

实验 4_2 报告

2016K8009909006

刘杰

一、实验任务（10%）

为 myCPU 增加例外与中断支持，完成功能测试，并支持运行一定的应用程序。需要完成如下任务：

- (1) 增加 BREAK 指令。
- (2) 增加 CP0 寄存器 COUNT、COMPARE、BADVADDR。
- (3) 增加 break 例外，地址错，整数溢出，保留指令例外支持。
- (4) 增加时钟中断例外支持，绑定在硬件中断 5 号上，CAUSE 对应的 IP7。增加 6 个硬件中断和 2 个软件中断支持，绑定在 CAUSE 的 IP6-IP0。
- (5) 完成所有的功能测试，运行电子表和记忆游戏。

二、实验设计（30%）

为优化例外处理的电路，所有例外在统一的流水级处理（定为写回级），即在例外发生后，先传递到写回级，在写回级提交例外，进行处理。在例外发生后的直到进入例外处理入口，所有进入流水线的指令将被设置为无效，即不改变 CPU 状态。实现方法为设置信号，该信号在例外发生时拉高，在例外传递过程保持不变，在例外提交时拉低，当该信号为高时，清空例外发生指令前的流水线，并将之后进入的指令设为无效。

在 lab4-1 的基础上，增加 3 个 CP0 寄存器 COUNT,COMPARE 和 BADVADDR。COUNT 寄存器每两个时钟周期自增 1，与 COMPARE 寄存器比较，如相等则触发时钟中断。此外，每次写 COMPARE 寄存器会将 CAUSE 寄存器 TI 位域置 1,在写 COMPARE 寄存器时清 0。BADVADDR 寄存器用于存储在触发地址错例外的指令地址。

为便于软硬件中断处理，将其固定的标在译码级，传递到写回级后报出。如果流水级上出现多个例外，先处理最先到达写回级的例外，之后的例外不做处理。

三、实验过程（60%）

（一）实验流水账

2018.11.24

9:00-10:00 阅读讲义和指令手册，编写代码。

2018.11.24

13:00-13:40 仿真，整数溢出例外出现 bug，修复 bug。

14:00-14:30 修复 bug 后仿真，在取指地址错例外发现 bug，修复 bug。

16:00-17:00 修复 bug 后仿真，在时钟中断发现 bug，修复 bug。

18:00-18:30 修复 bug 后仿真，在多个中断同时出现测试点发现 bug，修复 bug 后测试通过。

2018.11.25

18:00-20:00 完成实验报告。

（二）错误记录

1、错误 1

（1）错误现象

整数溢出例外发生之后，MFHI 指令报错。

（2）分析定位过程

MFHI 指令执行过程没有问题，HI 寄存器的值有误，往回查找最近一条写 HI 寄存器的指令，发现是 divu 指令后的一条 MTHI 指令，在 MTHI 没有将除法结果写入 HI 寄存器时，HI 寄存器的结果符合 trace 的比对结果。

（3）错误原因

除法指令是紧跟着发生溢出的 ADD 指令，ADD 指令之后的指令都不应该执行。但此时只清空流水线沿用 syscall 例外的处理方法，只清空了取指级的指令。而溢出例外发生时，译码级和取指级都有指令，取指级恰好为除法指令。

（4）修正效果

修改例外发生时流水线的清空电路逻辑，将例外发生指令前的流水级都清空，再次仿真通过 break 测试点。

（5）归纳总结（可选）

例外发生后，例外指令会‘堵住’大门，让之后的指令都无法执行。

2、错误 2

（1）错误现象

取指地址错发生后，例外处理的第一条指令错误，即 PC=0xbfc00380 的指令有误。

（2）分析定位过程

取指地址错例外处理时，0xbfc00380 处的指令执行有误，查看指令码发现为 0x00000000，进一步发现 PC=0xbfc00380 时取指使能未拉高。

（3）错误原因

取指地址错发生后，将取指使能拉低，直到进入例外处理时再拉高，但是因为拉高的判断逻辑为 PC 和 0xbfc00380 相等，故晚了一拍，所以 0xbfc00380 没有取出有效指令。

（4）修正效果

把 syscall 指令后的指令相关控制信号都设置为无效，确保 syscall 指令后的指令都未执行。

3、错误 3

(1) 错误现象

时钟中断标记指令 PC 值有误。

(2) 分析定位过程

时钟中断测试点，在没有重新设置 COMPARE 时，COMPARE 为 0。将 COUNT 置为 0 后，立即发生时钟中断，但此时是关中断的，所以这个时钟中断应不发生。

(3) 错误原因

没有关注 CP0_STATUS 寄存器开关中断的位域，修改控制通路，使得关中断时即使发生中断也不进行提交处理。

(4) 修正效果

软件写 CP0 寄存器相应位域可以实现开关中断。

4、错误 4

(1) 错误现象

同时发生多个例外，重复处理，即 PC 值连续跳转到 0xbfc00380。

(2) 分析定位过程

PC 值停留在 0xbfc00380 多个周期，查看例外提交信号也拉高多个周期，有多个例外同时发生并连续提交，每次提交后 PC 都会跳转到 0xbfc00380，这就造成 PC 停留的假象。

(3) 错误原因

在进入某个例外的处理后，没有屏蔽其他例外，但又没有实现例外嵌套。所以，在进入一个例外处理后，屏蔽其他例外的提交，使得不发生例外嵌套。

四、实验总结（可选）

这次写的例外处理的代码有点乱，可能要重构一下代码。