

### 1. mfhi mflo mfc0 后发生MEM级数据依赖，需要用MEM\_OutB的数据旁路

解决：在MEM级后面加一个ALUOut和OutB的多选器，用于选择旁路的数据

### 2. add检测出来了溢出异常（MEM级），之后如果跟一条写HILO的指令（如乘除法等，EXE级就写回），必须关掉HILO的写使能信号，（与其他指令不同，这两个都是在EXE级写回的）

解决：在HILO的写使能中加一个与，不发生异常才写回

### 3. 取地址没有对齐的异常检测（在MEM级判断PC是否对齐），这里填入的EPC就是出错的那个没有对齐的地址。

解决：在Exception中添加相应的取指令异常检测

### 4. 发生两次软件中断的写入

解决：发现在interrupt例外检测时漏写了判断status的【1】是否为1，要加一下，如果不是1才去写EPC和cause寄存器

### 5. 在发生异常后，后面一条指令是除法的情况，除法的状态机需要做出改变

### 6. store后面跟一条load的情况下，要阻塞一个周期（这里的阻塞不只是把前面的PC和IFID写使能关掉，还要把IDEXE寄存器清零。）

## 问题：

### 1. 软件中断之后，一定会跟着一条nop指令吗？一条软件中断指令，后面跟着一条R型指令，在我们的逻辑体系下，这条R型指令会被重复执行2次，可能存在问题。

```
0 bfc4c998: 2694c9c4 addiu s4,s4,-13884
1 /media/sf_nscscc2019/develop/trash/func_test_v0.03/soft/func/inst/n77_soft_int_ex.S:20
2 bfc4c99c: 3c070040 lui a3,0x40
3 bfc4c9a0: 34e7ff01 ori a3,a3,0xff01
4 bfc4c9a4: 2402ffff li v0,-1
5 bfc4c9a8: 3c03000f lui v1,0xf
6 bfc4c9ac: 3463f2ff ori v1,v1,0xf2ff
7 bfc4c9b0: 40804800 mtc0 zero,$9
8 bfc4c9b4: 40825800 mtc0 v0,$11
9 bfc4c9b8: 40876000 mtc0 a3,c0_sr
10 bfc4c9bc: 00000000 nop
11 bfc4c9c0: 40836800 mtc0 v1,c0_cause
12 bfc4c9c4: 1000ffff b bfc4c9c4 <n77_soft_int_ex_test+0x54>
13 bfc4c9c8: 00000000 nop
14 /media/sf_nscscc2019/develop/trash/func_test_v0.03/soft/func/inst/n77_soft_int_ex.S:21
15 bfc4c9cc: 16570078 bne s2,s7,bfc4cbb0 <inst_error>
16 /media/sf_nscscc2019/develop/trash/func_test_v0.03/soft/func/inst/n77_soft_int_ex.S:22
17 bfc4c9d0: 00000000 nop
18 /media/sf_nscscc2019/develop/trash/func_test_v0.03/soft/func/inst/n77_soft_int_ex.S:24
```

### 2. 为什么软件中断会存在两个周期，而wth改了一下Cp0中的写法，就过了测试点？有点想不明白，就只在CP0中加了一个if else end。但是去掉之后就不能过点？初步怀疑是因为某些写法导致的？