

## 体系结构实验课程第 二 次实报告

|      |                  |    |         |            |     |
|------|------------------|----|---------|------------|-----|
| 实验名称 | 静态 5 级流水线 CPU 实现 |    |         | 班级         | 李雨森 |
| 学生姓名 | 秦泽斌              | 学号 | 2212005 | 指导老师       | 董前琨 |
| 实验地点 | A303             |    | 实验时间    | 2024.10.12 |     |

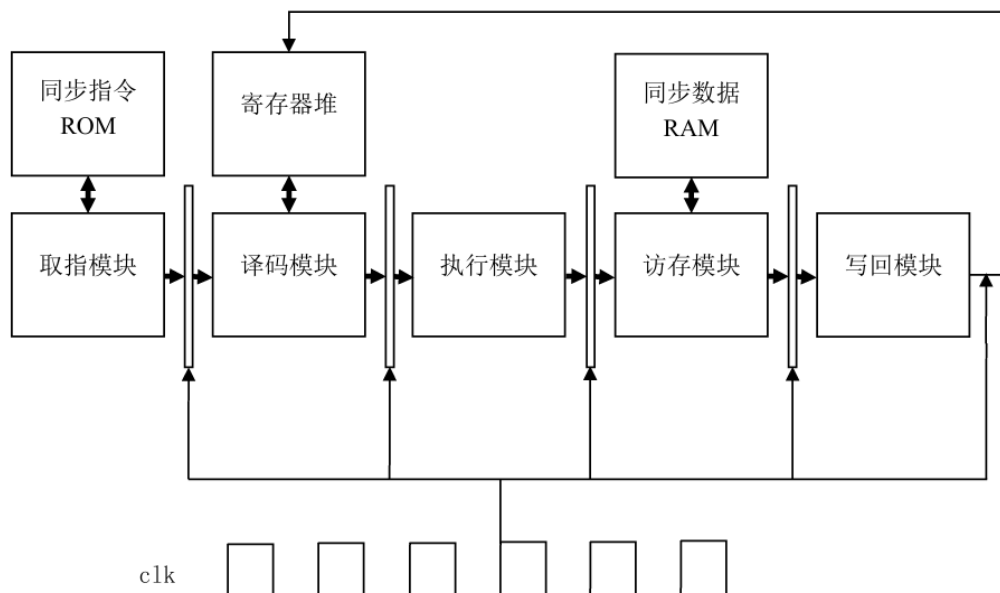
### 1、实验目的

1. 在多周期 CPU 实验完成的提前下，深入理解 CPU 流水线的概念。
2. 熟悉并掌握流水线 CPU 的原理和设计。
3. 最终检验运用 verilog 语言进行电路设计的能力。
4. 通过亲自设计实现静态 5 级流水线 CPU，加深对计算机组成原理和体系结构理论知识的理解。
5. 培养对 CPU 设计的兴趣，加深对 CPU 现有架构的理解和深思。

### 2、实验内容说明

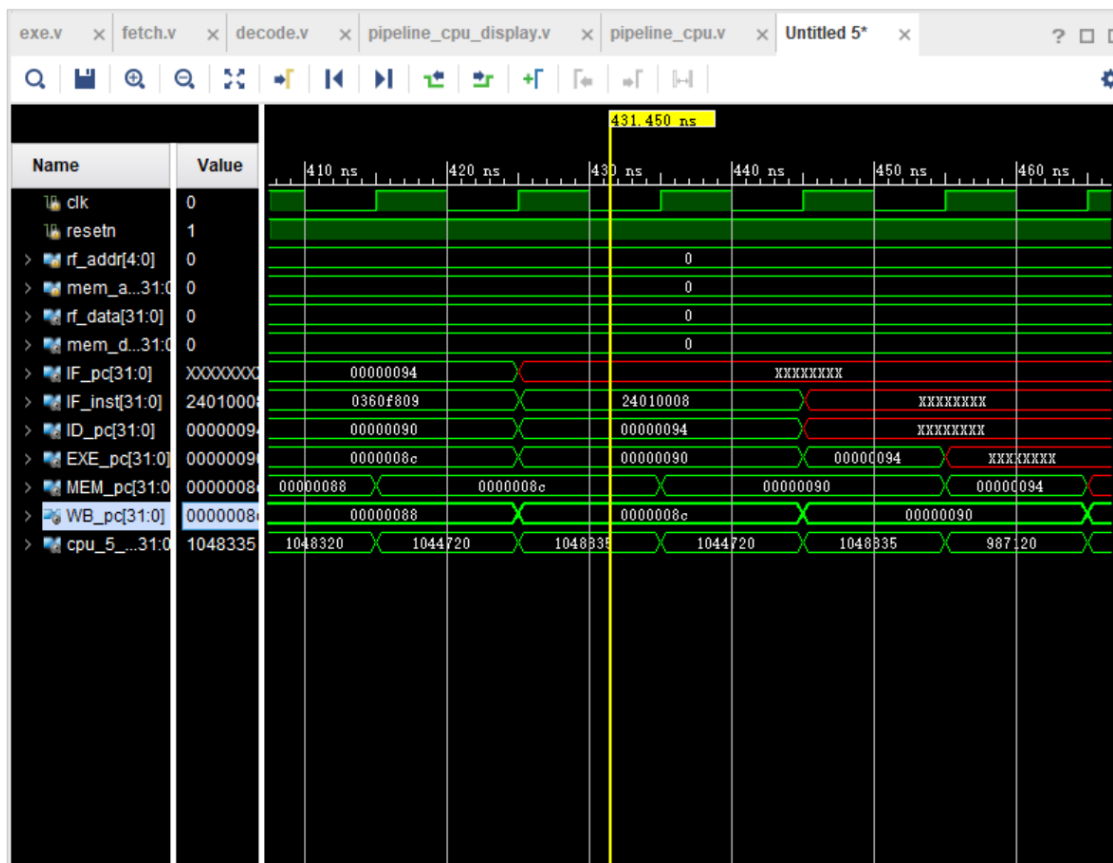
- 1、分析现有指令的执行过程，找到存在的读后写/写后读问题，并进行改进验证（推荐自己编写指令 rom 中的指令，针对存在的问题进行测试改进）然后写入实验报告。
- 2、在做 1 的过程中，寻找现有 CPU 的不足之处，提出一些自己的改进方案和想法，包括但不限于实验指导手册第十章中的优化部分，并初步讨论一下可行性，为最后一次实验做准备。
- 3、实验报告中的原理图放图 9.2 即可。注意要在实验报告中详细说明改进过程，验证时先描述改进前的情况，再说明改进后的情况，进行具体对比。

### 3、实验原理图



### 4、实验步骤及修改结果分析

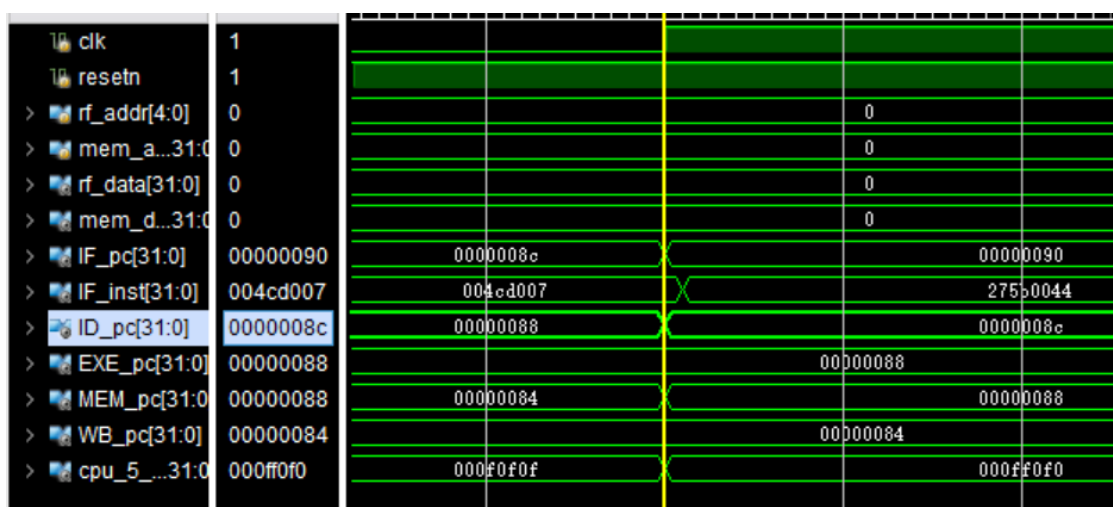
1. 分析初始波形图及指令列表可知



|     |                     |                                 |          |   |
|-----|---------------------|---------------------------------|----------|---|
| 8CH | addiu \$27,\$26,#68 | [\$27] = 0000_0050H             | 275B0044 | 0010_0111_0101_1011_0000_0000_0100_0100 |
| 90H | jalr \$27           | 跳转到 50H,<br>[\$31] = 0000_0098H | 0360F809 | 0000_0011_0110_0000_1111_1000_0000_1001 |

在 8CH 指令写回\$27 的值之前，90H 已经开始读取\$27 了，发生了读后写问题，发生错误。

2. 仔细分析波形图发现当译码阶段开始后 IF\_inst 仍处在上一条指令，导致错误，因而需要在 fetch 中加入气泡，使整个过程延后一个周期，消除读后写问题。



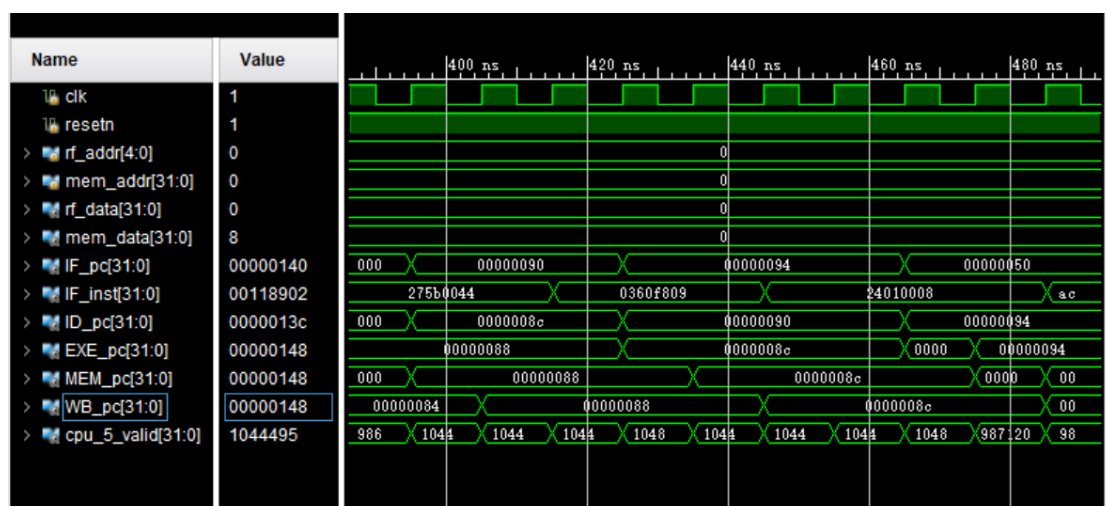
修改代码如下：使用 nop 强行延迟一个周期。

reg nop;

```

always @(posedge clk)
begin
    if (!resetn || next_fetch)
    begin
        IF_over <= 1'b0;
        nop<=1'b0;
    end
    else if(!nop)
    begin
        nop<=1'b1;
    end
    else
    begin
        IF_over <= IF_valid;
    end
end
End
正常运行

```



3. 又发现在 144H 指令处未能正确执行分支跳转，仔细分析波形图后，发现其仍然出现类似上面的问题，在 140H 写回前，144 已完成读取，发生读后写问题，因此，类似的在 mem 阶段增加气泡以解决



修改代码如下：

reg nop;

reg MEM\_valid\_r;

always @(posedge clk)

begin

if (MEM\_allow\_in)

begin

MEM\_valid\_r <= 1'b0;

nop <= 1'b0;

end

else if (nop == 1'b1)

begin

MEM\_valid\_r <= MEM\_valid;

end

else

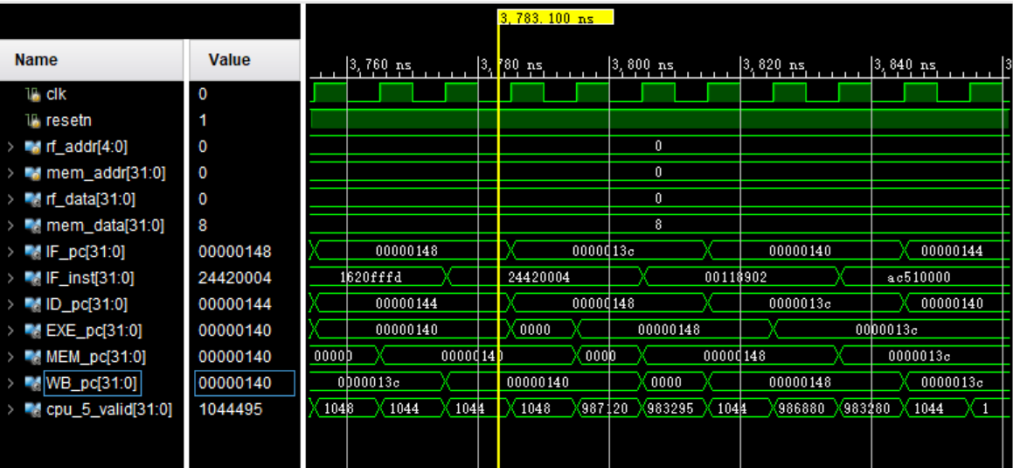
begin

nop <= 1'b1;

end

end

assign MEM\_over = inst\_load ? MEM\_valid\_r : MEM\_valid;  
结果正确



5、实验问题分析

1.分支预测改进

现有问题：  
许多 CPU 使用静态分支预测（如总是预测为跳转或不跳转），这可能导致较高的误预测率。

改进方案：  
动态分支预测：引入更复杂的动态分支预测机制，如局部历史预测和全局历史预测结合（如两层分支预测器），可以根据过去的分支行为动态调整预测结果。  
分支目标缓冲区（BTB）：使用 BTB 缓存跳转目标地址，以加快跳转指令的执行。通过在预测成功时直接从 BTB 中获取目标地址，可以减少分支延迟。

2. 提高流水线的处理能力

现有问题：  
分支指令可能导致流水线停顿，影响指令吞吐率。

改进方案：  
气泡插入策略：设计一个智能的气泡插入策略，在分支跳转发生时智能决定插入气泡的数量，尽量减少性能损失。  
增强的旁路机制：在已有的旁路机制基础上，引入更多的数据通路，使得指令在数据依赖时能够更灵活地进行旁路，减少因数据冒险导致的延迟。

3. 提高内存访问效率

现有问题:

分支指令往往会导致频繁的内存访问,影响 CPU 效率。

改进方案:

预取机制:引入智能预取机制,根据程序的运行行为预测即将使用的指令或数据,并提前加载到缓存中。通过分析局部性原则,优化内存访问。

多级缓存:设计更高效的多级缓存系统,优化数据的存取速度,尤其是在频繁分支的情况下。

## 6、总结感想

通过这次静态 5 级流水线 CPU 的实现实验,我对 CPU 流水线的概念有了更深入的理解。实验过程中,我遇到了多个技术挑战,尤其是在读后写问题和分支跳转的实现上。通过仔细分析波形图和指令执行过程,我学会了如何通过修改代码和插入气泡来解决这些问题。实验结果基本符合预期,但我也意识到在某些情况下,如分支预测的准确性和内存访问效率,还有很大的提升空间。我提出的改进方案,包括动态分支预测和增强的旁路机制,我认为这些方案能够显著提高 CPU 的性能和效率。