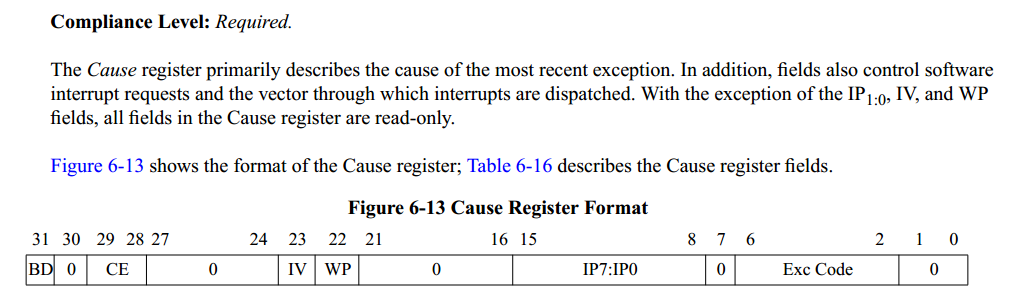
## Status

## 

EXL：exception level.

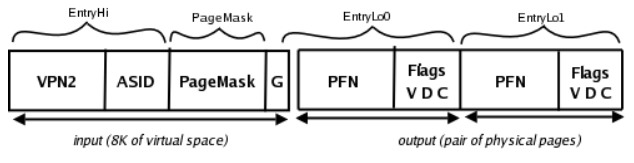
* 在发生除Reset/Soft Reset/NMI/Cache Error以外的异常时，置位
* 为1时，处于内核态，关中断

## Cause



Exc Code为2(TLBL)或3(TLBS)，代表读/写的TLB miss.

## 一个TLB数据项



VPN**2**有32-13位，因为对应的是两个页框，一个页4K，两个页8K。

（**2**指的是有两个页）

两个页的页框号在EntryLo0与EntryLo1的PFN域中，作为输出。

两个页框号（物理地址）**可以不连续**，但**虚拟地址是连续**的。

EntryLo0/Lo1的前6位可以用来存一些

## Context域



① PTEBase：页表起始地址。页表共占8M。（需要对齐。）

每一项占16字节，一共有219项，对应了4GB内存里的219对页面。

每一项有128位（16字节），对应一个TLB数据项（每个TLB数据项就是128位的）

每一项存一个虚拟地址→物理地址的转换。

② BadVPN2：需要在页表中查找的Index。

查找时，直接把Context的内容作为一个内存地址访问页表项。

## TLB/MMB控制指令

① tlbr：读tlb项，用index

② tlbwi：写tlb项，用index

③ tlbwr：写tlb项，用random

④ tlbp：查找当前VPN2、ASID跟EntryHi中匹配的TLB项，并把索引存到Index中。

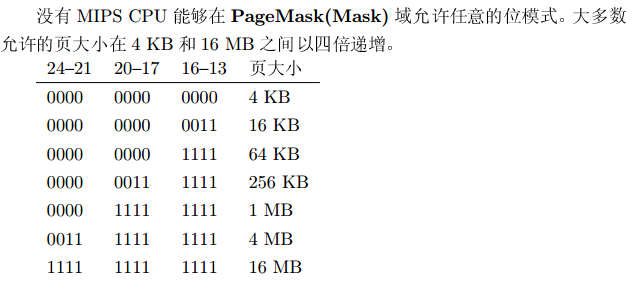
（先在EntryHi中存好VPN2、ASID，再查TLB项）。

作废某个TLB项时有用。

多于256个进程时，清除某个进程需要

因此TLB中不能用VPN2与ASID都相同的两项。

## PageMask寄存器



## 写入一个页面的过程

① 首次加载进内存时，页表项（内存）的D位清零。

② 试图写入时，发生一个异常，在驻留内存页表中设置一个”modified”位，即D位；同时也要设置TLB中的D位；

③ 以后就可以写入了。

## 发生TLB重填异常的处理过程

（**MIPS第三卷，5.2.9**；或**see MIPS run，14.4.8**）

① 发生异常时，

## 虚拟内存的含义

1、进程运行所需的内存空间大于内存总容量，进程觉得自己有很大一段空间，实际上只有很少的一部分是在内存中。

2、MMU的作用：把虚拟地址映射成物理地址（实地址）

## MIPS地址分配（参考“see MIPS run”14.4、2.8）

这里的地址都是**虚拟地址**。非粗的表示虚拟地址=物理地址。

1、0x8000\_0000 – 0xA000\_0000：Kernel Unmapped cached，kseg0。

无需TLB管理，直接映射；通过Cache。

2、**0x0000\_0000 – 0x7fff\_ffff：User mapped，通过MMU**。

从虚拟零地址开始的一两页不作地址映射。这样使用空指针就会被捕获。

用户堆用malloc()函数，从0向上增长；

栈从顶部向下增长。

自己设计一个malloc函数，给用户进程用。

3、0xA000\_0000 – 0xC000\_0000：Kernel Unmapped uncached，kseg1，不通过Cache访问；

4、**0xC000\_0000 – 0xffff\_ffff：Kernel Mapped，通过MMU**。

不考虑Supervised Mode（管理模式）。

## 代码相关

1、可以指定一个index变量（自行设计

## 用户态与内核态的转换过程

1、用户态向内核态：syscall/trap/中断/运行中异常

* 硬件：EXL=1。只要EXL=1或ERL=1，就处于内核态。
* 软件：编写相应的异常服务程序。

2、内核态向用户态：eret指令。

* 硬件：EXL=0。
* 软件：手动设置CU0=0。（可以一直设置为0）

需要解决的问题

**1、怎么开关MMU？（怎么设置MMU）**

**0x8000\_0000开始的虚拟地址默认不经过MMU。**

**用户态用Status的第31-28位来设置。所以每个用户进程只有2G的虚拟空间。**

2、页表的一项有几个字节？8个字节还是16个字节？

一个页表项有16个字节。(EntryLo0, EntryLo1, EntryHi, PageMask)

3、内核进程需要通过TLB转换地址吗？（用户进程是肯定要的。）

可以转，可以不转。

4、用户态、内核态转换？

syscall从用户态转换到内核态。

eret从内核态转换到用户态。

用户态：CU0=0, UM=1, EXL=ERL=0

5、做swap会不会很慢？

I/O操作是阻塞式的，会很慢。而且需要关中断。

6、异常/中断发生时，自动关中断。

7、syscall的过程中，EXL置1，关中断，自动变成内核态。