

Computer Architecture Lab1

PB19071501 李平治

1. 描述执行一条 XOR 指令的过程（数据通路、控制信号等）

IF: 根据IF的PC，从Instruction Cache中取出指令 `XOR rd,rs1,rs2`

ID: 寄存器堆 `Addr1`, `Addr2` 地址分别为 `rs1`, `rs2`，从中读取 `Reg1`, `Reg2`。`rd` 传给Controller Decoder，其将 `ALU Func` (选异或)，`Op1` (选 `Reg1`)，`Op2` (选 `Reg2`)，`RegWrite` (使能)，`WBSelect` (在MEM用于选 ALU计算的结果)，`Load NPC` (选 `ALU Out`) 传入Control寄存器。

EX: ALU执行异或运算，结果写入RESULT寄存器。

MEM: 多选器将RESULT传入WB DATA寄存器中，`WB Select` 选择Result寄存器

WB: `RegWrite` (使能)，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）

IF: 根据IF的PC，从Instruction Cache中取出指令 `BEQ rs1,rs2,imm`

ID: 寄存器堆 `Addr1`, `Addr2` 地址分别为 `rs1`, `rs2`，从中读取 `Reg1`, `Reg2`。`rd` 传给Controller Decoder，其将 `Op1` (选 `Reg1`)，`Op2` (选 `IMM`)，`Br Type` (选BEQ对应的)，`Jalr` (选1)，`Load NPC` (选PCE) 传入Control寄存器；`ImmType` (左移两位并符号扩展) 传入立即数生成器。加法器计算 `PC` 和 `Imm` 的和，并传入 `Br Target`寄存器

EX: 多选器选择PCE后写入RESULT寄存器。Branch Module对比Op1和Op2是否相同，产生BR信号

MEM: 无

WB: 无

3. 描述执行一条 LHU 指令的过程（数据通路、控制信号等）

IF: 根据IF的PC，从Instruction Cache中取出指令 `LHU rd,rs1,imm`

ID: 寄存器堆 `Addr1` 地址为 `rs1`，读出 `Reg1`。`rd` 传给Controller Decoder，其将 `Alu Func` (选加法)，`Op1` (选 `Reg1`)，`Op2` (选 `IMM`)，`Load NPC` (选 `ALU Out`)，`WBSelect` (在MEM用于选 ALU计算的结果)，`RegWrite` (使能)，`Load Type` (指令中 `func3` 字段对应宽度) 传入Control寄存器；`ImmType` (符号扩展) 传入立即数生成器。

EX: ALU计算加法，结果写入RESULT寄存器

MEM: RESULT结果写入WB DATA；Data Extension根据输入判断是否对齐，再根据LoadTypeM判断数据长度，进行扩展

WB: `RegWrite` (使能)，结果写回寄存器堆

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。

IF:

无

ID:

- 1、CSRRW: `Inst[19:15]` 进入Register File, 读 `rs1` 入 `Op1`
- 2、CSRRS, CSRRC: `Inst[19:15]` 进入Register File, 读取 `rs1` 进入 `Op1`; CSR进入 `Op2`
- 3、CSRWI, CSRSI, CSRCI: `Inst[19:15]` 进入 `RegAddr`

增加部件: 需完善立即数扩展模块, 加入对CSR的支持; 需添加CSR寄存器堆; Control Unit需产生CSR控制信号

EX:

- 1、CSRRW: `Op1` 和 `0` 作为ALU的两个操作数, 进行加法运算后进入RESULT
- 2、CSRWI: `Reg1Src` 和 `0` 作为ALU的两个操作数, 进行加法运算后进入RESULT
- 3、CSRRS, CSRRC, CSRSI, CSRCI: `Op1` (`rs1` 的值或 `Reg1SrcE` 经过zimm扩展后的值) 和 `Op2` (此时是CSR的值) 作为ALU的两操作数, 根据规则进行运算后, 结果进入RESULT

增加部件: 在 `Op2` 多路器处加入CSR

MEM/WB:

`CSRWrite` 为1, 直接将RESULT写回到CSR

增加部件: 写回CSR的线路

5. Verilog 如何实现立即数的扩展?

```
1 // I-Type
2 imm[31:0] = {20{inst[31]}, inst[31:20]};
3 //S-Type
4 imm[31:0] = {20{inst[31]}, inst[31:25], inst[11:7]};
5 //B-Type
6 imm[31:0] = {20{inst[31]}, inst[7], inst[30:25], inst[11:8], 1'b0};
7 //U-Type
8 imm[31:0] = {inst[31:12], 12'b0};
9 //J-Type
10 imm[31:0] = {12{inst[31]}, inst[19:12], inst[20], inst[30:21], 1'b0};
```

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 `addr[1:0]` 置为 `2'b00`，读取整个字；写回时，再根据后两位 `addr[1:0]` 对字节进行选择 and 符号拓展，最后存入寄存器。
- Store: 同理，把访存的地址最后两位 `addr[1:0]` 置为 `2'b00`，读取整个字；写回时，再根据后两位 `addr[1:0]` 对字节进行修改，最后写回。

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。

- 指示 Branch 指令是否进行跳转（当为1时跳转）
- 当跳转（为1）时，将PC改为跳转后的地址，并在Hazard Unit中进行相应处理

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

- `Jalr` 和 `Br` 的优先级高于 `Jal`，并且都高于 `PC+4`
- `Jalr` 和 `Br` 在EX阶段才计算出信号，而 `Jal` 在ID就产生信号，因此应当优先选择前者
- `Jalr` 和 `Br` 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：例如，load-use型冲突若在ALU计算时，所需操作数还未load出来，则需EX段stall，MEM段flush，即停顿1个周期
- 分支和跳转：在跳转时显然需要插入气泡，flush掉前面IF段的指令，停顿1个周期；在条件跳转时，需要flush掉IF段和ID段的指令，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?

在 `BR` 型号为1时，即发生跳转时，对IF和ID段寄存器进行flush；否则不flush/stall

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

会。需要在实现转发过程中对源寄存器是否是x0进行判断，否则若某指令对x0写入时，后续指令可能会接收到非0的x0值。