

预备知识

首先介绍一下本次试验需要用到的一些预备知识。这些在体系结构课程中基本都讲过，这里为大家重点梳理一下，讲的不够详细的地方，烦请参照体系结构教材。

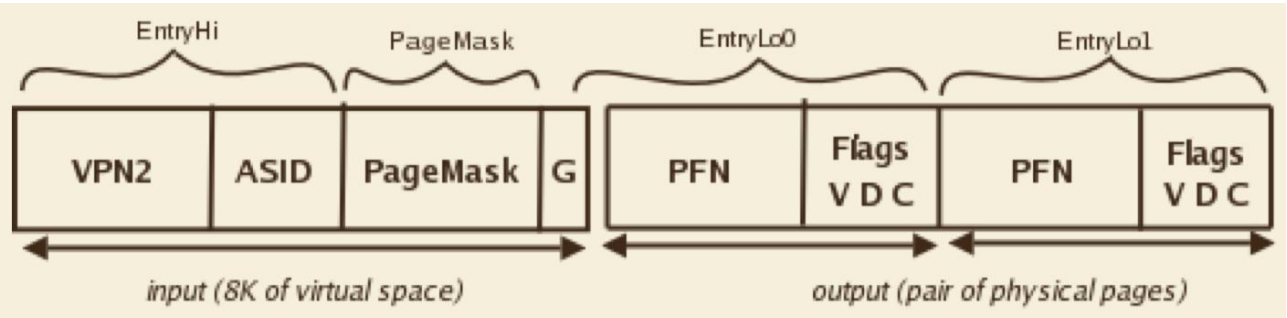
mips 内存映射：

mips 的一个进程的虚拟内存映射如图。

地址范围	容量	映射方式	cached
0xc0000000-0xffffffff	1GB	查找 TLB	Yes(TLB)
0xa0000000-0xbfffffff	0.5GB	地址-0xa0000000	No
0x80000000-0x9fffffff	0.5GB	地址-0x80000000	Yes(Config)
0x00000000-0x7fffffff	2GB	查找 TLB	Yes(TLB)

我们介绍的是 mips 32bit 的结构，因此程序看到的虚拟地址就是 $2^{32}=4\text{GB}$ ，空间大致分成了四块，0-2GB 是通过页表来实现虚实映射的，一部分页表缓存在 TLB 中，TLB 中保存的是一个一个的映射 $A \rightarrow B$ ，A 是虚拟地址，B 是物理地址。因此，访问过程是：处理器需要访问虚拟地址 A，在 TLB 中查找 A，得到物理地址 B，完成访存。中间两个 0.5GB 的空间比较特殊，他们的虚实转换不需要页表，直接默认虚拟地址减去一个固定的值即可。2-2.5GB 是过 cache 的，2.5-3GB 不过 cache。回想我们目前的代码都是在 2.5-3GB，用的是最“朴素”的映射方式，我们不需要关心虚实转换以及 cache。在本实验中，我们要需要使用 0-2GB 这块内存，因此我们需要实现虚实转换。另外为了简单，我们目前还是把 cache 关闭。

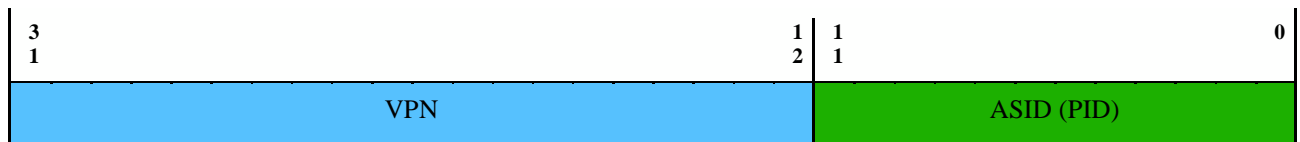
TLB 结构与操作



TLB 可以简单看做页表的缓存，可以看做是一个表，其表项结构如图：

mips 使用的是一个页表项表示连续的两个页面。pagemask 和 G 我们不用管，我们这里只关心 VPN2 和 ASID。TLB 查询是在 TLB 中查找虚拟页号(VPN)和 ASID(可以用进程号表示)一致的表项，如果存在，则 TLB 命中，否则 TLB 缺失。请注意图中的花括号上的寄存器的名字并不是说 TLB 表项里有寄存器，是说 TLB 表项是和这些寄存器相互传输数据的。这中间我们需要了解的与 TLB 有关的几个寄存器是：EntryHi，EntryLo0，EntryLo1，Index。以及 TLB 相关的几个汇编指令：tlbp, tlbr, tlbwr, tlbbwi。我们逐一介绍。

EntryHi：



EntryLo0-1



Index 就是普通的寄存器，表示 TLB 表中的 index。

几个指令介绍。

tlbp: tlb lookup

搜索虚拟页号及 ASID 跟当前 EntryHi 中的值相匹配的 TLB 项，并把该项的索引保存到 Index 寄存器。若没有找到匹配则 Index 会被设置成负值。

tlbr: read TLB entry at index

将 index 选中的 TLB 表项的内容传送到 EntryLo0 和 EntryLo1 中。

tlbwi: write tlb entry at index

将 EntryLo0 和 EntryLo1 中的内容写到 index 选中的 TLB 表项中。

tlbwr: write tlb entry selected by random

随机在 TLB 中选择一个位置，将 EntryLo0 和 EntryLo1 中的内容写入。