



第 4 章

存储器

存储器是计算机系统必不可少的基本组成部分,用于存放计算机工作所必需的程序和数据。计算机工作的本质就是执行程序的过程,因此计算机工作的大部分时间需要与存储器打交道,存储器性能的好坏在很大程度上影响着计算机系统的性能。本章在介绍当今高档微机系统的存储器体系结构、存储器芯片的选用原则和接口特性的基础上,重点介绍内存的构成原理,并简要介绍高速缓冲存储器、外存储器和虚拟存储器的工作原理等。

4.1 当今高档微机系统的存储器体系结构

当今高档微机系统中，存储器技术的发展始终是以实现低成本、大容量和高速度为其追求目标，而用单一工艺制造的单级半导体存储器往往难以同时满足这三方面的要求。为解决这一矛盾，目前高档微机系统中普遍采用分级存储器结构和虚拟存储器技术来组织整个存储器系统。

4.1.1 分级存储器结构

分级存储器结构的思想是把几种不同容量、速度的存储器按层次结构合理地组织在一起，使之能较好地满足大容量、高速度和低成本的要求。如图 4.1 所示，这种分级结构是在存储器的组织上将全部存储器从内到外分为内部寄存器组、高速缓冲存储器 (Cache)、内存储器和外存储器四级。它们按从内到外的顺序在存储容量上依次递增，而在存取速度和位价格上依次递减。

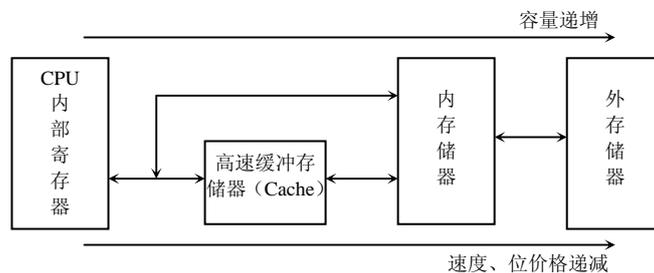


图 4.1 分级存储器结构示意图

1. 内部寄存器组

有些待使用的数据，或者运算的中间结果可以暂存在微处理器内部的寄存器中，这样，进行数据读写时，速度很快，一般在一个时钟周期即可完成。一个有众多通用寄存器的微处理器，只要充分利用并恰当安排这些寄存器，总可以在提高系统性能上获得好处，但受芯片面积和集成度的限制，寄存器的数量有限。

2. 高速缓冲存储器 (Cache)

高速缓冲存储器有时简称为“快存”，是为了解决 CPU 和内存之间速度匹配问题而设置的。它是介于 CPU 与内存之间的小容量高速存储器，容量只有几 KB~几百 KB，其存取速度足以与微处理器相匹配。这一级存储器一般只装载当前用得最多的程序或数据，使微处理器能以自己最高的速度工作。设置高速缓冲存储器是高档微型计算机中最常用的一种方法，其目的是把一个容量较大、而速度相对较慢的内存当作高速的存储器来使用。当今高档微处理器一般也将它们的全部或部分制作在 CPU 芯片中，有的还具有多级 Cache 结构。如 Pentium 内集成了 16KB 的一级 Cache (L1 Cache)，而把二级 Cache (L2 Cache) 放在主板上；Pentium II 以后的 CPU 则采用了全新的封装方式，把 CPU 内核与一级、二级缓存一起封装在芯片内。目前，已出现了带有三级缓存的 CPU。

3. 内存储器

内存储器用于存放运行的程序和数据。其速度比上两级存储器稍慢，但由于 CPU 大部分时间访问的是高速缓冲存储器 (Cache)，只有当程序或数据不在 Cache 中时，才需要访问内存并将相

关区域的程序或数据调入 Cache 中，这就降低了 CPU 对内存存取速度的要求，能以较低的成本实现大容量的内存，而对微机系统的性能并没有太大的影响。在当今高档微机系统中，内存一般都在几兆字节以上，甚至高达几百、上千兆字节，比过去的大中型机的内存还大。

4. 外存储器

外存是指磁带、软盘、硬盘和光盘等。外存容量很大，可达几十至几百吉字节（GB），但速度比内存慢得多。由于它的平均存储费用很低，所以大量用作后备存储器，存储各种数据和程序。在高档微机系统中，外存还广泛用作虚拟存储器的硬件支持。

上述四级存储器并非每个存储器系统都必备，应当根据系统的性能要求和微处理器的功能来设定。一般单片机系统和各种嵌入式系统中不设 Cache 和外存。

4.1.2 虚拟存储器技术

虚拟存储器是在分级存储器结构的基础上，通过综合应用硬件与软件技术，在内存与外存之间引入的一种假想存储器。它的引入，相当于把内存空间扩大到了外存那么大。这种存储器在物理上并不存在，但在逻辑上却确实可用，所以被称为虚拟存储器（Virtual Memory）。有了这种虚拟存储器技术，编程人员编写程序时就不必考虑计算机的实际内存容量，可以编写出比实际配置的物理存储器容量大很多的应用程序。编写好的程序预先存放在外存中，运行时由操作系统将部分程序调入内存储器，其余部分则仍在外存上，当要执行的这部分程序不在内存时，再由操作系统按一定的原则将内存中不常用的部分淘汰出内存，而将需要执行的部分从外存调入内存，这种调入和调出对用户来说是透明的。

在采用虚拟存储器的计算机系统中，存在着虚地址空间（或逻辑地址空间）和实地址空间（或物理地址空间）两个地址不同的空间。虚地址空间是程序可用的空间，而实地址空间是 CPU 可访问的内存空间。后者容量由 CPU 地址总线宽度决定，而前者由 CPU 内部软硬件结构决定。一般虚地址空间远远大于实地址空间，例如 80486 和 Pentium 微处理器的实地址空间为 $2^{32}=4\text{GB}$ ，而虚地址空间则可高达 $2^{46}=64\text{TB}$ ，两者相差极大。

综上所述，虚拟存储器结构把一个大容量的外存当作一个大容量的内存来使用；而分级存储器结构中，Cache 技术的引入则把一个容量较大而速度相对较慢的内存当作一个高速的内存来使用。综合两者，就使得当今高档微机系统的 CPU 可访问的存储器，相当于既具有外存的容量又具有高速缓存的速度，从而极大地提高了存储系统的性能，实质上也就等于提高了整个微机系统的性能。

4.2 半导体存储器的分类与选用原则

4.2.1 半导体存储器的分类

半导体存储器按制造工艺的不同，可分为双极型和 MOS 型两大类。双极型存储器由 TTL（Transistor-Transistor Logic）电路制成，其特点是存取速度快、集成度低、功耗大、位价格较高。MOS 型存储器由金属氧化物半导体电路制成，与双极型存储器比较，它的特点是集成度高、功耗小、价格便宜，但存取速度慢。

微型计算机中的内存储器和高速缓存器使用的一般都是 MOS 型存储芯片。从功能和应用角度，MOS 型半导体存储器主要分为只读存储器 ROM 和随机读写存储器 RAM 两类。

1. ROM 的类型

根据编程写入方式不同, ROM 可分为如下几种。

(1) 掩模 ROM

掩模 ROM 存储的信息由厂商按用户要求掩模制成, 封装后不能改写, 用户只能读出, 不能改写。

(2) PROM (Programmable ROM)

PROM 为一次可编程 ROM。其内容可由用户一次性编程写入, 写入后不能改写。

(3) EPROM (Erasable Programmable ROM)

EPROM 是一种紫外线可擦除 PROM。用户可多次改写内容, 改写方法一般可用紫外线擦除, 再编程写入, 有任一位错, 都需全片擦除、改写。紫外线照射约半小时, 所有存储位复原到“1”。

(4) EEPROM (Electrically Erasable Programmable ROM)

EEPROM 是一种电可擦除 PROM。可以字节为单位多次用电擦除和改写, 并可直接在机内进行, 无需专用设备, 故方便灵活。

(5) 闪速存储器 (Flash memory)

简称 Flash 或闪存。它与 EEPROM 类似, 也是一种电可擦除 PROM。但与 EEPROM 不同的是, 闪存不仅可按字节擦写, 还可按扇区或页面擦写, 速度更快。而更重要的是, 闪存内部还设置有命令、状态寄存器, 可在线编程, 具有数据保护、保密功能。

2. RAM 的类型

按存储电路结构不同, RAM 可分为如下几种。

(1) SRAM (Static RAM)

SRAM 是一种静态 RAM。存储单元电路以双稳为基础, 故状态稳定, 不掉电信息就不会丢失。

(2) DRAM (Dynamic RAM)

DRAM 是一种动态 RAM。存储单元电路以电容为基础, 故电路简单, 集成度高, 功耗小, 但不掉电也会因电容放电而丢失信息, 所以需定时刷新。

(3) IRAM

IRAM 称为组合 RAM。是一种附有片上刷新逻辑的 DRAM, 兼有 SRAM、DRAM 的优点。

(4) NVRAM (Non Volatile RAM)

NVRAM 是一种非易失性 RAM。由 SRAM 和 EEPROM 共同构成, 正常时为 SRAM, 掉电或电源故障时, 立即将 SRAM 中信息保存在 EEPROM 中, 使其不丢失。

4.2.2 半导体存储器的选用原则

存储器芯片的选用原则是由各种存储器芯片的不同特点所决定的, 通常有如下几个层次的选用。

1. RAM 和 ROM 的选用

RAM 是一种随机读写存储器 (Random Access Memory)。它的突出优点是读写方便, 使用灵活; 缺点是一旦停电所存信息就会丢失。一般用作各种二进制信息的临时或缓冲存储, 如存放当前正在执行的程序和数据、作为 I/O 数据缓冲存储器和用作堆栈等。此外, 在后备电源及掉电保护电路的支持下, 也可作为存放系统参数的存储器。

而 ROM 是一种只读存储器 (Read-Only Memory), 其特点是一旦写入, 在工作过程中就只能读出不能重写, 即使掉电内容也不会丢失。因此主要用于存放各种系统软件、应用程序和常数、

表格等。

2. RAM 类型的选用

SRAM 状态稳定, 接口简单, 速度快, 但集成度低, 成本高, 功耗也较大, 一般只用于高速缓存器和小容量内存系统。DRAM 集成度高, 功耗小, 价格低, 一般用它组成大容量的内存系统。IRAM 兼具 SRAM 和 DRAM 的优点, 是一种应用前景较广的产品。

3. ROM 类型的选用

掩模 ROM 和 PROM 只用于大批量生产的微机产品; 产品研制和小批量生产时, 宜选用 EPROM 和 EEPROM 芯片。闪存存储器 (Flash Memory) 兼具有 EEPROM 和 SRAM 的优点, 主要用来构成移动存储器 (如优盘) 和用作小型磁盘的替代品。目前, 闪存技术已大量用于便携式计算机、数码相机和 MP3、MP4 播放器等设备中。闪存也被用作内存, 用于内容不经常改变且对写入速度要求不高的场合, 如微机的 BIOS、IC 卡的数据记录单元等。

4. 芯片型号的选用

无论选用哪类具体芯片, 通常都应考虑存取速度、存储容量、结构和价格等因素。存取速度应取与 CPU 时序相匹配的芯片。否则, 如速度慢了, 要增加必要的时序匹配电路; 速度太快了, 又会造成不必要的浪费, 使成本增加。存储芯片的容量和结构直接关系到系统组成的形式和成本的高低。一般在满足存储系统总容量的限度内, 尽可能选用集成度高、存储容量大、字长等于或接近于存储器字长的芯片。这样使用芯片少, 总线负载轻, 也有利于简化接口电路设计, 提高系统可靠性。

4.3 存储器芯片与存储条的接口特性

存储器芯片的接口特性实质上就是指它有哪些与 CPU 总线相关的信号线, 以及这些信号线相互间的定时关系。了解存储器芯片的接口特性就是要弄清楚这些信号线与 CPU 三大总线的连接关系。

4.3.1 各类存储器芯片的接口共性

如图 4.2 所示, 除电源和地线外, 各种存储器芯片都有 4 类外部引脚: 地址线、数据线、片选线和读/写控制线。不同类型和型号的芯片, 这些引脚信号的含义和功能基本相同:

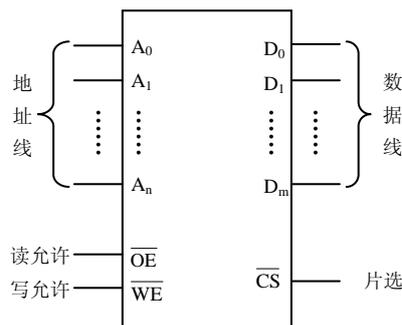


图 4.2 存储器芯片的通用引脚

- 地址线 $A_0 \sim A_n$ 用于选择存储器芯片中的存储单元, 差别在于不同容量和型号的芯片, 其地址线的数量可能不同。地址线的条数决定存储器芯片中存储单元的个数, 如有 10 根地

址线 ($A_0 \sim A_9$) 的存储器芯片通常有 1K 个存储单元; 有 20 根地址线 ($A_0 \sim A_{19}$) 的存储器芯片通常有 1M 个存储单元。

- 数据线 $D_0 \sim D_m$ 用于向存储器芯片写入或从存储器芯片读出数据。不同型号的芯片, 数据线的位数可能不同, 它决定于存储器芯片的字长。存储器芯片的字长通常有 1 位、4 位和 8 位等。
- 片选线 \overline{CS} (或芯片允许线 \overline{CE}) 用于选择芯片。各种存储器芯片都至少有一个片选线 (\overline{CS}) 或芯片允许线 (\overline{CE}), 只有在所有片选信号有效, 芯片被选中时, CPU 才可以对存储单元进行读/写操作。
- 读/写控制线 (\overline{OE} 、 \overline{WE}) 用于控制存储器芯片中数据的读出或写入, 差别在于不同种类存储器芯片的读/写控制线设置有所区别。掩模 ROM、PROM 和 EPROM 只有一根输出允许线 \overline{OE} ; EEPROM 和 Flash Memory 有输出允许线 (\overline{OE}) 和写允许线 (\overline{WE}); SRAM 的读/写控制线的设置方法通常有两类, 一类既有输出允许线 (\overline{OE}), 又有写允许线 (\overline{WE}), 另一类只有 1 根读/写控制线 (\overline{WE}), 利用 \overline{WE} 的两种状态 0 和 1 区分写和读。

图 4.3 至图 4.6 给出了部分常用 EPROM、SRAM, 以及典型 EEPROM 和典型 Flash Memory 芯片的外部引脚排列图。

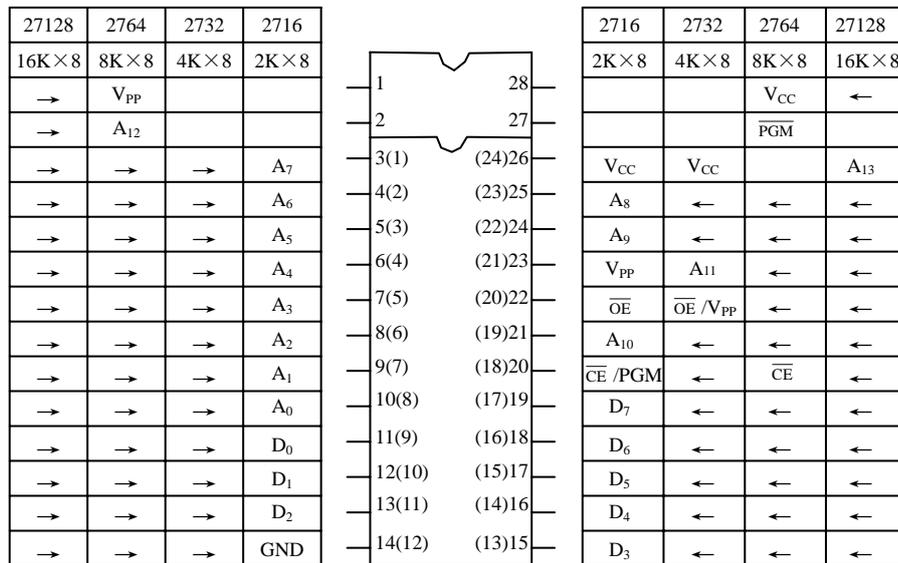


图 4.3 部分 EPROM 芯片的引脚排列

从图 4.3 至图 4.6 中可以看出, 就地址线、数据线、片选线和读/写控制线而言, 上述各类存储芯片确实大同小异。这些引脚信号线与 CPU 三大总线的接口方法也基本相同, 这就是存储器芯片的接口共性。它们与 CPU 的连接方法一般为:

- ① 地址线 $A_0 \sim A_n$ 通常与 CPU 的低位地址线 $A_0 \sim A_n$ 直接相连。
- ② 数据线 $D_0 \sim D_m$ 与 CPU 的某 (m+1) 位数据线直接相连。
- ③ 芯片允许线 \overline{CE} (或片选线 \overline{CS}) 与 A_{n+1} 位以上的 CPU 高位地址线经译码后产生的片选信号相连。

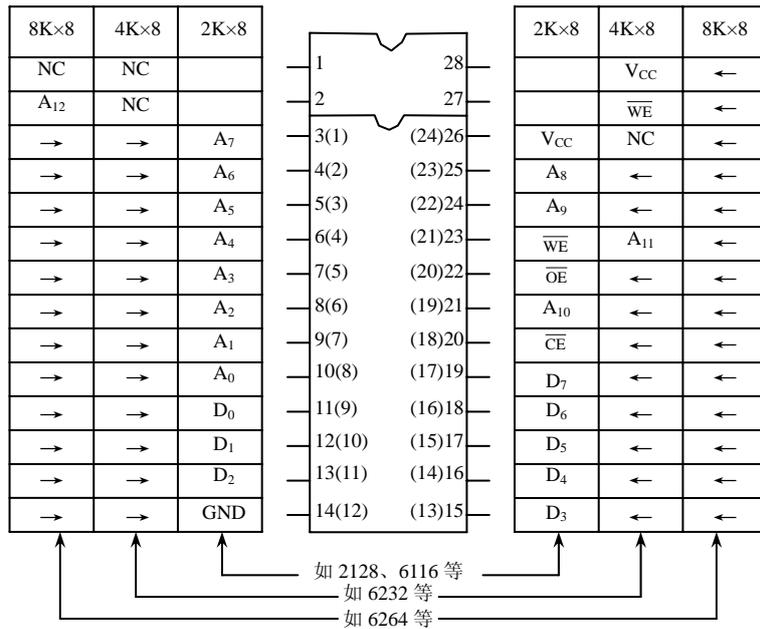


图 4.4 部分 SRAM 芯片的引脚排列

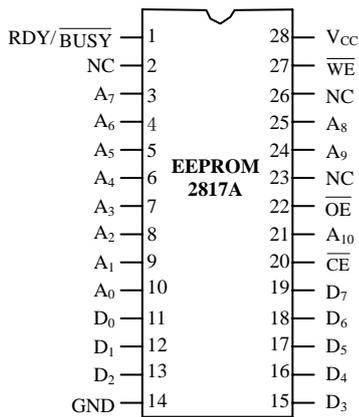


图 4.5 典型 EEPROM 芯片引脚信号

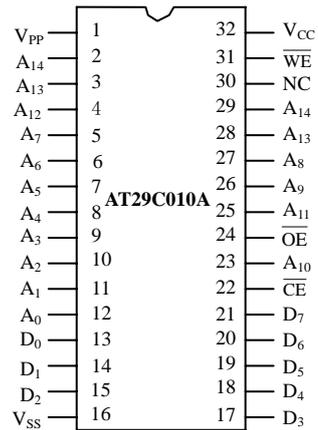


图 4.6 典型 Flash 芯片引脚信号

④ 读/写控制线在存储器芯片的存取速度与 CPU 匹配时，与 CPU 控制总线组合形成的读/写控制信号（如 \overline{MEMR} 和 \overline{MEMW} ）直连；对存取速度不匹配的存储器芯片，即不能在 CPU 的读写周期内完成数据读/写操作时，就需要引入时序匹配逻辑（等待信号产生电路），才能相连。

当然，具体的接口方法也不尽然，可以很灵活地组合。图 4.7 给出了这几种芯片的接口方法示例。分别设有读和写控制线的 SRAM 芯片与 EEPROM 和 Flash Memory 芯片的接口几乎完全相同，只是对 EEPROM 芯片编程时，通常要用 RDY/ \overline{BUSY} 状态产生中断请求信号或作为查询的状态信号。

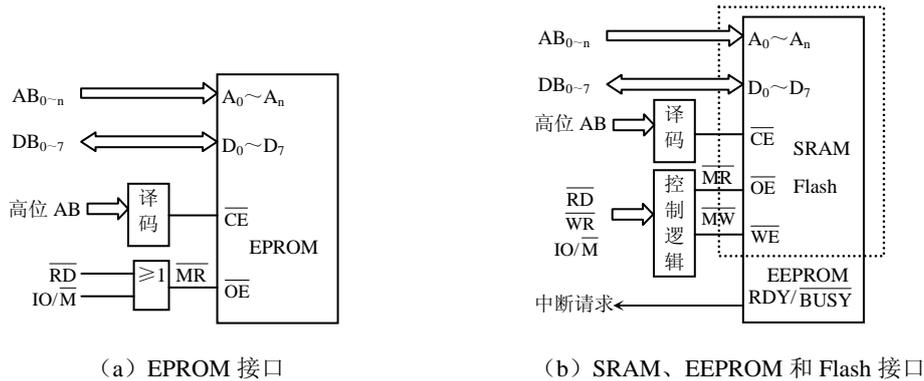


图 4.7 EPROM、SRAM、EEPROM 和 Flash 接口方法示例

4.3.2 DRAM 芯片与存储条的接口特性

1. DRAM 芯片的接口特殊性

DRAM 在原理上是靠电荷存储器件（或电容）存储信息，由于电容存在漏电现象，电荷会随着电容放电而泄漏，从而导致信息丢失，所以需定时刷新来补充电荷。此外，由于 DRAM 芯片集成度高，存储容量大，为节省外部引脚，其地址输入一般采用两路复用锁存方式，即把地址信号分为两组共用几根地址输入线，分两次把它们送入芯片内部锁存起来。因此，DRAM 在构造上和外部引脚的设置上与 SRAM 差别也较大，故与 CPU 接口时表现出更多的特殊性。

为说明 DRAM 芯片的接口特性，图 4.8 和图 4.9 分别给出了 64K×1 位的 DRAM 芯片 2164A 的外部引脚和内部结构。从图 4.8、图 4.9 可以看出 DRAM 芯片在结构和外部引脚设置上的一些特点：

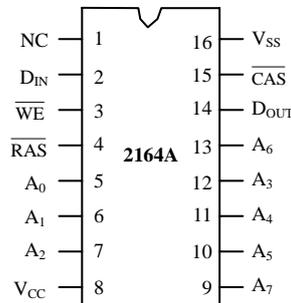


图 4.8 2164A 外部引脚

- 从内部结构看，DRAM 存储体一般由 4 个 N 行×N 列矩阵组成，如 2164A 由 4 个 128×128 的存储矩阵组成。每个存储矩阵由 N 条行地址线和 N 条列地址线进行选择。N 条行地址线经过行译码器选择 2^N 行，N 条列地址线经过列译码器选择 2^N 列。
- DRAM 芯片没有专门的片选线（ \overline{CE} ），而是设置了一条行选通线 \overline{RAS} 和一条列选通线 \overline{CAS} 。使用中可用行选通线 \overline{RAS} 和列选通线 \overline{CAS} 兼作片选信号。
- DRAM 芯片的数据线不是输入、输出公用的双向线，而是两根分设的输入、输出线 \overline{Din} 和 \overline{Dout} 。由 1 根读写控制信号线 \overline{WE} 控制读出和写入。当 \overline{WE} 为高电平时，读出；当 \overline{WE} 为低电平时，写入。

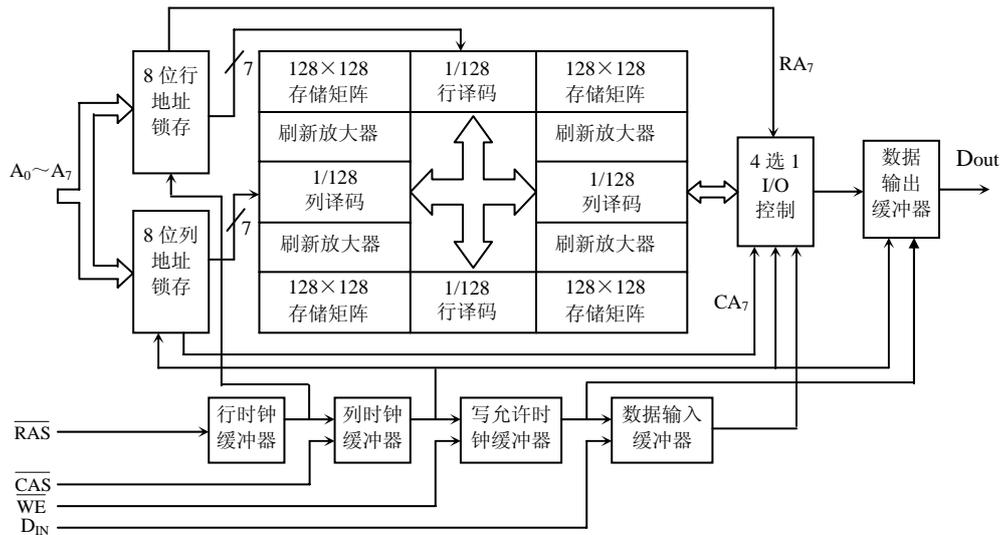


图 4.9 2164A 内部结构

- DRAM 为节省外部引线，其地址线采用二路复用工作方式。如要实现 64K 个存储单元的 DRAM 寻址，需要 16 条地址线，而 2164A 只有 $A_0 \sim A_7$ 这 8 条地址线。所以，来自 CPU 的地址信号要分为行地址和列地址分别在行选通信号 \overline{RAS} 和列选通信号 \overline{CAS} 的控制下送入芯片内部锁存，再经行地址和列地址译码，选中一个单元进行读写。与此同时，4 个矩阵也在行地址控制下刷新。芯片如果只加行选通信号 \overline{RAS} ，不加列选通信号 \overline{CAS} ，可以把行地址加到行译码器，使指定的 4 行存储单元只被刷新，而不读数写数，这时数据输出端为高阻态。

由此可见，DRAM 在与 CPU 接口时有两个特殊问题需要考虑：

- 一是刷新问题。由于 DRAM 芯片的刷新周期一般不能大于 2ms，利用芯片正常的读写实现刷新显然是不可靠的，需要加专门的定时刷新电路。这个电路能够在刷新时提供行选通信号，并且提供连续的行地址，保证在 2ms 以内将全部行地址循环一次。实现 DRAM 正常读/写和定时刷新的控制方法有多种，可以由 CPU 通过一定控制逻辑实现，也可以用 DMA 控制器实现，还可以用专用 DRAM 控制器实现。
- 二是地址信号输入问题。来自 CPU 的地址线需通过地址多路复用器转换成行地址和列地址，由行地址选通信号 \overline{RAS} 和列地址选通信号 \overline{CAS} 控制，分两次送给 DRAM。

图 4.10 所示为 PC/XT 微机的 DRAM 简化电路图，图中由 8 片 2164DRAM 组成 64KB 的存储器，LS158 是二选一数据选择器，LS245 为总线驱动器。当 CPU 读写存储器的某个单元时，首先由行列锁存信号电路送出行地址锁存信号 \overline{RAS} ，同时 $\text{ADDSEL}=0$ ，使 LS158 的 A 端口导通，CPU 将 8 位行地址信号 $A_0 \sim A_7$ 通过 LS158 的 A 口加到存储器芯片上，并由 \overline{RAS} 控制锁存于 2164 芯片内部的行地址锁存器。60ns 后， $\text{ADDSEL}=1$ ，使 LS158 的 B 端口导通，CPU 将 8 位列地址信号 $A_8 \sim A_{15}$ 通过 LS158 的 B 口加到存储器芯片上，延时 40ns 后，由 \overline{CAS} 控制锁存于 2164 芯片内部的列地址锁存器。最后在存储器读/写信号 $\overline{\text{MEMR}}/\overline{\text{MEMW}}$ 控制下，实现数据的读写。

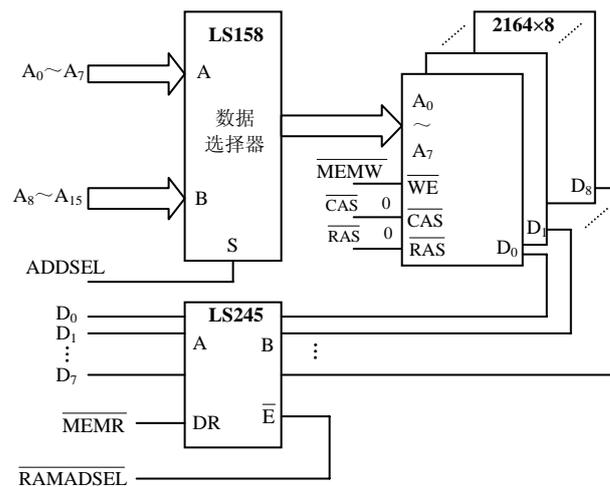


图 4.10 DRAM 读写简化电路示意图

图中省去了 DRAM 刷新控制电路。在 PC/XT 机中，DRAM 的刷新控制是利用 DMA 控制器 8237 的通道 0 来实现的。刷新过程是利用定时/计数器 8253 产生 15.12 μ s 的周期性定时信号向 8237 的 DMA 通道 0 发出 DMA 请求，当 8237 通道 0 响应该 DMA 请求时，在其 $\overline{\text{DACK0}}$ 端产生一个低电平有效的应答信号，使列地址信号 $\overline{\text{CAS}}$ 为高电平，而行地址信号 $\overline{\text{RAS}}$ 为低电平。最后由 DMA 控制器送出刷新的行地址，实现一次刷新。

为了进一步简化 DRAM 芯片的接口设计，方便用户，20 世纪 90 年代以来，新推出并广泛使用了两种新型先进的 DRAM 芯片：一种是将动态刷新逻辑和地址多路复用逻辑集成于 DRAM 芯片内的 IRAM（组合 RAM）芯片；另一种是在一般 DRAM 芯片内集成了动态刷新逻辑和一个小容量 SRAM Cache 的 EDRAM（增强型 DRAM）芯片。

2. DRAM 存储条的接口特性

目前，微机系统中使用的内存都是将多片 DRAM 芯片塑封在一个长条型印刷电路板上的 DRAM 内存条，以便于减小体积、扩充容量和更换模块。内存条有 SIMM（Single In-Line Memory Module，单列直插存储模块）、DIMM（Dual In-Line Memory Module，双列直插存储模块）和 RIMM（Rambus In-Line Memory Module）三种结构。

SIMM 的常用规格有 1MB、4MB、8MB、16MB、32MB 和 64MB 等。按外部引线主要有 30 线和 72 线两种，30 线 SIMM 的数据线有 8 位和 9 位（带 1 位奇偶校验位）两种，需要用四条 SIMM 组成一组来构成 32 位数据宽度的主存储器；72 线的 SIMM 可提供 32 位的有效数据位和 4 位奇偶校验位，用一条就可构成 32 位数据宽度的主存储器。72 线的 SIMM 是 486 微机和早期 Pentium 微机的主流配置，所用存储器基本上都是 EDO DRAM（Extended Data Output DRAM，扩展数据输出 DRAM），访问时间为 60~70ns。

DIMM 的常用规格有 32MB、64MB、128MB、256MB 和 512MB 等，外部引线为 168 线。这种内存条可提供 64 位的有效数据位和 8 位奇偶校验位，所以只需一个 DIMM 就可构成具有某种容量和 64 位数据宽度的主存储器。它所使用的存储器是 SDRAM（Synchronous DRAM，同步动态随机存储器），访问时间小于 10ns，是 Pentium II/Pentium III 微机中流行的标准内存配置。但随着处理器前端总线的不断提高，SDRAM 已经无法满足新型处理器的需要了，目前 DIMM 结构的

SDRAM 也已退出了主流市场。

最新式的 RIMM 接插结构的内存条外部引线为 184 线，也可提供 64 位的有效数据位和 8 位奇偶校验位，是目前 Pentium 4 微机的主流内存配置。它使用的存储器有两种：一种是 DDR SDRAM（Double Data Rate SDRAM，双倍速率同步 DRAM）或 DDR2 SDRAM（4 倍速率同步 DRAM），其实际数据传输速率是普通 SDRAM 的 2 倍或 4 倍；另一种是 RDRAM（Rambus DRAM），最高传输速率可达 3.2GB/s，多用于 Pentium 4 服务器和 workstation 中。

图 4.11 给出了 DIMM 结构的 SDRAM 和 RIMM 结构的 DDR SDRAM 内存条的实物样例。从外形上看 DDR 内存条与 SDRAM 相比差别并不大，它们具有同样的长度与同样的引脚距离。只不过 DDR 内存条有 184 个引脚，金手指中也只有一个缺口，而 SDRAM 内存条是 168 个引脚，并且有两个缺口。



(a) 168 线 256MB SDRAM 内存条



(b) 184 线 256MB DDR SDRAM 内存条

图 4.11 SDRAM 内存条和 DDR SDRAM 内存条实物样例

在选购和安装内存条时，除要关心内存条的访问速率外，还要特别注意内存条数据宽度与 CPU 数据总线宽度的关系。例如：用 72 线的 SIMM 构成 64 位的存储器，则要两条为一组；而用 168 线的 DIMM 或 184 线的 RIMM，一条就可构成 64 位的存储器。但要注意，同一台微机中，一般要么装 72 线 SIMM 内存条，要么装 168 线或 184 线 DIMM 内存条，不能混装。

4.4 内存储器

4.4.1 内存储器组织原理

内存储器的构成即是用存储器芯片构成存储器系统。主要任务包括存储器结构的确定、存储器芯片的选配和存储器接口的设计。

1. 存储器结构的确定

存储器结构的确定，主要指采用单存储体结构还是多存储体结构。微机系统中，存储器一般都按字节编址、以字节（8 位）为单位构成。对于 CPU 的外部数据总线为 8 位的微机系统（如 8088

系统), 其存储器只需用单体结构; 对于 CPU 的外部数据总线为 16 位的微机系统 (如 8086/80286 系统等), 为了支持 8 位字节操作和 16 位字操作, 一般需采用双体结构。如图 4.12 所示, 给出了 80286 的存储器结构。

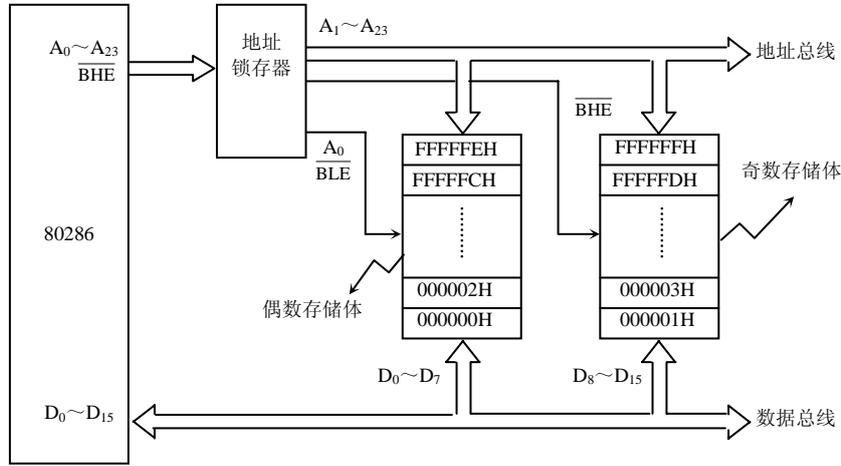


图 4.12 80286 存储器结构

图中, 80286 的 16MB 存储器被分成两个容量为 8MB 的存储体, 一个由偶数地址单元组成, 称为偶数存储体; 另一个由奇数地址单元组成, 称为奇数存储体。两个存储体的地址线连法相同, 均与 CPU 的地址总线 $A_1 \sim A_{23}$ 相连, 用于选择体内存储单元; 而数据线则分别与数据总线的 $D_0 \sim D_7$ 和 $D_8 \sim D_{15}$ 相连。高位允许信号 \overline{BHE} 和低位允许信号 \overline{BLE} (A_0) 分别用作奇数存储体和偶数存储体的选通信号, 这两个信号结合用于选择 8 位字节和 16 位字操作, 选择功能如表 4.1 所示。

表 4.1 \overline{BHE} 和 \overline{BLE} (A_0) 对 8 位和 16 位操作的选择控制表

\overline{BHE}	\overline{BLE} (A_0)	功能
0	0	允许两个存储体进行 16 位数据传送
0	1	允许奇数存储体进行 8 位数据传送
1	0	允许偶数存储体进行 8 位数据传送
1	1	两个存储体都未选中

当 \overline{BLE} 和 \overline{BHE} 中只有一个为低电平有效时, 由 $A_1 \sim A_{23}$ 选中偶数存储体或奇数存储体中的一个字节单元进行 8 位字节传送操作; 若 \overline{BLE} 和 \overline{BHE} 同时为低电平有效时, 将选择偶数存储体中的一个字节单元与奇数存储体中的一个字节单元组成 16 位的字, 进行字传送。这时又分两种情况, 若字地址是偶数, 即字对准时, 组成字的两个字节单元的地址除最低位不同外, 对应 $A_1 \sim A_{23}$ 的地址编码均相同, 即可由 CPU 地址总线 $A_1 \sim A_{23}$ 发出的地址编码同时选中这两个字节单元, 在一个总线周期内完成字传送; 若字地址不是偶数, 即字未对准时, 组成字的两个字节单元 (如 00001H 和 00002H) 的地址除最低位不同外, 对应 A_1 的地址编码也不相同, 即不可能由 CPU 地址总线 $A_1 \sim A_{23}$ 发出的地址编码同时选中这两个字节单元, 这时, CPU 就必须分两次发出不同的地址编

码分别对这两个字节单元进行读/写，并在 CPU 内部进行高低字节交换才能完成 16 位的字操作，即需两个总线周期才能完成字传送。

对于 CPU 的外部数据总线为 32 位的微机系统（如 80386/80486 系统），一般要使用 4 体结构，以支持 8 位字节、16 位字和 32 位双字操作；而对于 CPU 的外部数据总线为 64 位的微机系统（如 Pentium 系列机），一般要使用 8 体结构，以支持 8 位字节、16 位字、32 位双字和 64 位四字操作。图 4.13 给出了 Pentium 系统的存储器结构，它将整个存储器分成 8 个存储体，分别使用 $\overline{BE}_0 \sim \overline{BE}_7$ 作为体选控制信号，以构成 64 位数据。当 $\overline{BE}_0 \sim \overline{BE}_7$ 同时有效且四字对准时，在一个总线周期里就可以完成 64 位数据的存储器读写操作。

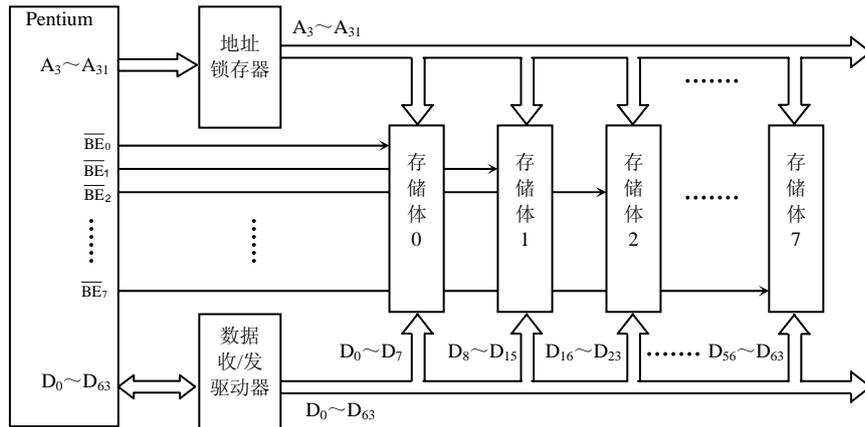


图 4.13 Pentium 存储器结构

无论是 80286 的双体结构还是 Pentium 的多体结构，不同存储体除数据线和体选控制线的连接有所不同外，地址总线的连接基本相同。所以，存储器的设计可归结为（8 位）单体存储器的设计。

2. 存储器芯片的选配

存储器芯片的选配包括芯片的选择和组配。在按 4.2.2 节的选用原则选定芯片后，需要根据芯片的结构和所需构成的存储器的容量，进行芯片的组配。而所谓存储器芯片的组配，实际上就是存储器位、字的扩展。

（1）存储器位扩展

位扩展是指增加存储芯片的数据位数。实际存储芯片的数据位数（字长）有 1 位、4 位和 8 位的，当用字长不足 8 位的存储芯片构成内存时，就需要进行位扩展，以构成具有 8 位字长的存储体。

例如，要用 $1K \times 1$ 位的存储芯片构成 1KB 存储器，就需要 8 个芯片连在一起，如图 4.14 (a) 所示。图中各存储芯片的对应地址线、读写控制线 (\overline{WE}) 和片选信号 (\overline{CS}) 分别并连在一起，而数据线则分别连接到数据总线的不同位线上。该芯片组可等效为图 4.14 (b) 所示的 $1K \times 8$ 位芯片。当 CPU 访问该 1KB 存储器时，其发出的地址和控制信号同时传给 8 个芯片，选中各芯片中具有相同地址的单元，8 个芯片的各数据位就组成同一个字节的 8 位，其内容被同时读至数据总线的相应位或数据总线上的内容被同时写入相应单元，完成对一个字节的读/写操作。

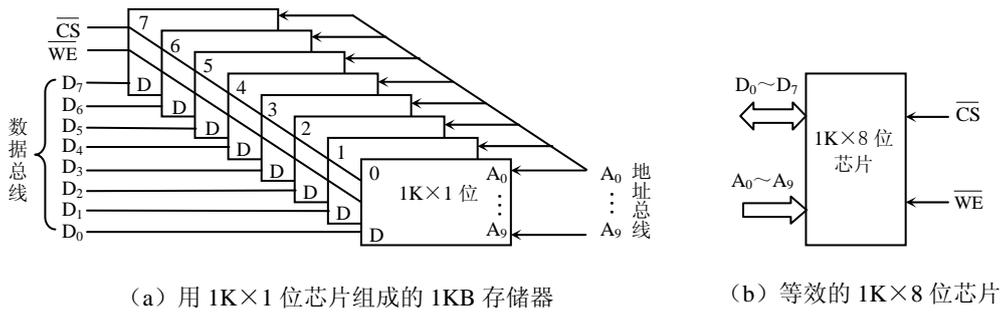


图 4.14 位扩展示例

(2) 存储器字扩展

字扩展是指增加存储器的字节数量。当用一片字长为 8 位的存储芯片或经位扩展后的一个 8 位芯片组不能满足存储器容量的要求时，就要进行字扩展，以满足字数（地址单元数）的要求。

例如，用 1K×8 位芯片（或芯片组）实现 4KB 存储器，需要 4 个芯片（或 4 个芯片组）进行字扩展，如图 4.15 所示。

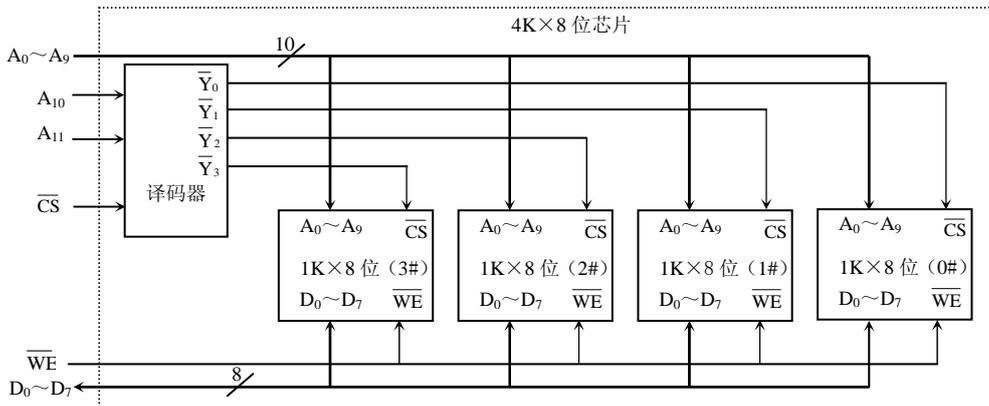


图 4.15 字扩展示例

各芯片或芯片组的地址线（ $A_0 \sim A_9$ ）、数据线（ $D_0 \sim D_7$ ）和读/写控制线（ \overline{WE} ）按信号名称分别对应并连，而片选线则分连到片选地址译码器的不同输出端。系统的高位地址线 A_{10} 和 A_{11} 作为译码器的输入。当 $A_{11}A_{10}=00$ 时，选中芯片 0#；当 $A_{11}A_{10}=01$ 时，选中芯片 1#；当 $A_{11}A_{10}=10$ 时，选中芯片 2#；当 $A_{11}A_{10}=11$ 时，选中芯片 3#。每个芯片有不同的片地址，即扩展了存储单元数。该存储器也可等效为一个 4K×8 位存储器芯片。

当选用的存储芯片的字长和容量均不满足存储器字长和容量的要求时，就需要同时进行位扩展和字扩展。这实际上是先对存储芯片进行位扩展以满足存储器字长的要求，然后对位扩展后的芯片组进行字扩展以满足存储单元数的要求。有关字位扩展的实例见 4.4.2 节中的例 4.2。

3. 存储器接口的设计

存储器接口的设计实际上就是要解决存储器同 CPU 三大总线的正确连接与时序匹配问题。其中数据线的连接比较简单，是与 CPU 的相应数据总线直连；读/写控制线在时序匹配时，也是与

CPU 相应控制线经译码产生的读/写控制信号（如 $\overline{\text{MEMR}}/\overline{\text{MEMW}}$ ）直连；而与地址总线的连接，本质上就是在存储器地址分配的基础上实现地址译码，以保证 CPU 能对存储器中的所有单元正确寻址。它又包括两个方面：芯片选择和片内单元选择。

通常，芯片内部的存储单元由 CPU 输出的低位地址线完成选择，而芯片选择信号则是通过 CPU 的高位地址线译码得到。地址总线高位、低位的划分，由芯片的地址单元数（字数）决定，如 $8\text{K} \times n$ 位芯片对应的低位地址线，一般为 $A_0 \sim A_{12}$ 共 13 位， $1\text{M} \times n$ 位芯片对应的低位地址线，一般为 $A_0 \sim A_{19}$ 共 20 位等，其余部分均为高位地址线。

根据对高位地址总线的译码方案不同，通常有线选法、局部译码法和全译码法三种片选控制方法。

（1）线选法

线选法的原理如图 4.16 所示，它是将余下的高位地址线分别作为各个存储器芯片的片选信号。这种方法译码简单，无需译码器，但用于片选的地址线（ $A_{13} \sim A_{10}$ ）在每次寻址时只能有一位有效，不允许同时有多位有效。主要缺点是存储空间利用率低，由于 A_{15} 和 A_{14} 未参与高端译码，每个芯片实际上都占据了 4 个 1KB 的地址空间。此外，各芯片使用的地址空间不连续。

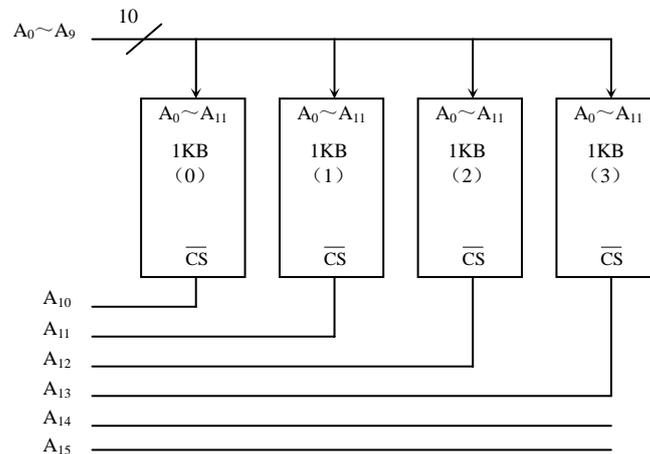


图 4.16 线选法

（2）局部译码法

局部译码法的原理如图 4.17 所示，它是对余下高位地址总线中的一部分进行译码，译码输出作为各存储器芯片的片选控制信号。这种方法由于部分高端地址线未参与译码，也存在地址区域重复使用，致使存储空间利用率不高的问题。一般在线选法不够用，而又不需要全部地址空间时，使用这种方法。

（3）全译码法

全译码法的原理如图 4.18 所示，它是对余下高位地址线全部译码，译码输出作为各存储器芯片的片选控制信号。与前两种译码方法相比，全译码法存储空间利用率最高且译出的地址连续，不存在地址重复使用问题，但译码电路最复杂。

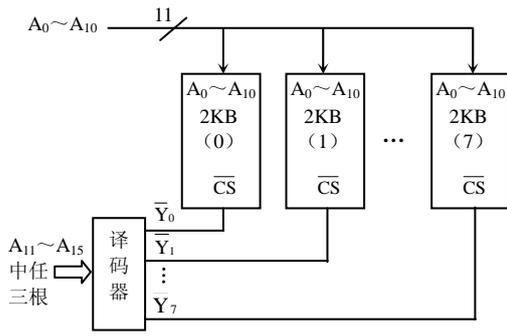


图 4.17 局部译码法

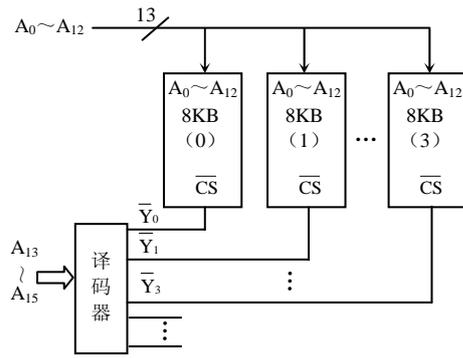


图 4.18 全译码法

无论是局部译码还是全译码，译码方案既可采用门电路或译码器芯片实现，也可采用 PROM 等可编程逻辑器件实现。

4.4.2 内存储器设计举例

例 4.1 试用 8K×8 位的 EPROM (2764) 和 8K×8 位的 SRAM (6264) 及 74LS138 译码器为某 8 位微机系统 (地址总线宽度为 20 位) 构成一个 16KB ROM 和 16KB RAM 的存储器系统, 要求 RAM 的起始地址为 00000H, ROM 的起始地址为 80000H。

本例 EPROM 和 SRAM 芯片均为 8K×8 位的存储器芯片, 无须进行位扩展。要构成 16KB 的 ROM 和 16KB 的 RAM 分别需要 2 片 2764 和 2 片 6264。地址总线的低 13 位 ($A_0 \sim A_{12}$) 要用作片内地址线选择, 余下的 7 根高位地址线 $A_{13} \sim A_{19}$ 经过地址译码来产生 4 个芯片的片选信号。为确定译码方案, 先列出各芯片的地址范围和存储器地址位分配, 如表 4.2 所示。

表 4.2 例 4.1 存储器地址位分配和芯片地址范围

芯片	地址位分配		地址范围
	$A_{19}A_{18}A_{17}A_{16}A_{15}A_{14}A_{13}$	$A_{12} \sim A_0$	
6264-1	0 0 0 0 0 0 0	0000~1FFFH	00000~01FFFH
6264-2	0 0 0 0 0 0 1	0000~1FFFH	02000~03FFFH
2764-1	1 0 0 0 0 0 0	0000~1FFFH	80000~81FFFH
2764-2	1 0 0 0 0 0 1	0000~1FFFH	82000~83FFFH

由表 4.2 可以看出, 虽然 2 片 6264 和 2 片 2764 的地址编码较分散, 但 4 个芯片的高位地址除 A_{19} 和 A_{13} 不同外, $A_{18} \sim A_{14}$ 均相同。存储器译码方案当然可以选用 2-4 译码器或门电路对 A_{19} 和 A_{13} 进行译码来产生 4 个芯片的片选信号。但此例要求用 3-8 译码器 74LS138 译码, 这时可选择 A_{19} 、 A_{14} 和 A_{13} 作译码输入, $A_{18} \sim A_{15}$ 作译码器的使能控制信号, 译码器的输出 \bar{Y}_0 、 \bar{Y}_1 、 \bar{Y}_4 和 \bar{Y}_5 分别用作 2 片 6264 和 2 片 2764 的片选信号。存储器扩展电路如图 4.19 所示。

例 4.2 试用 4K×4 位的 SRAM 芯片为某 8088 微机系统构成一个 16KB 的 RAM 存储器, RAM 的起始地址为 0BC000H。

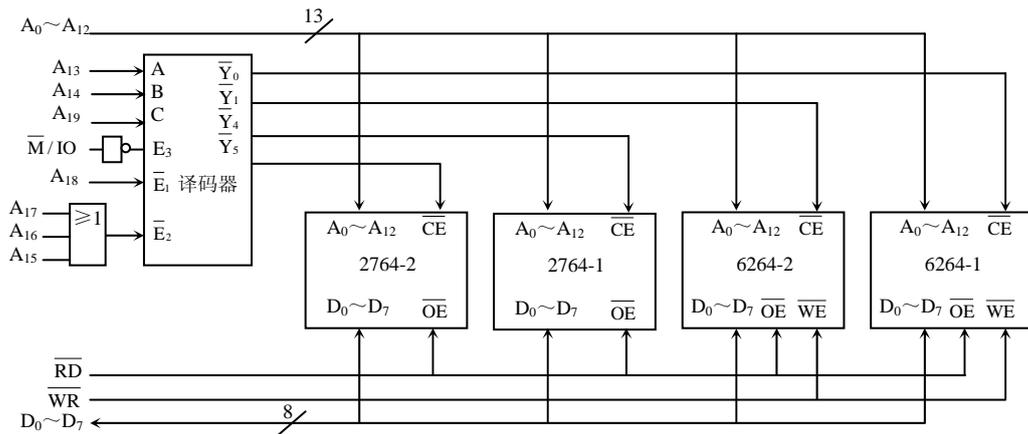


图 4.19 用 6264 和 2764 构成的存储器

该例 SRAM 芯片字长不足 8 位，需用 2 个芯片为一组进行位扩展，位扩展后每组存储容量为 4K×8 位。要构成 16KB 的 RAM 存储器还需用 4 组芯片进行字扩展，CPU 的低位地址线 A₀~A₁₁ 用于组内存储单元选择，余下的高位地址线 A₁₂~A₁₉ 经过地址译码来产生 4 个芯片组的片选信号。同样，为确定译码方案，先列出各芯片组的地址范围和存储器地址位分配，如表 4.3 所示。

表 4.3 例 4.2 存储器地址位分配和芯片组地址范围

芯片组号	地址位分配		地址范围
	A ₁₉ A ₁₈ A ₁₇ A ₁₆ A ₁₅ A ₁₄ A ₁₃ A ₁₂	A ₁₁ ~A ₀	
0	1 0 1 1 1 1 0 0	000~FFFH	0BC000~0BCFFFH
1	1 0 1 1 1 1 0 1	000~FFFH	0BD000~0BDFFFH
2	1 0 1 1 1 1 1 0	000~FFFH	0BE000~0BEFFFH
3	1 0 1 1 1 1 1 1	000~FFFH	0BF000~0BFFFFH

据此可确定译码方案：用 3-8 译码器对 A₁₄、A₁₃ 和 A₁₂ 地址信号进行译码来产生 4 个芯片组的片选信号，A₁₉~A₁₅ 用作 3-8 译码器的使能信号。相应的字位扩展设计如图 4.20 所示，每组芯片内部地址线、片选线和读/写控制线并连，数据线分连；各组间地址线、数据线和读/写控制线对应并连，而片选线分别与译码输出相连。

例 4.3 试用 16K×8 位的 SRAM 芯片为某 8086 微机系统设计一个 256KB 的 RAM 存储器系统，RAM 的起始地址为 00000H。

8086 为 16 位数据总线的微处理器，要支持 8 位和 16 位的数据传送操作，其存储器需采用双体结构，即要将 256KB 的存储器分为两个容量为 128KB 的偶数存储体和奇数存储体，每个存储体均需 8 个 16K×8 位的芯片组成。这时，两个存储体中各存储芯片的地址位分配如表 4.4 所示。

两个存储体中，对应芯片的地址位分配除 A₀ 不同外，其他位均相同，所以两个存储体的片选地址译码既可采用独立的地址译码，又可采用统一的地址译码。采用独立的地址译码时，各存储体使用相同的读/写控制信号，而用字节选择信号（ $\overline{\text{BLE}}$ 和 $\overline{\text{BHE}}$ ）作译码器的使能控制信号；采

用统一的地址译码时, 则用字节选择信号 (\overline{BLE} 和 \overline{BHE}) 与 CPU 的读/写信号组合产生各存储体的读/写信号。本例采用独立的地址译码方法, 用 3-8 译码器 74LS138 对 A_{17} 、 A_{16} 和 A_{15} 进行译码来产生 8 个芯片的片选信号, \overline{BLE} 和 \overline{BHE} 分别作为偶数存储体和奇数存储体译码器的使能控制信号, 存储器扩展电路如图 4.21 所示。

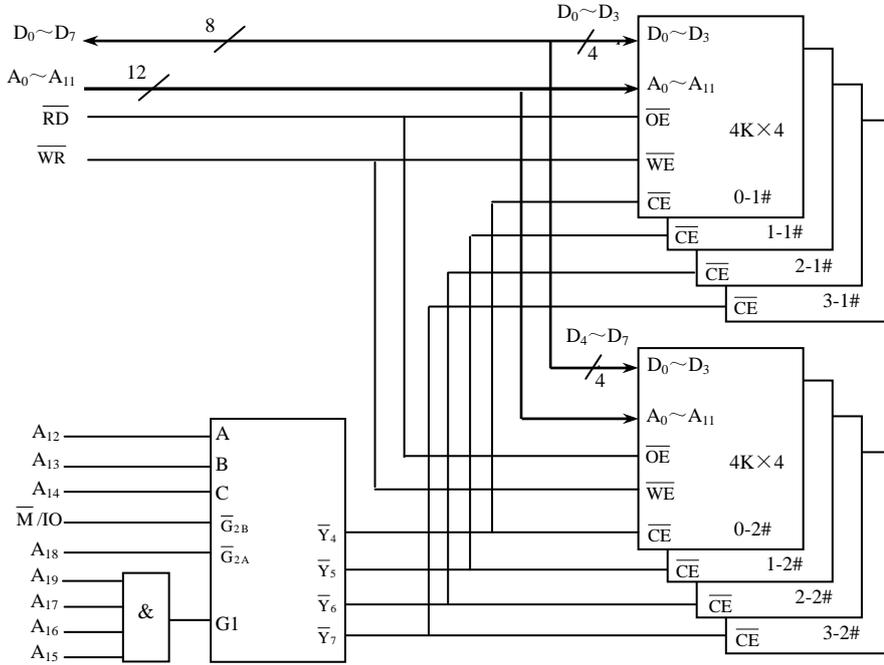


图 4.20 例 4.2 存储器连接图

表 4.4 双体结构中存储器地址位分配

偶数存储体				奇数存储体			
芯片	$A_{19}A_{18}A_{17}A_{16}A_{15}$	$A_{14} \sim A_1$	A_0	芯片	$A_{19}A_{18}A_{17}A_{16}A_{15}$	$A_{14} \sim A_1$	A_0
0	0 0 0 0 0	0000~FFFFH	0	0	0 0 0 0 0	0000~FFFFH	1
1	0 0 0 0 1	0000~FFFFH	0	1	0 0 0 0 1	0000~FFFFH	1
2	0 0 0 1 0	0000~FFFFH	0	2	0 0 0 1 0	0000~FFFFH	1
3	0 0 0 1 1	0000~FFFFH	0	3	0 0 0 1 1	0000~FFFFH	1
4	0 0 1 0 0	0000~FFFFH	0	4	0 0 1 0 0	0000~FFFFH	1
5	0 0 1 0 1	0000~FFFFH	0	5	0 0 1 0 1	0000~FFFFH	1
6	0 0 1 1 0	0000~FFFFH	0	6	0 0 1 1 0	0000~FFFFH	1
7	0 0 1 1 1	0000~FFFFH	0	7	0 0 1 1 1	0000~FFFFH	1

若此例改用统一地址译码, 则可省去一个译码器, 方法是将两个存储体中对应芯片共用一个译码输出信号, 而用图 4.22 所示电路来产生两个存储体的读/写控制信号。

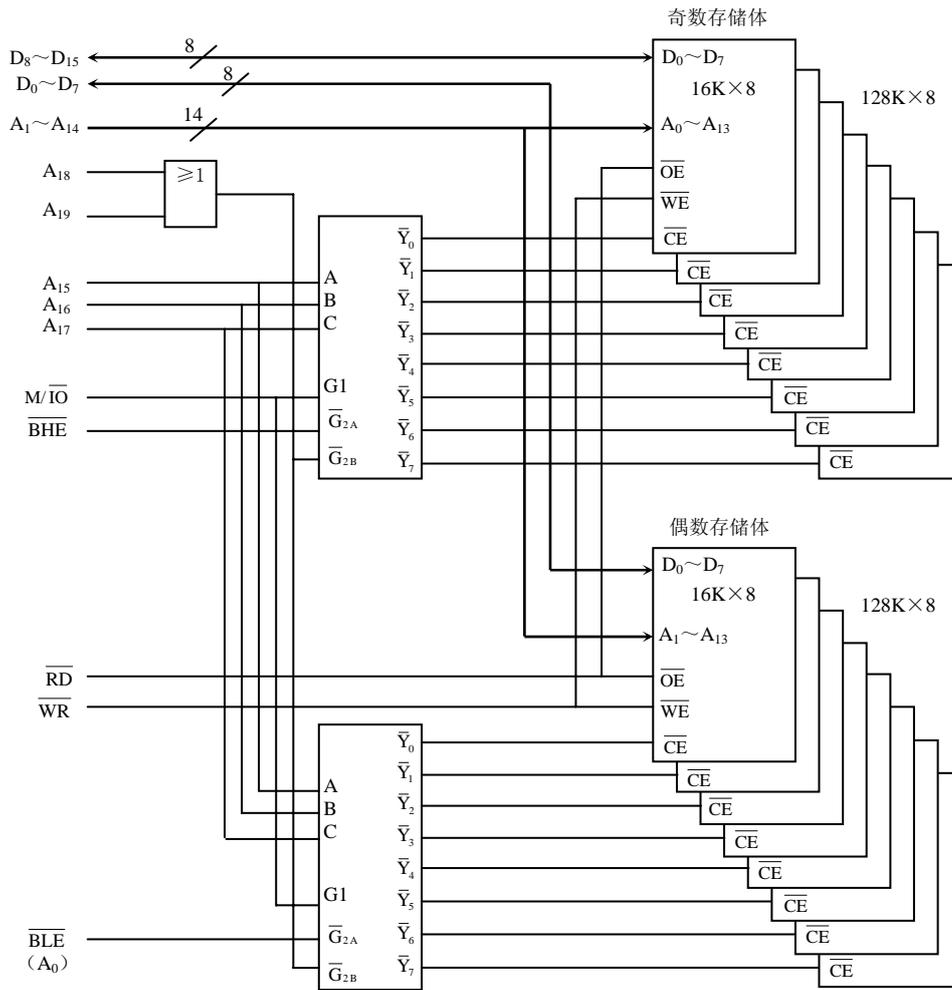


图 4.21 用 16K×8 位的 SRAM 芯片实现的 8086 存储器



图 4.22 统一译码时读/写控制信号产生电路

4.5 高速缓冲存储器基本原理

现代高档微机系统中，为了提高存储器系统的性能，普遍在 CPU 与内存之间采用高速缓冲存

存储器 (Cache) 技术。即通过在 CPU 与内存之间设置一个小容量的高速局部存储器, 而把由 DRAM 组成的大容量内存储器当作高速存储器来使用。

Cache 的有效性源于 CPU 对存储器的访问在时间和空间上所具有的局部区域性。大量的典型程序运行表明, CPU 从主存取指令或取数据, 在一定时间内, 只是对主存局部区域的访问。这是由于指令和数据在主存内都是连续存放的, 而且程序中子程序和循环程序往往要多次重复执行, 对数组的访问在时间上也相对集中, 也即 CPU 取指令和访问数据产生的内存访问地址分布不是随机的, 而是相对簇聚的, 使得 CPU 在执行程序时, 访存具有相对的局部性。因此, 只要将 CPU 近期要用到的程序和数据, 提前从主存送到 Cache, 那么就可以做到 CPU 在一定时间内只访问 Cache, 从而减少对内存的访问, 这也就降低了 CPU 对内存存取速度的要求。

4.5.1 Cache 的基本结构和工作原理

Cache 的基本结构如图 4.23 所示。它由 Cache 存储器、地址映像机构和置换控制器几部分组成。

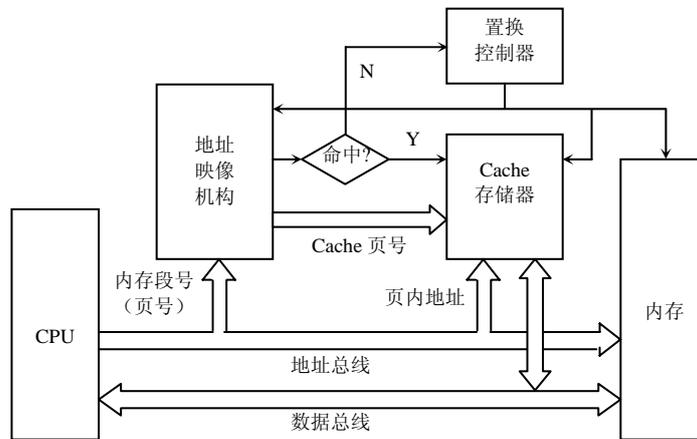


图 4.23 Cache 存储器结构

Cache 存储器是 Cache 的主体, 存放由内存复制过来的内容。Cache 在管理上将内存和 Cache 存储器划分成大小相同的页, 以页为单位交换信息。

地址映像机构用于将 CPU 送来的内存地址转换为 Cache 地址。由于内存和 Cache 的页大小相同, 它们有相同的页内偏移地址 (即相同的低位地址), 因此地址变换主要是内存的页号 (高位地址) 与 Cache 页号间的转换。这种地址变换是通过地址映像机构定义的转换函数来完成的, 具体采用何种转换函数取决于 Cache 与内存间采用的地址映像方式。CPU 访问内存时, 将地址总线送出的内存地址高位部分同存放在地址映像机构内部的地址标记相比较, 以判别 CPU 要访问的地址单元是否在 Cache 中。若在, 称为 Cache 命中, 由转换函数将内存页号 (高位地址) 转换为 Cache 页号, 得到 Cache 访问地址, CPU 可用极快的速度对它进行读/写操作; 若不在, 则称为 Cache 未命中, 这时就需要从内存中访问, 同时把与本次访问相邻近的存储区域 (一页) 内容复制到 Cache 中, 并在地址映像机构中进行标记。

置换控制器负责调页换页。当 Cache 内容已满, 无法接受来自内存页的信息时, 就必须由置换控制器按一定的置换算法来确定应从 Cache 内移走哪个页返回内存, 而把新的内存页调入 Cache

中。常用的置换算法有先进先出（FIFO）和最近最少使用（LRU）两种。

FIFO 算法总是将最先调入 Cache 的页置换出来，它不需要随时记录各页的使用情况，所以容易实现，开销小。但其缺点是可能把一些需要经常使用的程序（如循环程序）页也作为最早进入 Cache 的页而被置换出去。

LRU 算法是将近期最少使用的页置换出来。它需要随时记录 Cache 中各页的使用情况，以便确定哪个页是近期最少使用的页。LRU 算法的平均命中率比 FIFO 高，但实现起来比较复杂，系统开销较大。

也有的微处理器（如龙芯 2F 等）中采用随机置换算法，其显著优点是置换控制简单，但缺点也是显然的。

4.5.2 Cache 与内存的映像关系

从内存将某一部分内容调入高速缓冲存储器是以页为单位调动的。高速缓存中各页所存放的位置与主存中相应页的映像关系，决定于对高速缓存的管理策略。从原理上，可以把映像关系分为全关联方式、直接映射方式和分组关联方式三种。

1. 全关联方式

这种方式允许内存中任一页映像到 Cache 的任一页。假定内存地址为 $NA=N+M$ 位，Cache 容量为 2^{C+M} 字节，页面大小为 2^M 字节，全关联映像方式如图 4.24 所示。这种映像方式可以从被占满的 Cache 中置换出任一旧页，映射方法较灵活，Cache 的利用率和命中率较高，因而缩小了页冲突率。缺点是内存页号全部要用作地址映像机构的“标记”，这就使 Cache “标记”的位数较多，而且访问 Cache 时需要和 Cache 的全部“标记”位进行比较，才能判别出所访问的内存地址是否已在 Cache 内，所以地址转换速度较慢，而且需要采用某种置换算法将 Cache 中的内容调入调出，实现起来系统开销大。

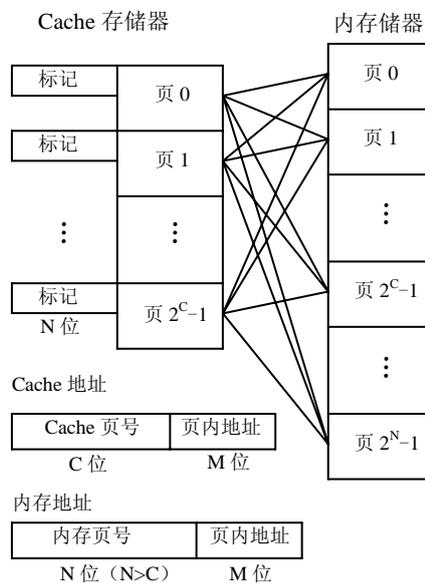


图 4.24 全关联映像

2. 直接映射方式

直接映射方式是将内存中每个页映射到某一固定的 Cache 页中,如图 4.25 所示。内存按 Cache 大小分为若干个段,段内再划分成与 Cache 相同的页,每段按对应的页号进行映射,也即 Cache 中的各页只接收主存中相同页号的内容。在这种映像方式中,地址映像机构只需保存内存段号作为 Cache “标记”。地址映像时,只需根据 CPU 地址总线送出的内存页号找到 Cache 页,然后根据“标记”是否与内存段号相符来判断,若相同且有效位为“1”,表示命中,可根据页内地址从 Cache 中取得信息;若不符或有效位为“0”,则表示未命中,就需要从主存读入新的页来置换旧页。

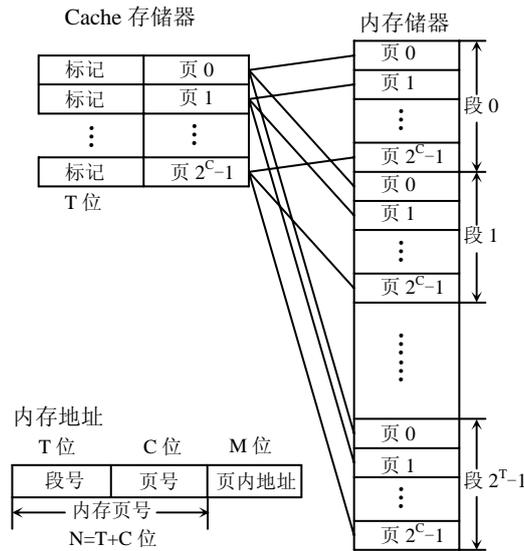


图 4.25 直接映射

直接映射方式的优点是实现简单,地址转换速度快;缺点是不够灵活,由于每个内存页只能固定地对应某个 Cache 页,即使 Cache 内有许多空闲页也不能占用,使 Cache 的存储空间得不到充分利用。此外,如果程序恰好要重复访问对应同一缓存位置的不同内存页,就要不停地进行置换,从而降低了命中率。

3. 分组关联方式

分组关联方式是前两种方式的折衷,映射原理是将 Cache 和内存都分为大小相同的若干组,组内直接映射,组间采用全关联映射。这种方式下,允许不同组中相同页号的内容同时存放在 Cache 中。

4.5.3 Cache 的读/写操作

Cache 的读操作过程如 Cache 工作原理所述, CPU 将主存地址送往主存、启动主存读的同时,也将主存地址送往 Cache,并将主存地址高位部分同存放在地址映像机构内部的地址标记相比较,若 CPU 要访问的地址单元在 Cache 中, CPU 可用极快的速度对它进行读操作,不访问主存;若不在,就需要从主存中访问,同时把与本次访问相邻近的一页内容复制到 Cache 中,并在地址映像机构中进行标记。

Cache 的写操作与读操作有很大的不同,这是因为在具有 Cache 的系统中,同一个数据有两

个拷贝，一个在主存，一个在 Cache。因此，当对 Cache 的写操作命中时，就会出现如何使 Cache 与主存内容保持一致的问题。针对这一情况，通常有如下几种解决方法。

- 通写 (Write-Through) 法。这种方法是在每次写入 Cache 的同时也写入主存，使主存与 Cache 相关页内容始终保持一致。它的优点是比较简单，而且 Cache 中任意页被随时置换，决不会造成数据丢失的错误；缺点是会增加访存次数，影响工作速度。
- 回写 (Write-Back) 法。回写法每次只是暂时将数据写入 Cache，并用标志将该页加以注明。当 Cache 中任一页数据被置换时，只要在它存在期间发生过对它的写操作，那么在该页被覆盖之前必须将其内容写回到对应主存位置中去；如果该页内容没有被改写，则其内容可以直接淘汰，不需回写。这种方法的速度比通写法快，但结构要复杂得多，而且主存中的页未经随时修改，可能失效。
- 只写主存。这种方法是只将数据写入主存，同时将相应的 Cache 页有效位置“0”，表明此 Cache 页已失效，需要时再从主存调入。

目前 80486 和 Pentium 系列微机系统中，以通写法和回写法应用较多，且一般一级 Cache 采用通写法，二级 Cache 采用回写法。

4.5.4 分级 Cache 结构与平均访存周期的估算

为了最大限度地提高 Cache 的命中率，目前高档微机系统中不仅普遍采用了一级 Cache，而且增设了二级 Cache，从而构成一种分级 Cache 结构。通常将一级 Cache 集成在 CPU 芯片中，而将二级 Cache 以芯片（卡匣）内置或外置的方式设计。例如 Pentium 系列微处理器中都集成了 16KB~128KB 的一级 Cache，同时芯片（卡匣）内置或外置了 128KB~2MB 不等的二级 Cache。在这种采用分级 Cache 结构的微机系统中，一般有 80% 左右的内存访问请求可在一级 Cache 中命中，剩下 20% 的内存访问请求大约又有 80% 可在二级 Cache 中命中。这样，整个 Cache 的命中率将达到 96% 左右，全部内存访问请求真正需要到内存去访问的大约只有 4%。

在有二级 Cache 的系统中，CPU 对内存的平均访问周期 T 大体可按下式估算：

$$T = T_1 \times H_1 + T_2 \times (1 - H_1) \times H_2 + T_M (1 - H_1) (1 - H_2) \quad (4.1)$$

式中： T_1 、 T_2 和 T_M 分别为一级 Cache、二级 Cache 和内存的存取周期， H_1 和 H_2 分别为一级 Cache 和二级 Cache 的命中率。

一般集成在 CPU 中的 Cache 和芯片（卡匣）内置的 Cache 具有同 CPU 内核相同的工作频率。假定一级 Cache 集成在 CPU 内，二级 Cache 内置于 CPU 卡匣中，这样它们的存取周期 T_1 和 T_2 便相等；设 $T_1 = T_2 = T_C$ ，同时假定它们的命中率也相等，且 $H_1 = H_2 = H_C$ ，则式 (4.1) 可变成：

$$T = T_C (2H_C - H_C^2) + T_M (1 - H_C)^2 \quad (4.2)$$

若系统中只有一级 Cache，则相当于式 (4.1) 中的 $H_2 = 0$ ，于是 CPU 的平均访存周期为：

$$T = T_1 \times H_1 + T_M (1 - H_1) \quad (4.3)$$

例 4.4 某微机由一级 Cache 和 DRAM 构成存储器。若 Cache 的存取周期为 5ns，DRAM 的存取周期为 100ns，Cache 的命中率为 90%，试求该存储器的平均访问周期。

解：由式 (4.3) 可求得平均访存周期

$$T = 5\text{ns} \times 90\% + 100\text{ns} \times 10\% = 14.5\text{ns}$$

例 4.5 若将例 4.4 微机系统的存储器由一级 Cache 改为两级 Cache，且两级 Cache 的存取周期和命中率都相同，仍分别为 5ns 和 90%，试求存储器的平均访问周期。

解：由式 (4.2) 可求得平均访存周期

$$\begin{aligned} T &= 5\text{ns}[2 \times 90\% - (90\%)^2] + 100\text{ns} \times (1 - 90\%)^2 \\ &= 5\text{ns} \times 0.99 + 100\text{ns} \times 0.01 = 5.95\text{ns} \end{aligned}$$

由例 4.4 和例 4.5 可以看出，系统中有没有 Cache，CPU 访问内存的速度有很大不同。当采用一级 Cache 时，访存周期缩短了 85.5%；采用两级 Cache 时，比一级 Cache 时又缩短了 58.97%，比不用 Cache 时缩短了 94.05%。

4.6 外存储器

外存储器是指需要通过设备接口与微机相连的存储器，也称辅存。与内存相比，外存容量大、价格低、能长期和脱机保存信息，但速度较慢，主要用作微机系统的后备存储器，用以存放计算机工作所需要的系统文件、应用程序、用户程序、文档和数据等，也用作虚拟存储器的硬件支持。目前，微机系统常用的外存储器有硬盘、光盘、移动硬盘、软盘和 U 盘等。软盘已基本不用，所以本节只对硬盘、光盘、移动硬盘和 U 盘作一简单介绍。

4.6.1 硬盘存储器

硬盘是微机系统最主要的外存储器，主要用作大容量的后备存储器和虚拟存储器的硬件支持。第一个商品化的硬盘是由美国 IBM 公司于 1956 年研制成功的。近五十年来，无论在结构还是在性能上，磁盘存储器都有了很大的发展和进步。

1. 磁盘存储器的记录原理

磁盘和磁带均属于磁表面存储器。磁表面存储器记录信息是通过在磁头和磁性记录介质之间作相对运动来实现读写操作的，如图 4.26 所示。磁头是实现电磁转换过程的关键装置，它是由软磁材料做铁芯，上面绕有读/写线圈的电磁铁。磁记录介质是在某种刚性（如硬盘）或柔性（如软盘、磁带）载体上涂有薄层磁性材料的物体，用于记录以磁状态表示的信息。

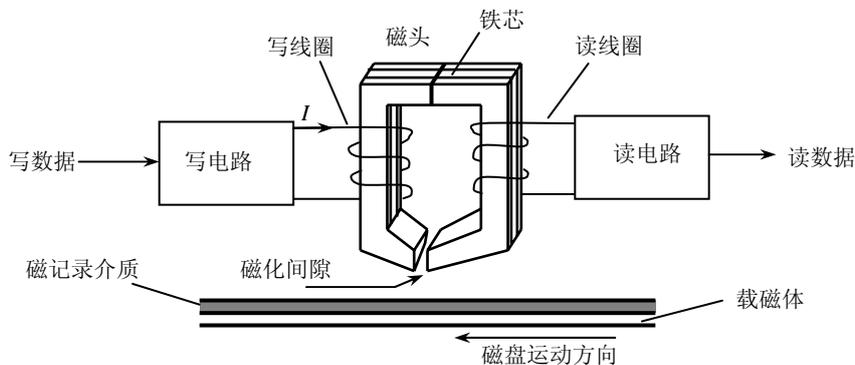


图 4.26 磁表面存储器记录信息原理图

(1) 磁存储器的写入过程

当向线圈提供一定方向和大小的电流时，将使磁头体被磁化，建立起有一定方向和强度的磁场，即在磁环内有磁力线产生，由于磁头的磁化间隙处磁阻较大，将产生漏磁，这个漏磁就是向磁记录介质中写入信息的信息源。当磁头前端与磁记录介质距离很近时，磁化间隙处的漏磁将把

处于附近的磁记录介质上的一小片磁性材料（磁化单元）磁化，而当磁头离去时，就在这个磁化单元保留了磁化状态，从而记录下写入的一位信息（完成了“电—磁”转换）。可以根据写入驱动电流的不同方向，使磁层表面被磁化的极性方向不同，以区别记录“0”或“1”。

（2）磁存储器的读出过程

当磁头前端与磁记录介质距离很近且高速经过时，若所经过的磁记录介质上的磁化单元已被磁化，这一磁化状态将在磁头的环体内产生磁力线，从而在磁头的线圈中感应出一个脉冲电流，这表示读出了记录在磁记录介质中的一位信息（完成了“磁—电”转换）。根据感应电流的方向不同，可以区分读出的是“0”还是“1”。

2. 硬盘存储器的组成原理

硬盘存储器由硬盘驱动器、硬盘控制器和盘片几大部分组成，如图 4.27 所示。

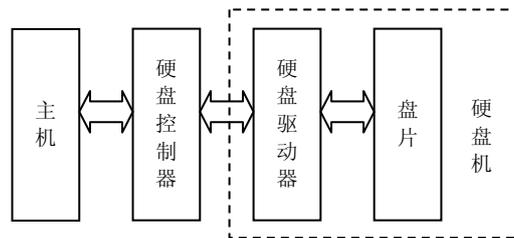


图 4.27 硬盘存储器的基本结构

微机系统中配置的硬盘均为可移动磁头固定盘片结构，这种结构的硬盘又称为温彻斯特磁盘，简称温盘。其特点是工作时磁头悬浮在高速转动的盘片上方，而不与盘片直接接触。如图 4.28 所示，它采用密封组合方式，将磁头、盘片、驱动部件以及读写电路等制成一个不能随意拆卸的整体，叫作“头盘组合体”。



图 4.28 揭开外盖后的温彻斯特磁盘

（1）硬盘驱动器

硬盘驱动器一般与盘片一起构成一个完整独立的设备，称为硬盘机（实物如图 4.28 所示）。它包括作为磁记录介质使用的磁盘和驱动磁盘匀速旋转的动力与驱动部件，完成读写功能的磁头和驱动磁头沿磁盘径向方向运动和准确定位的部件，以及控制逻辑电路等部件。其内部结构框图如图 4.29 所示，分为三个子系统：

一是主轴电机驱动电路。硬磁盘的盘片（组）被固定在硬盘机的主轴上，由主轴带动磁盘匀速旋转，而硬盘机的主轴则由一个主电机通过传动皮带带动旋转。为使读写数据正确，同时保证浮动磁头与磁盘表面有合理的距离，要求磁盘以一个额定的转速匀速旋转。为此在电机上用光电

一个硬盘控制器通常可以控制一台或几台驱动器。但一般说来，硬盘机都有自己的控制器，不同的硬盘机，在处理硬盘机本身的驱动器与硬盘控制器之间的功能划分方面，会有一些不同的安排。图 4.30 给出了硬盘控制器与硬盘驱动器功能划分的示意表示。

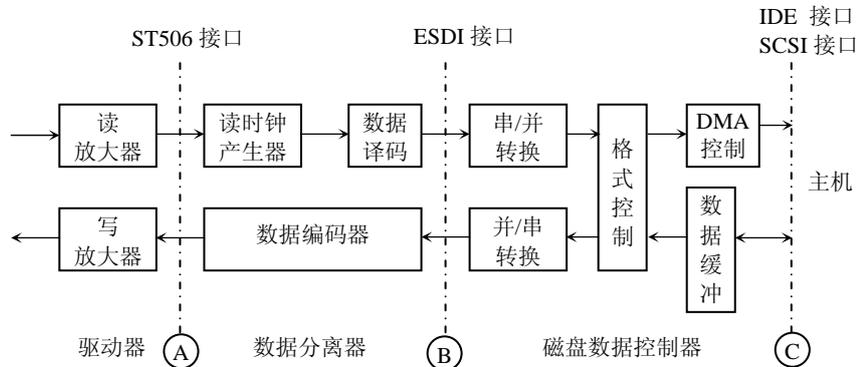


图 4.30 硬盘控制器与硬盘驱动器的功能划分

如采用 ST506 接口标准，硬盘控制器与驱动器的界面可设在图 4.30 的 A 处。这时硬盘读/写等逻辑电路主要划分在硬盘控制器上，而硬盘驱动器部分只保留了读写和放大电路。如采用 ESDI（Enhanced Small Device Interface，增强型小型设备接口）标准，则将界面设在 B 处，把数据分离电路和编码、译码电路划入硬盘驱动器中，控制器仅完成串/并或并/串转换、格式控制和 DMA 控制等逻辑功能。

而采用当前流行的 IDE（Integrated Drive Electronics，集成驱动电子部件）或 SCSI（Small Computer System Interface，小型计算机系统接口），则硬盘控制器的功能全部划归到设备之中。这种把盘体与控制器集成在一起的做法减少了硬盘接口的电缆数目与长度，数据传输的可靠性得到了增强，硬盘制造起来变得更容易，因为厂商不需要再担心自己的硬盘是否与其他厂商生产的控制器兼容，对用户而言，硬盘安装起来也更为方便。

3. 硬盘上的信息组织

磁盘片是磁存储器的信息记录载体，它的上下两面都可用于记录信息。硬盘一般采用多片结构的磁盘组，读这样的磁盘上的信息时，必须指出该信息在磁盘的哪个盘片的哪一记录面。由于磁盘上的信息必须由磁头读出或写入，所以记录面的数量与磁头数是一样的。一般就用磁头（head）号来代替记录面号。

在同一个磁盘记录面上，信息被写在许多个同心圆上，每个同心圆为一个磁道，不同磁道用磁道号表示。磁道的编址是从外向内依次编号，最外一个同心圆叫 0 磁道，最里面的一个同心圆叫 n 磁道。不同记录面上的同一磁道被叫做一个柱面（Cylinder），柱面个数正好等于磁道数，所以磁道号就是柱面号。

在同一个磁道上，信息被组织为固定大小的区段，称为扇区（sector），即把一个圆周等分成若干部分，每部分就构成一个扇区，每个扇区的一个磁道，用于存储一定数目的二进制信息（一般为 512 字节），如图 4.31 所示。扇区的一个磁道通常是磁盘进行读写的最小信息单位。在磁盘寻址时，首先确定柱面，再确定磁头，最后找到扇区。不同扇区用扇区号表示，为此必须有办法标识一个磁道的起始位置，以便表明第一个扇区的开始。请注意，在一个磁道上，只有一部分区

域用于记录有用的信息, 还有很多区域用于标记磁道的开始、结束、扇区位置(编号)、磁头号等, 以及用于保存数据校验与纠错处理的冗余信息(常用的是 CRC 校验码), 可能还有一些必要的间隙部分。所以一个磁盘上的可用存储容量, 不是简单地用磁道数乘上每个磁道理论上可写入的最多信息数目(可磁化的单元数)。对一个磁盘片(组), 在使用之前要进行格式化操作, 即在每个磁道上完成区域划分, 写入各种标记信息, 建立标明磁盘记录面使用情况的信息位图等。磁盘的存储容量是指在磁盘完成格式化操作之后, 留给用户实际可用的存储空间, 通常用字节数表示。

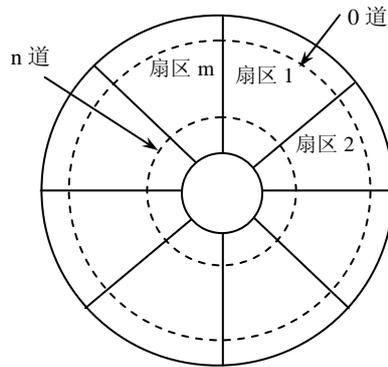


图 4.31 磁盘的磁道和扇区格式示意图

4.6.2 光盘存储器

光盘存储器是随着多媒体计算技术的兴起而出现的一种新型外存储设备, 一经出现, 就以其记录密度高、存储容量大、信息保持时间长、价格低廉、经久耐用和便于携带等优点而受到计算机用户的特别青睐, 目前已成为高档 PC 机系统必不可少的标准配置之一。光盘存储器的功能部件组成与磁盘存储器相似, 也由盘片和盘机(驱动器)两部分组成。在信息分布上, 光盘也与磁盘相似, 盘片划分为若干光道, 每个光道又划分为若干扇区, 每个扇区存放一定长度的数据块。光盘直径有 12 英寸、5.25 英寸、4.75 英寸和 3.5 英寸等多种规格。在 12 英寸的盘面上, 约有 31000 个光道, 每道的扇段数有 25、32、64 等规格, 位密度约为 $3 \times 10^8 \text{bpi}$, 一个盘片的容量超过 1GB。

光盘驱动器(Optical Driver)常被简称为光驱, 它和软盘、硬盘驱动器不一样, 对光盘的读写既不是用磁头作接触式读写, 又不是从外向里读写, 而是采用光读写头把激光束汇集成一个光点, 由里向外螺旋式地对光盘进行非接触式的读写。由于光读写头比磁读写头复杂得多, 加之盘上的光道数很多, 因此光盘的存取速度目前仍低于硬盘, 大致与软盘相当。

1. 光盘存储器分类

目前光盘存储器主要分为三类: 只读型、一次写入型和可擦写型。这与半导体只读存储器分成掩模 ROM、PROM 和 EPROM/EEPROM 三类很相似。

(1) 只读型光盘(CD-ROM)

CD-ROM 光盘上的信息是由厂家在母版上刻录好的, 用户只能按需选购已记录信息的光盘, 并在 CD-ROM 驱动器上读出, 不能进行信息写入、更改和擦除。一张 5.25 英寸的 CD-ROM 盘片, 存储容量约为 600MB, 位成本低, 易于分发, 便于保存, 且不会受病毒干扰, 所以是多媒体应用的首选存储载体。因为 CD-ROM 是只读的, 因此非常适合于存储不允许更改擦除的文件资料, 目

前已在出版业和其他企事业单位广泛使用，用于制作各种电子文字声像出版物和产品说明书等。

CD-ROM 最先是由 Philip 和 Sony 两家公司发明的，随后相继制定和推行了一系列 CD 的编码和数据组织格式的标准（CD 是 Compact Disc 的缩写，本意是指一种注塑成形的镀铝盘，用 μm 量级的沟槽表示数据，用激光读出数据）。这些标准被广泛采用或借鉴修订，成为 CD 平台间相互兼容的通用国际标准。

按 CD 标准生产的 CD-ROM 早期产品采用的都是与 CD 唱机和激光视盘机相似的单速光盘驱动器，传输速率仅为 150KB/s。由于这种光盘驱动器的速度较慢，所以在 1994 年以后市场就过渡到以双倍速光驱为主流产品，其一般传输速率为 300KB/s。1995 年以后，三倍速、四倍速以及更高倍速的光驱也逐渐普及，目前市场上的主流光驱产品已普遍达到 40 倍速以上，即数据传输速率在 6000KB/s 以上。

但是仅在原有 CD-ROM 标准上提高速度总不是最佳的办法，于是在业界的强力推动下，一种称为 DVD (Digital Versatile Disc, 数字通用光盘) 的新型大容量高速度光盘标准在 1995 年 9 月应运而生了。按 DVD 标准制造的 DVD-ROM 新型光盘和光盘驱动器也于 1996 年开始投放市场。这种 DVD-ROM 光盘可存储 5GB 以上的信息，读取速度可达 4MB/s 以上，为原有 CD-ROM 光盘存储器的 10 倍以上。

DVD 光盘根据其容量和格式不同大致可分成四种，如表 4.5 所示。目前市面上比较常见的是 DVD-5 和 DVD-9 两种单面 DVD 碟片，DVD-10 和 DVD-18 两种双面 DVD 碟片则由于涉及到盘片换面的问题，而且容量也太大，实际中的需求并不多，因此暂时还不多见。采用 DVD 的影视光盘不仅播放时间长，可以达到单盘 135 分钟以上，而且可以获得比 CD 和 VCD 更好的图像播放质量。

表 4.5 DVD 光盘 4 种规格

名称	格式	容量
DVD-5	单面单层	4.7GB
DVD-9	单面双层	8.5GB
DVD-10	双面单层	9.4GB
DVD-18	双面双层	17GB

(2) 一次写入型光盘 (WORM)

WORM (Write Once Read Many-time) 光盘也叫 CD-WO (CD-Write Once) 盘。它提供用户一次写入机会，包括允许在一次未写完的剩余空间中追加信息。但信息一旦写入，便与 CD-ROM 一样只能读出，不能擦除、重写。WORM 光盘的记录密度高，存储容量大（每面可达 759MB~3.4GB），记录信息稳定可靠，保存时间长（10~15 年），位价格较低。缺点是读写时间长（一般需 200~600ms），数据传输率不高。

WORM 光盘在制造材料、结构型式和存储原理上与 CD-ROM 光盘很相似，都是采用形变型记录原理，通过母版压制或激光照射，在盘片记录薄膜上形成凹坑（小孔）或微小气泡而制成的。光盘采用三层结构：基片、反射层和记录层。基片使用一种耐热的有机玻璃 PMMA 做成，基片上涂一层铝膜作为反射层，反射层上再涂一层碲合金薄膜作为记录介质，最后涂一层透明的保护膜。写入时，能量集中的激光束照射某个特定区域，使该处加热到熔点温度，记录介质蒸发，形成一

个凹坑,代表记录了1;不发激光束,记录层完好如初、未形成凹坑的特定区域,则代表记录了0。读出时,用低功率激光束沿光道扫描,当扫到记录1的凹坑时,光束直接射到反射层,反射光强,经光探测器转换为1读出信号;当扫到无凹坑处时,反射光很弱,表示0读出信号。

WORM 写入必须用专门的 WORM 驱动器,也叫 CD-R 刻录机。但是写入了信息的 WORM 光盘既可以在 WORM 驱动器上读出,也可以在 CD-ROM 驱动器上读出。因此, WORM 的出现对多媒体光盘出版系统是一大推动,对政府部门、图书馆、档案馆、会议、培训和广告等场合也很适用。

为了提高一次写入型光盘的存取速度和存储容量,和推出与 CD-ROM 对应的 DVD-ROM 相似,业界也推出了与 CD-R (即 WORM) 光盘存储器对应的 DVD-R 光盘存储器。

(3) 可擦写型光盘 (CD-RW)

CD-RW (CD-ReWritable) 光盘允许用户在盘片上任意地进行信息的写入、修改、擦除和读出,因此备受用户特别是研究开发型用户的青睐。相应的盘驱动器俗称 CD-RW 刻录机。

CD-RW 光盘按记录原理可分为磁光型盘 (CD-MO) 和相变型盘 (PCD) 两种。

1) CD-MO 光盘

这是一种制造技术已成熟、目前采用较多的可重复读写光盘,它是将磁记录和激光技术结合起来进行读、写、擦除和重写的。它的双面格式化容量可达 1GB 左右,数据传输率可达 1MB/s,存取时间较短,对环境温度、电磁干扰等不敏感,盘片寿命在 40 年以上。缺点是需使用相应的磁光盘驱动器。不过磁光盘驱动器的使用特点与传统的磁盘驱动器完全相同,而且比之多了介质可换、可靠性高、存储容量大的优势,因此它在与磁盘驱动器的竞争中发展,有望成为计算机存储设备市场的主流产品。

CD-MO 光盘以磁性材料为记录介质,利用热磁效应写入,利用磁光效应读出,通过恢复原有磁化状态擦除。写入前,在外加磁场作用下使记录介质呈某种磁化方向。写入时,用较强激光束照射的特定微小区域温度升高,磁化强度下降,在外加磁场作用下使该区域的磁化方向翻转,即记录了一个 1;未被照射的区域则保持磁化方向不变,表示记录了 0。读出时,用较弱的激光束沿光道扫描,根据磁光效应,反射光的偏转角度与介质的磁化方向有关,只要适当安置光检偏器的角度,使记录 1 处的反射光可以通过检偏器,而记录 0 处的反射光则因角度不同不能通过,便可识别是 1 还是 0。擦除时,用激光照射记录介质以加热,同时外加磁场,使记录介质恢复到写入前的磁化状态,即清除了写入的信息。

2) PCD (Phase Change Disc) 光盘

这是一种只用光技术来记录、读出、擦除和重写信息的可读写光盘。写入时用激光加热的办法使介质内部在结晶和非结晶两种形态间发生改变,一旦写入,在常温下便保持不变,除非重写。读出时则利用这两种状态具有不同的反射率来区分 0 和 1。PCD 光盘的存储容量可达到 1.5GB 以上,平均读写时间小于 50ms,数据传输速率大于 1MB/s,是另一种前景看好的计算机存储设备的主流产品。

PCD 光盘是利用晶相结构(结晶状态)的可逆性变化而制成的。写入时,用较强的激光束照射记录介质薄膜,被照射的微小区域突然加热,晶粒直径变大;由于照射时间极短,已变化了的直径来不及再缩回去,就保持扩大了直径不变。晶粒直径的变化使光折射率相应改变,于是照射处(写 1 处)与未照射处(写 0 处)在光反射率上存在明显差别。读出时,用较弱的激光束沿光道扫描,利用光反射率的差别识别该处记录信息是 1 还是 0。擦除时,用激光束照射记录介质,

使其在加热后缓慢冷却（即退火），于是晶粒直径由大变小，恢复原来的结晶状态，反射率差异随之消失，从而使记录信息被清除。

和推出与 CD-ROM 对应的 DVD-ROM，与 CD-R（即 WORM）对应的 DVD-R 相似，业界为提高可擦写型光盘的存储容量和存取速度，也推出了与 CD-RW 光盘存储器对应的 DVD-RW 光盘存储器。

2. 光驱的组成结构及工作原理

光驱是读写光盘信息的设备。和上述光盘存储器的类型相对应，光驱也有 CD-ROM 驱动器、CD-R 驱动器、CD-RW 驱动器和 DVD-ROM 驱动器、DVD-R 驱动器、DVD-RW 驱动器之分，当然也有可运行任一种光盘的多功能光驱产品。但是，无论哪种光驱，其基本组成结构是相似的，基本上都由激光头及伺服控制系统、机械传动机构、数字信号处理系统及接口、面板背板及控制系统等部分组成。

（1）激光头及伺服控制系统

激光头是光盘驱动器与光盘片的耦合部件，是光驱的心脏，也是其最精密的部件。激光头的原理性结构如图 4.32 所示。

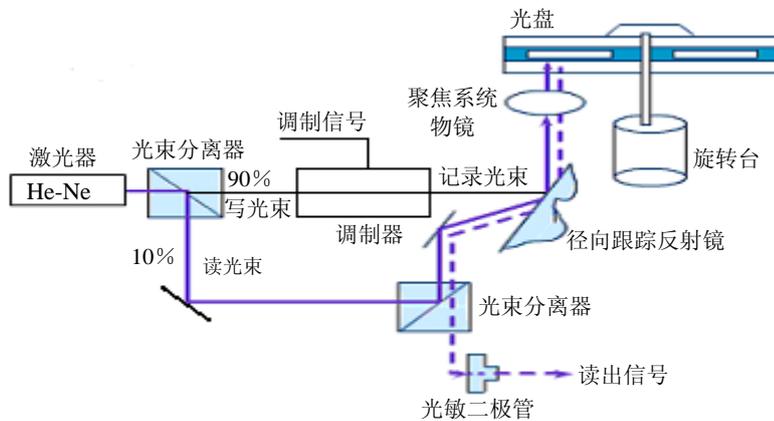


图 4.32 激光头原理性结构示意图

由图可见，它主要由激光器、光束分离器、光聚焦镜、反射镜、调制器和光电二极管等部件组成。激光器（图中为氦-氖激光器）是光源部件，用于产生一定强度的激光束；光束分离器把接收到的激光束分成 90% 的写光束和 10% 的读光束两路；调制器用于控制是否让写光束通过，以产生记录光束；记录光束通过跟踪反射镜和聚焦镜实现寻道定位，把写光束聚焦为很细的光束打到指定的光盘位置，从而在那里“刻”出一个小凹坑，表示写入“1”。读出时，读光束也会照射到指定的光盘位置，但它的能量很弱，不会在盘片记录介质上形成凹坑，从而改变其反射特性，而只会把所照射之处的有无小凹坑状态通过反射光的强弱表现出来，反射光被送到光敏二极管，从而确定读出来的是“1”还是“0”。

为了最大限度地减小激光头的聚焦、跟踪误差，确保经物镜聚焦后的激光束准确地落在盘面上（聚焦）和信道中央（跟踪），目前的高档光驱都采用了悬挂式激光头结构，并对光束聚焦和光道跟踪分别引入了伺服控制系统。光束聚焦控制系统通过四象限光电检测器产生聚焦误差信号控制物镜上下移动，从而确保聚焦效果；光道跟踪控制系统采用三光束法等技术产生光道跟踪误差信号控制光头水平移动，以确保跟踪效果。

激光头中还有一个由微型电位计控制增益的激光强度(功率)调节器,通过调节控制,可获得读、写、擦除的不同功率,同时也可在改善读盘能力和延长激光管使用寿命之间取得较好折中。一些新型光驱还可通过伺服控制系统,根据实际使用情况动态地调整激光头的功率,以获得尽可能高的读盘能力和尽可能长的激光管使用寿命。

(2) 机械传动机构

机械传动机构包括主轴旋转机构、托盘机构和平衡机构等部分。

主轴旋转机构的核心部件是电机,作用是带动托盘和光盘在光驱中按照一定规律高速旋转。光驱产品的不断提速,实际上是意味着主轴转速的不断提升,当主轴旋转速度提高到 10000 转/s 左右后,读盘的时候会产生巨大的震动及噪声。为此,目前市场上的 40 倍速以上光驱产品中,普遍在其托盘与前仓盖之间引入了橡胶支架以减震,并使用两个抗震动装置与动态阻尼器构成的双动态抗震悬吊系统(DDSS, Double Dynamic Suspensory System)来有效吸收主轴电机高速旋转时产生的震动。

为了提高光驱读取密度不均匀的盘片时的性能,并保护光驱,现在的高档光驱中还大多采用了一种先进技术——自动平衡系统(ABS, Auto Balance System),即在托盘下面放置滚珠,当光驱读写密度不均的盘片时,滚珠在离心力的作用下,会滚动到质量轻的那一边,从而使整个盘片的质量得到平衡,盘片在数据读写过程中的转速也更稳定。一些高档光驱如三星、雄兵 48 倍速光驱等,为了防止爆盘,还引入了智能监测调整机制(又称智能安全技术),在其固件中固化了监测调整程序,用以识别盘片上的划痕和裂纹,并记录裂纹生长情况,据此调整主轴电机转速,从而杜绝爆盘现象的发生。这种智能安全技术也有助于顺利读取各种坏盘和有效控制光驱震动及其带来的噪声。

(3) 数字信号处理系统及接口

数字信号处理系统的主要功能是将激光头读取的 0、1 信号,转换为符合输出接口标准的连续数据流,然后通过输出接口传送给主机。其中还有音频解码单元,用于还原 CD 模拟音频信号,以支持光驱独立的 CD 播放能力。不过由于 CD 播放毕竟不是光驱的主要功能,所以其音频解码单元比较简单,因此播放的音质较差。

(4) 面板背板及控制系统

面板及控制系统的主要作用是通过按键控制光盘托盘的进出,通过面板指示灯动态指示光驱的运行状态,通过带开关旋钮控制 CD 的播放与播放音量。除此之外,面板上还有紧急弹出孔和耳机插孔,前者用于停电或出现故障时通过插入一曲别针等细钢丝将托盘弹出,取出光盘,后者用于插入耳机听 CD 音乐。有的光驱(主要是中国台湾地区生产的光驱)面板上还有向前、向后搜索等按钮,用于选音乐曲目。

背板上主要提供一些接口和插座,包括:

数字音频输出接口(Digital Audio Output Connector)——用于连接到数字音频系统。

模拟音频输出接口(Analog Audio Output Connector)——用于通过音频线与声卡相连。

主/从盘跳线器(Master/Slave Jumper)——用于设置主盘或从盘。

数据线插座(Interface Connector)——用于连接数据线(数据线另一端连接主机板上的 CD-ROM 控制器)。

电源线插座(Power-In Connector)——用于连接电源线,为光驱提供电源。

图 4.33 所示为某 CD-ROM 驱动器的面板、背板结构图。



图 4.33 某 CD-ROM 驱动器的面板、背板结构图

早期光驱的机芯有相当大一部分是由塑料元件构成的。在光驱主轴电机速度提高后，随着工作时间的增长，光驱内部积蓄的热量越来越多，其核心部分的温度可达到相当高，致使塑料元件老化速度加快，光驱使用寿命缩短。因此，现在的光驱大多采用金属机芯。另外，为了散热，普遍在光驱中电路板的主控芯片（电机驱动芯片）与底部金属壳之间安装了一个具有良好导热功能的硅胶散热片，或者在主控芯片上涂以硅脂。由于硅胶材料独特的柔软性，硅胶散热片除能散热外，还能有效缓冲对芯片的意外直接硬性撞击。

3. 光驱的接口

目前市面上的光驱与主机的接口，主要有 IDE、EIDE、SCSI、SCSI-2 四种。一般说来，后两种接口的光驱在性能上要优于前两种接口的光驱，表现为数据传输率较高、工作稳定性更好、CPU 占用率较低。但相比之下，它们的价格也较贵，且安装较复杂，必须通过专门的 SCSI 接口卡才能与主机相连，因此它们主要用在服务器和 workstation 等高档微机上。对一般终端用户使用的微机而言，采用 IDE 或 EIDE 接口的光驱，既安装方便（可直接连在主机的 IDE 接口上），又同样可满足一般多媒体应用对数据传输率和稳定性的需求，并且物美价廉，兼容性好，所以现在大多数用户采用的光驱都是 IDE/EIDE 接口型的。

其实，现在 IDE 接口光驱的各项性能也在快速提高，尤其是对 CPU 的占用率已经与 SCSI 接口方式大体持平了。目前 32 倍速以上的 IDE 接口光驱均采用了 Ultra DMA/33 标准，该标准采用 DMA 方式传输数据，光盘与内存之间的数据传输在 DMA 控制器的控制下直接进行，不必 CPU 参与，因此既降低了 CPU 占用率，又提高了数据传输率。这样一来，IDE 接口光驱的市场竞争优势就更加突出了。

目前市场上的光驱特别是刻录机，也有采用 USB 接口的。采用这种接口的光驱具有方便的热插拔和即插即用功能，且安装方便，可跨平台使用；传输速度和 CPU 占用率等性能也不错。

无论哪种连接标准的接口，就其内部适配电路及功能而言，其实是大同小异的，基本上都由数据输入缓冲器、记录格式器、编码器组成的写盘通道和译码器、读出格式器、数据输出缓冲器等组成的读盘通道所构成；主要功能是把计算机输出的数据信息转换成光驱所能接受的信息格式，以及把光驱从光盘中读出的信息转换成计算机所能接受的信息格式，同时对计算机与光驱之间的速度差异起同步协调作用。

4.6.3 移动存储器

目前流行的移动存储器主要有移动硬盘和 U 盘。

1. 移动硬盘

移动硬盘是随着多媒体技术和宽带网络的发展,人们对移动存储的需求越来越高而发展起来的一种便携式、大容量移动存储设备。

(1) 移动硬盘的组成结构

移动硬盘实际上是将硬盘和一些外围控制电路集成在一起,并封装在硬脂塑料外壳内,通过外部接口与主机相连的一种移动存储设备。图 4.34 给出了一种移动硬盘的实物样例和内部结构。由于采用硬盘为存储介质,因此移动硬盘在数据读写模式和存取原理方面与标准 IDE 硬盘是相同的。所以,对移动硬盘的读写,关键是需要通过移动硬盘内的控制器(如 USB 控制芯片)实现标准 IDE 接口数据与主机接口数据(如 USB 接口数据)之间的转换。

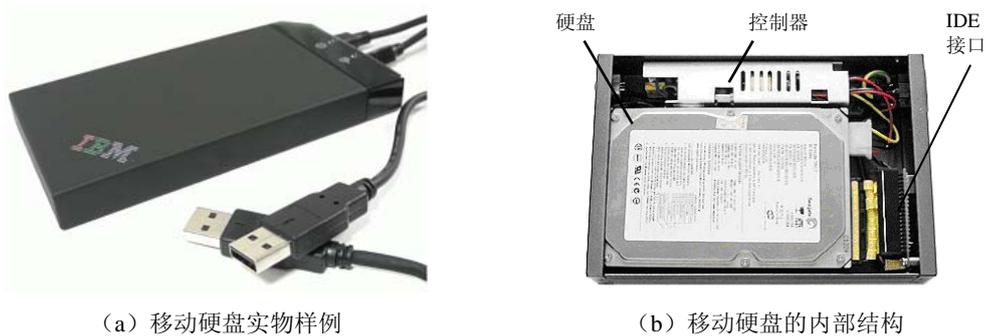


图 4.34 移动硬盘实物样例和内部结构

(2) 移动硬盘的接口标准

移动硬盘外置接口方式主要有并行接口、IEEE 1394 接口和 USB 接口三种。并行接口移动硬盘出现较早,通过并行打印机接口与主机相连,由于其数据传输率较低且不支持即插即用功能,目前已被淘汰。IEEE 1394 接口也称 Fire wire (火线)接口,其数据传输速度理论上可达 400Mb/s,并支持热插拔。但只有一些高档 PC 机主板才配有 IEEE 1394 接口,所以普及性较差。USB 接口已成为移动硬盘的主流接口,它支持热插拔,传输速度高达 480Mb/s (USB2.0)。

(3) 移动硬盘的特点

移动硬盘作为 PC 机的一个重要外设,近年来得到了蓬勃的发展,主要是因为它具有如下特点:

- 容量大。移动硬盘可以提供相当大的存储容量,是一种较具性价比的移动存储产品。目前大容量“U 盘”价格还无法被用户所接受,而移动硬盘则在用户可以接受的价格范围内,提供给用户较大的存储容量和不错的便携性。目前市场中的移动硬盘能提供 10GB、20GB、40GB 等容量,一定程度上满足了用户的需求。
- 传输速度快。移动硬盘大多采用 USB、IEEE 1394 接口,能提供较高的数据传输速度。
- 使用方便。现在的 PC 基本都配备了 USB 功能,主板通常可以提供 2~8 个 USB 口,一些显示器也提供了 USB 转接器,USB 接口已成为个人电脑中的必备接口。USB 设备在大多数版本的 Windows 操作系统中,都可以不需要安装驱动程序,具有真正的“即插即用”特性,使用起来灵活方便。
- 可靠性提升。数据安全一直是移动存储用户最为关心的问题,也是人们衡量该产品性能好坏的一个重要标准。移动硬盘以高速、大容量、轻巧便捷等优点赢得许多用户的青

睐，而更大的优点还在于其存储数据的安全可靠性。移动硬盘与笔记本电脑硬盘的结构类似，多采用硅氧盘片。这是一种比铝、磁更为坚固耐用的盘片材质，并且具有更大的存储量和更好的可靠性，提高了数据的完整性。采用以硅氧为材料的磁盘驱动器，以更加平滑的盘面为特征，有效地降低了盘片可能影响数据可靠性和完整性的不规则盘面的数量，更高的盘面硬度使 USB 硬盘具有很高的可靠性。

2. U 盘

U 盘是随着闪速存储器（Flash Memory）技术和 USB 接口技术而发展起来的一种新型移动存储器。U 盘是由我国朗科公司发明的，目前已作为新一代的存储设备在几十个国家申请了发明专利，在国内外被广泛使用。

U 盘的存储介质是 Flash 存储器，这种存储器既可在不加电的情况下长期保存信息，具有非易失性，又能在线进行快速擦除与重写（可重复擦写达 100 万次以上）。U 盘实际上是将 Flash 存储器和一些外围控制电路焊接在电路板上，并封装在颜色比较亮丽的半透明硬脂塑料外壳内的一种小型便携式移动存储器。如图 4.35 所示，给出了一种 U 盘的实物样例，它通过 USB 接口与主机相连。有的 U 盘内部还设计了用来显示其工作状态的指示灯和提供类似软盘的写保护。写保护是用一个嵌入内部的拨动开关来实现的，它可以控制对 U 盘的写操作，从而可减少由于操作失误而造成数据丢失的机会。



图 4.35 U 盘实物样例

(1) U 盘的内部结构

U 盘的内部结构和电路连接分别如图 4.36 和 4.37 所示。它主要由 U 盘控制器和 Flash 存储器芯片构成。此外，还包含一个电压调节芯片，用于将 USB 接口输入的 5V 电压转换为 3.3V。

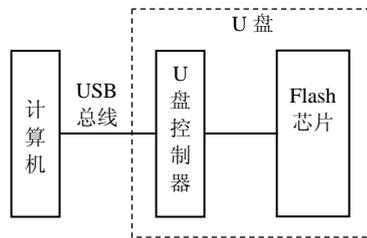


图 4.36 U 盘的内部结构

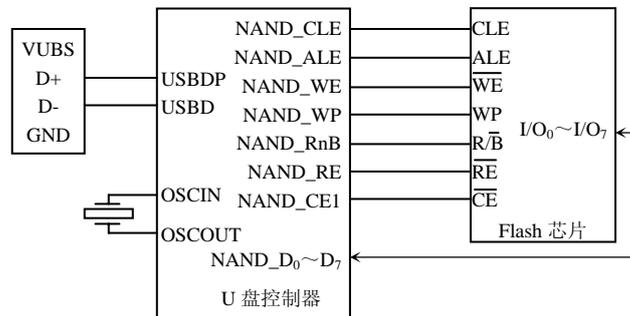


图 4.37 U 盘电路

Flash 存储器是 U 盘的主体, 用于存储信息。U 盘控制器则作为 USB 总线和 NAND Flash 芯片的接口, 从 USB 总线上接收来自计算机的各种命令, 将计算机对 U 盘的读/写操作转换为对 Flash 芯片的操作。

U 盘控制器实际上是一个被微缩的计算机, 它包含一个 CPU、程序和内存。U 盘控制器执行的程序被称为固件 (Firmware)。U 盘固件的主要工作流程如图 4.38 所示。

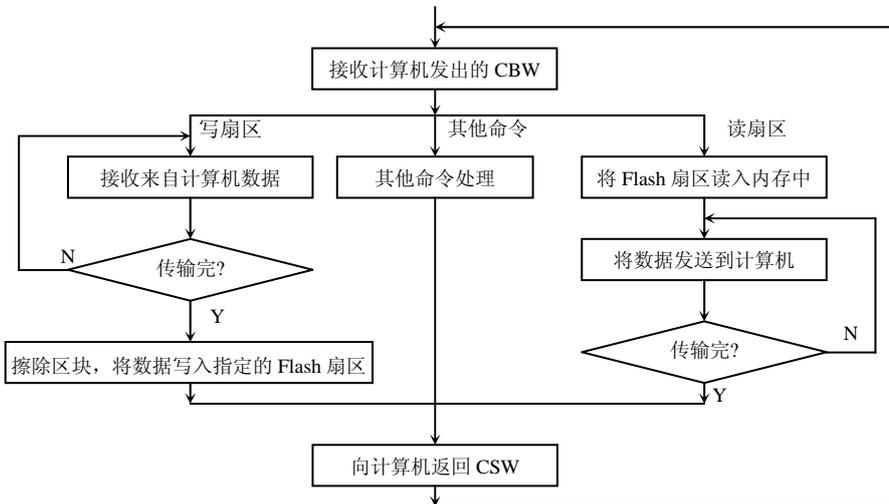


图 4.38 U 盘固件的主要工作流程

有三种数据类型在 USB 和 U 盘之间传输: CBW、CSW 和普通数据。

① CBW (Command Block Wrapper): 命令块分组。它是从计算机发送到 U 盘设备的命令, U 盘控制器从 CBW 中提取命令, 执行相应的操作。

② CSW (Command Status Wrapper): 命令状态分组。它是 U 盘控制器完成命令后, 向计算机发出的当前命令执行状态, 计算机根据 CSW 来决定是否继续传送下一个 CBW 或数据。

③ 数据: 根据 CBW 的不同, U 盘可以向计算机发送数据, 或接收来自计算机的数据。

(2) U 盘的特点

U 盘之所以被广泛用作便携式移动存储器, 是因为它优越的性能和特点:

① U 盘不使用驱动器, 不仅方便文件共享与交流, 还可节省开支。

② U 盘采用 USB 接口, 无须外接电源, 支持即插即用和热插拔, 只要用户的主板上有 USB 接口就可以使用 U 盘。而且使用 Windows Me/2000/XP 等操作平台时, 不用安装驱动程序就可以使用 U 盘。使用时, 只需将 U 盘插在 USB 口, 过几秒钟, 系统会自动显示 U 盘所使用的盘符。使用者完全可以将它视为一般的活动硬盘来使用。

③ U 盘的存取速度快, 大约是软盘的 15 倍, 读为 750KB/s, 写为 450KB/s。由于 U 盘读写大文件要比小文件快, 因而 U 盘特别适合于传送大型文件。

④ U 盘的体积非常小且重量轻 (重量仅相当于一支圆珠笔), 便于携带。而且由于采用无机机械装置、结构坚固, 所以 U 盘有很好的防震性能。

目前, U 盘的常用规格有 64MB、128MB、256MB、512MB、1GB、2GB 等, 以后随着 Flash 存储器容量的提高, 还会有更大容量的 U 盘。总之, U 盘作为新一代的存储设备不仅目前应用广泛, 而且具有广阔的应用发展前景。

4.7 虚拟存储器管理机制

虚拟存储器的概念如前所述，它是在存储体系层次结构基础上，由操作系统的存储管理软件对内存和外存进行统一管理，将内存和外存统一编址，形成一个比内存空间大得多的虚拟存储空间。虚拟地址空间的大小取决于 CPU 所采用的存储器管理机制，并由一个大容量的快速硬盘或光盘存储器支持。虚拟存储器的地址称为虚拟地址或逻辑地址。在采用虚拟存储器技术的系统中，用户编程时使用逻辑地址，这使得用户编写程序时可以不考虑内存空间大小的限制在虚拟空间内自由编程。运行时首先由操作系统将需要执行的部分程序调入内存，其余部分则仍留在外存上，此后需要执行的程序段不在内存时，由操作系统动态地进行调入和调出，从而实现在具有小内存空间的系统中运行大容量的程序。

由于用户编程使用的是逻辑地址，而 CPU 能访问的是物理存储器地址，所以实现虚拟存储器的关键是自动而快速地完成虚拟地址向物理地址的变换，通常也称为地址映像或程序定位。目前普遍采用的地址映像都是使用驻留在存储器中的各种表格，规定各自的转换函数，在程序执行过程中动态地完成地址变换。具体采用何种地址映像方法取决于虚拟存储器的管理策略，常用的虚拟存储管理有以下三种：

- 页式管理。页式管理的基本原理是将虚拟存储空间和内存空间划分成固定大小的若干页，并为每个页按顺序指定一个页号。然后以页为单位来分配、管理和保护内存，每个任务或进程对应一个页表，保存在内存中。CPU 访问某个页时，通过页表将虚拟页号转换为物理页号，从而实现虚拟地址到物理地址的变换；若该页不在内存，就需要由操作系统的存储管理软件按一定的算法淘汰和调页。
- 段式管理。段式管理是以虚拟存储器和内存的分段来作为内存分配、管理和保护的基础。段的大小取决于程序的逻辑结构，每个任务或进程对应一个段表驻留在内存中，用于记录段的有关信息，如段号、段基址、段长度和段装入等。CPU 通过访问段表，判断该段是否已调入内存，并完成虚拟地址到物理地址的变换。
- 段页式管理。这种管理策略是综合页式和段式管理的优点而产生的一种折中方法。它首先将程序按其逻辑结构划分为若干个大小不等的逻辑段，然后再将每个逻辑段划分为若干个大小相等的逻辑页；主存空间也划分为若干个同样大小的物理页。每个任务或进程对应一个段表，每段则对应有自己的页表，系统以页为单位进行地址映像。

目前各种 16 位、32 位微机系统中，大多采用了虚拟存储器技术。下面以 Pentium 微处理器为例，详细介绍 Pentium 的分段分页管理机制，以及虚拟地址向物理地址转换的原理。

4.7.1 分段分页管理机制

Pentium 的虚拟存储器管理与 80386/80486 基本相同，采用分段分页管理策略。分段分页管理的基本思想如前所述：首先将虚拟地址空间分成若干个大小不等的逻辑段，逻辑地址由间接指向段基址的 16 位段选择符和 32 位段内偏移量两部分组成，并由分段机制将 48 位逻辑地址转换为 32 位线性地址。然后再将线性地址空间等分为固定大小的若干页，将线性地址用页基址和页内偏移量表示，以页为单位进行地址映射，并由分页机制将 32 位线性地址转换为 32 位物理地址。这种分段分页机制的原理示意如图 4.39 所示。

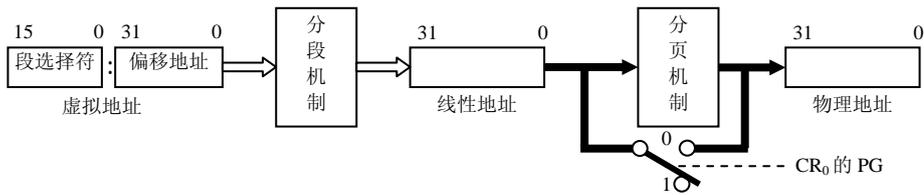


图 4.39 80486/Pentium 的分段分页机制示意图

虚拟地址空间是二维的，它所包含的段数最大可到 16K 个（由 GDT 和 LDT 定义），每个段最大可到 4GB（分段分页时，只分段时为 1MB），从而构成 $16K \times 4GB = 64TB$ 容量的庞大虚拟地址空间。线性地址空间和物理地址空间都是一维的，容量为 $2^{32} = 4GB$ 。

分段机制是 Pentium 虚拟存储管理的基础，而分页机制则是可选的。当控制寄存器 CR₀ 的 PG 位为 0 时，分页机制被禁止，这时分段机制产生的线性地址即是物理地址。

4.7.2 虚拟地址向实地址的转换原理

Pentium 的分段和分页部件支持段式、页式和段页式三种虚拟存储管理策略。其虚拟地址向实地址的转换首先由分段机制将虚拟地址转换为线性地址，分页时再由分页机制将线性地址转换为物理地址。

1. 虚拟地址向线性地址的转换

Pentium 的虚拟地址由 16 位段选择符和 32 位偏移量两部分构成。段选择符由 CS、SS、DS、ES、FS 和 GS 六个段寄存器提供，32 位偏移量由指令中的寻址方式给出。在分段机制中，每个逻辑段都由三个参数定义：

- 段的基址。它是线性空间中段的开始地址。
- 段的界限。是指段内可以使用的最大偏移量，它指明该段的长度。
- 段的属性。如可读出或写入段的特权级等。

以上三个参数存储在段的描述符中。所有任务所共享的段的描述符组织在一起构成全局描述符表 GDT，而每个任务所私有的段的描述符则存放在各自的局部描述符表 LDT 中。将虚拟地址转换成线性地址是根据驻留在内存中的全局描述符表 GDT 和局部描述符表 LDT 进行转换的，转换原理如图 4.40 所示。

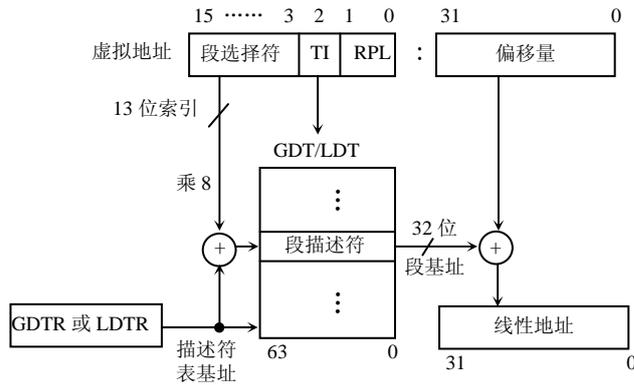


图 4.40 虚拟地址向线性地址的转换

由段选择符中的表指示符 TI 找到段描述符所在的描述符表 GDT/LDT，再由描述符索引字段乘 8 计算出段描述符相对于 GDT/LDT 表基址的偏移量，找到该段对应的描述符，描述符中 32 位段基址加上 32 位段内偏移量即得 32 位线性地址。

2. 线性地址向物理地址的转换

Pentium 采用段式管理，即未启用分页机制（CR₀ 中的 PG 位等于 0）时，分段机制产生的线性地址即为物理地址；采用段页式管理，即同时启用分段分页机制时，段部件产生的线性地址空间被划分成大小固定的页，由分页机制把线性地址空间中的任一页映射到物理空间的一页，将线性地址转换成物理地址。

与 80486 不同的是，Pentium 除支持 4KB 分页外，还支持 4MB 分页。新的 4MB 页由控制寄存器 CR₄ 的 PSE 位选择。无论 4KB 分页还是 4MB 分页，将线性地址转换成物理地址都是由驻留在内存中的页表来完成的。

(1) 4KB 分页时的地址变换

Pentium 的 4KB 分页与 80386/80486 的分页机制相同，为节省页表所占用的内存空间，也是采用两级页表机构：第一级页表由 2^{10} 个表项构成页目录表，每项 4 个字节，占 4KB 内存，其物理基址由 CR₃ 控制寄存器提供；第二级也是由 2^{10} 个表项构成页表，每项 4 个字节，占 4KB 内存。这样，两级页表组合起来只占 8KB 内存，即可描述 2^{20} 个表项。这种分页机制的转换原理如图 4.41 所示。

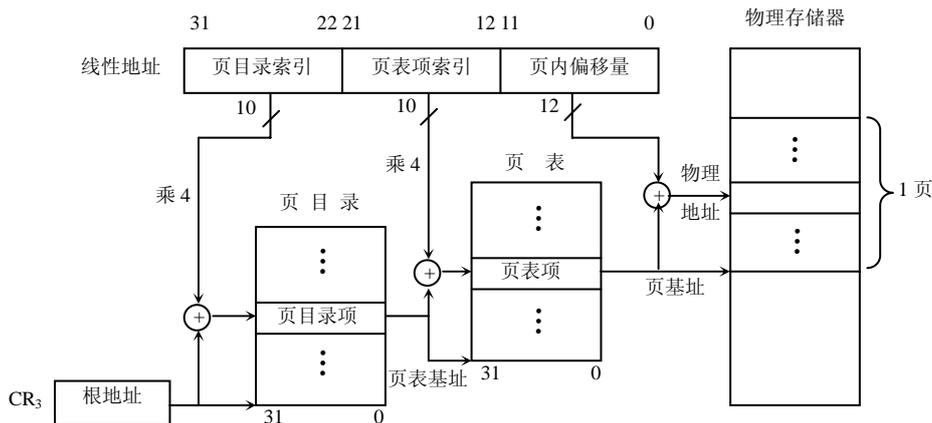


图 4.41 线性地址到物理地址的转换

为了与两级页表机构相适应，将 32 位线性地址分成页目录索引（10 位）、页表项索引（10 位）和页内偏移地址（12 位）三个字段。由页目录索引乘 4 计算出页目录项相对于页目录的偏移量，找到该页对应的页表的首地址（页目录项），再用页表项索引乘 4 计算出页表项相对于页表的偏移量访问页表，即可找到与线性地址相对应的物理地址所在页的页基址（低 12 位为 0），页基址与页内偏移量相加即为物理地址。

例 4.6 假定 CR₃ 中值为 00000000H；访问页组前，内存中已有 3 页被访问过并已定位；访问此页前，内存已有 40 页被定位。设线性地址为 01234567H，试通过页目录表和页表将其转换为物理地址。

将线性地址 01234567H 分解为页目录索引、页表项索引和页内偏移地址的形式如下：



则通过两级页表进行物理地址变换的过程如图 4.42 所示。

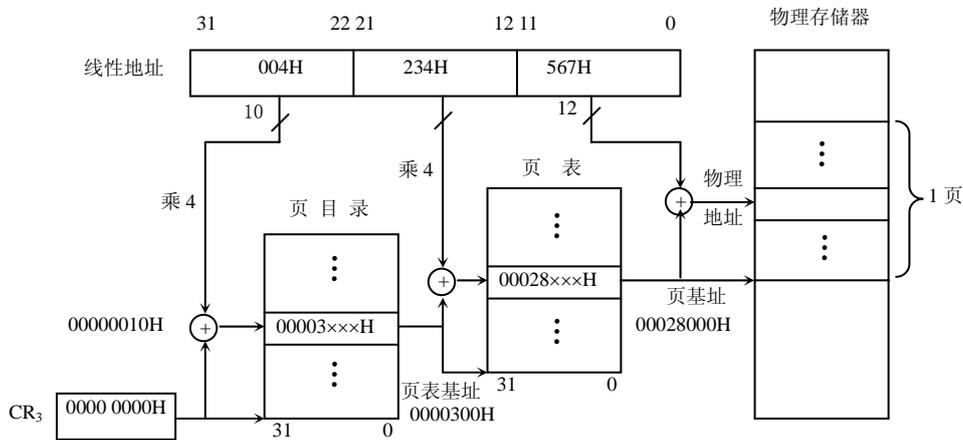


图 4.42 线性地址到物理地址的转换

第一步：根据项目索引和 CR_3 给出的项目基址计算出项目表的物理地址：

$$\text{项目表物理地址} = CR_3 \text{ 中基址} + \text{项目索引} \times 4 = 00000010H$$

以该地址从项目表中取出页表基址为：00003000H

第二步：根据页表项索引和得到的页表基址计算出页表项的物理地址：

$$\text{页表项物理地址} = 00003000H + 234H \times 4 = 000038D0H$$

以该地址从页表中取出页基址为：00028000H

由此得到物理地址为：

$$\text{物理地址} = \text{页基址} + \text{页内偏移量} = 00028000H + 567H = 00028567H$$

(2) 4MB 分页时的地址变换

Pentium 采用 4MB 分页时，只需要单一的一个页表，从而大大地减少了内存用量。图 4.43 给出了线性地址 00400002H 在 4MB 页中定位的转换过程。由项目索引乘 4 计算出项目项相对于项目表的偏移量，访问一级项目表，即可找到与线性地址相对应的物理地址所在页的页基址（低 22 位为 0），页基址与页内偏移量相加即为物理地址。

3. 转换后援缓冲器 TLB

Pentium 使用 4KB 分页时，采用两级页表机构节省了内存，但处理器进行地址变换时，需访问两级页表，从而降低了地址变换速度。为解决这一问题，Pentium 微处理器与 80486 一样，在页部件中设置了一个转换后援缓冲器 TLB。TLB 是一个可容纳 32 个页表项的高速缓存，它存放着最近访问过的 32 个页面所对应的页表项。

TLB 由两个字段组成：标记字段存放线性地址对应的虚页号；页表数据字段则存放着该页面所对应的页表项（页基址的高 20 位）。页部件将线性地址变换为物理地址时，用虚页号同时在二级

页表和 TLB 中查找，若在 TLB 中找到，说明 TLB 命中，立即停止在二级页表中的查找，直接使用页表数据字段作为页基址；若 TLB 未命中，就继续在二级页表中查找，如果找到，就取出页基址（页表项），同时将它复制到 TLB 中；如果仍未找到，则产生一个页面故障，供操作系统处理。这种查表转换过程如图 4.44 所示。

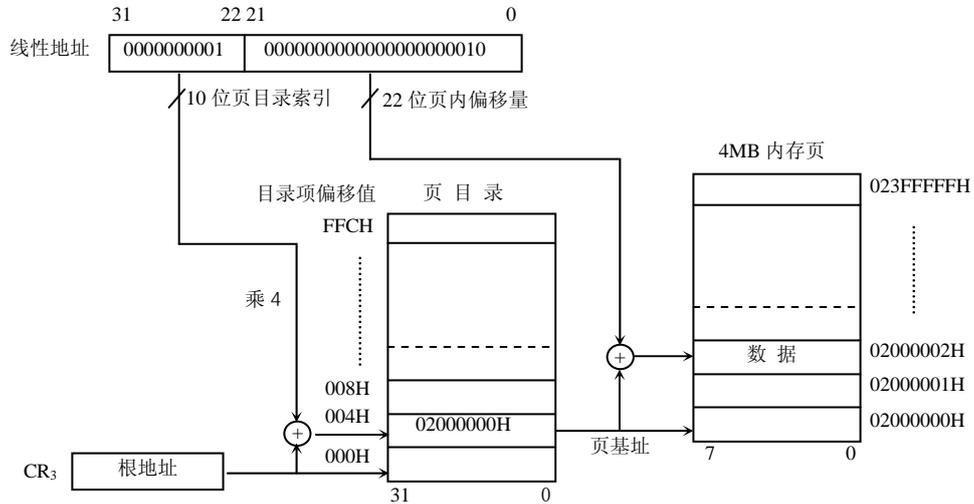


图 4.43 线性地址 00400002H 在 4MB 页中重新定位到 02000002H

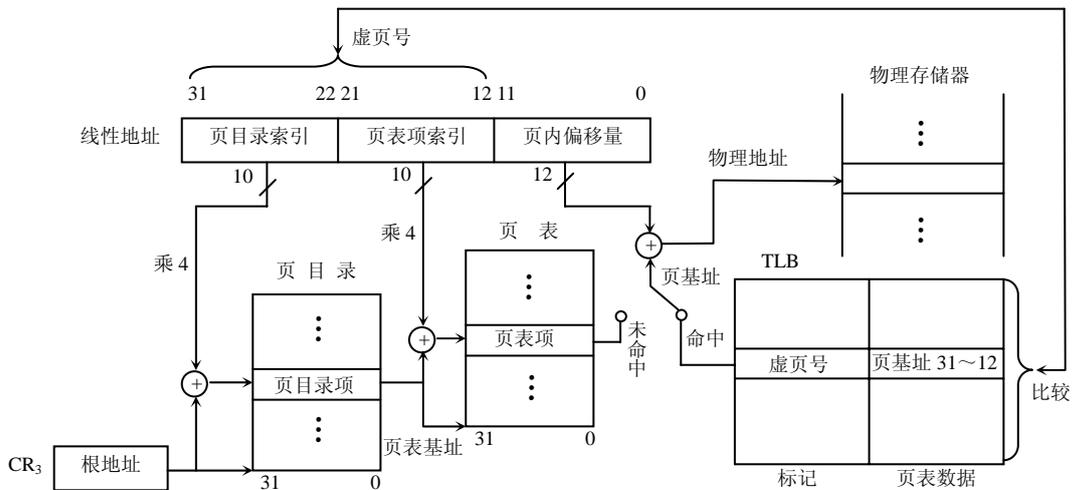


图 4.44 经二级页表和 TLB 的地址转换

4. Pentium 的页式管理

Pentium 的页式管理不使用段部件，此时所有逻辑段映射到同一线性地址空间（逻辑地址对应的线性地址实际上就是段内偏移地址），线性地址到物理地址的转换由页部件完成，原理同上。这种方式下，一个任务拥有的虚拟地址空间仅为 4GB。

- (2) 可直接存取 16M 字节内存的微处理器, 其地址总线需_____条。
A. 24 B. 16 C. 20
- (3) 设存储器的地址线为 20 条, 存储单元为字节, 使用全译码方式组成存储器, 该系统构成最大存储器容量需要 64K×1 位的存储器芯片的数量是_____。
A. 32 B. 64 C. 128
- (4) 在 16 位以上微机系统中, 存储器一般都_____构成。
A. 按 16 位编址、以 16 位为单位
B. 按字节编址、以字节为单位
C. 按字长编址、以字长为单位
- (5) 一个具有 24 根地址线的微机系统中, 装有 16KB ROM、480KB RAM 和 100MB 的硬盘, 说明其内存容量为_____。
A. 496KB B. 16MB C. 100.496MB
- (6) Cache 是介于寄存器组与主存储器之间的一级存储器, 其存储主体一般由_____构成。
A. SRAM B. DRAM C. EPROM
- (7) 某 SRAM 芯片的容量是 1K×8 位, 除了电源和地外, 该芯片引出线的最小数目是_____。
A. 21 B. 20 C. 18
- (8) 80386/80486 等 32 位微处理器为支持 8 位字节、16 位字和 32 位的双字操作, 其内存应采用_____。
A. 单体结构 B. 双体结构 C. 4 体结构
- (9) PROM 存储器是一种_____。
A. 可以随机读写的存储器
B. 可以由用户一次性写入的存储器
C. 只能读出不能写入的存储器
- (10) 在研制某一个计算机应用系统的过程中, 存储其监控程序一般应选用_____。
A. RAM B. PROM C. EPROM
- 4.2 RAM 和 ROM 这两类存储器有什么不同? 它们在计算机中各有什么主要用途?
- 4.3 各类半导体存储器芯片与 CPU 的接口特性有什么共性? 接口的基本原则是什么?
- 4.4 试述动态 RAM 的工作特点。与静态 RAM 相比, 动态 RAM 有什么长处和不足之处。说明它的使用场合。
- 4.5 Intel 系列 PC 机中内存存储器的编址有什么特点? 什么情况下用单体存储器, 什么情况下用多体存储器? 为什么要用多体存储器?
- 4.6 常用的存储器片选控制方法有哪几种? 它们各有什么优缺点?
- 4.7 在对静态存储器进行读/写时, 地址信号要分为几部分? 分别产生什么信号?
- 4.8 什么是位扩展? 什么是字扩展? 当用户购买内存条进行内存扩充时, 是在进行何种存储器扩展?
- 4.9 某微机系统的存储器地址空间为 A8000H~CFFFFH, 若采用单片容量为 16K×1 位的 SRAM 芯片构成, 回答以下问题:

- (1) 系统存储容量为多少?
- (2) 组成该存储系统共需芯片多少个?
- (3) 整个系统应分为多少个芯片组?

4.10 某 8088 微机系统的内存接线如图 4.46 所示。

- (1) 试分析两块内存区域的地址范围。
- (2) 试编写一段汇编语言程序将内存 6264 中首地址开始的 20 个字节清零。

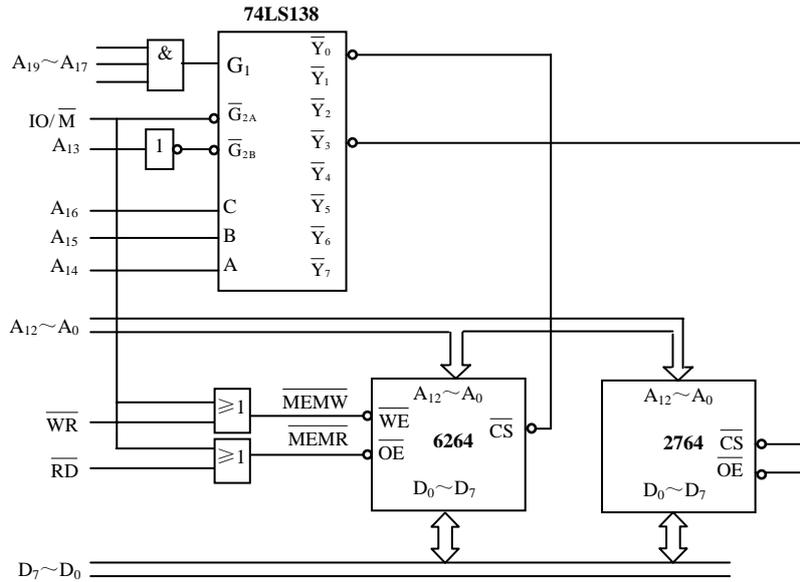


图 4.46 题 4.10 图

4.11 一个微机系统按图 4.47 所示电路扩充了 8K ROM 存储器，假定 CPU 有 16 条地址线，8 条数据线，试指出它是什么地址译码方式，及每片存储器芯片的地址范围（要求所有地址连续）？并指出有无地址重叠，为什么？

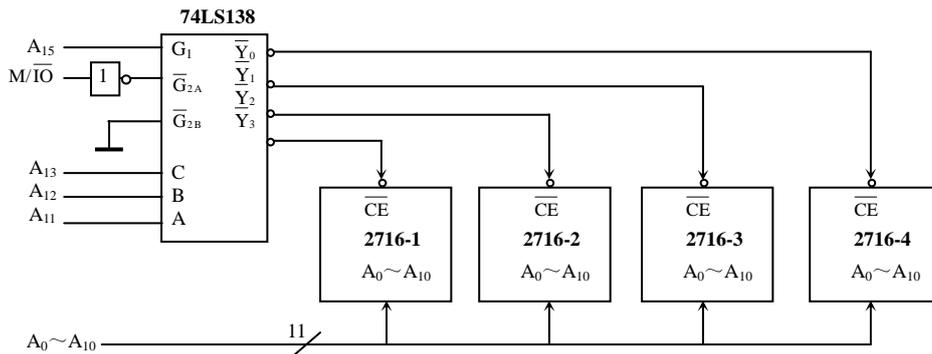


图 4.47 题 4.11 图

4.12 已知某 8086 单板机系统的 SRAM 如图 4.48 所示。

(1) 试分析芯片 (1)、芯片 (2) 和整个存储器的地址范围分别为多少？

(2) 若要将芯片 (1)、芯片 (2) 分别作为偶数存储体和奇数存储体，联合构成一个 $64\text{K}\times 16$ 位的存储器，要求总的地址范围不变，应如何改变图中的接法？请画出新的连线图。这时芯片 (1)、芯片 (2) 的地址范围分别为多少？

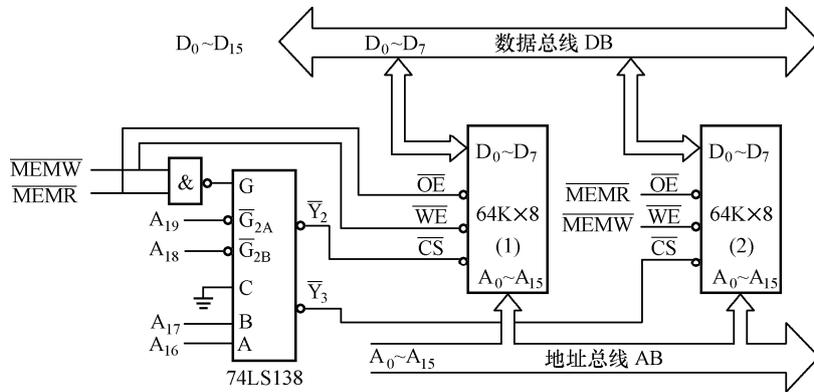


图 4.48 题 4.12 图

4.13 某 8 位微机(地址线为 $A_{19}\sim A_0$)需要配置 SRAM 存储器系统。若选用 $16\text{K}\times 4$ 位的 SRAM 芯片，地址范围为 $09\text{C}000\text{H}\sim 09\text{F}FFF\text{H}$ 。设计译码电路，并画出存储器电路的连接图。

4.14 用 $16\text{K}\times 8$ 位的 SRAM 芯片构成 $\text{C}8000\text{H}$ 到 $\text{D}7\text{F}FF\text{H}$ 的内存：

(1) 利用 3-8 译码器输出作为片选信号，画出连接图。

(2) 若规定用 ROM 作为译码器，说明该如何去做。

4.15 用 $16\text{K}\times 8$ 位的 SRAM 芯片构成 8086 的从 $\text{E}8000\text{H}\sim \text{E}FFFF\text{H}$ 的内存，问：

(1) 共需要几个存储器芯片才能满足上述要求？

(2) 试画出存储器连接图。

(3) 从内存 $\text{E}8000\text{H}$ 开始，顺序将 00H 、 01H 、 02H 、...直到 FFH 重复写满上面构成的内存，编写一个程序段实现该功能。

4.16 简述 Cache 的含义、特点和在存储体系中的作用。

4.17 Cache 和内存中的地址映像方式有哪几种？各有什么特点？

4.18 在有 Cache 的系统中，通写法、回写法是针对什么问题而提出的？它们的含义是什么？

4.19 磁盘组有六片磁盘，每片有两个记录面，存储区域内径 22 厘米，外径 33 厘米，道密度为 40 道/厘米，内层密度为 400 位/厘米，转速 2400 转/分，问：

(1) 共有多少存储面可用？

(2) 共有多少柱面？

(3) 盘组总存储容量是多少？

(4) 数据传输率是多少？

4.20 磁表面存储器和光盘存储器记录信息的原理有何不同？

4.21 什么叫虚拟地址？试简述虚拟存储器的基本工作原理。

4.22 虚拟存储器中常用的地址映像方法有哪几种？试分别说明它们的工作原理。

- 4.23 高档微机中使用 Cache 和虚拟存储器有什么好处? 试比较它们有什么相似之处。
- 4.24 试简述 80386/80486 的虚拟存储管理机制, 并说明它的虚拟地址—物理地址变换原理。
- 4.25 Pentium 的存储器分页机制中, 是怎样来保证既节省内存又不降低线性地址—物理地址变换速度的?
- 4.26 Pentium 的虚拟存储管理机制与 80386/80486 有什么区别?
- 4.27 目前流行 PC 系列机存储器管理机制的保护功能是如何实现的?