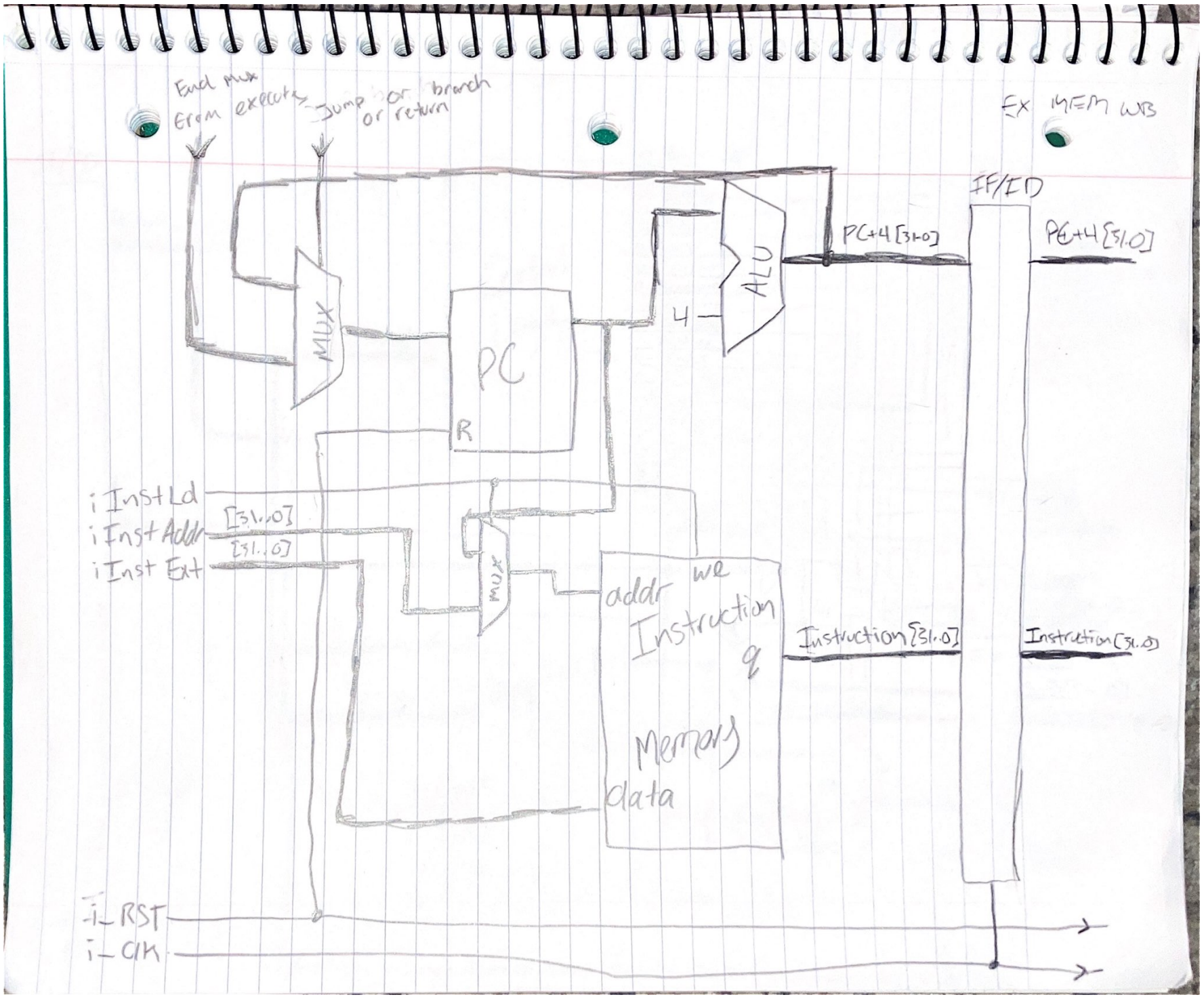
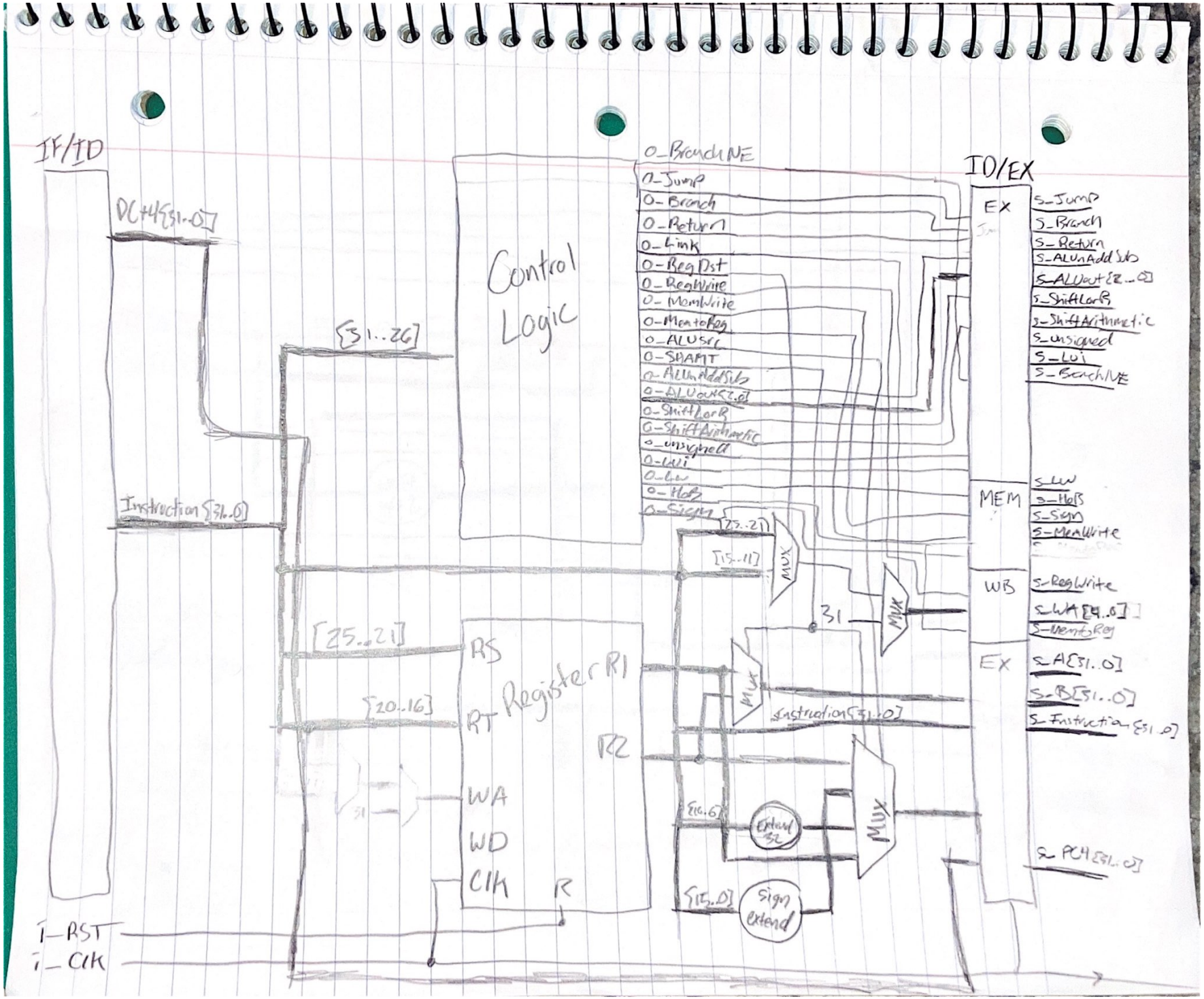
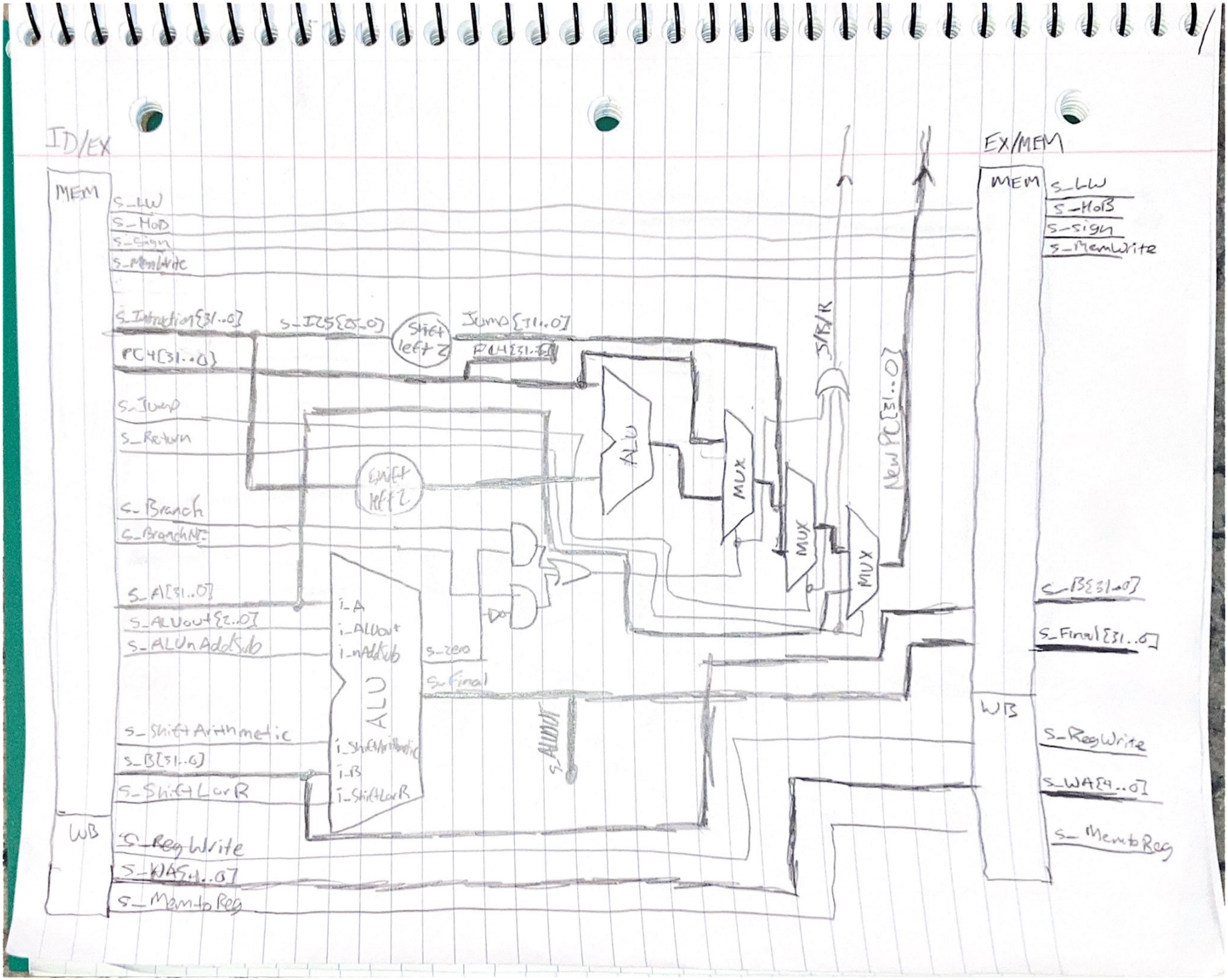
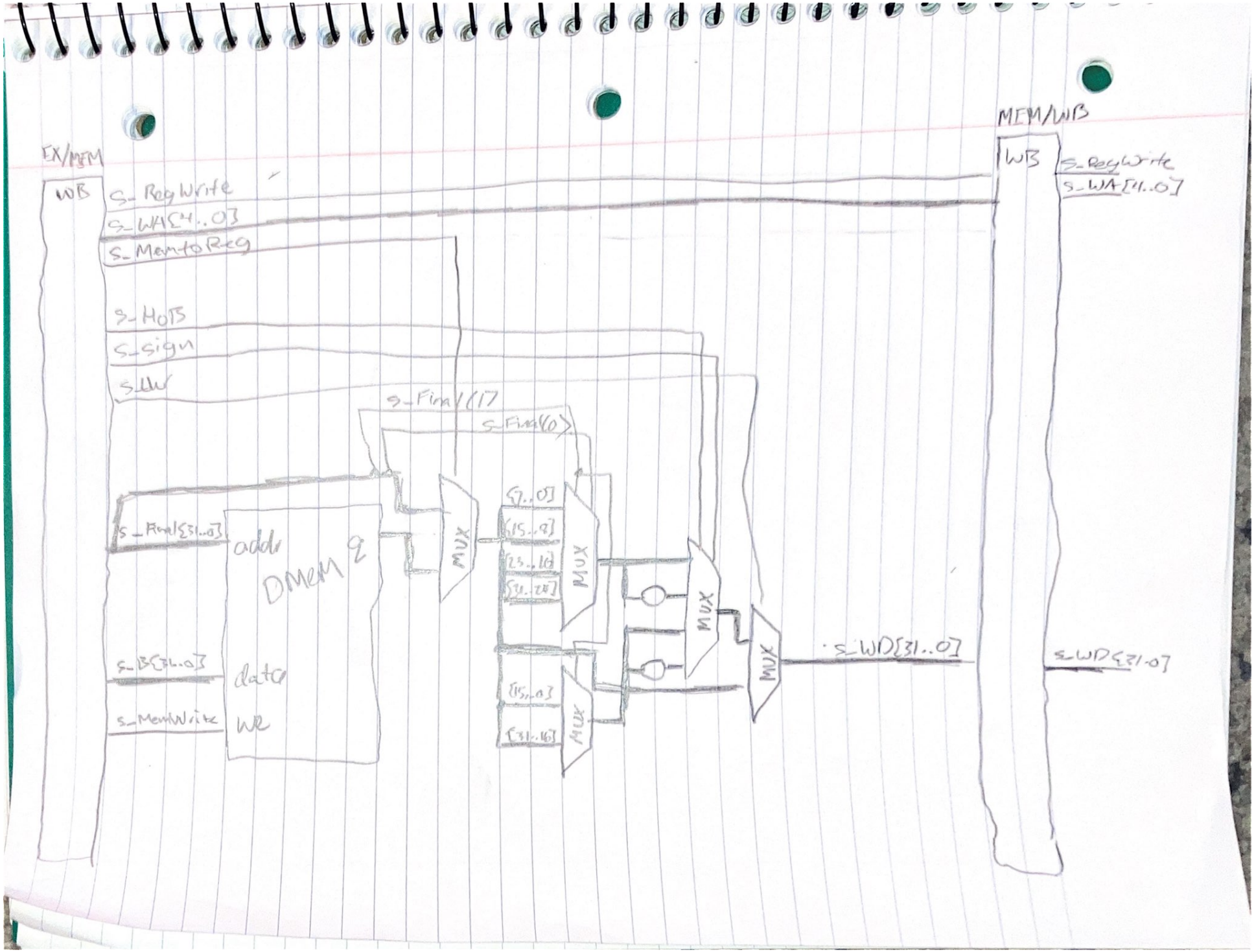

Exmem mux
From execute
Jump or branch
or return
EX MEM WB
IF/ID
PC+4 [31:0]
PC+4 [31:0]
ALU
4
MUX
PC
R
iInstLd
iInstAddr [31:0]
iInst Ext [31:0]
MUX
addr
we
Instruction
q
Memory
data
Instruction [31:0]
Instruction [31:0]
i_RST
i_CLK


IF/ID
PC+4[31..0]
[31..26]
Instruction[31..0]
Control Logic
O_BranchNE
O_Jump
O_Branch
O_Return
O_Link
O_RegDst
O_RegWrite
O_MemWrite
O_MemtoReg
O_ALUSrc
O_SHAMT
O_ALUmAddSub
O_ALUout[2..0]
O_ShiftLorR
O_ShiftArithmetic
O_unsigned
O_LUI
O_Lw
O_Hob
O_Sign
ID/EX
EX
MEM
WB
S_Jump
S_Branch
S_Return
S_ALUmAddSub
S_ALUout[2..0]
S_ShiftLorR
S_ShiftArithmetic
S_unsigned
S_LUI
S_BranchNE
S_LW
S_HoB
S_Sign
S_MemWrite
S_RegWrite
S_WA[4..0]
S_MemtoReg
S_A[31..0]
S_B[31..0]
S_Instruction[31..0]
S_PC4[31..0]
[25..21]
[20..16]
[15..11]
[25..0]
RS
RT
RD
WA
WD
Clk
R
Register RI
MUX
31
Extend 32
Sign extend
[15..0]
[16..6]
Instruction[31..0]
I_RST
I_Clk


ID/EX
MEM
s_LW
s_MoB
s_Sign
s_MemWrite
s_Instruction[31..0]
s_I25[25..0]
Shift left 2
Jump[31..0]
PC4[31..0]
s_Jump
s_Return
Shift left 2
s_Branch
s_BranchNE
s_A[31..0]
s_ALUout[2..0]
s_ALUnAddSub
s_ShiftArithmetic
s_B[31..0]
s_ShiftLorR
WB
s_RegWrite
s_WA[4..0]
s_MemtoReg
i_A
i_ALUout
i_nAddSub
i_ShiftArithmetic
i_B
i_ShiftLorR
ALU
s_zero
s_Final
s_ALUout
ALU
MUX
MUX
MUX
S/R/R
New PC[31..0]
EX/MEM
MEM
s_LW
s_MoB
s_Sign
s_MemWrite
s_B[31..0]
s_Final[31..0]
WB
s_RegWrite
s_WA[4..0]
s_MemtoReg


EX/MEM
WB
s_RegWrite
s_WA[4..0]
s_MemtoReg
s_HOTS
s_sign
s_LW
s_Final[17]
s_Final[0]
s_Final[31..0]
addr
DMem 2
s_B[31..0]
data
s_MemWrite
we
MUX
[7..0]
[15..8]
[23..16]
[31..24]
[15..0]
[31..16]
s_WD[31..0]
MEM/WB
WB
s_RegWrite
s_WA[4..0]
s_WD[31..0]