

它的一部分读入内存时，可以把CPU交给另一个进程使用。

3.3.1 分页

大部分虚拟内存系统中都使用一种称为分页 (paging) 的技术，我们现在就介绍这一技术。在任何一台计算机上，程序引用了一组内存地址。当程序执行指令

```
MOV REG, 1000
```

时，它把地址为1000的内存单元的内容复制到REG中（或者相反，这取决于计算机的型号）。地址可以通过索引、基址寄存器、段寄存器或其他方式产生。

由程序产生的这些地址称为虚拟地址 (virtual address)，它们构成了一个虚拟地址空间 (virtual address space)。在没有虚拟内存的计算机上，系统直接将虚拟地址送到内存总线上，读写操作使用具有同样地址的物理内存字；而在使用虚拟内存的情况下，虚拟地址不是被直接送到内存总线上，而是被送到内存管理单元 (Memory Management Unit, MMU)，MMU把虚拟地址映射为物理内存地址，如图3-8所示。

图3-9中一个简单的例子说明了这种映射是如何工作的。在这个例子中，有一台可以产生16位地址的计算机，地址范围从0到64K，且这些地址是虚拟地址。然而，这台计算机只有32KB的物理内存，因此，虽然可以编写64KB的程序，但它们却不能被完全调入内存运行。在磁盘上必须有一个可以大到64KB的程序核心映像的完整副本，以保证程序片段在需要时能被调入内存。

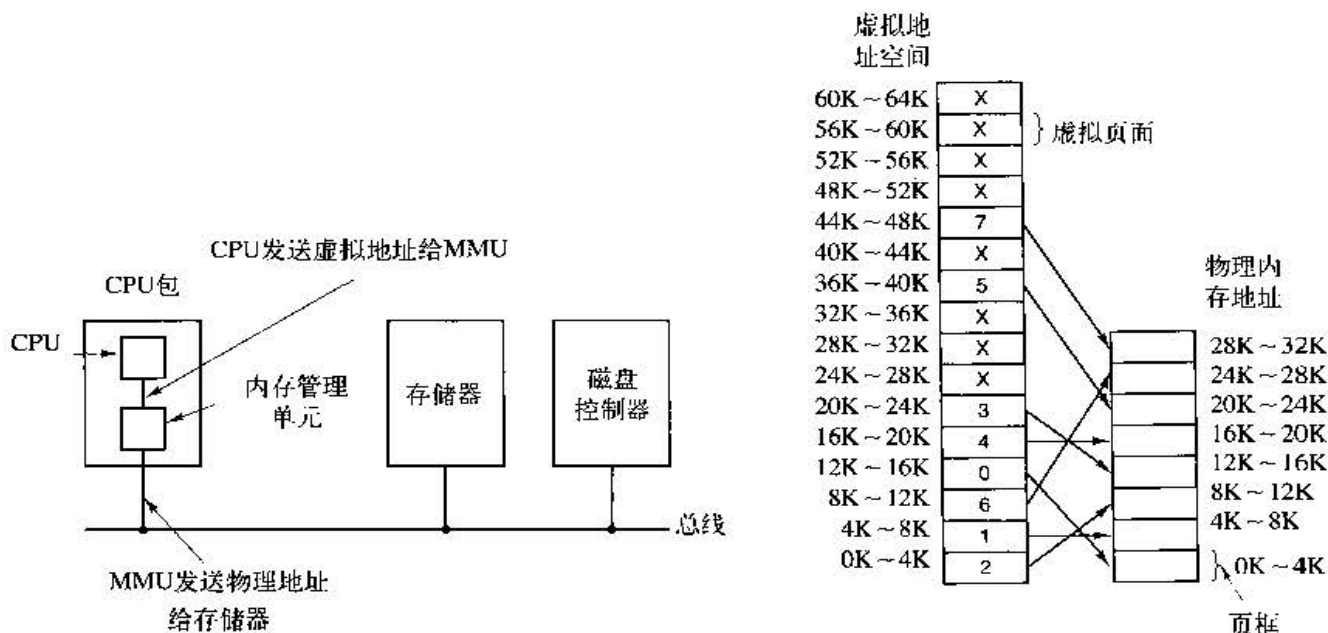


图3-8 MMU的位置和功能。这里MMU作为CPU芯片的一部分，因为通常就是这样做的。不过从逻辑上看，它可以是一片单独的芯片，并且早就已经这样了

图3-9 页表给出虚拟地址与物理内存地址之间的映射关系。每一页起始于4096的倍数位置，结束于起址加4095，所以4K到8K实际为4096~8191，8K到12K就是8192~12287

虚拟地址空间按照固定大小划分成称为页面 (page) 的若干单元。在物理内存中对应的单元称为页框 (page frame)。页面和页框的大小通常是一样的，在本例中是4KB，现有的系统中常用的页大小一般从512字节到64KB。对应于64KB的虚拟地址空间和32KB的物理内存，我们得到16个虚拟页面和8个页框。RAM和磁盘之间的交换总是以整个页面为单元进行的。

图3-9中的标记符号如下：标记0K~4K的范围表示该页的虚拟地址或物理地址是0~4095。4K~8K的范围表示地址4096~8191，等等。每一页包含了4096个地址，起始于4096的整数倍位置，结束于4096倍数减1。

当程序试图访问地址0时，例如执行下面这条指令

```
MOV REG, 0
```