新的工作

# CP0寄存器：

SR

Cause

EPC

BadVAddr

EntryHi

EntryLo0

EntryLo1

Index

Count

Compare

EBase

## 几个问题：

### 软件在维护Compare寄存器时做了哪些工作？

根据时间片的分割需要修改Compare寄存器以进入时钟中断。

### 写入Compare寄存器会如何清除中断？

### 如何判断是否进入时钟中断

Process(clk)

if(reg\_cmp = reg\_cnt) then 是下个周期进入时钟中断吗？

Anyway, count <= count + 1

进入中断以后count还是否需要每次+1，如果不要，从什么时候开始不需要+1

### 如何判断进入硬件中断？

If EXL = ‘0’ and IE = ‘1’ and IM(certain bit) = ‘1’ and IP(certain bit) = ‘1’ then interrupt

### 如何进入中断以及进入中断后的操作

Clk’event and clk = ‘1’

If (KSU = ‘0’ or EXL = ‘1’) and not\_masked 进入中断（中断屏蔽是干嘛的？）

EPC <= current\_PC;

BadVAddr <= mem\_addr;

异常原因如何获得？ Cause <= … TLB的中断号是？

SR(EXL) <= ‘1’;

If TLB\_missing then EntryHi <= BadVAddr(VPN)

PC <= EBase+0x180;

More: mem\_en <= ‘0’

### 用户态下的限制的疑问

在用户态下不能执行CACHE/ERET/TLBWI/MFCO/MTCO，这些指令的译码直接在核心态译码，但是如果在用户态执行的话，是否需要报异常或者其他？还是直接当做NOP读过去，进入下一条指令的读取状态？

另外，是否需要限制在用户态下的访存虚拟地址在0x00000000, 0x7FFFFFFF之间？

### 中断号存在哪个寄存器中？

### 中断总线是指什么？IM IP分别做哪些工作？

IM, IP是被谁赋值的？

### 如何进行中断返回操作？

一旦译码得到ERET指令，就进行中断返回，具体是：

If ERET is called then

SR(EXL) <= ‘0’;

EPC <= PC;