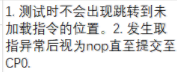
需要解决的问题

一、进入异常处理程序的操作

1、异常信号时使能的判断：（异常前）

发生取指异常应该视为nop。

问：mult异常，但是会影响E级里的乘除，但是不应该有该影响.

核心思想：异常指令包括之后应该不会改变GRF,HI,LO,DM,CP0的值。

参考指令序列:E级异常——E->mult 原因：取指异常|未知指令|中断|算数溢出（除0）故ALU\_Exc应该为ALU的输出结果

M级异常——E->mult(不异常)、M->异常指令 原因均有可能

解决方法：En要或

E\_En= M\_En&!(E\_IFU\_Exc|E\_undefined\_code|ALU\_Exc)

M\_En=!(M\_IFU\_Exc|M\_undefined\_code|Read\_Exc|Save\_Exc|ALU\_Exc|if\_Int)

If\_Int=HW\_Int…….

注：写入CP0的Cause应该不会触发异常吧？不会。

2、EPC的写入的判断：

W->W写入的时候有中断，此时W级指令是否执行效果？

据pb说，不执行。即中断的受害指令以M级为准。

那么情况有: M->为延时槽指令

M->不为延时槽指令

M->为气泡，实际指令应为延时槽指令

M->为气泡，实际指令不为延时槽指令

解决方法为：插入气泡时PC+8不清零，if\_delay不清零。

即向CP0传输虚假信息。

EPC=(if\_delay)? PC+8-8:PC+8-8-4;

另一种方法：追溯方法，追溯第一个PC+8不为0的数写入EPC，从M级开始追溯。

3、Cause写入的判断:

ExcCode的编写：会存在冲突情况么？ 不会，中断优先，四个异常不会冲突

关键原因：取指异常会传nop指令。实现取值异常传递nop：（默认nop为定义指令）

实现方法：assign true\_IF=(IFU\_Exc)? 0:IF;

注：目前未检查异常信号的来源是否正确。ALU\_Exc无错

4、SR写入的判断：

无

5、清除流水寄存器里的数据（除了PCadd8与if\_delay）。也清，采用了追溯方法

6、IFU中PC值得跳转（异常中断时得跳转地址具有最高优先级）。

二、从异常中退出：

1、EPC->PC,EPC有较高优先级。

（1）注意可能会对EPC造成修改的指令序列：

M->mtC0(对EPC进行修改)， D->eret

E->mtC0(对EPC进行修改)， D->eret

关键原因：未处理新加三条指令的转发暂停

解决方案：增加暂停条件：

stop=()||(M\_CP0\_En&&M\_IF[`rd]==5’d14)||( (E\_CP0\_En&&E\_IF[`rd]==5’d14))

注：以上作废，采用点3方案。

2、保证eret后面指令不执行

解决方案：eret->D\_clear(控制信号)

3、EXL置为0立刻生效，（不行）

冲突原因：若写入非NPC指令，则不会执行

解决方案：eret到M级再进行效果，即采用eret信号右控制信号生成。

三、协处理指令得暂停、转发

暂停：莫得，写好Tuse，Tnew即可；

转发：W级多了个CP0\_out；

四、测试程序编写

1、基本测试：用取指异常

2、考虑异常值的暂停情况：