

实验四报告

2016K8009907007

黄熠华

一、实验任务（10%）

增加 BREAK 指令，地址错、整数溢出、保留指令例外支持。

硬件上增加 CP0 寄存器 COUNT、COMPARE。

增加时钟中断支持，将其绑定在硬件中断 5 号上，也就是 CAUSE 对应的 IP7 上。

增加 6 个硬件中断支持以及两个软件中断支持，完成功能测试以及运行 lab3 的电子表程序，实现相同功能。

完成 lab4_fun_2 功能测试。

二、实验设计（30%）

在实现任务要求例外的过程中，我并没有采用老师上课讲授的思路，即给例外指令记上标记，统一到某个阶段提交的方式。而是根据例外发生情况尽早提交例外，减少气泡指令。因为提交例外的时间并不确定，我用一组 cancel 信号来进行阶段指令的取消，保证精确例外。需要注意的是，某些例外需要将自己取消掉，比如整形溢出例外需要将该条运算指令置为 invalid，保证其不写回寄存器，某些例外则不必要取消自己，比如 syscall。

取址例外是由 jump 指令跳一个非法地址产生的，jump 指令并不报例外，其延迟槽指令也必须执行完。因此，该 jump 指令不能取消掉其下一条指令，而是取消掉其下下条指令。

LB, LH, LW 等指令的访存地址有着不同的要求，触发地址例外也有不同的条件。

延迟槽指令发生例外时，EPC 写地址是前一条指令的地址，因此需要用额外的寄存器来为延迟槽指令记上标记，令其 EPC 和 Cause 寄存器的 bd 位能够正确赋值。

三、实验过程（60%）

（一）实验流水账

11.24: 写完 cp0 系列寄存器以及各种中断

11.26: 调试通过

（二）错误记录

1、错误 1

（1）错误现象

Epc 内容发生错误。

(2) 分析定位过程

发现指令取址例外和 syscall 同时发生，结果 EPC 写地址出错。

(3) 错误原因

同时触发例外的情况没有处理好 EPC 的赋值问题。

(4) 修正效果

处理好 EPC 的赋值问题之后，EPC 内容正确。

(5) 归纳总结（可选）

本次实验使用了触发例外的阶段不固定的方式，可能会同时触发多种例外，需要做好 cp0 寄存器赋值的处理。

1、错误 2

(1) 错误现象

LB 指令触发例外。

(2) 分析定位过程

发现 LB 指令后跳转到了例外处理地址。

(3) 错误原因

LB 指令意外的触发了地址错例外。

(4) 修正效果

综合考虑 LH, LB, LW, SB, SH, SW 各指令后，修改地址错例外的判定条件。

(5) 归纳总结（可选）

四、实验总结（可选）

本次实验设计难度较小，因此我写代码速度较快，导致了后续 debug 工作费时较长。以后写代码还是要设计好清晰的架构再开始。