

# 体系结构实验报告\_lab1

学号 PB18000290 姓名 胡毅翔

## CPU设计

[logisim实现](#)

## 回答下列问题

1. 描述执行一条 XOR 指令的过程（数据通路、控制信号等）。

取指：根据 PCF，从 Instruction Memory 中取出指令（XOR rd,rs1,rs2）。

译码：寄存器堆 A1,A2 地址分别为 rs1,rs2, 读出待异或的值 RegOut1D 和 RegOut2D。

rd 传给 RdD。

指令 Instr 传给 Control Unit, Control Unit 把 AluControlD（选异或），AluSrc1D（选 RegOut1），AluSrc2D（选 RegOut2），RegWriteD（使能），ResultSrcD（在 MEM 用于选 ALU 计算的结果）传入段寄存器。

执行：ALU 执行异或计算，结果写入 AluOutE。RdE，RegWriteE 写入段寄存器。

访存：多选器将 AluOutM 传入 ResultM。RdM，RegWriteM 写入段寄存器。

写回：MemToRegW 选择 ResultW，RegWriteW 使能，将结果写回寄存器堆。

2. 描述执行一条 BEQ 指令的过程（数据通路、控制信号等）。

取指：根据 PCF，从 Instruction Memory 中取出指令（BEQ rs1,rs2,imm）。

译码：寄存器堆 A1,A2 地址分别为 rs1,rs2, 读出待比较的值 RegOut1D 和 RegOut2D。

指令 Instr 传给 Control Unit, Control Unit 把 AluSrc1D（选 RegOut1E），AluSrc2D（选 ImmE），BranchTypeD（选 BEQ 的对应类型），AluControlD（选加法）传入段寄存器。Control Unit 输出 ImmTypeD（左移两位并符号扩展）。

将 Imm 左移两位并符号扩展，传入 ImmD。

执行：Branch Decision 计算 Reg1 - Reg2，若结果为 0，BrE 使能，Hazard Unit 输出相应控制信号，否则继续执行。ALU 计算 PC 加上 ImmE 作为 BrT。

访存：无。

写回：无。

3. 描述执行一条 LHU 指令的过程（数据通路、控制信号等）。

取指：根据 PCF，从 Instruction Memory 中取出指令（LHU rd,rs1,imm）。

译码：寄存器堆 A1 地址为 rs1, 读出待计算的值 RegOut1D。

指令 Instr 传给 Control Unit, Control Unit 把 AluSrc1D（选 RegOut1），AluSrc2D（选扩展后的 Imm），RegWriteD（使能），MemToRegD（使能），ResultSrcD（在 MEM 用于选 ALU 计算的结果）传入段寄存器。Control Unit 输出 ImmTypeD（符号扩展）。

执行：ALU 计算 RegOut1E + ImmE，存入 AluOutE。

访存：从 Data Memory 读取地址为 {AluOutM[31:2], 2'b00} 的值，结果写入段寄存器。

AluOutM 写入 ResultM。

写回：根据 `Resultw[1:0]` 即 `LoadedBytesSelect` 选取要取的半字，并进行无符号拓展，多选器选择拓展后的结果写回 `Regfile`。

4. 如果要实现 CSR 指令 (`csrrw`, `csrrs`, `csrrc`, `csrrwi`, `csrrsi`, `csrrci`)，设计图中还需要增加什么部件和数据通路？给出详细说明。

`csrrw`：读后写控制状态寄存器， $t = \text{CSRs}[\text{csr}]$ ;  $\text{CSRs}[\text{csr}] = x[\text{rs1}]$ ;  $x[\text{rd}] = t$

故需要额外的寄存器用于保存  $t$ 。

`csrrwi`：立即数读后写控制状态寄存器， $x[\text{rd}] = \text{CSRs}[\text{csr}]$ ;  $\text{CSRs}[\text{csr}] = \text{zimm}$

故需要 Immediate Operand Unit 实现 5 位立即数的零扩展。

`csrrci`：立即数读后清除控制状态寄存器， $t = \text{CSRs}[\text{csr}]$ ;  $\text{CSRs}[\text{csr}] = t \& \sim \text{zimm}$ ;  $x[\text{rd}] = t$

`csrrs`：读后置位控制状态寄存器， $t = \text{CSRs}[\text{csr}]$ ;  $\text{CSRs}[\text{csr}] = t | x[\text{rs1}]$ ;  $x[\text{rd}] = t$

`csrrc`：读后清除控制状态寄存器， $t = \text{CSRs}[\text{csr}]$ ;  $\text{CSRs}[\text{csr}] = t \& \sim x[\text{rs1}]$ ;  $x[\text{rd}] = t$

`csrrsi`：立即数读后置位控制状态寄存器， $t = \text{CSRs}[\text{csr}]$ ;  $\text{CSRs}[\text{csr}] = t | \text{zimm}$ ;  $x[\text{rd}] = t$

如果要实现 CSR (Control Status Register) 指令,需要增加 `regfile`，提供给状态寄存器。在 EX 段寄存器的输入段需要加入多选器，确定读的数是来自 `csr` 还是原来的寄存器堆。增加立即数扩展的功能及逻辑控制，因为 `csr` 指令的立即数扩展与原来不同。在 Control Unit 中要加入 CSR 相关的控制逻辑，多选器等信号控制。

5. Verilog 如何实现立即数的扩展？

无符号扩展：在立即数的剩余高位补零。例：

```
assign ex_data={16'h0000,instruct[15:0]};
```

有符号扩展：对于位数 `len` 的立即数 `imm[len-1:0]` 扩展到 `len_extend` 位，只需根据立即数的最高位进行扩展，一个实现的例子如下：

```
assign ex_data=instruct[15]?{16'hffff,instruct[15:0]}:
{16'h0000,instruct[15:0]};
```

6. 如何实现 Data Memory 的非字对齐的 Load 和 Store？

Load：把访存的地址最后两位置为 `2'b00` (`addr&32'hffff_fffc`)，把整个字读取出来，在写回时，进行字节的选择和符号拓展。

Store：把访存的地址最后两位置为 `2'b00` (`addr&32'hffff_fffc`)，把整个字读取出来，根据 `addr` 后两位 (`addr[1:0]`)，控制需要修改的部分，再写回 Data Memory。

7. ALU 模块中，默认 `wire` 变量是有符号数还是无符号数？

无符号数。

8. 简述 BranchE 信号的作用。

若 `BranchE` 信号为 1，则说明条件转移指令的条件为真，则下条指令地址应为 `BrT`。同时，若 `BranchE` 信号为 1，Hazard Unit 也需要进行相应控制。

9. NPC Generator 中对于不同跳转 target 的选择有没有优先级？

`Ja1rT` 和 `BrT` 的优先级高于 `Ja1T`。因为若 `Ja1D` 使能的同时，`Ja1rE` 或 `BrE` 使能，说明 `Ja1r` 跳转指令或 `Br` 条件转移指令先于 `Ja1` 跳转指令发生，故有优先级的区别。`Ja1rT` 和 `BrT` 同优先级，二者的使能 `Ja1rE` 和 `BrE` 不可能同时有效。

10. Harzard 模块中，有哪几类冲突需要插入气泡，分别使流水线停顿几个周期？

RAW 类冲突需要插入气泡，如 Load 和 ALU 指令，在 ALU 计算时，操作数还未读出来，EX 段 Stall，MEM 段 flush，故需要停顿 1 个周期。

还有控制相关的冲突，在跳转时，需要插入气泡，flush 掉 IF 段取的指令，停顿 1 个周期。

在条件转移时，需要插入气泡，flush 掉 IF，ID 段取的指令，停顿 2 个周期。

11. Harzard 模块中采用静态分支预测器，即默认不跳转，遇到 branch 指令时，如何控制 flush 和 stall 信号？

发生条件转移时，即 BrE 信号为1后，对IF，ID段寄存器输出flush。

12. 0 号寄存器值始终为 0，是否会对 forward 的处理产生影响？

会，否则按照设计，可能使用forward的结果非零。所以在forward中需要加入判断是否为0号寄存器，控制多选器的输出，若为0号寄存器，则输出0。