**实验九报告**

学号 2017K8009929032 2017K8009929034

姓名 杨程远 杨宇恒

箱子号 15

一、实验任务

实验八的基础上增加：

BREAK指令和断点例外支持；地址错、整数溢出、保留指令例外支持；CP0寄存器COUNT、COMPARE、BADVADDR；时钟中断支持；6个硬件中断支持和2个软件中断支持。

完成lab9的功能测试，并运行记忆游戏程序。

二、实验设计

（一）总体设计思路

1、从取指级开始，设置一个表示存在例外（任何例外）的信号，这用来帮助判断执行级是否要禁止写HI/LO寄存器和store写主存的效果。设置一个表示例外类别的信号，这用来给CP0做判断。

2、在取指级判断指令地址读错误AdEL（用rine表示），在执行级地址算出后利用处理Store指令的st\_decode判断数据写AdES地址错例外，在访存级利用处理Load指令的ld\_decode部件判断数据读AdEL（用rdae表示）地址错例外；在访存级根据例外类型计算出出错的地址BADVADDR。

3、在执行级判断ADD、ADDI、SUB的整型溢出例外；在ALU中添加判断溢出的逻辑并输出。在译码级判断断点例外。在译码级判断保留指令例外。中断例外从CP0模块中根据寄存器进行判断，之后直接传到译码级，标记在译码级的指令上。

4、在顶层模块增加位宽为6的外部硬件中断ext\_int\_in信号，并恒置为0。

（二）重要模块1设计：CP0\_reg模块

1. 工作原理

沿用实验八的模块。将CP0寄存器堆、CP0操作译码、是否有待处理中断判断逻辑封装到一起。没有分成多个模块是因为需要CP0寄存器堆里的信息，来判断硬件对CP0寄存器堆的操作，放到一个模块里可以避免一个信号在两个模块间穿梭。

1. 接口定义

| **­**  **类别** | | **名称** | **方向** | **位宽** | **功能描述** |
| --- | --- | --- | --- | --- | --- |
| 软件的指令操作 | | cp0\_addr | IN | 8 | 软件读写地址 |
| cp0\_rdata | OUT | 32 | 软件读数据 |
| cp0\_wen | IN | 1 | 软件写使能 |
| cp0\_wdata | IN | 32 | 软件写数据 |
| 硬件例外或ERET自动完成的操作 | 对所有例外有效 | exc\_type | IN | 8 | 每一位表示一种例外发生 |
| PC | IN | 32 | 发生例外指令的地址 |
| is\_slot | IN | 1 | 例外发生在延迟槽指令 |
| 对特定例外有效 | int\_num | IN | 5 | 中断号 |
| bad\_vaddr | IN | 32 | 内存访问错误地址 |
| EPC | OUT | 32 | EPC输出 |
| int\_happen | OUT | 1 | 有需要处理的中断 |
| eret | IN | 1 | eret指令 |

1. 重要内部中间结果：硬件CP0操作的译码结果

| **名称** | **位宽** | **功能描述** |
| --- | --- | --- |
| cp0\_XX\_XX\_set | 1 | 对应寄存器硬件自动置位信号 |
| cp0\_XX\_XX\_clear | 1 | 对应寄存器硬件自动清零信号 |
| cp0\_XX\_XX\_wen | 1 | 对应寄存器硬件自动写有效 |
| exccode | 5 | 例外编码，由例外类型译出 |

1. 功能描述

按照讲义实现寄存器更新，每一位写的优先级为rst，set，clear，wen，cp0\_wen；Cause\_ti寄存器除外，其clear优先级高于set。在本模块中完成中断IP被EXL、IE、IM屏蔽后是否有待处理的中断的逻辑判断，生成是否有待处理中断的int\_happen信号。

（三）重要逻辑设计：地址错例外的BADVADDR生成逻辑

1. 工作原理

当发生地址错例外时，需要将出错的虚地址BADVADDR记录并传输给CP0寄存器。出错的虚地址有两个可能的来源：取指发生地址错时PC的值；访存指令中由ALU计算出的访存地址。其中PC值已经包含在每一级流水线中；而ALU计算的结果也被送到访存级的流水线中。因此，可以在访存级再根据地址错类型计算出BADVADDR，并在流水线中传递给写回级。

1. 功能描述

在访存级根据例外类型的信息进行判断：若有取值的地址错例外AdEL（记作rine），则将BADVADDR赋值为当前PC的值；否则（对应访存指令的地址错）将BADVADDR赋值为ALU的计算结果。

三、实验过程

（一）实验流水账

2019年11月4日16:00-22:00：完成所有模块的设计，通过测试。

2019年11月5日14:00-16:00：完成实验报告。

（二）错误记录

1、错误1：整数溢出例外的寄存器写回没有被取消

（1）错误现象

仿真报错，如图1。

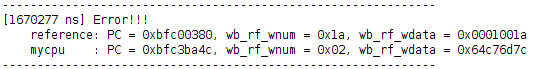


图1 仿真报错信息

（2）分析定位过程

在反汇编程序中找到这个写回出错的指令（地址为0xBFC3BA4C），如图2。这是一条ADD指令；在图3的波形中，可以发现检测出了整数溢出例外（ws\_exc\_type的第0位拉高），因此这条指令不应该写回寄存器；但寄存器的写使能信号却还是4’b1111（0xF），出现了错误。

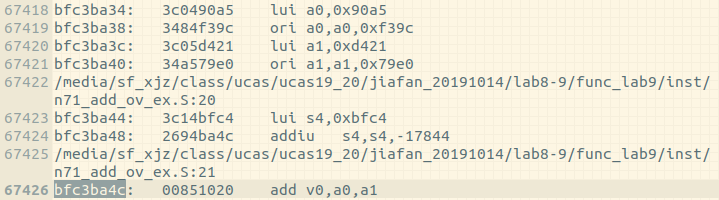


图2 出错的ADD指令

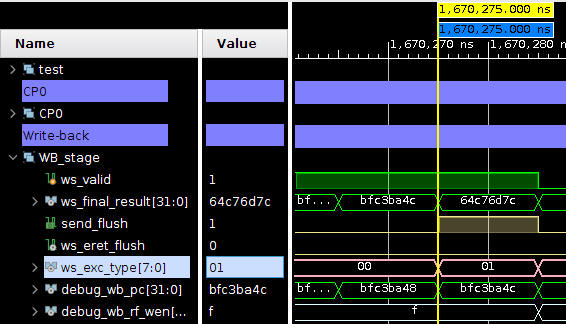


图3 检测出了整数溢出例外，但寄存器堆写使能信号仍然拉高

（3）错误原因

当写回级的指令有例外时，没有调整寄存器堆的写使能信号，导致写回了不该写回的值。

（4）修正效果

将写回级模块生成的寄存器堆写使能信号rf\_we按位与上表示有例外的信号ws\_exc的非，表示当该指令发生例外时不写回寄存器，如图4。进行修正后通过测试。

bug1_cor

图4 在寄存器堆写使能信号上加上例外判断

（5）归纳总结

在可能改变机器状态的信号上（如寄存器、内存的写使能），需要对例外信号进行判断，以正确地实现精确例外。