

RV32I 指令集流水线 CPU 设计报告

PB16111245 王立峰

一、主要模块的设计思想

1、RV32Core.v

此模块为 RV32I 指令集 CPU 的顶层模块，声明了各个主要模块单元，以及总线排布情况；

2、ALU.v

算术逻辑单元模块；

3、BranchDecisionMaking.v

分支预测模块；

4、ControlUnit.v

控制单元模块，根据输入的 Op, fn3, fn7 信号来出所有控制信号，分别对每个输出信号分析是在那些指令会用到，对应的值是多少；

5、DataExt.v

该模块是用来处理非字对齐 load 的情形，同时根据 load 的不同模式对 Data Mem 中 load 的数进行符号或者无符号拓展，组合逻辑电路，根据输入的 RegWrite 确定相应的是符号拓展还是零拓展；

6、HazardUnit.v

该模块用来处理流水线冲突，通过插入气泡，forward 以及冲刷流水段解决数据相关和控制相关，组合逻辑电路；主要是数据转发，stall 和 flush；

7、ImmOperandUnit.v

立即数模块，利用正在被译码的指令的部分编码值，生成不同类型的 32bit 立即数；

8、NPC_Generator.v

PC 计数器模块，用来生成 Next PC 值得模块，根据不同的跳转信号选择不同的新 PC 值；

9、Parameters.v

定义常量值模块；

10、RegisterFile.v

寄存器文件模块；

11、IFSegReg.v

IF-ID 段寄存器模块；

12、IDSegReg.v

IF-ID 段寄存器模块；

13、EXSegReg.v

EX 段寄存器模块，本模块是支持同步清零的段寄存器，当 EN==0 时寄存器状态保持不变（也不会执行清零）；

14、MEMSegReg.v

EX-MEM 段寄存器模块；

15、WBSegReg.v

WB 段寄存器模块；

16、DataRam.v

数据存储器模块；

- 17、InstructionRam.v
指令存储器模块；

二、回答问题

- 1、为什么将 DataMemory 和 InstructionMemory 嵌入在段寄存器中？
直接进入下一个流水阶段，不用等待下一个周期
- 2、DataMemory 和 InstructionMemory 输入地址是字（32bit）地址，如何将访存地址转化为字地址输入进去？
地址使用 A[31,2]
- 3、如何实现 DataMemory 的非字对齐的 Load？
进行选位和拓展
- 4、如何实现 DataMemory 的非字对齐的 Store？
WE 写使能信号控制
- 5、为什么 RegFile 的时钟要取反？
不同步读，便于五级流水
- 6、NPC_Generator 中对于不同跳转 target 的选择有没有优先级？
有优先级
- 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、AluSrc1E 执行哪些指令时等于 1'b1？
AUIPC
- 9、AluSrc2E 执行哪些指令时等于 2'b01？
SLLI, SRAI, SRLI
- 10、哪条指令执行过程中会使得 LoadNpcD==1？
JALR, JAL
- 11、DataExt 模块中，LoadedBytesSelect 的意义是什么？
选位
- 12、Harzard 模块中，有哪几类冲突需要插入气泡？
LOAD 相关
- 13、Harzard 模块中采用默认不跳转的策略，遇到 branch 指令时，如何控制 flush 和 stall 信号？
FlushD=1, FlushE=1
- 14、Harzard 模块中，RegReadE 信号有什么用？
判断是否需要转发
- 15、0 号寄存器值始终为 0，是否会对 forward 的处理产生影响？
可能会

三、总结

还没有开始写 Verilog 代码，目前只是各个模块看了一下，把设计图细细地观察了一下，向同学了解了一下大致流程，总的感觉想要完成整个项目十分困难，有点慌……