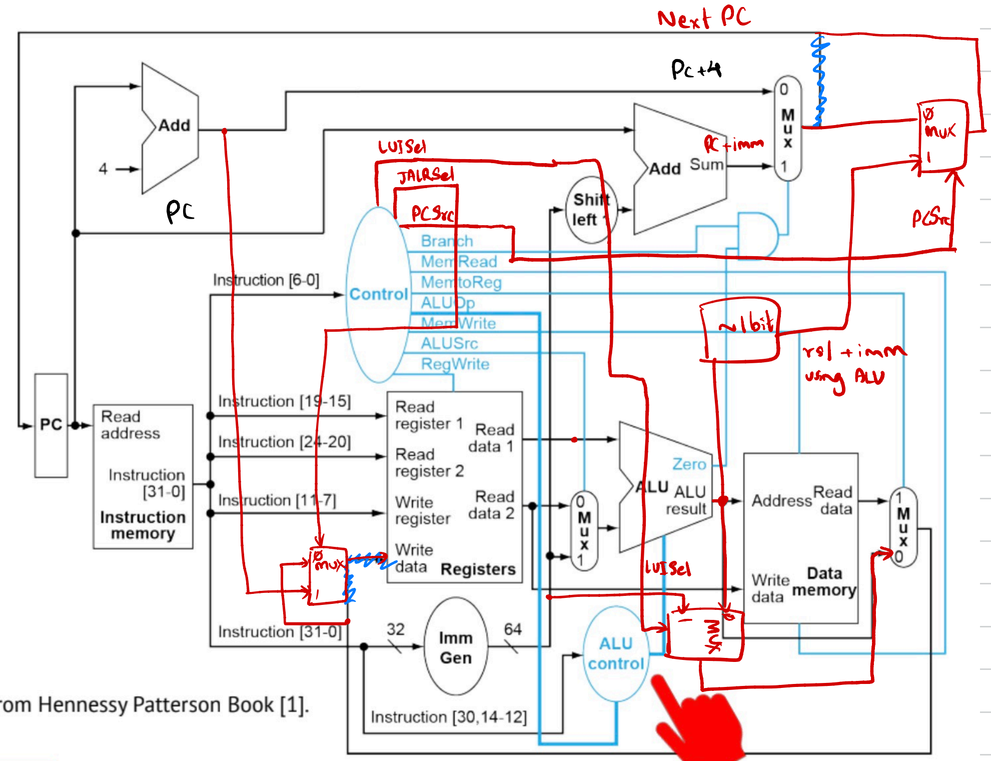
M116C - CA1 Report

KHANT NYI HLAING

Datapath Design



*Images were taken from Hennessy Patterson Book [1].

I used the design given in one of the lecture slides and started from there. Most instructions are solvable with the standard datapath and control signals table from the lecture.

For BNE, I added two 2-to-1 MUXes for PC selection: the first MUX chooses between PC+4 and PC+immediate based on the branch condition (Branch signal AND !zero flag), and the second MUX handles JALR. I also added a PCSrc control signal to coordinate these.

For JALR, I added a JALRSel control signal to select PC+4 as the writeback data (instead of ALU result or memory data) to save the return address. I extended the PC update logic with a second MUX level that selects (rs1+immediate)&~1 as the jump target when (PCSrc==2). This allows jumping to an address stored in a register.

For LUI, I added a LUISel control signal that, when active, bypasses the ALU and uses the immediate value directly as the result, since LUI only needs to load the immediate into the upper 20 bits of the destination register.

Control Signals

Instruction	Opcode	RegWrite	AluSrc	Branch	MemRe	MemWr	MemtoReg	ALUOp	LUI Sel	JALRSel	PC Src
R-type	0110011	1	0	0	0	0	0	10	0	0	0
I-type	0010011	1	1	0	0	0	0	00	0	0	0
lw, lbu	0000011	1	1	0	1	0	1	00	0	0	0
sw, SH	0100011	0	1	0	0	1	~~0~~ 1	00	0	0	0
~~beq~~	~~1100011~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~-~~	0	0	0
BNE	1100011	0	0	1	0	0	1	01	0	0	1
JALR	1100111	1	1	0	0	0	0	00	0	1	2
LUI	0110111	1	1	0	0	0	0	00	1	0	0

R-type : ADD, SUB, SRA, AND

I-type : ADDI, ORI, SLTIU