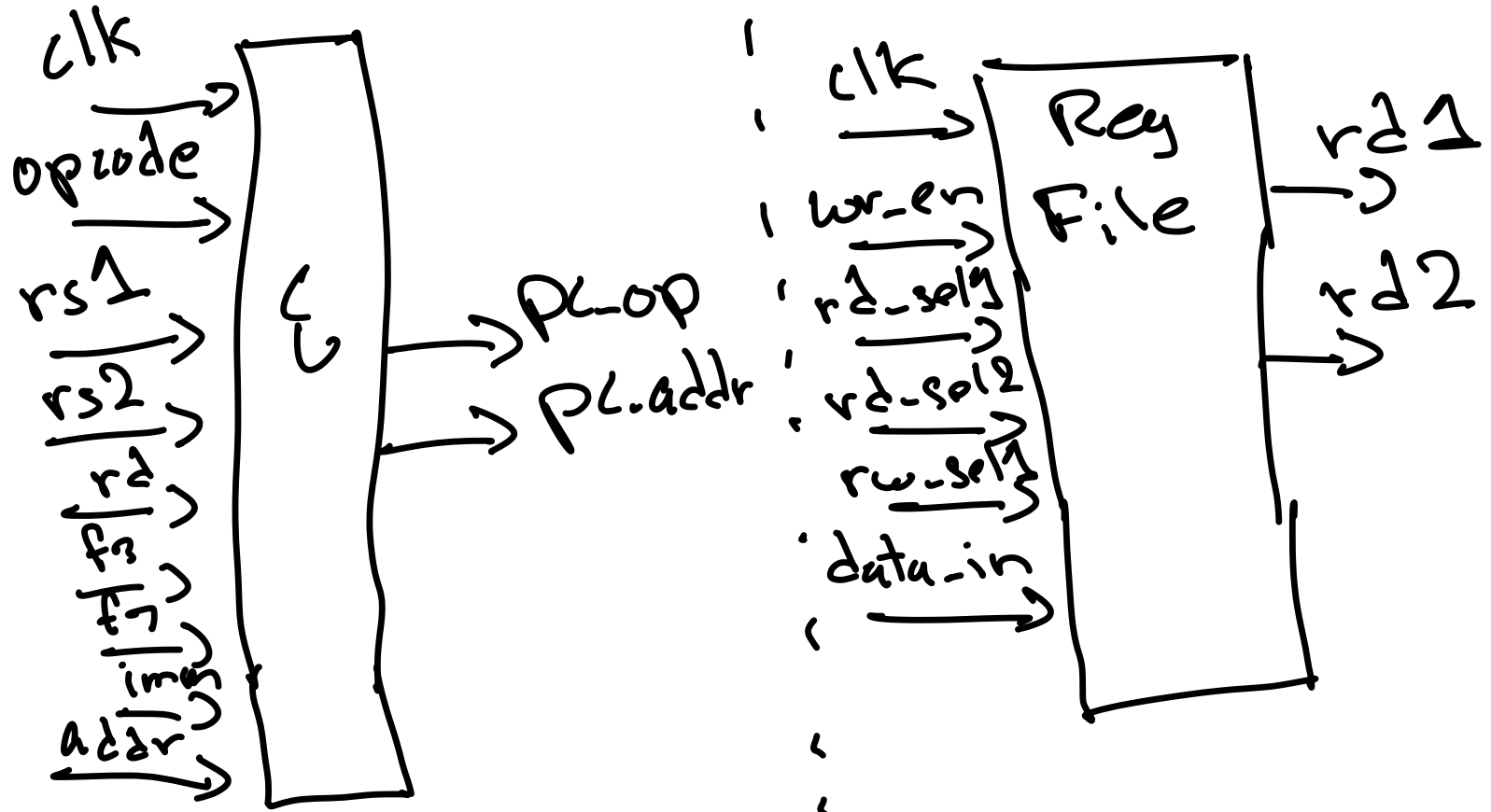
# Decoder



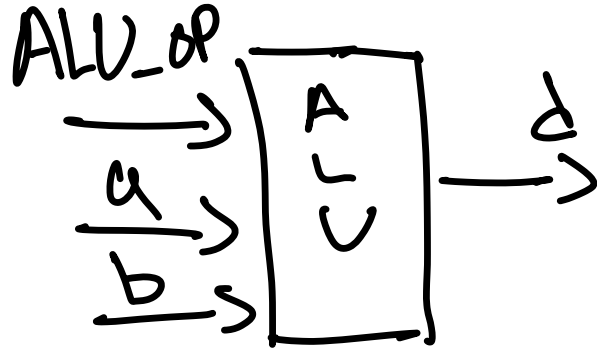
# Imm gen



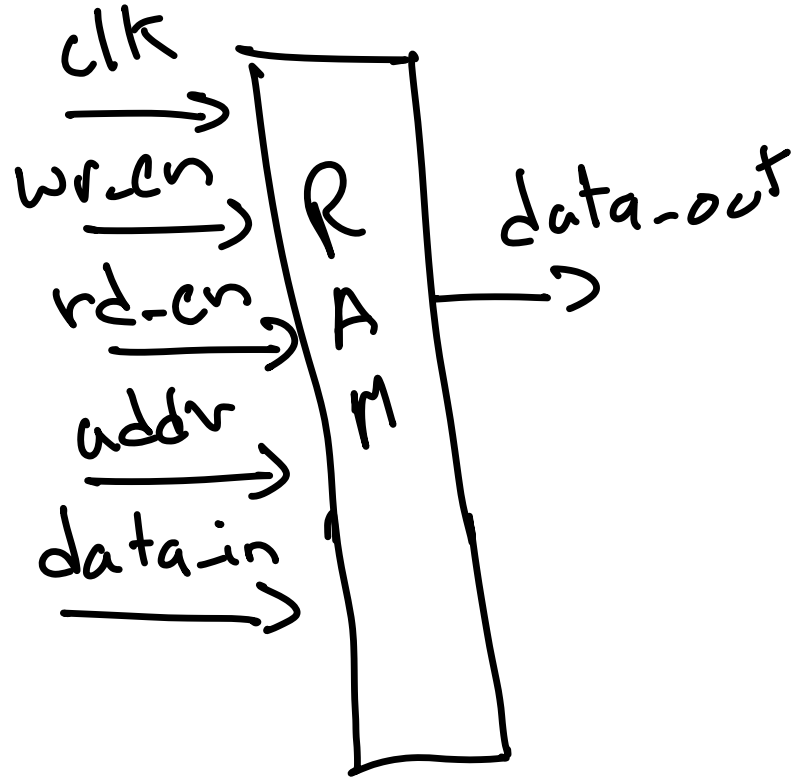
# Control Unit | Reg File



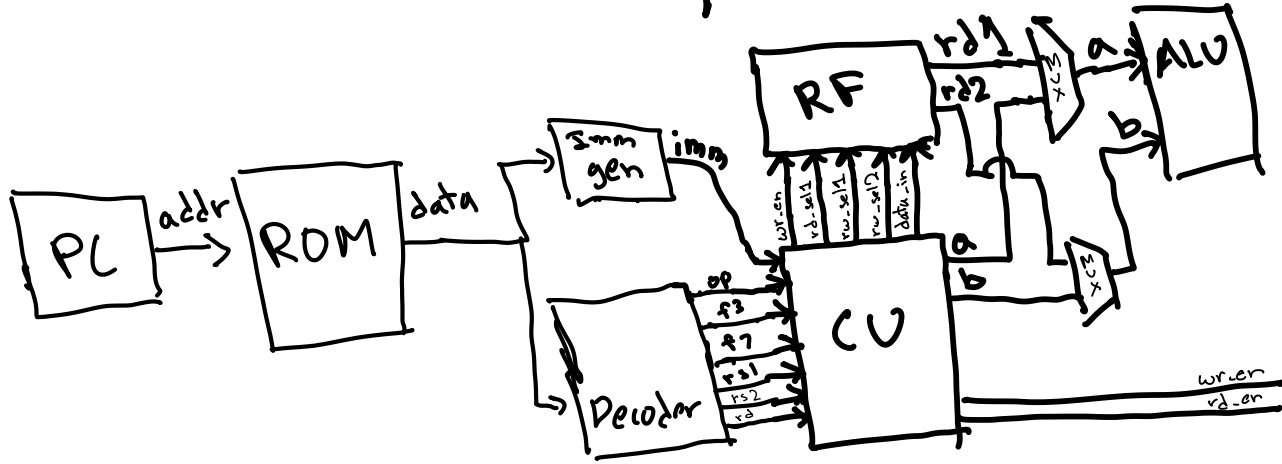
# ALU



# RAM

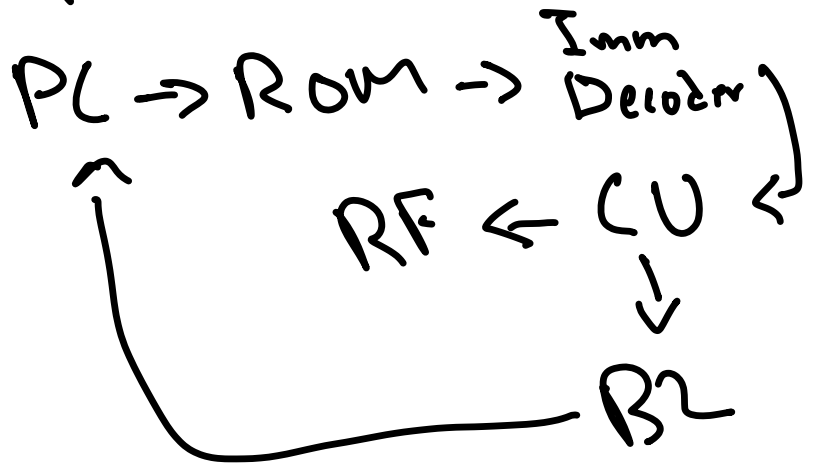


Try 2



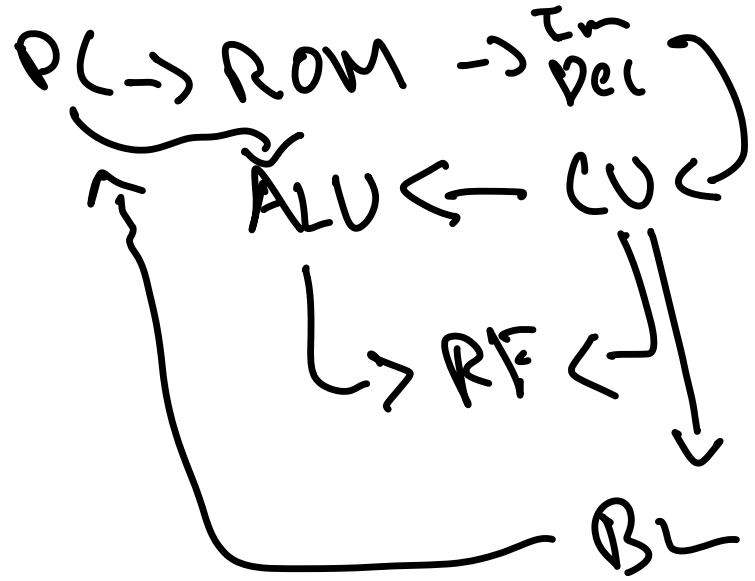
LUI rd, imm

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



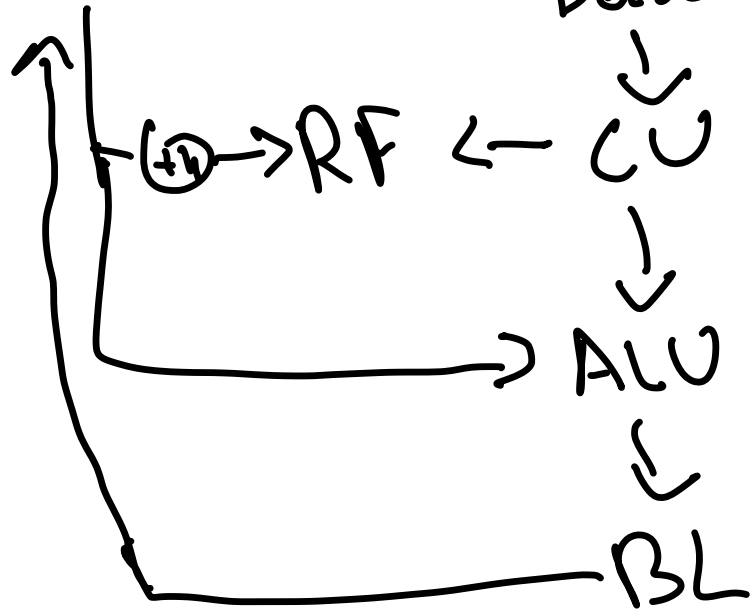
AUI PC rd, imm

adds PC + imm << 12  
and stores in rd



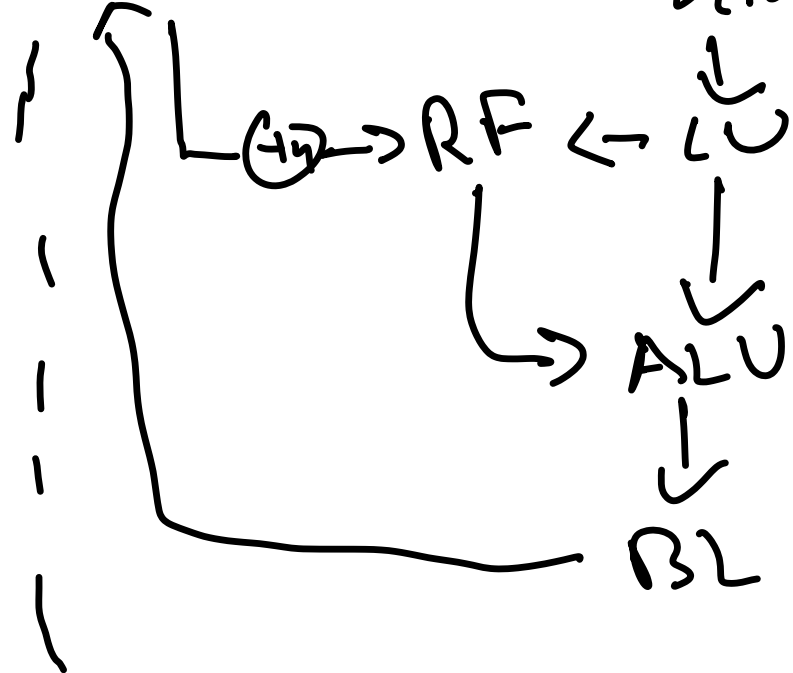
JAL  $rd, imm$   
jumps to  $PC + imm$ ,  
writes  $(PC + 4)$  to  $rd$

$PC \rightarrow ROM \rightarrow Imm$   
Decoder



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

$PC \rightarrow ROM \rightarrow Imm$   
Decoder



```

graph TD
    PC[PC] --> ROM[ROM]
    ROM --> BD[Branch Decoder]
    BD --> RF[RF]
    BD --> BB[Branch 'by'it]
    RF --> BB
    BB --> PC
  
```

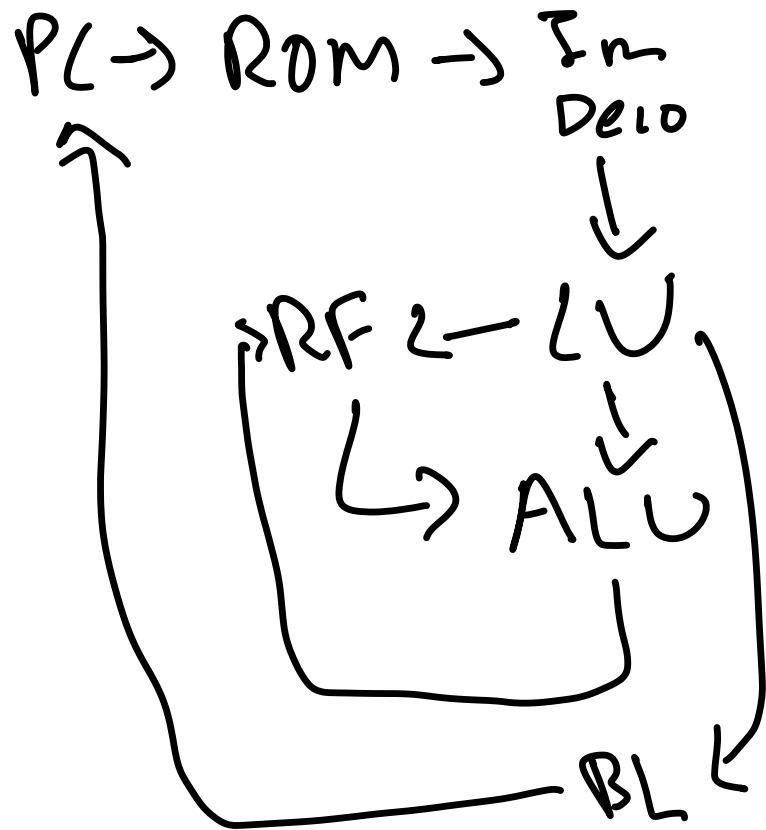
The diagram illustrates a feedback loop in a CPU. It starts with a 'PC' (Program Counter) which points to 'ROM' (Read-Only Memory). From 'ROM', the flow goes to a 'Branch Decoder'. The 'Branch Decoder' has two outputs: one to 'RF' (Register File) and another to 'Branch 'by'it'. The 'RF' also has an output to 'Branch 'by'it'. Finally, the 'Branch 'by'it' unit has a feedback loop that returns to the 'PC'.

```
graph TD; PL[PL] --> ROM[ROM]; ROM --> Imm[Imm Decoder]; Imm --> CU[CU]; CU --> RF[RF]; CU --> ALU[ALU]; CU --> RAM[RAM]; CU --> BL[BL]; RF --> ALU; RF --> RAM; RF --> PL; ALU --> RAM; RAM --> BL; BL --> PL;
```

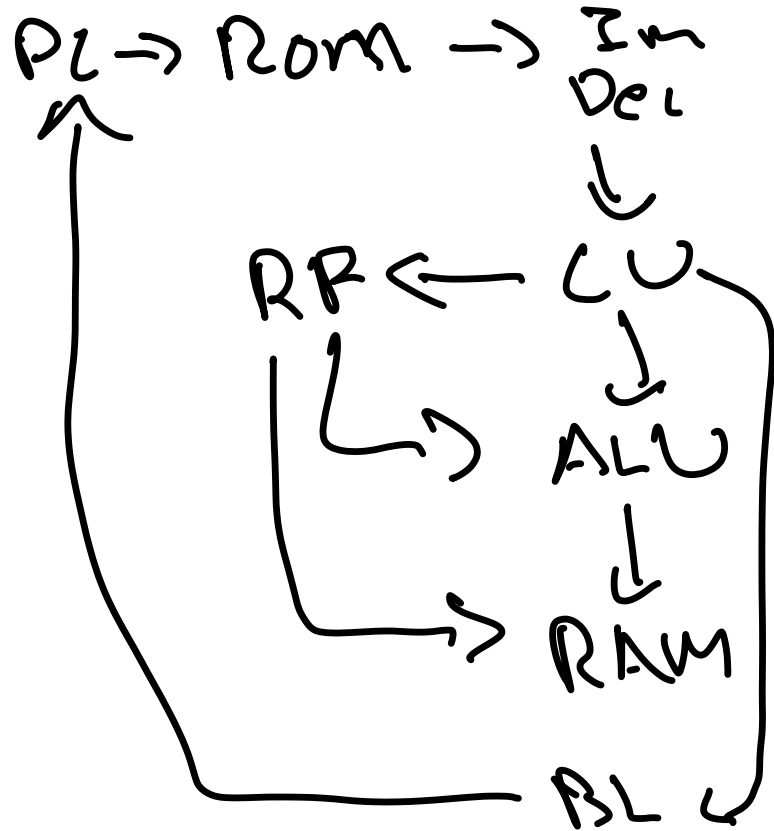
A hand-drawn block diagram of a computer system architecture. At the top left, 'PL' (Program Loader) has an arrow pointing to 'ROM' (Read-Only Memory). From 'ROM', an arrow points to 'Imm Decoder' (Immediate Decoder). Below 'Imm Decoder' is 'CU' (Control Unit), with a downward arrow from the decoder. To the left of 'CU' is 'RF' (Register File), with an arrow pointing from 'CU' to 'RF'. Below 'CU' is 'ALU' (Arithmetic Logic Unit), with a downward arrow from 'CU'. Below 'ALU' is 'RAM' (Random Access Memory), with a downward arrow from 'ALU'. At the bottom is 'BL' (Bus Logic), with a downward arrow from 'RAM'. A large curved arrow on the right side connects 'CU', 'ALU', 'RAM', and 'BL' back to 'PL'. Additionally, an arrow points from 'RF' to 'ALU', and another from 'RF' to 'RAM'. A long arrow also originates from 'RF' and points directly back to 'PL'.



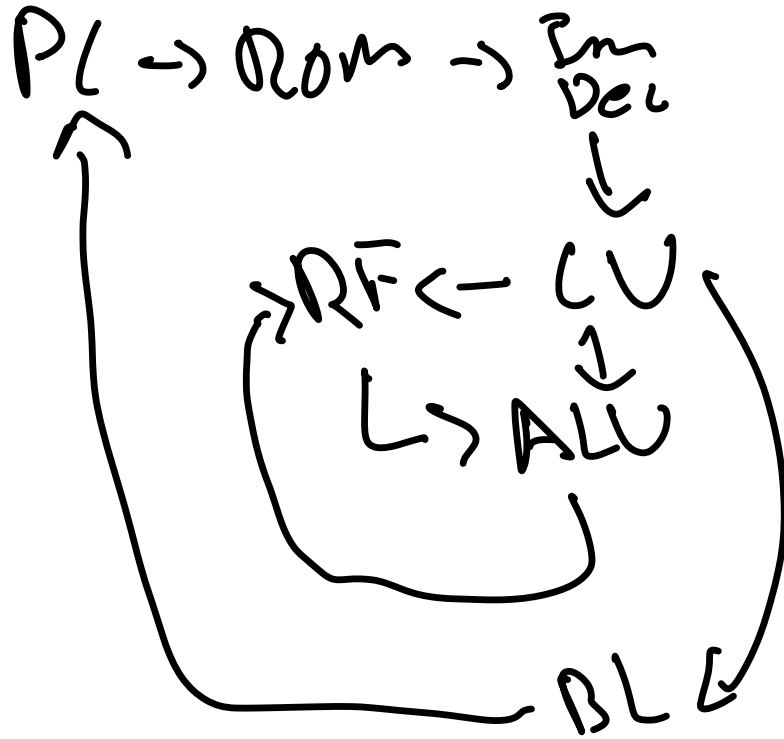
Register-imm instructions  
 $\dots I \ rd, rs, im$



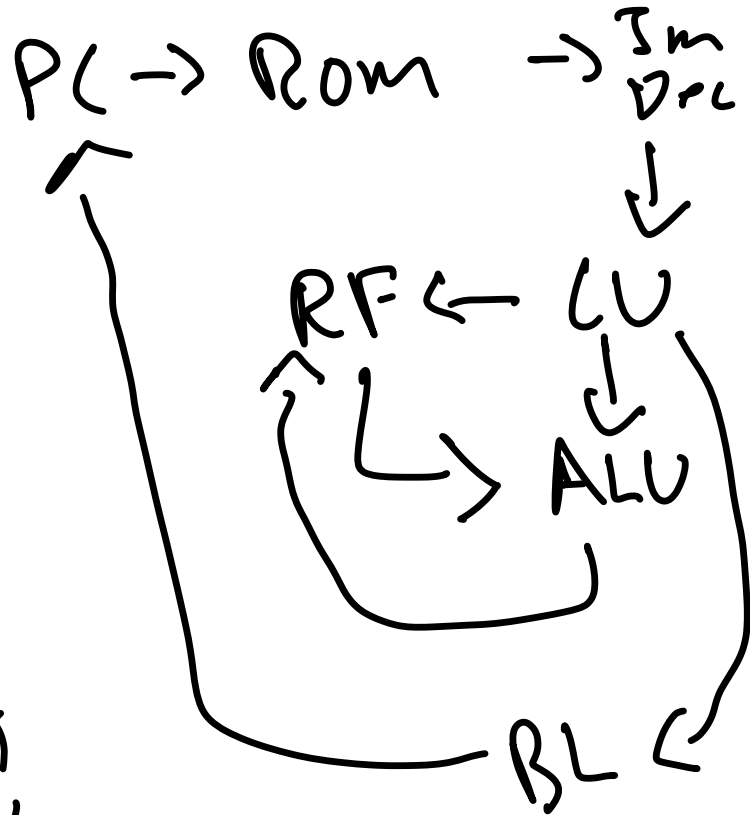
Mem-store instructions  
 $S \dots rs2, im(rs1)$



Shift instructions  
S — rd, rs1, shamt



R-type  
... rd, rs1, rs2



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

