Pentium 芯片中有两个独立的 cache，每个 8K。P2 芯片中也有两个独立 cache，分别是 8K 字节的四路组相联指令 cache 和 8K 字节的两路组相联的数据 cache，L2 cache 为 256KB～1MB。P3 芯片中包含 16K 字节指令 cache 和 16K 字节数据 cache。P4 芯片中包含 12KB 指令 cache 和 8KB 数据 cache，L2 cache 为 256KB。AMD 公司的 Athlon 4，L1 cache 为 128KB(64KB 指令 cache 和 64KB 数据 cache)，L2 cache 为 256KB。

4.6 虚拟存储器

4.6.1 虚拟存储器的工作原理

存储系统设计的基本目标是设计一个访问速度快、存储容量大的存储系统。在存储系统中采用 cache 大大提高了存储系统的访问速度，然而它不能解决存储系统的容量矛盾：一方面，受技术和成本的限制，主存空间不可能无限扩大，最大主存空间受限于 CPU 可寻址主存空间；另一方面，程序员总希望能够有一个大于主存空间的编程空间，这样编写程序不受实际主存大小的限制；再者，计算机系统中同时运行着多个进程，每个进程都需要有自己的地址空间，不可能为每个进程都提供整个地址空间的存储器。虚拟存储器是解决上述问题的有效方法，它由英国曼切斯特大学的 Kilburn 等人于 1961 年提出，经过 20 世纪 60 年代到 70 年代的发展和完善，目前，几乎所有的计算机都采用了虚拟存储器系统。

在存储系统的层次结构中，虚拟存储器处于“主存-辅存”存储层次，通过在主存和辅存之间增加部分软件(如操作系统)和必要的硬件(如地址映射与转换机构、缺页中断结构等)，使辅存和主存构成一个有机的整体，就像一个单一的、可供 CPU 直接访问的大容量主存。程序员可以用虚拟存储器提供的地址(称为虚拟地址)进行编程，这样，在实际主存空间大小没有增加的情况下，用户编程不再受实际主存空间大小的限制，因此，把这种存储系统称为虚拟存储器。图 4.46 给出了实现虚拟存储器的典型组织结构。

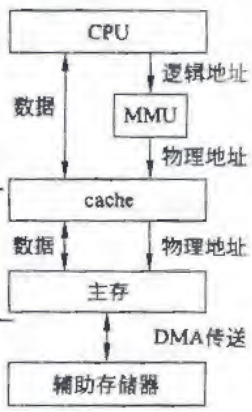


图 4.46 虚拟存储器的组织结构

在虚拟存储系统中程序运行时，CPU 以虚拟地址访问主存，由存储管理控制部件 MMU(Memory Management Unit)找出虚地址和实地址之间的对应关系，并判断这个虚地址对应的内容是否已经在主存中。如果已经在主存中，则通过 MMU 的地址变换机构将虚拟地址变换成物理地址，CPU 直接访问主存单元；如果不在主存中，则把包含这个字的一页或一个程序段(与采用的虚拟存储器的类型有关)调入主存，并向 MMU 中填写相关的标志。

根据虚拟存储器中对主存逻辑结构划分的粒度不同，虚拟存储器可分成 3 种不同的类型，分别是页式、段式和段页式虚拟存储器。本书将重点介绍页式虚拟存储器。

4.6.2 虚拟存储器的地址映射与变换

在虚拟存储器中有3种地址空间，第一种是虚拟地址空间，也称为虚拟空间，它是应用程序员用来编写程序的地址空间；第二种地址空间是主存的地址空间，也称物理地址空间或实地址空间；第三种地址空间是辅存地址空间，也就是磁盘存储器的地址空间。与这3种地址空间相对应，有3种地址，即虚拟地址、主存地址和磁盘存储器地址（也称磁盘地址或辅存地址）。

我们知道，cache中的地址映射是将主存中的数据按照某种规则调入cache；虚拟存储器中的地址映像也有类似的功能，它把虚拟地址空间映像到主存空间，也就是将用户利用虚拟地址访问的内容按照某种规则从辅助存储器装入到主存储器中，并建立虚地址与实地址之间的对应关系。而地址变换则是在程序被装入主存后，在实际运行时，把虚拟地址变换成实地址或磁盘存储器地址，以便CPU从主存或磁盘中读取相应的信息。

4.6.3 页式虚拟存储器

以页（Page）为逻辑结构划分和信息传送单位的虚拟存储器称为页式虚拟存储器。在页式虚拟存储器中，虚拟空间和主存空间均划分成固定大小的页，不同类型的计算机对页大小的划分不同，有的页大小只有几百字节，也有的计算机系统中页的大小达几千字节。

1. 页式虚拟存储器的地址划分

在页式虚拟存储器中，虚拟地址被划分成虚拟页号（Virtual Page Number，VPN）和虚拟页偏移量（Virtual Page Offset，VPO）；同时物理地址被划分成物理页号（Physical Page Number，PPN）和物理页偏移量（Physical Page Offset，PPO）两部分。VPN和PPN分别构成虚拟地址和物理地址的高位部分；VPO和PPO分别构成虚拟地址和物理地址的低位地址，其位数决定了页面大小。因为物理页和虚拟页大小相同，因此VPO和PPO的位数相同。

2. 地址转换的基本原理

物理地址由PPN和PPO两部分构成，虚拟地址到物理地址的映射本质上就是如何将VPN转换成对应的PPN。页虚拟存储器中虚拟地址到物理地址转换的基本原理如图4.47所示。

页式虚拟存储器中虚拟地址与物理地址之间的转换基于页表进行。页表是一张保存虚拟页号和物理页号（也称实页号）之间对应关系的表格，由若干个表项组成，每一个表项又包括有效位和物理页号两部分。页表常驻内存，并通过虚地址中的虚页号作为索引来访问，因此，页表项的数量与虚拟地址中虚页号字段的位数有关，如虚页号字段为20位，则页表中表项的数量为2^{20}。利用虚页号实现虚拟地址转换成物理地址的过程如图4.48所示。

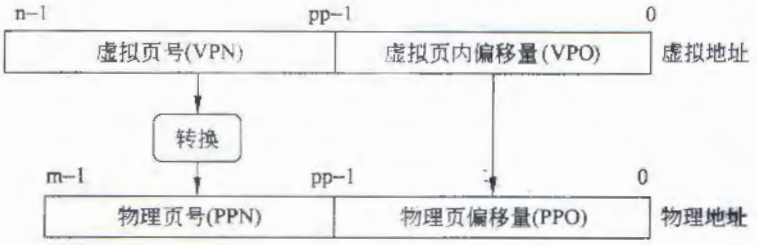


图 4.47 虚拟地址到物理地址的映射

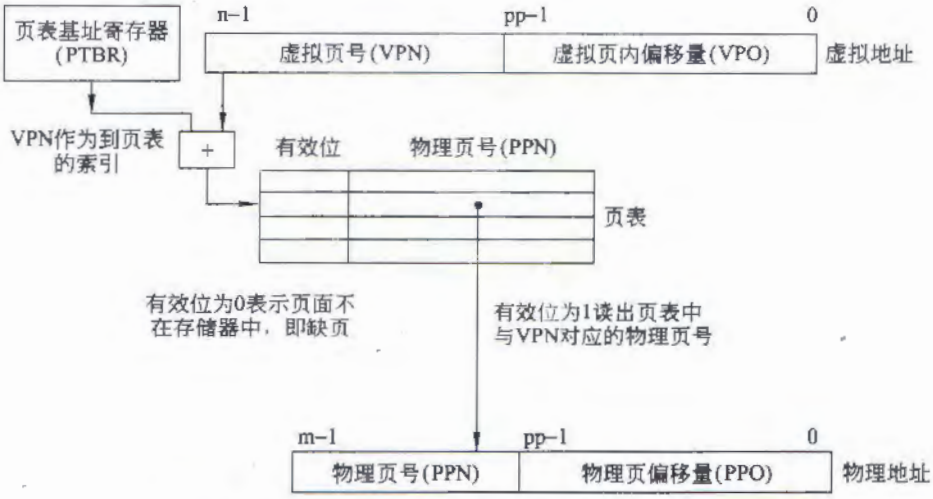


图 4.48 利用虚页号实现的虚拟地址与物理地址转换

图 4.48 中的页表基址寄存器(Page Table Base Register，PTBR)用于指出页表在内存中的位置，通过与 VPN 相结合就能访问到页表中与虚拟页号 VPN 相对应的物理页号 PPN，从而实现虚拟地址到物理地址的转换。

例 4.11 设主存容量为 16MB，按字节寻址。虚拟存储器容量为 4GB，采用页式虚拟存储器，页面大小为 4KB。完成下列各问：

(1) 计算物理页号、页内偏移字段、虚拟页号字段各为多少位。

(2) 计算页表中页表项的数量。

(3) 若部分页表内容如表 4.9 所示，求对应于虚拟地址$(00015240)_H$和$(03FFF180)_H$的物理地址。

表 4.9 部分页表内容

虚页号	有效位	物理页号	虚页号	有效位	物理页号
00010H	0	002H	03FFFH	0	153H
00015H	1	035H	…	…	…
…	…	…			

解：(1) 由于页面大小为 4KB，因此页偏移量字段为 12 位，物理页号字段的位数为 24 位－12 位＝12 位，虚物页号字段的位数为 32 位－12 位＝20 位。

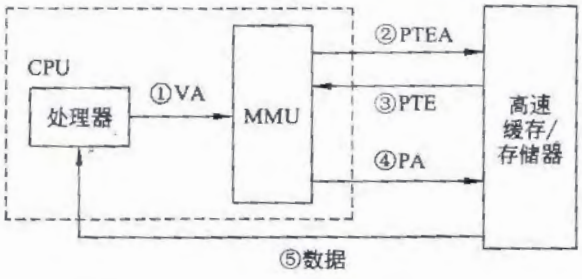
(2) 页表项数量与虚页号字段的位数相关，虚页号字段为 20 位，则页表项数为 2^{20}＝1024K 项。

(3) 从虚拟地址$(00015240)_H$和$(03FFF180)_H$中分离出的 20 位虚页号分别为 00015H 和 03FFFH。查页表可知与虚页号 00015H 对应的物理页号为 035H，而与虚页号 03FFFH 对应的页不在主存中，因为对应的有效位为 0。

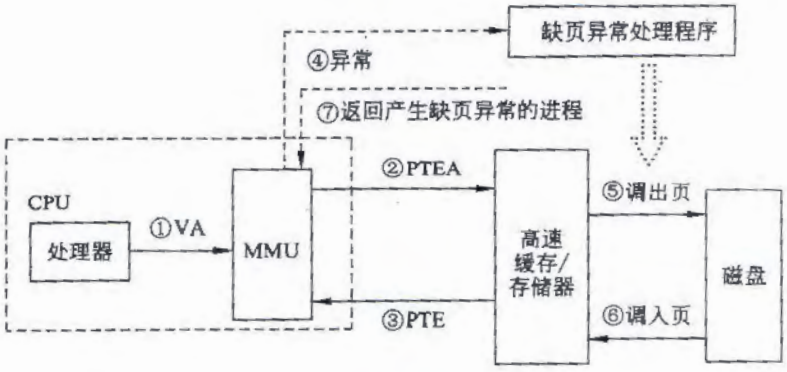
与虚拟地址$(00015240)_H$对应的物理地址为$(035240)_H$，其中的物理页内偏移字段 240H 直接来自虚拟地址的虚拟页内偏移字段。

3. 结合 cache 的页式虚拟存储器地址转换

在任何既使用虚拟存储器又使用 cache 的存储系统中，一般采用物理地址访问 cache。图 4.49 展示了一个结合 cache 的页式虚拟存储器的地址转换过程。



(a) 页面命中



(b) 缺页

图 4.49　结合 cache 的页式虚拟存储器地址转换

图 4.49 中所用符号的意义如下。

- VA(Virtual Address)：虚拟地址。
- PTEA(Page Table Entry Address)：页表项地址。
- PTE(Page Table Entry)：页表项。

• PA(Physical Address)：物理地址。

图 4.49(a)展示了页面命中时，CPU 硬件执行的步骤：

(1) 处理器生成一个虚拟地址，并把它传送给 MMU。

(2) MMU 利用页表基址寄存器 PTBR 和虚页号生成页表项地址 PTEA，访问存放在高速缓存/主存中的页表，请求与虚拟页号对应的页表项内容。

(3) 高速缓存/主存向 MMU 返回页表项 PTE，以构成所访问信息的物理地址。

(4) 若返回的 PTE 中有效位为 1，则 MMU 利用返回的 PTE 构造物理地址，并利用构造出的物理地址访问高速缓存/主存。

(5) 高速缓存/主存返回所请求的数据给处理器。

图 4.49(a)中当返回的 PTE 中有效位为 0 时，表示 CPU 要访问的页不在主存中，此时将出现页面不命中的情况，此时，将不能按照图 4.49(a)的处理流程来访问存储系统。处理缺页要求硬件和操作系统内核协作完成，图 4.49(b)给出了页面不命中的处理流程：

(1) 处理器生成一个虚拟地址，并把它传送给 MMU。

(2) MMU 利用页表基址寄存器 PTBR 和虚页号生成页表项地址 PTEA，访问存放在高速缓存/主存中的页表，请求与虚拟页号对应的页表项内容。

(3) 高速缓存/主存向 MMU 返回页表项 PTE，以构成所访问信息的物理地址。

(4) 若 PTE 中有效位为 0，则表明所访问的页不在主存，MMU 触发一次异常，调用操作系统内核中的缺页异常处理程序。

(5) 缺页处理程序根据替换算法确定出将被淘汰的页，如果这个页面的修改位为 1，则把该页面换出到磁盘，否则直接丢弃。

(6) 缺页处理程序从磁盘中调入新的页，并更新存储器中的 PTE。

(7) 缺页处理程序返回到原来的进程，驱使导致缺页的指令重新启动。CPU 将引起缺页的指令重新发送给 MMU，然后再重新执行图 4.49 (a)中的②～⑤步。

4. 利用 TLB 加速虚拟存储器地址转换

从上面的分析不难发现，即使不发生缺页异常，CPU 必须访问一次高速缓存/主存获得 PTE 后才能实现虚拟地址到物理地址的转换；如果发生缺页异常，则需要访问高速缓存/主存 3 次才能实现虚拟地址到物理地址的转换。为了降低虚拟存储器地址转换的开销，根据局部性原理，现代处理器都维护着一个转换旁路缓冲器 TLB(Translation Look-aside Buffer)作为页表映像的一部分。TLB 的组织与 cache 的组织方式类似，一般具有较高的关联度，大多采用全相联或组相联方式。当采用组相联方式时，按照组相联 cache 的地址划分方法，将虚页号划分成 TLB 标记和 TLB 索引两部分，便于快速判断所要访问的页面是否在主存，当页命中时还能在 MMU 中利用 TLB 返回的 PTE 完成虚拟地址到物理地址的转换。

图 4.50 展示了利用 TLB 加速虚拟存储器地址转换的基本流程。

图 4.50(a)展示了 TLB 命中时所包括的步骤：

(1) 处理器生成虚拟地址，并把它传送给 MMU。

(2) MMU 利用虚页号 VPN 查询 TLB。

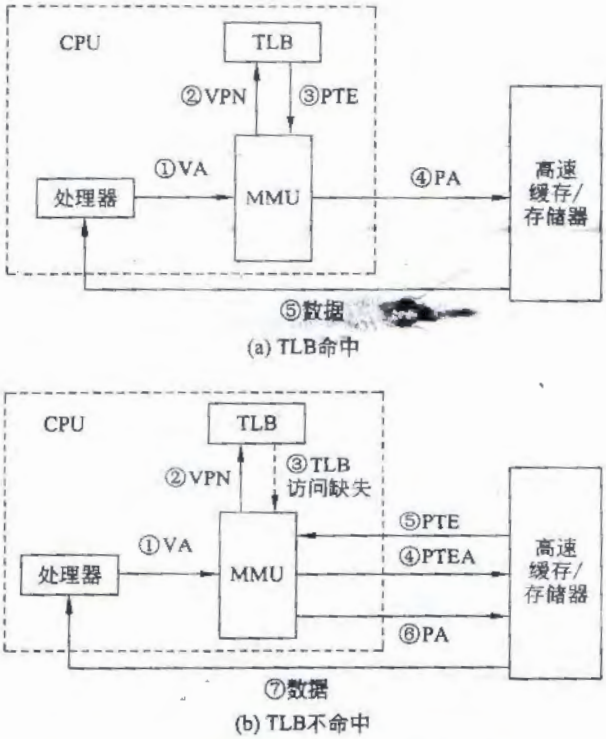


图 4.50　TLB 命中和不命中的操作视图

(3) 如果 TLB 访问命中(页表项在 TLB 中且相应 PTE 中有效位为 1),则 TLB 向 MMU 返回与 VPN 对应的 PTE。

(4) MMU 利用返回的 PTE 构造物理地址,并利用构造出的物理地址访问高速缓存/主存。

(5) 高速缓存/主存返回 CPU 请求的数据给处理器。

图 4.49(b)展示了 TLB 不命中(假定访问主存命中)时所包括的步骤:

(1) 处理器生成一个虚拟地址,并把它传送给 MMU。

(2) MMU 利用虚页号 VPN 查询 TLB。

(3) 如 TLB 访问不命中,则向 MMU 返回 TLB 访问缺失信息。

(4) MMU 利用页表基址寄存器 PTBR 和虚页号生成页表项地址 PETA,访问存放在高速缓存/主存中的页表,请求与虚拟页号对应的页表项内容。

(5) 高速缓存/主存向 MMU 返回页表项 PTE,以构成所访问信息的物理地址。

(6) 若返回的 PTE 中有效位为 1,则 MMU 利用返回的 PTE 构造物理地址,并利用构造出的物理地址访问高速缓存/主存。

(7) 高速缓存/主存返回所请求的数据给处理器。

对比图 4.49(a)和图 4.50(a)不难发现,采用 TLB 且访问 TLB 命中时,能减少一次访问主存的次数,因此又将 TLB 称为快表,而将存放在主存中的页称为慢表。

图 4.51 给出了页式虚拟存储系统中 TLB 和 cache 命中时详细的存储访问过程。

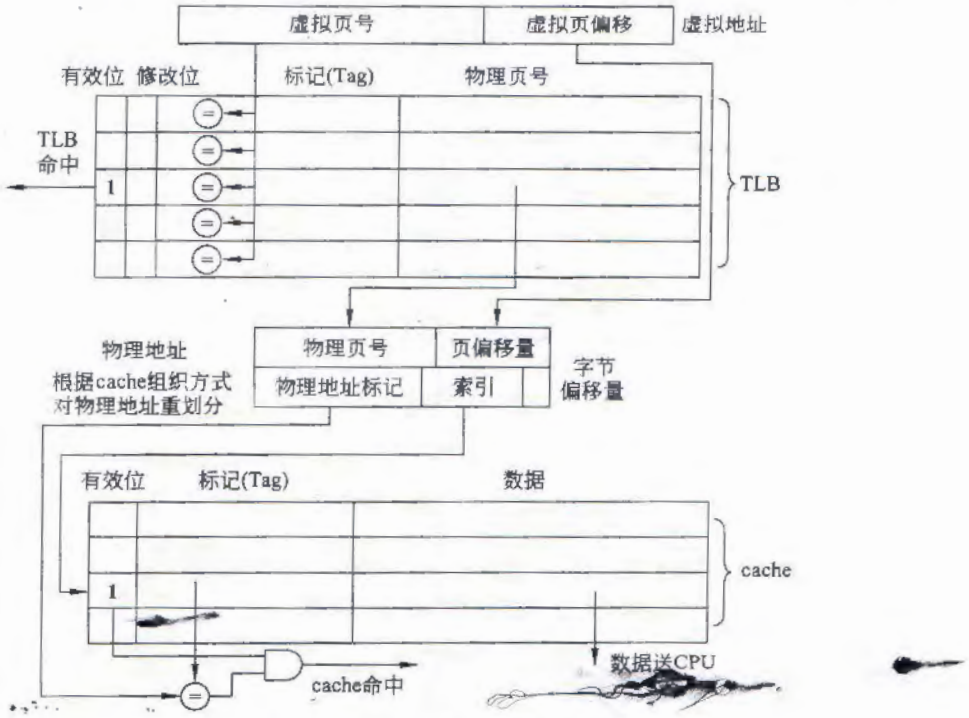


图 4.51 TLB 和 cache 都命中时存储访问的详细过程

图 4.51 中的 TLB 采用全相联，因此 TLB 中的标记部分保存的是虚页号。cache 采用直接映射。TLB 根据 MMU 发过来的虚页号从快表中读出对应的物理页号，并与来自虚拟地址的页内偏移地址构成物理地址访问 cache。访问 cache 时，根据其组织方式将物理地址重新划分成标记和索引两部分，并利用索引字段将 cache 中特定行(组)的标记值取出，与地址中的标记值进行比较，当 cache 命中时，将命中行中的数据取出送 CPU，该次存储访问完成。

需要说明的是，在包含快表和慢表的页式虚拟存储系统中，在进行地址变换时，往往同时查快表和慢表，如果查快表命中，则从快表中获得与虚页号对应的实页号同时终止查慢表的过程，这样就可以在几乎不降低主存访问效率的情况下访问虚拟存储器。

页式虚拟存储器中，页的大小固定且都取 2 的整数次幂个字节，因此，在页式虚拟存储系统中可以将虚拟存储空间和实存空间进行静态的固定划分，与所运行的程序无关，即页式虚拟存储器对程序员透明。页式虚拟存储器面向存储器自身的物理结构分页，有利于存储空间的利用与调度。但页式虚拟存储器不能反映程序的逻辑结构，一个在逻辑上独立的程序模块被划分成很多个页面，给程序的执行、保护和共享等都带来了不便。

4.6.4 段式虚拟存储器

以段作为基本信息单位在主存与辅存之间进行传送和定位的虚拟存储器称为段式虚拟存储器。一般将主存空间按照实际运行程序中的段来划分，因此，段长可大可小。段式

虚拟存储器中使用段表实现虚拟地址到物理地址的转换。段表项除包含段基址和装入位外，为了适应段长的可变特性，还需要增加一个段长字段。段基址是该段在主存中的起始地址，指向段中的第一个字；装入位表示该段装入主存与否；段长表示程序段占用了多少主存空间，它可用来检查访问地址是否越界。段表中还可以设置其他控制字段，如读、写和只能执行等，以作为对段保护的依据。

段式虚拟存储器虚拟地址由段号和段内偏移地址两部分组成。基于段表的虚拟地址与物理地址变换过程如图 4.52 所示，程序运行时根据段表基址寄存器的值与虚拟地址中的段号访问段表，获取该段的段基址。当装入位有效时，段基址与段内地址拼接组成物理地址。对于多道程序环境，一般为每道程序分配一个段表。

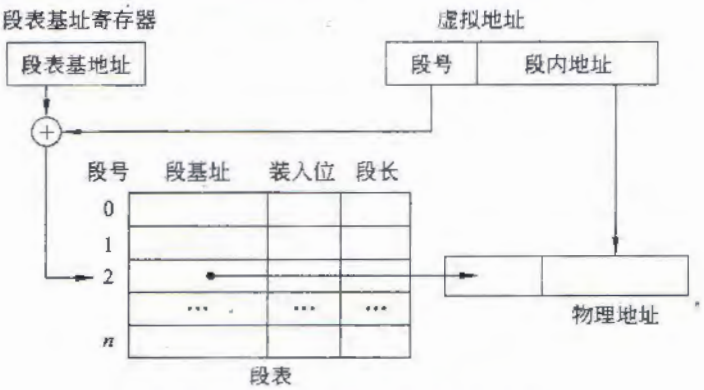


图 4.52　段式虚拟存储器地址变换

与页式虚拟存储器不同，段式虚拟存储器面向程序的逻辑结构分段，一个在逻辑上独立的程序模块可以作为一段来处理，从而使存储空间的划分与程序的自然段对应。以大小可变的段为单位进行调度、传送与定位，有利于对程序的编译、执行、共享与保护。另一方面，由于段的大小可变，不利于存储空间的管理、调度与优化。同时，由于段内必须连续，而各段的首尾地址又没有规律，必然导致地址计算比页式虚拟存储器要复杂。

4.6.5　段页式虚拟存储器

段页式虚拟存储器是页式虚拟存储器和段式虚拟存储器的结合。它将程序段分成固定大小的页。段页式虚拟存储器把程序按逻辑功能分段以后，再把每段分成固定大小的页。数据在主存与辅存之间的调入、调出以页为单位进行。每道程序由若干段组成，每段又由若干页组成，因此，段页式虚拟存储器的虚拟地址分成基号、段号、页号、页内偏移地址等 4 个字段。段基址表中每一个表项描述了一道程序含多少段，并给出段表基址；段表每道程序一个，每个段表项反映了该段由多少页组成(段长)，装入情况，并给出页表基址；页表每段一个，给出了该段中所有页的页基址和装入情况，其中一个页表项给出一页的页基址和该页的装入情况。

段页式虚拟存储器的地址变换过程如图 4.53 所示。具体过程包括：

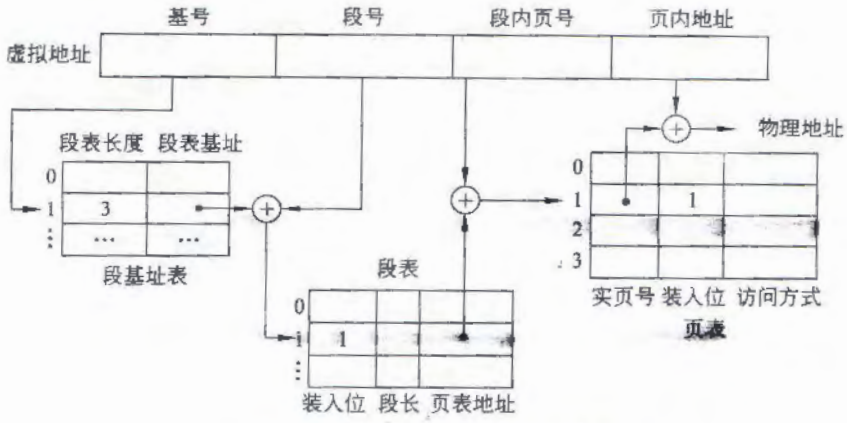


图 4.53 段页式虚拟存储器地址变换

(1) 从虚拟地址中分离出基号,并利用基号查段基址表。

(2) 从段基址表中得到该程序在段表中的起始地址,即段基址。

(3) 利用段基址与从虚拟地址中分离出的段号访问段表,同时判断该段是否已经装入主存。

(4) 若段已装入主存,则利用段表中的页表地址和从虚拟地址中分离出来的段内页号访问页表。

(5) 判断页是否在主存中,若在则通过页表中的实页号和虚拟地址中的页内地址获得与虚拟地址对应的物理地址。

当段或页不在主存中时,将按照一定的算法进行段的调度,详细内容请参见操作系统中的相关内容。

段页式虚拟存储器的引入是为了综合页式和段式虚拟存储器的优点,在这种方式下,将程序段按其逻辑结构分段,每段再分成固定大小的页,主存空间也划分成固定大小的页。相应地建立段表与页表,分两级分别查段表和页表实现虚拟地址到物理地址的转换,以页为单位调进或调出主存,以段共享与保护程序与数据。

4.7 存储保护

为了保证计算机系统能正确运行,当多个用户共享主存时,应防止由于一个用户程序出错而破坏其他用户的程序和系统软件,以及一个用户程序访问不是分配给它的主存区域。

在多道程序运行的情况下,为保证各用户信息资源的安全,系统应提供存储保护。具体来说,在系统中有多道程序处理时,某一程序改写或占用其他程序的存储空间,以及非法的写操作都是不允许的。

在虚拟存储器中,通常采用页保护、段保护和键保护方式。页、段的保护属同一类型,每道程序都有自己的段表和页表。段表、页表都有自己的保护功能,当虚拟的段号或页号出错时,在段表和页表中将找不到相应的实段或页,也就访问不了主存,也就不会侵犯其

他程序空间。

这种段表、页表给出的保护是在形成主存地址前的保护。若是在地址变换中出现错误，则产生了错误的主存地址此时可采用键保护。

键保护的基本思想是，为主存每一页分配一个键，称为存储键，每个用户的实存页面的键都相同。每道程序都有访问键，当数据要写入某一页时，访问键要与存储键比较。若两键符合，则允许访问该页，否则拒绝访问。

对于没有采用虚拟存储器的主存储器而言，可采用越界保护方式进行存储保护。系统对每个用户程序划定存储区域，用确定上、下界地址值的方式实现区域保护，用户程序不能改变上、下界值，所以它如果出现错误，也只可能破坏自身的程序，影响不到别的用户程序和系统程序，这是一种禁止越界的保护方式。

4.8 辅助存储器

4.8.1 磁表面存储器

磁表面存储器是在金属或塑料的表面涂上一层薄薄的磁性材料，再在磁性材料上存储信息的存储器。计算机系统中使用的磁表面存储器包括磁盘存储器、磁带存储器和早期的磁鼓存储器等。

磁表面存储器的主要特点是：存储容量大、位价格低、非破坏性读出、可长期保存和反复使用、存取速度较慢、对工作环境(如电磁场、温度、湿度、灰尘等)要求较高。它主要作外存储器(或称辅助存储器)使用，一般用来存储暂时不用的信息或需长期保存的大量信息。

1. 读写原理

磁表面存储器利用磁性材料剩磁的两种磁化方向(S-N 或者 N-S)来记录信息，包括写入和读出两个过程。磁表面存储器读写信息的基本原理如图 4.54 所示。

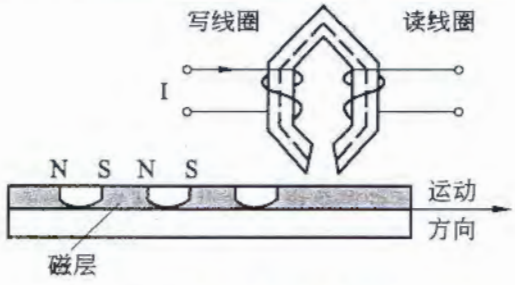


图 4.54 磁表面存储器读写信息的基本原理

写入信息时，在读写线圈中通上脉冲电流(电流的方向不同，则写入的信息不同)，磁头气隙处的磁场把它下面一小区域的磁层向某一方向磁化(S-N 或 N-S)，形成某种剩磁

由于翻转磁畴磁化方向的速率有限，故磁光盘需两次动作才能完成信息的写入。即第一次擦除，第二次写入新信息。

3. 光盘存储器的组成

光盘存储器由盘片、驱动器和控制器组成。驱动器主要由光头和控制电路组成。

光头部分包括：大功率激光器、光学系统、光电探测系统、调焦跟踪执行机构及快速径向移动机构等几部分。

控制电路部分包括：主轴恒角速度控制、光盘自动加载控制、点调焦跟踪伺服控制、快速存取控制、激光器读写功率控制、信号处理及内部系统控制等几部分。

光盘控制器与磁盘控制器相似，具有在驱动器和主机之间的接口功能。它主要进行驱动器与主机之间命令、数据的传送及纠错。为了加快存取速度，目前都配有缓冲存储器，其容量一般在 32～256KB 之间。

图 4.58 是一次写入型光盘的光学系统示意图。氦氖激光器产生的光束经分离器分成两束光。90%的光束用作写入光束，10%的光束用作读出光束。

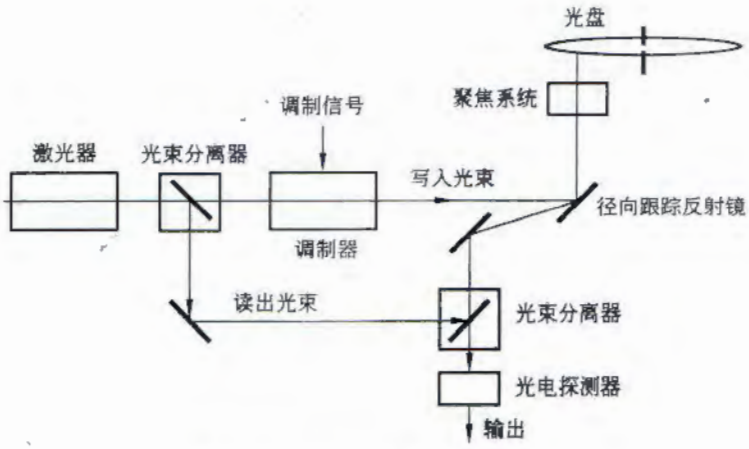


图 4.58 光盘的光学系统示意图

写入时，被调制信号送调制器，将写入光束调制。调制后的光束由跟踪反射镜反射至聚焦系统再射向光盘，在光盘记录介质上记录信息。

读出时，写入光束不起作用。小功率读出光束经光束分离器将从光盘反射回的读出光信号导入光电探测器，由光电探测器输出电信号。

4.8.3 冗余磁盘阵列

大容量存储系统中往往包含几十至几百块磁盘，为提高存储系统的性能、可靠性和数据的可用性，美国加州大学伯克利分校的 D. A. Patterson 教授提出了一种基于多磁盘冗余的存储系统，称为廉价冗余磁盘阵列 RAID(Redundant Array of Inexpensive Disk)或独立冗余磁盘阵列 RAID(Redundant Array of Independent Disk)，简称磁盘阵列。它将

多块独立的普通磁盘按照一定的方式组织与管理，构成一个大容量、高速度、高容错的存储系统。该存储系统具有以下基本特征。

(1) 在操作系统支持下，将一组磁盘视为一个独立的大型存储设备。

(2) 数据分布在这一组磁盘上。

(3) 提供良好的容错能力。在某个磁盘出现问题后，可以继续工作。

根据不同的数据组织与管理，RAID 具有多个不同的级别，不同级别的 RAID 具有不同的特性。

1. RAID 0

该方式下磁盘存储区被划分成条带(Strip)，条带大小可以是一个物理块，也可以是磁盘的一个扇区或其他存储单元。将待存放的文件也分成与条带区容量相同的块，然后按块依次轮流地存放在不同磁盘的条带区，如图 4.59 所示。

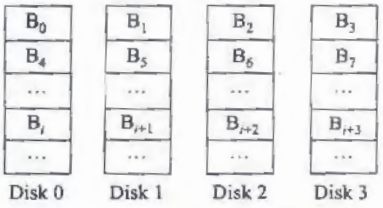


图 4.59 RAID 0 的数据分布

RAID 0 具有如下技术特点。

(1) 无数据冗余、无数据校验功能，因此它不具备数据的容错能力，数据的可靠性不高。

(2) 从 RAID 0 的数据分布看，其本质上是多磁盘体交叉存储(类似于主存的多体交叉存储)，多个磁盘可并行工作，存储系统的访问速度高。

(3) 条带的大小影响 RAID 0 的性能与应用。

① 条带大小对数据传输率的影响

小条带可将数据分配到更多的磁盘上，通过更多磁盘的并行工作可提高存储系统的数据传输率。

② 条带对 I/O 请求响应速度的影响

在面向事务处理的应用中，可能同时存在上百个 I/O 请求。此时，用户对 I/O 请求的响应时间比较关注。通过选择小而适中的条带，使得一次事务请求所传送的数据刚好集中在一个条带中，就能大大减少每个 I/O 请求的响应时间。

(4) 磁盘利用率高，由于 RAID 0 中没有冗余数据，所有的磁盘存储空间都可保存工作数据。

RAID 0 主要应用于对访问速度要求高，但对数据的可靠性要求不高的场合。

2. RAID 1

RAID 1 采用镜像盘冗余的方法将每份数据分配到两个不同的磁盘中，以提高数据的安全性和可靠性。RAID 1 数据分布如图 4.60 所示。

RAID 1 具有如下技术特点。

(1) 每个磁盘都有一个镜像磁盘，图 4.60 中备份磁盘 i 就是磁盘 i 的备份盘。

(2) 读请求时，可由包含该数据的两个磁盘中的任一个提供；写请求时，需同时更新两个磁盘中相应的数据块。

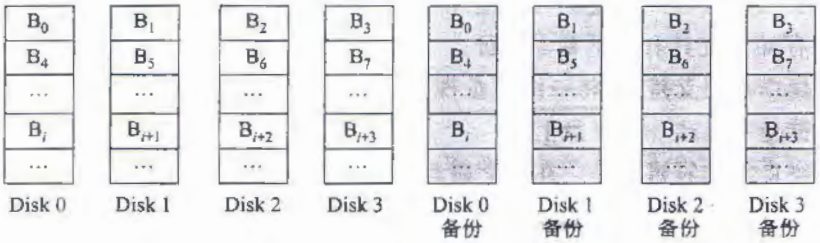


图 4.60 RAID 1 的数据分布

(3) 当一个磁盘被损坏时，数据仍可从另一磁盘获取。因此具有很高的安全性。

(4) 存储系统中磁盘的利用率只有 50%。

(5) 无数据校验功能。

(6) 对大批读请求来说，RAID 1 可以从对应的盘中并行读出。但对于写，其效率并不高。

由于 RAID 1 的读性能优于写性能，因此，RAID 1 主要应用于对数据的可用性要求高，且读操作所占比重较高的场合。

3. RAID 2

RAID 2 采用基于海明校验的磁盘体位交叉存取技术，即按照海明码校验技术对各数据盘上的相应位计算，并将计算出的校验位存储在多个校验盘的对应位。其中校验盘的数量与采用的海明校验技术有关，如果使用具有纠正一位错误并能检测两位错误的海明校验码，则校验盘的数量 r 与数据盘的数量 k 应该满足公式(4.2)。

$$2^r \geqslant r + k \tag{4.2}$$

例如数据盘 $k=4$ 时，校验盘的数量 $r=3$，该条件下 RAID 2 的数据分布如图 4.61 所示。

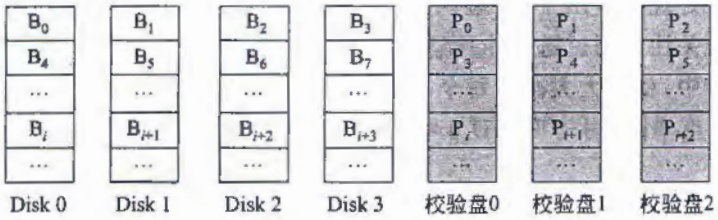


图 4.61 RAID 2 的数据分布($k=4, r=3$)

RAID 2 具有如下技术特点。

(1) 条带容量小，按位交叉存储，因此每个 I/O 请求都会访问到多个磁盘，导致 I/O 响应速度慢。

(2) 每个 I/O 请求都会访问到多个磁盘；对于单个读，所有磁盘同时读取，数据和相应的纠错码被送至控制器，若出现一位错，则由控制器立即识别并纠正。对于单个写，所有数据盘和校验盘都要进行写操作。

(3) 采用海明校验，具有纠错和检错功能，数据的可靠性高；但控制复杂。

(4) 冗余存放校验位，其数量与使用的数据盘的数量成正比。

(5) 由于按位存取，在 I/O 过程中所有磁盘上的磁头在任何时刻都处于同一位置，具有空间并行处理能力，数据传输率高。

受成本的影响，目前 RAID 2 很少被使用。

4. RAID 3

RAID 3 与 RAID 2 类似，也采用按位交叉存取和驱动器轴同步旋转技术。与 RAID 2 不同的是不论数据盘有多少，RAID 3 只需一块冗余盘来存放校验信息；另外，RAID 3 不采用纠错码，只用奇偶校验，即对所有数据盘上同一位置的一组位进行奇偶校验，奇偶校验位存于校验盘的相应位置。RAID 3 的数据分布如图 4.62 所示。

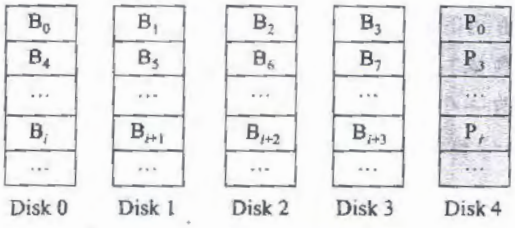


图 4.62 RAID 3 的数据分布

当某一磁盘损坏时，保存该盘上的数据可通过奇偶校验盘和其余磁盘上的数据进行恢复。

设 Disk0～Disk3 上保存数据，Disk4 为奇偶校验盘。采用偶校验时第 i 位的校验信息计算如公式(4.3)所示：

$$P_4(i) = D_3(i) \oplus D_2(i) \oplus D_1(i) \oplus D_0(i) \tag{4.3}$$

假设 Disk1 损坏，则其上的数据可通过公式(4.4)计算恢复：

$$D_1(i) = P_4(i) \oplus D_3(i) \oplus D_2(i) \oplus D_0(i) \tag{4.4}$$

RAID 3 的技术特点与 RAID 2 类似，不同点主要有两方面，其一是采用奇偶校验而不是海明校验，其二是校验盘只有一个，磁盘的利用率高。

与 RAID 2 相比，RAID 3 控制器简单，但由于仍按位处理，因此，RAID 3 不适合 I/O 请求较多的面向事务处理的应用场合。另外，由于所有的校验位都存放在同一个磁盘上，该盘容易成为访问的瓶颈，RAID 3 也不适合应用于大量写操作的场合。

5. RAID 4

RAID 4 与 RAID 3 类似，也采用奇偶校验和单个冗余盘。与 RAID 3 的不同主要表现在两方面：其一，RAID 4 不是按位校验，而是采用较大条区，一般为一个磁盘扇区；其二，各驱动器轴不再同步旋转，而是独立操作，因此 RAID 4 也称为独立存取技术。RAID 4 的数据分布如图 4.63 所示。

RAID 4 的技术特点如下。

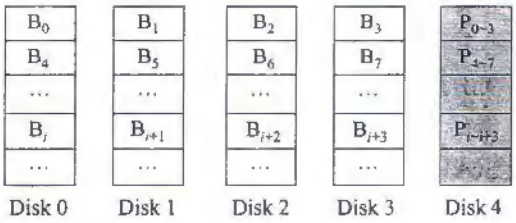


图 4.63 RAID 4 的数据分布

(1) 采用大条带区，I/O 请求响应速度快，但数据传输率不高。

(2) 采用奇偶校验技术。

(3) 各盘采用独立存取技术。

(4) 磁盘利用率高。

(5) 校验盘成为写访问的瓶颈。

下面将重点分析小数据写对 RAID 4 性能的影响。

当写操作只涉及修改一个磁盘条带，例如修改 Disk1 上的一个条带，此时校验盘上的校验信息按照公式(4.5)计算得到(仍以偶校验为例)：

$$P'_4(i) = D_3(i) \oplus D_2(i) \oplus D'_1(i) \oplus D_0(i) \tag{4.5}$$

其中 $P'_4(i)$和 $D'_1(i)$分别表示修改新计算的校验信息和修改后数据信息中的第 i 位。根据异或运算的特点，在公式(4.5)的右边异或 $D_1(i) \oplus D_1(i)$后，等式仍然成立。得到公式(4.6)：

$$\begin{aligned} P'_4(i) &= D_3(i) \oplus D_2(i) \oplus D'_1(i) \oplus D_0(i) \oplus (D_1(i) \oplus D_1(i)) \\ &= D_3(i) \oplus D_2(i) \oplus D_1(i) \oplus D_0(i) \oplus (D'_1(i) \oplus D_1(i)) \\ &= P_4(i) \oplus D'_1(i) \oplus D_1(i) \end{aligned} \tag{4.6}$$

其中从公式(4.6)不难看出，更新一次校验信息涉及两次读操作和两次写操作。两次读操作分别是读 $P_4(i)$和 $D_1(i)$(更新前的校验信息和数据信息)，两次写操作分别是写 $P'_4(i)$和$D'_1(i)$(更新后的校验信息和数据信息)。

当写操作涉及所有磁盘的条带时，不需要读原来的数据和校验信息，直接利用更新后的数据直接计算校验信息并写入冗余盘即可。

每次写数据时，校验盘成为数据传输的瓶颈。RAID 4 也不适合应用于大量写操作的场合。

6. RAID 5

RAID 5 与 RAID 4 类似，也采用大条带交叉存储和磁盘独立操作技术；不同的是 RAID 5 不将校验信息保存在一块磁盘上，而是将校验数据以循环方式放在每个磁盘中，从而可以有效避免校验盘成为写操作瓶颈的可能。RAID 5 的数据分布如图 4.64 所示。

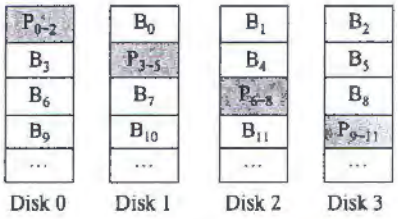


图 4.64 RAID 5 的数据分布

RAID 5 具有如下技术特点。

(1) 采用大条带区，I/O 请求响应速度快。

(2) 采用奇偶校验技术。

(3) 各盘采用独立存取技术。

(4) 校验信息在不同磁盘中循环存放，克服了 RAID 4 中校验盘成为写瓶颈的不足。

(5) 磁盘利用率高。

可以认为 RAID 5 是对 RAID 4 的改进，对大、小数据的读写都具有较好的性能，具有比较广泛的应用。

7. RAID 6

RAID 6 采用了按块交叉存放和双磁盘容错技术，其基本思想是对相同的数据进行两种不同的校验算法并将校验码分别存于两个磁盘中。这样，即使有两块盘同时出错也能将数据恢复出来，从而提高了数据的完整性和有效性。RAID 6 的数据分布如图 4.65 所示。

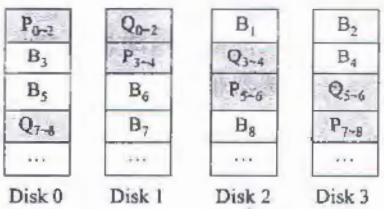


图 4.65 RAID 6 的数据分布

P 和 Q 是两个不同的校验算法。这种方案中，即使两个磁盘都出故障，数据仍可再生。因此，它提供了极高的数据可用性。相对 RAID 5 而言，其缺点是在组成中增加了一个磁盘，而且每次写都要进行 P 和 Q 两种校验以形成两个奇偶校验块。

以上各级 RAID 中，目前最常用的是 RAID 0，RAID 1，RAID 3，RAID 5 以及由 RAID 0 分别与 RAID 1，RAID 3 和 RAID 5 组合而成的 RAID 10，RAID 30 和 RAID 50 等模式。

RAID 既可采用硬件实现，也可采用软件实现。当采用硬件实现时，对应的磁盘阵列就称为硬磁盘阵列；当 RAID 基于软件实现时，对应的磁盘阵列就称为软磁盘阵列。硬磁盘阵列的性能比软磁盘阵列的性能高。

本章小结

根据冯·诺依曼计算机“存储程序”的设计思想，存储器是计算机系统中必不可少的用来记录信息的设备。设计一个大容量、高速度、成本低的存储器是存储系统的目标。本章主要介绍存储器分类、存储器分级结构、半导体存储器的工作原理、存储器组织、高速存储技术、虚拟存储器以及辅助存储器等主要内容。本章的主要学习目标包括：

(1) 熟悉存储器(含主存和辅助存储器)的技术指标。

(2) 熟悉主存中数据的存放模式。

(3) 熟悉存储系统的分级结构。

(4) 掌握静态和动态存储单元的工作原理；掌握动态存储器刷新的工作原理；了解新型存储器的发展方向。