

实验报告

[文档副标题]



2020-9-2



# 流水线CPU设计实验报告

## 实验目的

* 深入理解计算机相关理论
* 熟练掌握计算机软硬件工作原理
* 学习并掌握计算机接口设计
* 学习并掌握RISC指令集的处理器设计
* 熟悉HDL语言、EDA工具应用
* 培养计算机系统能力

## 实验环境

硬件开发工具：Vivado 2019.2

硬件实验平台：EES-338口袋计算机（Xilinx Artix-7 FPGA）

MIPS交叉编译、模拟运行软件：MARS4.5

## 设计思路

### 指令信息

#### 逻辑运算指令(8条)

And,andi,or,ori,xor,xori,nor,lui,

#### 移位指令（6条）

Sll,sllv,sra,srav,srl,srlv

#### 移动指令（2条）

Movn，movz

#### 算术指令（12条）

Add，addu，sub，subu，slt，sltu，addi，addiu，slti，sltiu，clo，clz

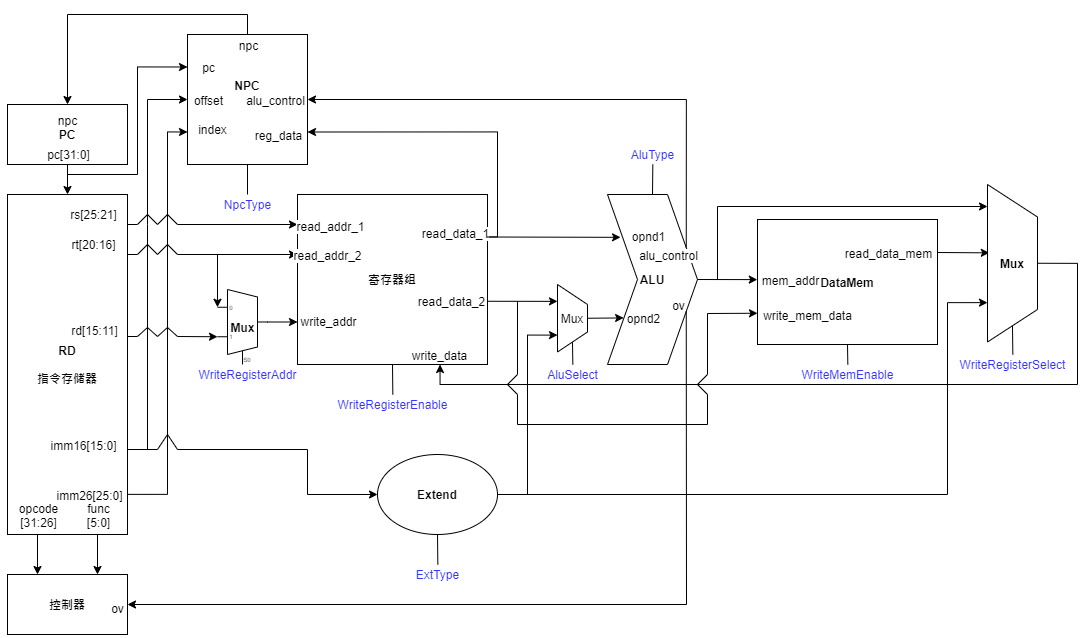
#### 转移指令（14条）

Jr，jalr，j，jal，b，bal，beq，bgez，bgezal，bgtz，blez，bltz，bltzal，bne

#### 空操作指令（1条）

Nop

### 数据通路图

参考课程示例结合我抽取到的指令，通过绘图软件绘制了如下的数据通路图。

### 译码控制信号

译码阶段，针对不同的指令，选择了不同的控制信号，这些控制信号将决定在执行、访存、写回等阶段CPU的行为，主要的控制信号及其含义如下：

|  |  |
| --- | --- |
| 信号名称 | 含义 |
| aluop | ALU的运算类型 |
| alusel | ALU的输出来源 |
| wreg\_enable | 是否允许写寄存器 |
| cu\_npc\_op | 下一条指令地址的获取方式 |
| next\_inst\_in\_delayslot | 下一条指令是否在延时槽 |
| is\_in\_delayslot | 本条指令是否在延时槽 |

## 具体实现

### 控制代码

### 溢出判断

### 分支预测

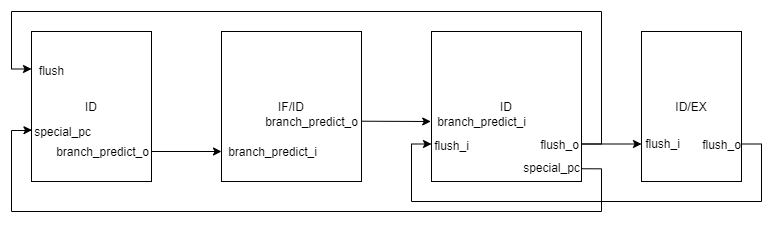
#### 提前判断

#### 动态预测

上方提前判断的方法能够只经过一个周期就判断出指令是否需要跳转，但是也仍然需要一个周期，这就使得分支指令后一条指令必然会被执行，这就需要在编码或者编译器层面为分支指令之后插入一条延时槽指令，或者在分支指令之后对流水线进行flush清空指令，无论哪一种方法在我们看来都不够完美。前者增加了额外的编译成本且破坏了人们对于分支指令的理解，后者会为流水线引入一个Bubble造成一个周期的延迟。对此，我们结合计算机体系结构课程的学习，决定采用动态分支预测技术，通过动态预测分支指令是否执行，对于预测跳转的分支指令提前取跳转后的指令，这样在预测成功的情况下就不再需要清空流水线，即使预测失败也不过是和之前一样造成一个周期的延迟。

##### 数据流图修改

为了达到这个效果，我们引入了几个额外的信号，需要对数据流图做以下修改：



信号含义如下：

Branch\_predict\_i/o:分支预测的结果，1跳转0不跳转

Flush\_i/o:是否需要引入bubble，即（对ID来说flush\_o控制下一条指令，flush\_i控制本条指令）让指令变为NOP。

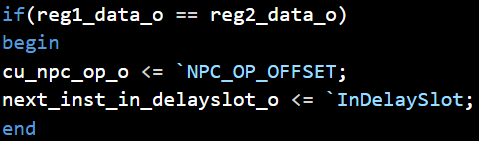
Special\_pc:预测失败时可能需要的指定npc值

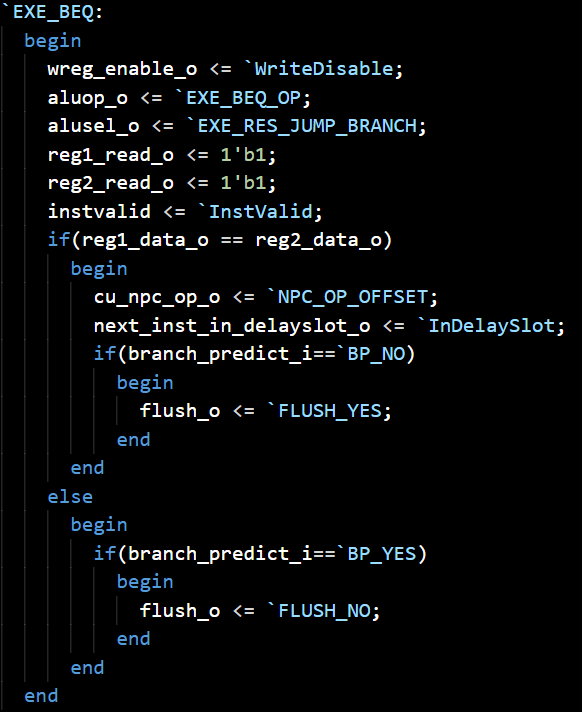
##### 代码逻辑修改

1. 根据数据流图添加对应的输入输出
2. 修改ID模块内部逻辑，引入新的wire变量inst\_sel，将原来直接接入输入指令inst\_i的逻辑全部转到inst\_sel上，inst\_sel控制逻辑如下：

如果本条指令需要flush，则改为32位全0的指令，对应sll $0, $0,$0，即NOP。

1. 修改ID模块内部逻辑，分支指令译码过程修改，以BEQ为例，修改部分如下：

修改前:

修改后：

可以看到如果分支预测不跳转但是实际跳转的话，需要对下一条指令flush，反之如果预测跳转而结果不跳转，也需要flush下一条指令。

1. 啊

## 仿真分析

### 初始数据&测试程序

### 波形分析

### 程序验证

## 问题与解决

开发过程不是一帆风顺，出现了各种问题，以下是其中比较有代表性的：

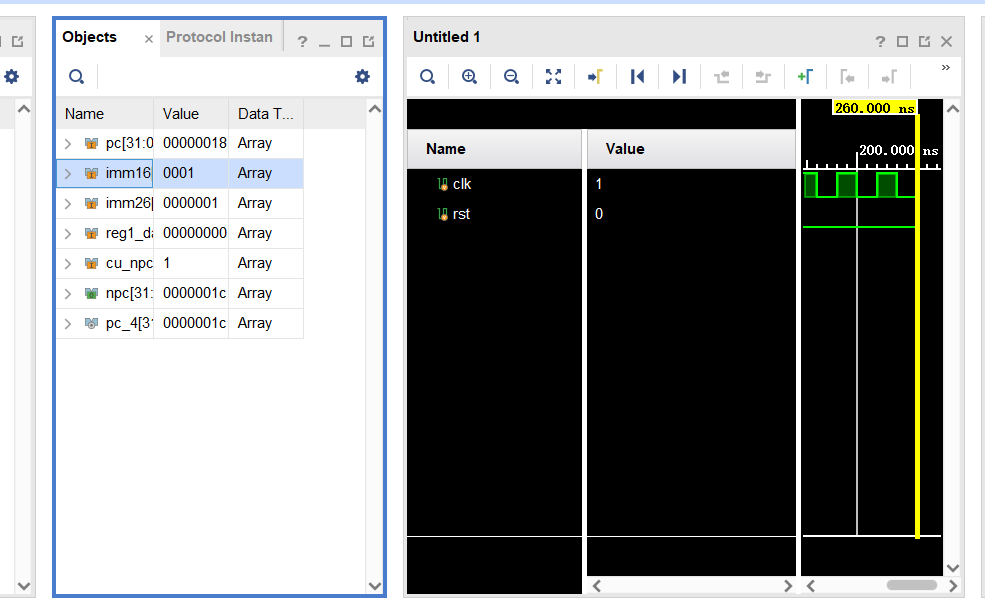
* 问题一 Entend模块的扩展信号输入

xxxx

解决方法

xxxx

* 问题二 NPC指令imm计算值和控制信号不同步 多次模拟数据不一致



解决方法

xxxx

## 设计总结

在本次实验中，

## 心得体会

1. CPU的开发是比较复杂的工程，需要长期的经验和技术积累。我国在芯片领域的薄弱正是因为缺少这种积累，中国“芯”路任重道远。
2. 硬件开发，包括嵌入式开发和一般的应用程序设计有非常大的不同点，这是我以前不怎么关注的一个领域。如果有机会深入探究，必将能够拓宽我的技术视野，改变我思考很多问题的方式。
3. CPU开发乍看起来对于我们来说简直不可能，但是实际上只要综合运用我们数字逻辑、计算机组成原理和体系结构课程中组合逻辑、时序逻辑和关于编码和运算等的知识，还是能够完成的。可见哪怕是看上去很困难的事，只要认真分析扎实学习，都能够有不错的收获，在此对龙芯杯参赛的队友表示由衷的敬佩。