



计算机系统贯通课程-计算机系统 I

——Project-1 单周期 CPU 控制单元设计

洪奕迅 3230102930@zju.edu.cn

史璐欣 3220104390@zju.edu.cn

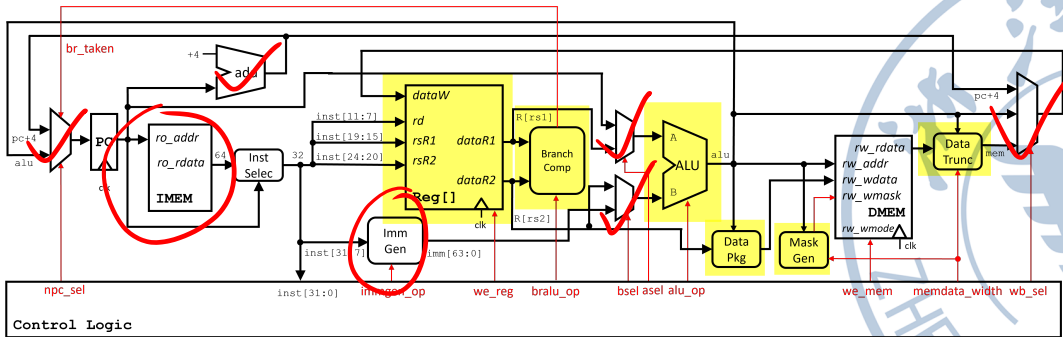
计算机学院
浙江大学

2025 年 5 月 15 日

注意事项

- 本次实验一直持续到本学期最后一周，每周实验课都可以验收，因此不再接受 ddl 以内的线上验收（最后一周实验课结束后会再验一会，课后则全部视为迟交）
- 本次实验非常非常非常重要，单周期 CPU 的代码大家会一直修修改改用到 sys3（甚至于说你们下学期的 Lab0 就是这学期的 project，这学期写好了可以白拿分），当然写不好的话后续所有硬件实验几乎都无法完成，因此请大家认真对待，并且尽量把代码写得规范一些（如果你不想下学期看不懂自己的代码）

CPU 数据通路



- 红线：控制流，Controller 把指令译码为一系列的控制信号（这节实验课要讲的内容）

指令的功能

31	27	26	25	24	20	19	15	14	12	11	7	6	0	
funct7				rs2		rs1		funct3		rd		opcode		R-type
imm[11:0]						rs1		funct3		rd		opcode		I-type
imm[11:5]				rs2		rs1		funct3		imm[4:0]		opcode		S-type
imm[12 10:5]				rs2		rs1		funct3		imm[4:1 11]		opcode		B-type
imm[31:12]										rd		opcode		U-type
imm[20 10:1 11 19:12]										rd		opcode		J-type

图: RISC-V 定长指令的具体位划分

我们可以从指令中知道的信息:

- 确认具体指令和译码控制信号的信息: opcode, funct3, funct7
- 寄存器号: 读寄存器 rs1, rs2 和写回寄存器 rd
- 立即数: 为了保证指令定长和规范性, 立即数的存放位置会首先选择复用字段, 所以看起来比较凌乱, 在使用前需要按照规范重新整理并进行符号扩展

立即数的规范

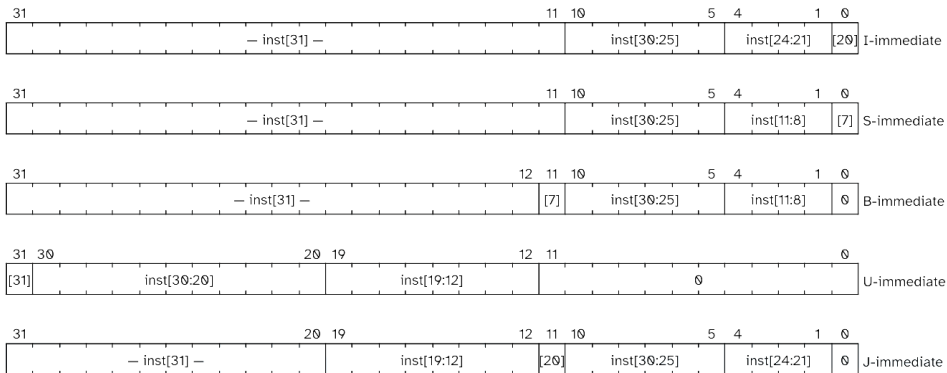


Figure 1. Types of immediate produced by RISC-V instructions.

图：具体的立即数规范



我们需要哪些控制信号

分阶段而言：

- IF 阶段：需要信号指导 PC 顺序/跳转 (`npc_sel` or `br_taken`)
- ID 阶段：产生控制信号，此时可能还需要先对指令的 `imm` 进行重新组合编码 (`immgen_op`)
- EXE 阶段：对运算单元 ALU 和 CMP 而言，我们需要知道运算类型 (`alu_op`, `cmp_op`)。且对于 ALU 操作数并不一定来自寄存器堆（可以考虑 `addi`, `add`, `jal` 的区别），因此还需要对操作数进行选择 (`alu_a_sel`, `alu_b_sel`)
- MEM 阶段：对所有与内存相关的部件，需要关心的都是数据类型 (`mem_op`)，数据正确性由其他部件计算确保
- WB 阶段：需要考虑写回值的类型 (`wb_sel`)，可以考虑指令 `add`, `ld`, `jal` 在写回值上的差异

具体示例 - I 型指令

ADDI x5, x6, -20(Reg[5] <- Reg[6] - 20)

- 二进制指令: 111111100000(imm) 00110(rs1: x6) 000(ADDI) 00101(rd: x5) 0010011(I-type)
- immgen_op: I-type, 直接符号扩展到 64 位即可
- alu_op: ALU_ADD; cmp: CMP_NO
- alu_a_sel: ASEL_REG; alu_n_sel: BSEL_IMM
- mem_op: MEM_NO
- wb_sel: alu_res