We would make our words as structs, as suggested by the doc. This way, we won't have to manually set the 32-bit value, but instead only the signals we want. In essence, we are indirectly creating 32-bit values

We traced the stages each instruction would go through. Obviously all instructions need to be fetched, decoded and then executed. Only select instructions need to interface with memory (cache or physical, but that is abstracted to us at this moment) and only select instructions need to go through the writeback stage (write to registers). We then used these stages to determine what relevant signals would need to be set at any point of the instruction's execution.

Opcode	Stages	Control Word (Signals set)
LUI	IF → ID → EX → WB	ld_mar marmux_sel = pc_out ld_mdr mem_read = 1 load_ir ld_regfile regfilemux_sel = u_imm ld_pc pcmux_sel = pc_plus4

AUIPC	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = u_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

JAL	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out

		alumux2_sel = j_imm
		ld_regfile
		regfilemux_sel = pc_plus4
		ld_pc
		pcmux_sel = alu_mod2

JALR	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = pc_plus4
		ld_pc
		pcmux_sel = alu_mod2

BEQ	IF → ID → EX	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = b_imm
		ld_pc
		pcmux_sel = {{1'b0}, br_en}

BNE	IF → ID → EX	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = b_imm
		ld_pc
		pcmux_sel = {{1'b0}, br_en}

BLT	IF → ID → EX	ld_mar marmux_sel = pc_out ld_mdr mem_read = 1 load_ir

		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = b_imm
		ld_pc
		pcmux_sel = {{1'b0}, br_en}

BGE	IF → ID → EX	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = b_imm
		ld_pc
		pcmux_sel = {{1'b0}, br_en}

BLTU	IF → ID → EX	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = b_imm
		ld_pc
		pcmux_sel = {{1'b0}, br_en}

BGEU	IF → ID → EX	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = pc_out
		alumux2_sel = b_imm
		ld_pc
		pcmux_sel = {{1'b0}, br_en}

LB	IF → ID → EX → MEM → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out

		alumux2_sel = i_imm ld_mar marmux_sel = alu_out ld_mdr mem_read = 1 ld_regfile **regfilemux_sel = lb** ld_pc pcmux_sel = pc_plus4

LH	IF → ID → EX → MEM → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_mar
		marmux_sel = alu_out
		ld_mdr
		mem_read = 1
		ld_regfile
		regfilemux_sel = lh
		ld_pc
		pcmux_sel = pc_plus4

LW	IF → ID → EX → MEM → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_mar
		marmux_sel = alu_out
		ld_mdr
		mem_read = 1
		ld_regfile
		regfilemux_sel = lw
		ld_pc
		pcmux_sel = pc_plus4

LBU	IF → ID → EX → MEM → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1

		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_mar
		marmux_sel = alu_out
		ld_mdr
		mem_read = 1
		ld_regfile
		regfilemux_sel = lbu
		ld_pc
		pcmux_sel = pc_plus4

LHU	IF → ID → EX → MEM → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_mar
		marmux_sel = alu_out
		ld_mdr
		mem_read = 1
		ld_regfile
		regfilemux_sel = lhu
		ld_pc
		pcmux_sel = pc_plus4

SB	IF → ID → EX → MEM	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = s_imm
		ld_mar
		marmux_sel = alu_out
		ld_data_out
		mem_write
		mem_byte_enable = 0001 (shifted accordingly)
		ld_pc
		pcmux_sel = pc_plus4

SH	IF → ID → EX →	ld_mar

	MEM	marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = s_imm
		ld_mar
		marmux_sel = alu_out
		ld_data_out
		mem_write
		mem_byte_enable = 0011 (shifted accordingly)
		ld_pc
		pcmux_sel = pc_plus4

SW	IF → ID → EX → MEM	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = s_imm
		ld_mar
		marmux_sel = alu_out
		ld_data_out
		mem_write
		mem_byte_enable = 1111
		ld_pc
		pcmux_sel = pc_plus4

ADDI	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SLTI	IF → ID → EX → WB	ld_mar marmux_sel = pc_out ld_mdr

		mem_read = 1
		load_ir
		
		cmpop = blt
		cmpmux_sel = i_imm
		ld_regfile
		regfilemux_sel = br_en
		ld_pc
		pcmux_sel = pc_plus4

SLTIU	IF → ID → EX → WB	ld_mar marmux_sel = pc_out ld_mdr mem_read = 1 load_ir **cmpop = bltu** cmpmux_sel = i_imm ld_regfile regfilemux_sel = br_en ld_pc pcmux_sel = pc_plus4

XORI	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_xor
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

ORI	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_or
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

ANDI	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_and
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SLLI	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_sll
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SRLI	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_srl
		alumux1_sel = rs1_out
		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SRAI	IF → ID → EX → WB	ld_mar marmux_sel = pc_out ld_mdr mem_read = 1 load_ir **aluop = alu_sra** alumux1_sel = rs1_out

		alumux2_sel = i_imm
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

ADD	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_add
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SUB	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_sub
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SLL	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_sll
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SLT	IF → ID → EX → WB	ld_mar marmux_sel = pc_out

		ld_mdr mem_read = 1 load_ir **cmpop = blt** cmpmux_sel = rs2_out ld_regfile regfilemux_sel = br_en ld_pc pcmux_sel = pc_plus4
SLTU	IF → ID → EX → WB	ld_mar marmux_sel = pc_out ld_mdr mem_read = 1 load_ir **cmpop = bltu** cmpmux_sel = rs2_out ld_regfile regfilemux_sel = br_en ld_pc pcmux_sel = pc_plus4

XOR	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_xor
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

SRL	IF → ID → EX → WB	ld_mar marmux_sel = pc_out ld_mdr mem_read = 1 load_ir **aluop = alu_srl** alumux1_sel = rs1_out alumux2_sel = rs2_out ld_regfile regfilemux_sel = alu_out ld_pc

		pcmux_sel = pc_plus4

SRA	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_sra
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

OR	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_or
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4

AND	IF → ID → EX → WB	ld_mar
		marmux_sel = pc_out
		ld_mdr
		mem_read = 1
		load_ir
		
		aluop = alu_and
		alumux1_sel = rs1_out
		alumux2_sel = rs2_out
		ld_regfile
		regfilemux_sel = alu_out
		ld_pc
		pcmux_sel = pc_plus4