

Verilog 实验 1 实验报告

PB19071405 王昊元

2022 年 03 月 26 日

实验问题回答

基于助教提供的设计图进行回答，且省略部分有关传入段寄存器的说明。

1. 描述执行一条 XOR 指令的过程（数据通路、控制信号等）

- **IF** NPC Generator 和 IF 产生 PC，通过 Instruction Cache 和 IR 拿到指令 (XOR rd, rs1, rs2)。
- **ID** 从 General Register File 中读出两个操作数 Reg1 和 Reg2，Immediate Generate 不产生立即数。Inst[11:7] 为 RegDstD，传入 REG ADDR 段寄存器，写回数据时使用。Func7、Opcode 和 Func3 传入 Controller Decoder，控制信号 ALU Func 为异或，Op1 和 Op2 选择 Forwarding 结果（这里为寄存器读取结果），使能 RegWrite，WB Select 选择 RESULT 段寄存器结果（这里为 ALU 计算结果），Load NPC 选择 ALU 计算结果。
- **EX** OP1 Sel 和 OP2 Sel 分别选择 Reg1 和 Reg2，控制信号 Op1 和 Op2，使得从寄存器中取出的操作数经选择后作为 ALU 的操作数，ALU Func 为异或，ALU 执行异或操作，Load NPC 选择 ALU 计算结果，存入 RESULT 段寄存器。
- **MEM** WB Select 选择 RESULT 段寄存器结果（这里为 ALU 计算结果）作为 WBData。
- **WB** RegWrite 使能使得将 WBData 写入寄存器 WBAddr 处。

2. 描述执行一条 BEQ 指令的过程（数据通路、控制信号等）

- **IF** NPC Generator 和 IF 产生 PC，通过 Instruction Cache 和 IR 拿到指令 (BEQ rs1, rs2, imm)。
- **ID** 从 General Register File 中读出两个待比较的数 Reg1 和 Reg2，生成立即数，左移一位后与 PC 相加，作为 BR Target，Func7、Opcode 和 Func3 传入 Controller Decoder，控制信号 Op1 和 Op2 为选择 Forwarding 结果（这里为寄存器读取结果），BrType 为 BEQ。
- **EX** Op1 和 Op2 选择寄存器读取结果，并与 BrType 一起传入 Branch Module，产生 BR 信号，BR 和 BR Target 传入 NPC Generator，Hazard Module 产生相应信号。
- **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** NPC Generator 和 IF 产生 PC，通过 Instruction Cache 和 IR 拿到指令 (LHU rd, rs1, imm)。
- **ID** 从 General Register File 中读出待计算的数 Reg1, Func7、Opcode 和 Func3 传入 Controller Decoder，控制信号 Op1 为选择 Forwarding 结果（这里为寄存器读取结果），Op2 为选择立即数，ALU Func 为加法，ImmType 为符号扩展，Load NPC 选择 ALU 计算结果，使能 RegWrite，Load Type 选择内存数据，WB Select 选择扩展后的数据。
- **EX** Reg1 与扩展后的立即数相加后传入 RESULT 段寄存器。
- **MEM** result（ALU 计算结果）作为 ADDR 从 Data Cache 中读取数据，进行扩展后作为 WBData 传入段寄存器。
- **WB** 将 WBData 写入寄存器 rd 中。

4. 如果实现 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来保存 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。例如：assign ex_data = 16'h0000, imm[15:0]。
- 有符号扩展：根据立即数的最高位进行扩展。例如：assign ex_data = imm[15] ? 16'hffff, imm[15:0] : 16'h0000, imm[15:0]。

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

- 分两次 Load 或 Store 即可。
- 先读取 addr & 32'hfff_ffc，即低位字节所在地址，然后根据 addr[1:0] 选择低位数据写入或读取；
- 再读取 (addr + 32'h0000_0010) & 32'hfff_ffc，即高位字节所在地址，然后根据 addr[1:0] 选择高位数据写入或读取。

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。

- 用来指示 Branch 类型的指令是否跳转，Branch Module 会根据传入的 Op1、Op2 和 BrType 来判断当前 Br 指令是否成功跳转，并输出相应的控制信号 BranchE。
- 同时 BranchE 信号还会使得 Hazard Module 产生不同的控制信号。

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

- Br 和 Jalr 优先级最高，Jal 优先级次之，PC + 4 优先级最低。
- 因为 Br 和 Jalr 在 EX 阶段计算并返回信号，而 Jal 在 ID 段计算并返回信号，如果 NPC Generator 同时收到 Br/Jalr 和 Jal 的信号（Br 和 Jalr 不会同时有效），则意味着 Br/Jalr 是 Jal 前一条指令，则应先执行 Br/Jalr，故 Br 和 Jalr 优先级高于 Jal。
- PC + 4 只有在没有任何跳转发生时执行，故优先级最低。

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？

- 数据相关冲突：如 Load 指令后执行 ALU 指令，需要插入气泡，在 ALU 计算时，操作数还未读出来，EX 段 stall，MEM 段 flush，故需停顿 1 个周期。
- 控制相关冲突：在跳转时，需要插入气泡，flush 掉 IF 段的指令，停顿 1 个周期。
- 条件转移：需要插入气泡，flush 掉 IF、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？

- 遇到 branch 指令时，默认不跳转，当 branch 指令执行到 EX 段时，Branch Module 会产生 BranchE 信号（是否跳转）。如果 BranchE 有效，那么当前 ID 段的 PC + 4、IF 段的 PC + 8（PC 为 branch 指令的 PC）不应被执行，使能 FlushD 和 FlushE，清空对应段间寄存器，执行 Branch Target 指令；如果 BranchE 无效，说明跳转失败，继续执行命令即可，即不使能 Flush 信号。

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

- 会。
- 如果不判断，forward 过程中可能将 0 号寄存器的值改为非零值。
- 例如 XOR 0, rs1, rs2; XOR rd, 0, rs2; 此时，前一指令试图修改 0 号寄存器，但其实并不能修改，如果按照正常的 forward 处理，将计算结果（WBData）转发给后一条指令的第一个操作数，就会产生结果错误。
- 故 forward 中要加入是否为 0 号寄存器的判断，控制多选器的输出。

其他

如果所做回答中有任何不正确的地方，或实现不妥当有其他风险、表述有问题的地方，还烦请助教指出，这可以让我学到更多知识，也会对我之后的实验实现有所帮助。