第3章 存 储 器

冯·诺依曼计算机结构是以存储器为中心的。存储器用于存放程序以及与程序相关的数据。各类存储器以及管理存储器的软硬件构成了计算机的存储系统。主存储器是存储系统的核心,它用来存储计算机运行期间所需要的程序和数据,CPU 可以直接对其进行控制和访问。在现代计算机系统中,CPU 和主存储器间的速度差异已经成为整个计算机系统速度提升的主要瓶颈;同时,主存储器在容量上也难以满足程序和数据的存储要求。由于位价格、集成度等因素的限制,主存储器在速度和价格上短期内很难有较大程度的改善,因此现代计算机系统一般通过将多种不同性能的存储器构成层次化存储系统的方法来改善存储系统的性能。

本章主要内容包括:

(1) 存储器分类与存储系统的层次结构;

(2) 相联存储器原理;

(3) 高速缓冲存储器原理与设计;

(4) 辅助存储器组成与工作原理。

3.1 主存储器概述

3.1.1 存储器的分类

存储器是计算机系统中的记忆设备,用来存放程序和数据。在存储器中存放信息的基本单位是二进制位,用于存放二进制位的基本器件(或电路)被称为存储元或者存储元件,构成存储元的物理器件需要具有两种明显区分的物理状态,用来分别表示二进制位的“1”和“0”。

存储器按照在计算机系统中的作用、构成存储元的器件以及存取方式等的不同,有多种分类方法,目前常用的有以下三种分类方法。

1. 按在计算机系统中的作用分类

1) 内部存储器

内部存储器(简称内存)是一种可以和 CPU 直接交换信息的存储器。内存主要包括高速缓冲存储器(Cache)和主存储器(简称主存)。主存用来存放 CPU 当前正在执行的程序和数据;高速缓冲存储器位于 CPU 和主存之间,用于解决主存与 CPU 之间的速度差异,存放正在执行的部分程序和数据。由于高速缓冲存储器的内容是主存部分内容的副本,所以内存的容量只由主存储器的容量来决定,与 Cache 的大小无关。

2）外部存储器

外部存储器也被称为辅助存储器（简称外存或辅存），一般用来存储大量的、暂时不运行的程序和数据，以及一些需要永久性保存的信息。外存不能直接被 CPU 访问，它的数据需要被调入内存后才能被 CPU 访问。可以认为，外部存储器是内部存储器的后援存储器。

3）离线存储器

离线存储器又被称为备份存储器，主要用于对在线存储数据进行备份。

4）控制存储器

控制存储器也称为微程序存储器，它用于在微程序控制的计算机系统中存放微程序。控制存储器位于 CPU 的内部。

2. 按存储介质分类

1）半导体存储器

半导体存储器是一种以半导体电路作为存储介质的存储器，现代的半导体存储器普遍采用大规模集成电路工艺制成芯片。按照制造工艺的不同又可将半导体存储器分为 MOS（金属氧化物）型存储器和双极型存储器两大类。MOS 型存储器具有集成度高、功耗低、价格便宜等特点，但存取速度较慢；双极型存储器具有存取速度快、集成度较低、功耗较大、成本较高等特点。传统的半导体存储器存储的信息会因为断电而丢失，即具有电易失性。近年来出现的使用非电易失性材料制成的半导体存储器不会因断电而丢失信息，即具有非电易失性。半导体存储器主要用作内部存储器。

2）磁表面存储器

磁表面存储器是在金属或塑料基体的表面上涂一层磁性材料作为记录介质，工作时磁层随载磁体高速运转，用磁头在磁层上进行读/写操作。根据磁体形状的不同，又可分为磁盘存储器和磁带存储器。其主要特点为容量大、价格低、存取速度慢。磁盘常被用作辅助存储器，磁带则常被用作离线存储器。

3）光盘存储器

光盘存储器用磁光材料作为存储介质，利用激光进行信息的存取。光盘具有存储容量大、存取方便、位价格低和易于保存等优点。按照是否可以写入光盘可分为只读式、一次写入式和可改写式三种；按照所采用的激光类型不同，还可分为 CD 光盘、DVD 光盘、BD 光盘等。

3. 按存取方式分类

1）随机存储器

随机存储器（Random Access Memory，RAM）中任何存储单元的内容都能被随机存取，且存取时间与存储单元的物理位置无关。RAM 又可分为动态 RAM（DRAM，Dynamic RAM）和静态 RAM（SRAM，Static RAM）两种。其中，DRAM 中存储的信息若长时间不被访问会发生变化，因此需要定时对其进行刷新来维持信息的稳定；SRAM 在加电情况下所存储的信息能够稳定保持。

2）只读存储器

只读存储器（Read-Only Memory，ROM）采用的存取方式也是随机存取，但 ROM 在正常工作时，只能读不能写，写入操作需要通过特殊的手段完成。ROM 的最大特点：非易失性和高可靠性。ROM 常被作为主存储器的一部分，用来存放系统程序和各种固件。按照

写入方式不同ROM又可分为：可编程ROM(Programmable ROM,PROM)可进行一次性编程；可擦除可编程ROM(Erasable Programmable ROM,EPROM)可借助于紫外线或者其他方式多次擦除和编程；电可擦除可编程ROM(Electrically Erasable Programmable ROM,EEPROM)的工作原理与EPROM类似,区别在于擦除过程中不需要借助与紫外线等其他方式。闪速存储器(Flash memory)具有EEPROM的特点,但擦除和读写速度比EEPROM快得多。

3) 顺序存取存储器

顺序存取存储器(Sequential Access Memory,SAM)中信息的读/写顺序完全按照其在存储介质上的顺序进行。在顺序存取存储器中,信息的存取时间与信息所在的物理位置有关。由于读/写操作只能顺序地按数据所在物理位置进行查找,所以这种存储器的平均存取速度比较慢。磁带是典型的顺序存储器。

4) 直接存取存储器

直接存取存储器(Direct Access Memory,DAM)的工作原理与SAM类似,其读/写操作分两步进行：首先快速定位到信息所在的一个小区域,然后在此小区域内按照顺序对信息进行准确定位。这种存储器的存取特性及速度介于RAM和SAM之间,也被称为半顺序存取存储器。磁盘存储器和光盘存储器是典型的直接存取存储器。

5) 相联存储器

前面的四种存储器都是按地址访问的存储器,相联存储器(Associative Memory,AM)是按照存储单元中存放的全部或部分内容进行访问的存储器,因此也被称为按内容访问存储器(Content Addressed Memory,CAM)。相联存储器主要应用于需要快速检索的场合。

目前,主存储器主要采用半导体器件作为存储介质,3.2节将重点介绍半导体主存储器。下面首先介绍一下半导体存储器的分类。

1. 按制造工艺分类

(1) 双极型存储器,由TTL晶体管逻辑电路构成。该类存储器件的工作速度快,与CPU处在同一数量级,但集成度低、功耗大、位价格高,常被用作高速缓冲存储器。

(2) 金属氧化物半导体存储器,即MOS型存储器。这类存储器有多种制造工艺,如NMOS、HMOS、CMOS、CHMOS等,可用来制造多种半导体存储器件,如SRAM、DRAM、EPROM等。MOS型存储器的集成度高、功耗低、位价格低,但速度比双极型慢。计算机系统中的主存储器主要由MOS型存储器构成。

2. 按信息的可保存性分类

(1) 电易失型半导体存储器。这类存储器断电后数据会丢失,大多数RAM都属于这种类型。

(2) 非电易失型半导体存储器。这类存储器断电后数据仍保存,各种ROM都属于这种类型,也有少数RAM是非电易失型的。

通常,主存由ROM和RAM两部分组成。ROM中存储的内容只能读不能写,并且断电后信息仍保留,一般用于存放系统程序；RAM可读可写,但断电后信息会丢失,主要用于存储正在运行的用户程序、数据以及系统程序运行中的临时信息。RAM和ROM共享主存的地址空间,ROM占据较小的主存地址空间,而RAM占据了绝大多数的主存地址空间。

RAM存储器按照其工作机理不同又可细分为静态随机存储器、动态随机存储器和非

易失性随机存储器。

1）静态随机存储器(SRAM)

SRAM 的存储电路由 MOS 管触发器构成，用触发器的导通和截止状态来表示信息“0”或“1”。SRAM 的特点是速度快、信息稳定，不需要刷新电路，使用方便灵活。但由于它所用 MOS 管数量较多，其集成度低、功耗较大、成本也高。在计算机系统中，SRAM 常用作小容量的高速缓冲存储器。

2）动态随机存储器(DRAM)

动态随机存储器 DRAM 的存储电路是利用 MOS 管的栅极分布电容的充放电来保存信息。例如，充电后表示“1”，放电后表示“0”。DRAM 的特点是集成度高、功耗低、价格便宜，但由于电容存在漏电现象，电容电荷会因为漏电而逐渐丢失，因此必须定时对 DRAM 进行充电(称为刷新)。在计算机系统中，主存储器主要由 DRAM 构成。

3）非易失性随机存储器(NVRAM)

非易失性随机存储器 NVRAM(Non Volatile RAM)的存储电路由 SRAM 和 EEPROM 共同构成。在正常运行时 NVRAM 与 SRAM 的功能相同，可以随机读写。但在掉电或电源发生故障的瞬间，它可以立即把 SRAM 中的信息保存到 EEPROM 中，使信息得到自动保护。NVRAM 多用于掉电保护来保存存储系统中的重要信息。

随着集成电路技术的不断发展，半导体存储器也得到迅速发展，不断涌现出新型存储器芯片。动态 RAM 有快速页模式 DRAM(Fast Page Mode DRAM，FPM DRAM)、扩充数据输出 RAM(extended data output RAM，EDORAM)、同步 DRAM(Synchronous DRAM，SDRAM)、双倍速率同步 DRAM(Double Data Rate SDRAM，DDR SDRAM)。

3.1.2 存储系统的层次结构

冯·诺依曼计算机是以存储器为中心的结构，存储器的性能是影响计算机系统整体性能的主要因素之一。计算机对存储器的基本要求是速度快、容量大和价格低，但是这三个要求是相互矛盾的。一般来说，存储器的速度越快，位价格就越高；存储器的容量越大，速度就不可能很快。

为了解决容量、速度和价格三者之间的矛盾，通常把各种不同存储容量、不同存取速度和位价格的存储器，按一定的结构组织起来，形成一个多层次的存储系统。图 3-1 描述了这种存储系统的基本框架。

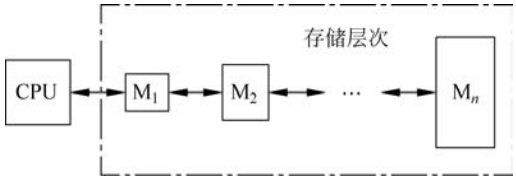


图 3-1 多层次存储系统

图 3-1 所示的多层次存储系统由 n 个不同类型的存储器构成，这些存储器越靠近 CPU，速度越快、容量越小、位价格越高。计算机存储管理的目标就是将这 n 个存储器对 CPU 呈现一个逻辑整体，并且整个存储系统在速度上接近 M_1，容量上接近 M_n，位价格上也

接近 M_n。信息在这种层次化的存储系统中进行存储时需要满足以下两个基本原则。

(1) 包含性原则。指在上层存储器中(靠近 CPU 的层次被称为上层)所存放的内容一定是其下层存储器内容的一部分。也就是说,上层存储器 M_i 存储的信息是其下层存储器 M_{i+1} 部分内容的副本。

(2) 一致性原则。由于同一信息在多个存储器层次中都存有副本,因此当含有该信息的最上层存储器中的内容被修改后,也必须修改该信息保存在其他下层存储器的所有副本,以保持信息的一致性。

从图 3-1 中可以看出,CPU 直接访问的是存储器 M_1,但 M_1 的容量非常小根本无法保证 CPU 需要的全部信息都能找到(称为"命中")。最坏情况下,CPU 需要访问的数据保存在距离 CPU 最远的存储器 M_n 中。在多层次的存储系统中命中率是一个重要的性能指标,命中率常用 H_i 来表示,H_i 表示在存储器 M_i 中找到被访问信息的概率。为了保证 CPU 对 M_1 有足够高的命中率,需要把程序和数据按其使用的急迫性和频繁程度,合理地调入不同层次的存储器中,这个调度过程由计算机硬件和软件自动地统一管理。调度的基本原则是把 CPU 最近一段时间内需要访问的信息存储在距离 CPU 较近的存储器中,而把那些暂时不使用的信息保存在远离 CPU 的存储器中。

很明显,如果命中率太低或调度工作太频繁,层次存储系统也就失去了意义。程序访问的局部性原理保证了命中率能满足 CPU 的基本要求。程序运行的局部性原理主要表现在以下两个方面。

(1) 时间局部性。在一小段时间内,最近被 CPU 访问过的程序和数据很可能再次被访问。例如,对于程序中的循环结构,会被重复执行多次。

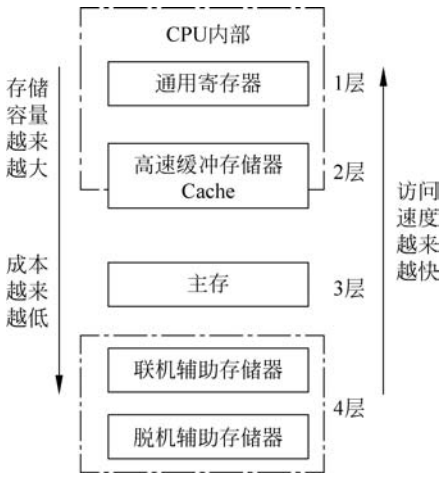


图 3-2　现代存储系统的层次结构

(2) 空间局部性。一段时间内被 CPU 访问的指令和数据往往集中在一小片存储区域内。例如,程序中对数组的操作,数组元素集中存放在存储器的连续单元内。并且,指令大多情况下是按顺序执行的,据统计顺序指令和转移指令在程序中的比率大约为 5∶1。

在现代计算机的多层次存储系统中,存储器的层次结构如图 3-2 所示。

在图 3-2 所示的层次结构中,最上层的通用寄存器组通常被集成在 CPU 芯片内部,保存在寄存器中的数据可以直接参与运算器的操作。最下层的脱机辅助存储器一般为磁带、光盘、U 盘等大容量存储设备,用于保存短时间内不会访问的数据和程序,通常是为了进行备份使用。为了提高主存储器的访问速度和存储容量,存储器的层次结构中最重要的是以下两个层次。

(1) 高速缓冲存储器(Cache)。设计这个层次的目的是提高存储系统的访问速度。Cache 层次使得 CPU 在对主存进行访问时,速度可以接近 Cache 的速度,容量可以达到主存的容量。高速存储器和主存之间的管理任务主要由硬件实现,所以这个层次对于系统程序和用户程序来说都是透明的。随着大规模集成电路技术的发展,Cache 通常被设计为多

级结构，最靠近 CPU 的为 L1 级 Cache，依次增大，绝大多数 Cache 集成在 CPU 芯片内部。

(2) 虚拟存储器(联机辅助存储器)。设计这个层次的目的是提高存储系统的容量。虚拟存储技术使得用户在使用存储系统时，可以获得接近于辅存的容量，而访问速度上接近于主存。虚拟存储器由软件和硬件相结合实现，虚拟存储器需要由系统程序(操作系统)来进行管理，但它对应用程序员是透明的。因本书篇幅有限，将不再展开讨论，读者可以参考《操作系统》相关内容。

多层次存储系统虽然提高了存储器的性能，但是由于数据由原来存放在单一存储器中，变为下层存储器中的部分副本被保存在上层存储器中。这种存储器的层次结构也带来了以下问题需要解决。

(1) 按照多层次存储系统的包含性原则，处于上层存储器的信息一定被包含在各个下层存储器中，即上层存储器中的全部信息是下层存储器中所存部分信息中的副本。那么，下层存储器中的哪些数据的副本是需要保存在上层存储器中的？这些数据又被保存在上层存储器的什么位置，即要解决上层副本所在单元地址与下层地址的对应关系的问题，即地址映射问题。

(2) 按照多层次存储系统的一致性原则，存放在不同存储器中的同一数据，在不同存储器中要保持相同的值。那么如何保证不同层次存储器中存放数据的一致性？特别是在执行写操作时，需要将数据同时写到哪些层次的存储器，即数据一致性问题。

(3) CPU 访存时给出的是主存单元地址，这个地址如何转换为在不同层次存储器中的地址，即地址变换问题。

(4) 当某层存储器已经存满数据，又有新数据要装入时，应将哪些数据替换出去，即替换策略问题。

以上这些问题将在 3.4 节进行详细介绍。

3.2 主存储器

本节将介绍常见的半导体存储器的存储元基本工作原理，存储元构成存储体的方法及主存储器与 CPU 的连接方法等。

3.2.1 主存储器的基本组成

主存由存储单元构成的存储体和一组控制电路组成，如图 3-3 所示。主存主要由以下功能部件组成。

(1) 存储体。存储体也被称为存储矩阵或存储阵列，它是存储单元的集合，用来存储二进制信息。但是，为了提高访问效率，主存与计算机系统中的其他部件进行数据交换(写入和读出操作)时，并不是以二进制位为单位进行的，而是采用并行传输方式。即将若干个存储元组成一个存储单元，存储单元中的所有(或者部分)二进制位作为一个整体被写入或者读出，所以每个存储单元具有独立的地址。通常，存储单元的位数(即存储字长)是字节的整数倍。一个存储器的存储体就是由很多这样的存储单元所构成。

(2) 寻址系统。由驱动器、译码器和 MAR 组成。地址译码器接收到来自 MAR 的 n 位地址后进行译码，经过译码和驱动后形成 2^n 个地址选择信号，每次选中一个存储单元，对所

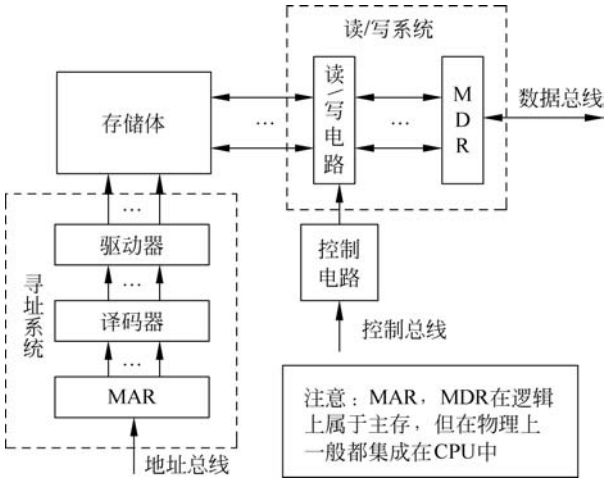


图 3-3　主存的基本结构

选中的存储单元进行读操作或者写操作。每条译码线都需要与它控制的所有存储单元相联,其负载很大,驱动器就是为了提高译码线驱动负载的能力而增加的。地址译码线的输出接到驱动器的输入端,由驱动器的输出端连接译码线所控制的所有存储单元。

(3) 读/写系统。由读写电路和 MDR 组成。MDR 用来缓存 CPU 送来的数据,或从存储单元中读出的数据。读写电路接收到 CPU 的读/写(R/W)控制信号后,产生存储器内部的读写控制信号,将指定地址单元的信息从存储体读出,再送到 MDR 供 CPU 使用,或将来自 CPU 并已经存入 MDR 的信息写入存储体的指定存储单元中。

(4) 控制电路。控制电路根据 CPU 发来的读/写命令,产生存储器各部件的时序控制信号。

下面分别以主存的读操作和写操作为例,示意性地说明主存各个功能部件是如何协同工作的。

在进行读操作时,CPU 首先在地址总线上给出要访问的存储单元的地址;主存的寻址系统对该地址进行译码,并通过驱动器选中指定的存储单元;随后,CPU 在控制总线上发出“读”命令;最后,主存的读/写电路将存储单元中存放的数据发送到数据总线上,并通过数据总线送到 CPU。

在进行写操作时,CPU 首先在地址总线上给出要访问的存储单元的地址;主存的寻址系统对该地址进行译码,并通过驱动器选中指定的存储单元;与此同时,CPU 将需要写入的数据发送到数据总线上;最后,CPU 在控制总线上发出“写”命令,这样通过读写电路就可以将数据总线上的数据写入指定的存储单元中。

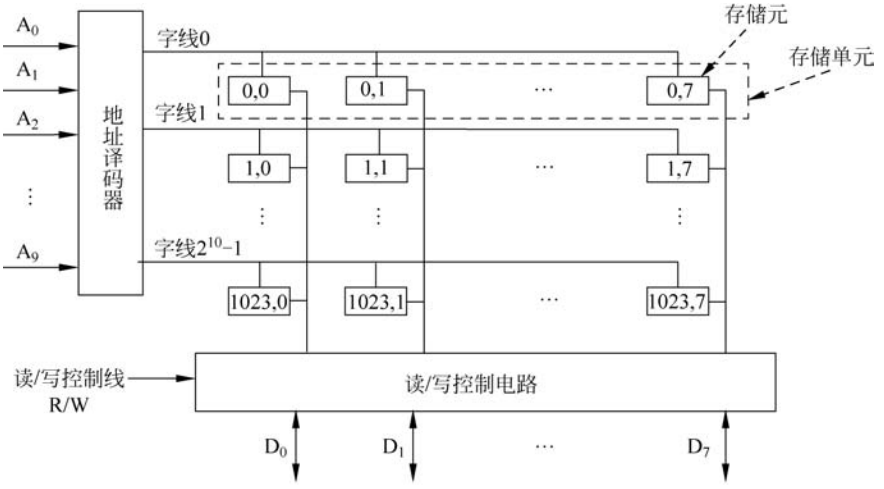
存储器芯片的容量一般使用 $M \times N$ 位的方式来表示。其中,N 表示存储器芯片中每个存储单元的字长(通常被称为片字长),这些存储单元是最小可寻址的单位;M 表示存储单元的个数。一般存储器芯片的片字长为 k 位,k 一般取 1、4、8、16…,每个存储单元由存储体中的 k 位组成。

在主存的寻址系统中,译码器进行地址译码的方式主要有线选法和重合法两种,如图 3-4 所示。

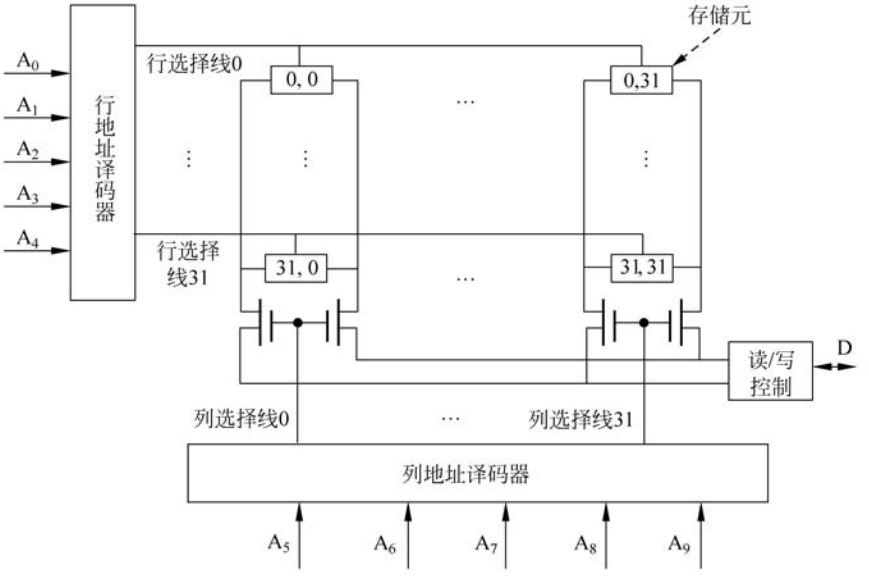
(1) 线选法(单译码结构)。图 3-4(a)给出了一个采用线选法的 1K×8 位的存储器芯片逻辑结构,采用 1024×8 位的存储矩阵。采用线选法译码方式的译码结果将直接选中一个存储单元的所有位。例如,图 3-4(a)中芯片地址信号为 $\log_2{}^{1024}=10$ 位,如果采用线选法,需要完成 10 条地址线到 $2^{10}=1024$ 个存储单元的译码,并且每根译码线需要一套驱动电路才能保证对每个存储单元的驱动,这样就需要 1024 套驱动电路。

这种译码方式的特点是译码结构简单、速度快;但器件用量大,当存储器容量较大时,成本过高。因此,仅适合于高速小容量存储器。

(2) 重合法(双译码结构)。图 3-4(b)给出了一个采用重合法的 1K×1 位的存储器芯片逻辑结构,采用 32×32 位的存储矩阵。重合法译码方式将地址码分成行地址(*X* 向)和



(a) 线选法



(b) 重合法

图 3-4　主存的地址译码方式

列地址(Y向)两组,分别由行地址译码器和列地址译码器进行译码,行、列译码的重合点即为所选中的存储元,这些存储元构成芯片的一个存储单元。例如,图3-4(b)中地址线宽度为10位,如果采用重合法,行、列地址各5位,行、列译码器都只需要完成5条地址线到$2^5=32$个存储单元的译码,每个方向译码器的输出线各需要一套驱动电路,共需要32+32=64套。

与线选法相比,重合法可以减少译码输出线的根数,并且器件用量也相应减少,可以有效降低存储器的成本,因此得到了广泛使用。

3.2.2 主存储器的性能指标

主存的性能主要从容量和速度两个方面进行评价,主要性能指标包括存储容量、存取时间、存储周期和存储器带宽。

1. 存储容量

存储容量是指主存可以容纳的存储单元的总数。存储容量的表示与主存的寻址方式相关。若采用按字编址,则存储容量=存储单元个数×存储字长,单位是位(b);若采用按字节编址,则存储容量等于所存放的字节数,单位是字节(B)。

需要注意的是,若主存按字编址时,存储容量通常书写为两数相乘的形式,即存储单元个数×存储字长(位),表示主存单元数及单元长度。例如,1024×16位,这两个数乘开写成16384位不符合主存操作特点。由于字节长度固定为8位,所以按字节编址时,主存容量可以直接用字节数表示,如1024B。

为了能表示更大的容量,常采用K、M、G、T等单位。下面以用字节表示的存储容量为例,说明各单位的换算关系。1KB=2^{10}B,1MB=2^{20}B,1GB=2^{30}B,1TB=2^{40}B。

2. 存取速度(读/写速度)

主存的存取速度通常用存取时间和存取周期表示。

存取时间又被称为存储器的访问时间(Memory Access Time,MAT),指启动一次存储器读(或写)操作到完成该操作所需要的时间。存取时间按照操作类型可分为读出时间和写入时间。读出时间是指从存储器收到有效地址开始,到产生有效输出所需要的时间;写入时间是指从存储器收到有效地址开始,到数据写入被选中单元所需要的时间。通常,取存储器的读出时间与写入时间相等,故读/写时间被统称为存储器的存取时间。

存取周期(Memory Cycle Time,MCT)是指连续启动两次存储器操作所需的最小时间间隔。通常,存取周期略大于存取时间。

3. 存储器带宽

存储器带宽是指单位时间内存储器存取的信息量,单位为W/s、B/s或b/s。存储器带宽是衡量数据传输率的重要技术指标。其计算公式为:

$$\text{存储器带宽}=\text{每个存取周期可访问位数}/\text{存取周期}$$

存储器带宽决定了以存储器为中心的计算机系统获得信息的速度,它是改善计算机系统瓶颈的一个关键因素。通常采用以下措施来提高存储器的带宽:缩短存取周期、增加存储字长、增加存储体等方式。

此外,还经常采用以下指标来评价存储器的性能。

(1) 可靠性。通常用平均无故障时间(Mean Time Between Failures,MTBF)来衡量主存的可靠性,MTBF表示两次故障之间的平均时间间隔。

（2）功耗与集成度。功耗反映了存储器件耗电多少；集成度标识单个存储芯片的存储容量。一般期望功耗小、集成度高，但两者往往是矛盾的，必须进行折中。

（3）性能价格比。性能主要包括存储容量、存储周期、存取时间和可靠性等。价格包括存储芯片和外围电路的成本。通常希望性能和价格之比越高越好。

3.2.3 SRAM

简单地说，静态随机存储器（SRAM）中“静态”的含义是指只要保持电源供电，SRAM就能稳定地保存数据。构成SRAM的存储元实质上是一个双稳态的触发器。

1. SRAM存储元

存储器中把能够存储一位二进制信息的电路称为存储元，它是构成存储器的基础和核心。SRAM的基本特征是使用一个双稳态触发器作为存储元。加电时，存储元将保持记忆的二进制位，直到再次写入，读出操作不会影响所存储的信息；断电时，存储的信息丢失。图3-5给出了一个由六个MOS晶体管组成的SRAM存储元电路。

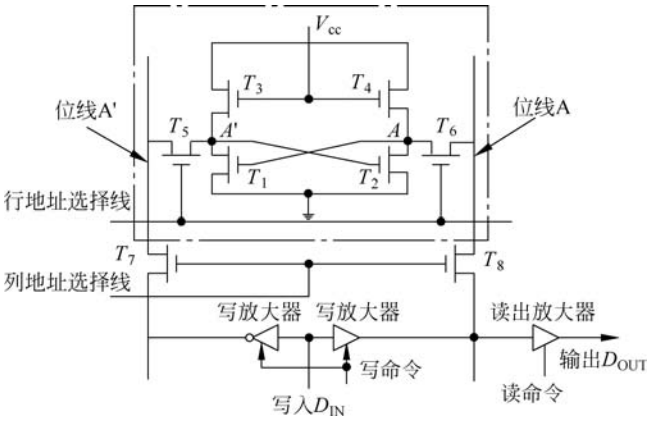


图3-5　六个MOS晶体管组成的SRAM存储元

在图3-5中，T_1、T_2是工作晶体管，T_3、T_4是负载晶体管，T_5、T_6受行地址选择信号控制，这六个MOS晶体管共同构成了一个双稳态触发器，即一个存储元。T_7、T_8受列地址选择信号控制，分别与位线A'和A相连，它们并不包含在基本存储元电路内，而是为芯片内同一列各个存储元所共用。

1）信息保持

当T_1截止时，A'点为高电位，此时T_2导通，而A点的低电位又使得T_1截止，因此这是一个稳定状态。同样，当T_1导通时，A点为高电位，此时T_2截止，而A点的高电位又使得T_1导通，这也是一个稳定状态。这种电路有两种稳定状态，并且A点和A'点电位始终是相反的，因此这个电路可以表示一位二进制的0或1。例如，A点高电位时表示“1”，A点低电位时表示“0”。

2）信息读出

读出时，地址译码后使得被选中存储元的行、列地址选择信号均为有效，这样T_5、T_6、T_7、T_8均导通。假设存储元原来存储的信息是“1”（A点为高电平），A点高电平通过T_6到达位线A，又通过T_8后作为读出放大器的输入信号，在读命令的作用下，将“1”信号读出。

同样,假设存储元原来存储的信息是“0”(A 点为低电平),A 点低电平通过 T_6 到达位线 A,又通过 T_8 后作为读出放大器的输入信号,在读命令的作用下,将“0”信号读出。

3) 信息写入

写入时,地址译码后使得被选中存储元的行、列地址选择信号均为有效,这样 T_5、T_6、T_7、T_8 均导通。不论存储元原来的状态如何,只要将写入代码送至 D_{IN} 数据输入端,在写命令有效时,经两个写放大器,使两端输出为相反电平。这样,A 点和 A' 点电平完全相反,可以将欲写入的信息写到该存储元中。如欲写入“1”,即 $D_{IN}=1$,经两个写放大器使位线 A 为高电平,位线 A' 为低电平,结果使 A 点为高,A' 点为低,即写入了“1”信息。同样的方法可以写入“0”信息。

2. SRAM 芯片举例

图 3-6 给出了赛普拉斯公司的 CY7C1011CV33 SRAM 芯片的逻辑结构,该芯片存储容量为 128K×16 位。数据线 $I/O_0 \sim I/O_{15}$,片使能(Chip Enable)$\overline{CE}$、输出使能(Output Enable)$\overline{OE}$、写使能(Write Enable)$\overline{WE}$ 共同控制数据的输入/输出。由于单个存储器芯片通常在容量、字长等指标上达不到主存要求,因此主存一般由若干个存储器芯片共同组成,$\overline{CE}$ 信号完成片选功能,即本次存储器访存操作是否涉及它。另外,由于此芯片字长为 2 字节(16 位),而 CPU 在访问存储器时,有可能只需要访问其中 1 个字节,所以芯片设置了高字节使能(Byte High Enable,BHE)和低字节使能(Byte Low Enable,BLE)信号,分别用于访问 16 位数据中的高字节和低字节。

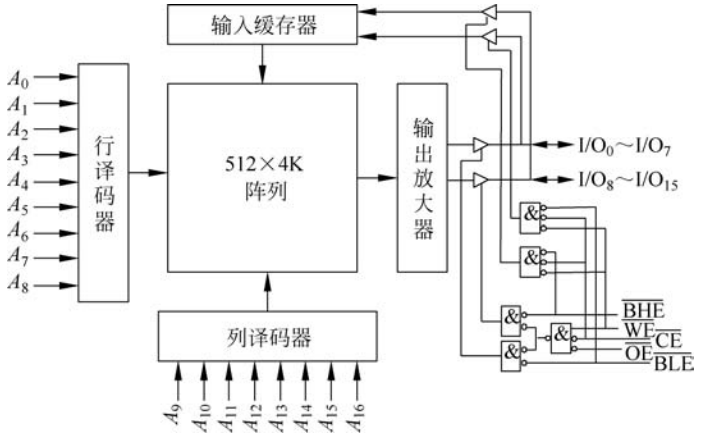


图 3-6 CY7C1011CV33 SRAM 芯片逻辑结构

当 $\overline{CE}$ 和 $\overline{WE}$ 均为低电平时,$\overline{BHE}$ 和 $\overline{BLE}$ 分别控制高低字节的输入三态门打开,来自数据线的数据被写入存储器。当 $\overline{CE}$ 和 $\overline{OE}$ 均为低电平时,$\overline{BHE}$ 和 $\overline{BLE}$ 分别控制高低字节的输出三态门打开,存储器中的数据输出到数据线。

描述 SRAM 芯片的外特性常用逻辑符号框图,在框图的内部标明芯片的型号、容量、引端名等,在框图的外部标明芯片的引脚线、信号名等。引端名是芯片引脚的逻辑定义,由厂家给出;信号名为引脚上所加信号的名称,两者不一定一样。引脚线包括电气引脚和逻辑引脚,例如电源线和地线这样的电气引脚一般在框图中被省略,逻辑引脚包括数据线、地址线和控制线。图 3-7 给出了 CY7C1011CV33 SRAM 芯片的逻辑符号图和实际引脚图。

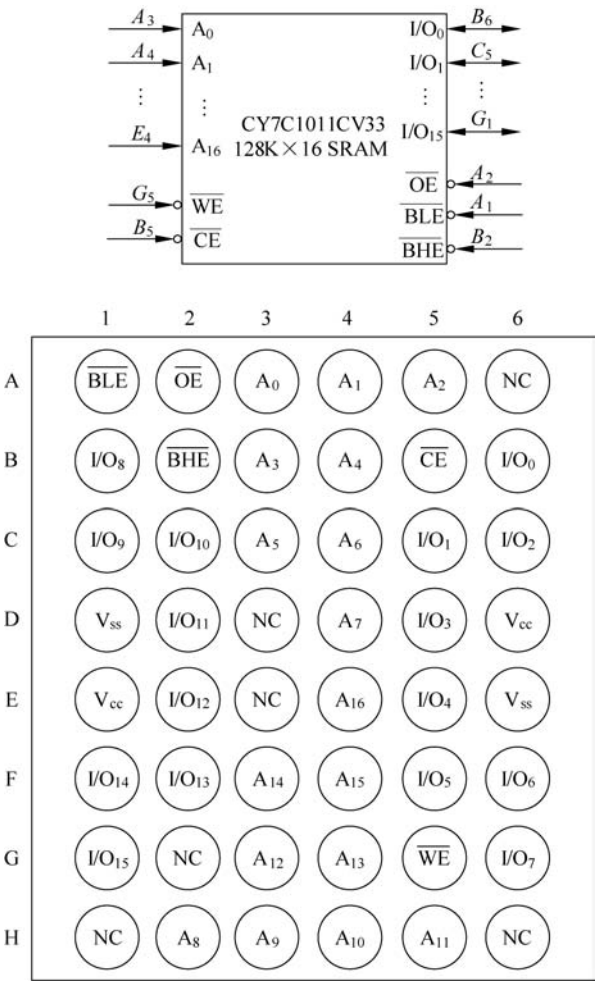


图 3-7　CY7C1011CV33 SRAM 芯片逻辑符号图和实际引脚图

3. SRAM 的读/写时序

将某存储器芯片用于一个计算机系统时，存储器在读写过程中各类信号之间的时序关系非常重要，它决定该存储器芯片是否可以满足计算机系统对时间性的要求，以及与其他部件间的配合关系。对于确定的存储器芯片来说，其读写周期的时序关系是确定的。通常可以查阅芯片技术手册获得详细且精确的时序图。这里通过 SRAM 芯片实例，希望读者理解其读写过程中的基本时间配合关系。

图 3-8 示意性地给出了 CY7C1011CV33 SRAM 的读周期和写周期时序图。其中，D_{OUT} 和 D_{IN} 分别表示数据线 $I/O_0 \sim I/O_{15}$ 上的读出数据和写入数据。

图 3-8(a)给出了 CY7C1011CV33 读周期时序。下面按照地址线、控制线和数据线三组信号线有效的先后顺序说明其读过程。

首先，CPU 在地址总线上发出有效地址，并且通过 $\overline{CE}$ 选中该芯片(有关 $\overline{CE}$ 信号的产生方法参见 3.2.7 节)，此时存储器通过译码电路进行地址译码，并选中相应存储单元；经过一段时间后，CPU 发出 $\overline{OE}$ 信号，依据 $\overline{BHE}$、$\overline{BLE}$ 信号是否有效，经过一段延迟后输出相

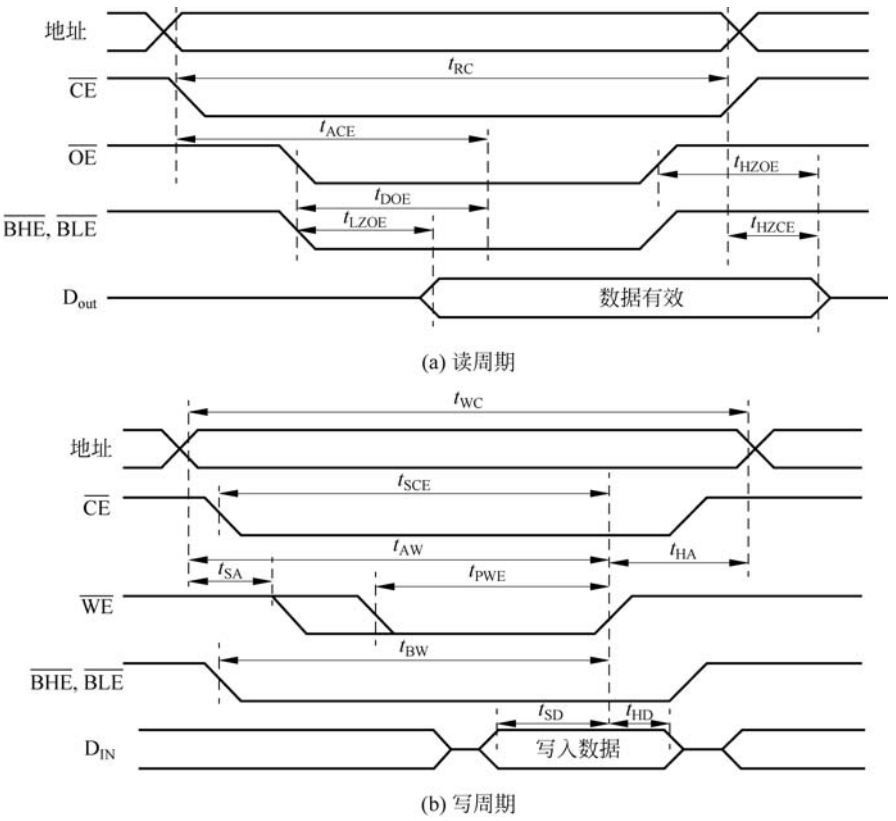


图 3-8　CY7C1011CV33 的读/写周期时序

应的高低字节。

读出过程中存在两个约束：第一，从 $\overline{CE}$ 有效到数据线上输出有效数据的最长时间为 t_{ACE}；第二，从 $\overline{OE}$ 有效到数据线上输出有效的数据的最长时间为 t_{DOE}。从表 3-1 看，t_{ACE} 最小值为 10ns，因此，在应用此存储芯片时，若 CPU 发出 $\overline{CE}$ 到读取数据之间的间隔小于 10ns，则可能无法读取到有效数据。CPU 在读取到有效数据后，要撤销 $\overline{CE}$、$\overline{OE}$ 信号。

撤销过程中也存在两个约束：第一，从撤销 $\overline{CE}$ 到存储芯片在数据线上撤销有效数据的时间为 t_{HZCE}；第二，从撤销 $\overline{OE}$ 到存储芯片在数据线上撤销有效数据的时间为 t_{HZOE}。在数据撤销前这段时间内，数据线仍然保持着上次读出的数据，即可以认为数据线被“占用”。从 $\overline{CE}$ 有效到 $\overline{CE}$ 撤销这段时间为存储芯片的读出时间 t_{RC}，t_{HZCE} 是存储器的恢复时间。

在具体实现访存逻辑中，只要不违反以上述约束，均可以实现正确的读操作。比如，可以使 $\overline{CE}$ 和 $\overline{OE}$ 同时有效，然后在 t_{ACE} 时间之后读取数据。

图 3-8(b)给出了 CY7C1011CV33 的写入周期时序，下面同样按照地址线、控制线和数据线三组信号线有效的时序说明其写入过程。

首先，CPU 在地址线上发出有效地址，然后发出片使能信号 $\overline{CE}$ 和写命令 $\overline{WE}$ 进行存储器写操作。要求写命令 $\overline{WE}$ 比有效地址信号晚最少 t_{SA} 时间。为保证写入成功，$\overline{WE}$、$\overline{CE}$ 和地址信号的最小保持时间分别为 t_{PWE}、t_{SCE} 和 t_{AW}。写入数据要在 $\overline{WE}$ 失效前 t_{SD} 时间出现在数据线上，并在 $\overline{WE}$ 失效后保持一段时间 t_{HD}，以保证数据的可靠写入。在 $\overline{WE}$ 失

效后，有效地址还需要保持 t_{HA} 时间。在写操作过程中，从本次地址有效到下次地址有效的时间间隔被定义为写周期时间，在图 3-8(b)中记为 t_{WC}。

表 3-1 列出了 CY7C1011CV33 读周期和写周期的主要时间参数及其指标所示，表中每项时间参数指标只有最长或者最短的时间。

表 3-1　CY7C1011CV33 读/写周期时间参数

参数	描　　述	最小值	最小值	单位
读周期				
t_{RC}	读周期时间	10	—	ns
t_{ACE}	从 $\overline{CE}$ 有效到读出数据有效时间	—	10	ns
t_{DOE}	从 $\overline{OE}$ 有效到读出数据稳定时间	—	6	ns
t_{LZOE}	从 $\overline{OE}$ 有效到读出数据有效时间	0	—	ns
t_{HZOE}	从 $\overline{OE}$ 无效到读出数据无效时间	—	5	ns
t_{HZCE}	从地址无效读出数据无效时间	—	5	ns
写周期				
t_{WC}	写周期时间	10	—	ns
t_{SCE}	从 $\overline{CE}$ 有效到写数据结束时间	7	—	ns
t_{AW}	从地址有效到写数据结束时间	7	—	ns
t_{HA}	写数据结束后地址保持时间	0	—	ns
t_{SA}	从地址有效到写开始时间	0	—	ns
t_{PWE}	$\overline{WE}$ 脉冲宽度	7	—	ns
t_{SD}	从数据有效到写数据结束时间	5	—	ns
t_{HD}	写数据结束后数据保持时间	0	—	ns
t_{BW}	从字节使能到写数据结束时间	7	—	ns

3.2.4　DRAM

1. DRAM 存储元

DRAM 存储元是由 MOS 晶体管和电容组成的基本电路，常见的有三管式和单管式两种，如图 3-9 所示。无论单管式 DRAM 还是三管式 DRAM 存储元都是通过电容存储电荷来记录二进制信息的“0”和“1”。例如，通常假设电容存有足够多的电荷时表示“1”，而无电荷时表示“0”。但是由于漏电的存在，电容上的电荷通常只能维持一段时间，即保存在电容上的电荷会慢慢地泄漏，为此必须定时为电容补充电荷，这个过程通常称作为刷新。这也是 DRAM 被称为动态存储器的原因。

图 3-9 给出了单管式和三管式 DRAM 存储元的基本电路。

图 3-9(a)所示的三管式 DRAM 存储元中 T_1、T_2 和 T_3 为工作管，T_4 为预充电管。

三管式 DRAM 存储元在进行读操作时，首先对 T_4 置充电信号，使得读数据线为高电平 V_{DD}；然后由读选择线高电平使 T_2 导通。如果存储元存储的是“1”，即 T_1 的极间电容 C_g 有足够电荷，则 T_1 导通。由于 T_1 和 T_2 导通接地，使得读数据线为低电平(读出“0”)；如果存储元存储的是“0”，即 C_g 的极间电容无电荷，则 T_1 不导通，读数据线维持高电平(读

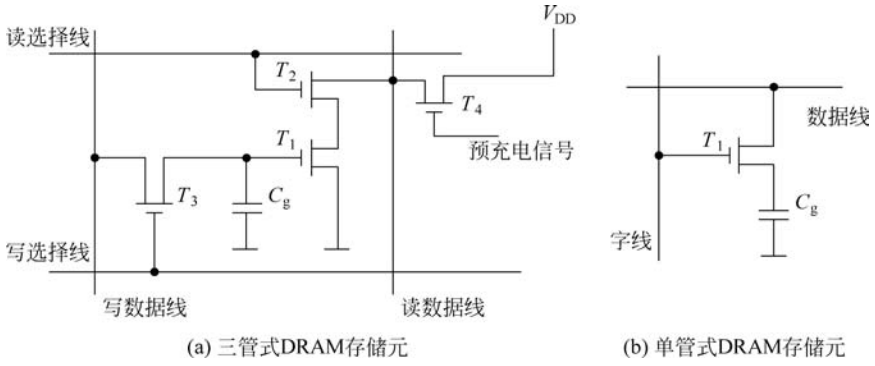


图 3-9　三管式及单管式 DRAM 存储元

出“1”)。可看出,读出数据与原存信息反向,因此还需要增加反向电路来保证读出数据与存储数据一致。

三管式 DRAM 存储元在进行写操作时,首先在写数据线上加上要写入的数据信号,然后在写选择线上加高电位,这样 T_3 导通,C_g 随写入信号进行充电或者放电操作,从而达到写入“1”或者“0”的目的。

为了提高系统的集成度,将三管式存储元电路简化成如图 3-9(b)所示的单管电路。读出时,字线上的高电平使 T_1 导通,若 C_g 有电荷,经 T_1 晶体管在数据线上产生电流,视为读出“1”。若 C_g 上无电荷,则数据线上无电流,视为读出“0”。

单管式 DRAM 存储元在执行写操作时,字线为高电平,T_1 导通。若数据线为高电平(写入“1”),则对 C_g 进行充电操作;若数据线为低电平(写入“0”),则对 C_g 进行放电操作。由此达到写入“1”和“0”的目的。

对 DRAM 存储器来说,由于在执行读操作后,被读单元的内容会被清“0”,因此读出操作是一种破坏性读出,在读出后必须进行再生操作。所谓再生,简单地说就是把刚读出的内容立即写回去。这要求读出放大器必须同时是再生放大器,一般利用双稳态结构,在读出过程中建立起稳态,然后该稳态再自动写回存储元。DRAM 存储器的这种特性,也影响了 DRAM 存储器的工作效率。

2. DRAM 刷新

DRAM 存储元中的电容会因为电荷泄漏而引起 DRAM 所存信息的衰减,如果不及时进行补充,就会造成存储元的信息丢失。因此,必须定期对电容补充电荷,这种充电操作被称为刷新操作。把 DRAM 存储器能维持信息的最长时间称为最大刷新间隔,一般为 2ms、8ms、64ms 等。

由于在进行读操作时,DRAM 会进行再生操作,因此刷新就可以采用“读”的方法进行,即对存储元进行读操作,通过读操作时进行的再生操作对电容进行重新充电。需要注意的是,此时数据并不输出到数据线上。一次刷新操作所需要的时间相当于一次读操作时间,称为刷新周期。为了提高刷新效率,刷新操作通常是按存储矩阵的行进行的,刷新时只需送出行地址和刷新信号,这样同一行的所有存储元都被选中进行刷新操作。刷新时的行地址不需要外部提供,DRAM 芯片的内部有一个刷新计数器(也被称为行地址生成器),用于产生刷新时所需要的行地址。

为了保证 DRAM 存储器的正常工作,必须在其最大刷新间隔内对所有存储元安排一次刷新。刷新是按行进行的,安排存储体中各行的刷新时间的策略,被称作刷新方式。常见

的刷新方式有集中式、分散式和异步式三种。

1）集中式刷新

集中式刷新是指将全部存储单元的刷新操作集中在一段时间内完成。这样，在整个最大刷新时间间隔中，前一部分用于正常的存储器读、写和保持操作，后一部分用于对所有存储单元的刷新操作。在进行集中刷新操作时，由于对存储体逐行进行刷新，因此不能进行正常的存储器读、写操作，故被称为访问存储器的“死时间”，或访存“死区”。由于“死时间”的存在，极大地影响了系统的执行效率，目前很少使用这种刷新方式。

假设 64K×1DRAM 芯片，若存储体为 256×256 矩阵，最大刷新间隔为 2ms，存取周期为 0.5μs。2ms 内共包含 4000 个存取周期，对 256 行集中刷新共需 256 个存取周期（128μs），其余的 3744 个存取周期用来进行存储器的正常读、写或保持操作。

2）分散式刷新

分散式刷新是指将刷新操作分散到每个存取周期内完成，即每个存取周期的前半段用于读、写或保持信息，后半段用于刷新操作。分散式刷新虽然不存在“死时间”，但是由于将每个存取周期都扩大一倍，存在时间上的浪费，因此也很少采用。

上述 64K×1DRAM 芯片，采用分散式刷新时，相当于将存取周期由 0.5μs 增加到 1μs，在最大刷新间隔 2ms 内共安排了 2000 次刷新。

3）异步式刷新

异步式刷新是前两种方式的结合，它将所有行的刷新操作平均分配在最大刷新间隔时间内，使得在一个最大刷新间隔内，每一行会且只会被刷新一次。这样既可克服了“死时间”问题，又充分利用了最大刷新间隔。

上述 64K×1DRAM 芯片，采用异步式刷新时，每隔 2000μs÷256＝7.8μs 刷新一行。

除了以上三种方法外，还有一些其他策略来安排刷新的时机。例如，将 DRAM 的刷新安排在 CPU 对指令的译码阶段，由于这个阶段 CPU 不访问存储器，所以这种方案既不会加长存取周期，也不会出现“死时间”，从而可以提高系统的工作效率。

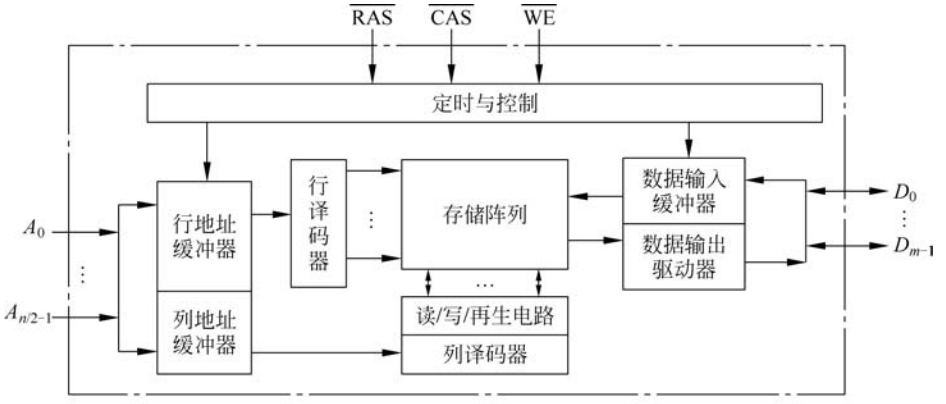
3. DRAM 存储器芯片举例

由于 DRAM 芯片集成度高，所以容量一般比较大，这导致了地址引脚数的大幅度增加，这对芯片的集成又带来了困难。为此，DRAM 芯片通常将地址分为行地址和列地址两部分，行地址和列地址分时使用同一组地址引脚，这样可以将地址引脚的数量减少为原来的一半。为了保证行、列地址的正确输入，又引入了行地址选通信号（$\overline{\text{RAS}}$）和列地址选通信号（$\overline{\text{CAS}}$）。这还意味着 DRAM 芯片每增加一根地址引脚，相当于行、列地址各增加一位（共增加了两位地址），使片容量扩大 4 倍。

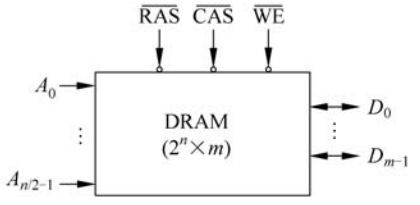
同时，为了进一步减少引脚的数量，DRAM 芯片通常还省略了 $\overline{\text{CE}}$ 引脚，将片选功能由 $\overline{\text{RAS}}$ 引脚兼任，即 $\overline{\text{RAS}}$ 信号与片选译码信号复合使用。

图 3-10(a)给出了一个典型的 DRAM 存储器的内部结构，图 3-10(b)则给出了一个 DRAM 存储器芯片的逻辑符号图。在进行芯片封装时，除了需要行列选择信号 $\overline{\text{RAS}}$ 和 $\overline{\text{CAS}}$ 外，写允许信号 $\overline{\text{WE}}$ 用来说明是写操作还是读操作。

下面以 KM44C16000B 芯片为例说明 DRAM 芯片的内部结构，如图 3-11 所示。KM44C16000B 是 16M×4 位的 DRAM 芯片，存储阵列为 8K(行)×8K(列)，采用重合法译码结构，行地址 13 位、列地址 11 位。该芯片的存取时间为 45～60ns，工作时最大功耗为



(a) DRAM芯片的内部结构



(b) DRAM芯片的逻辑符号图

图 3-10　DRAM 存储器芯片及内部结构与逻辑符号图

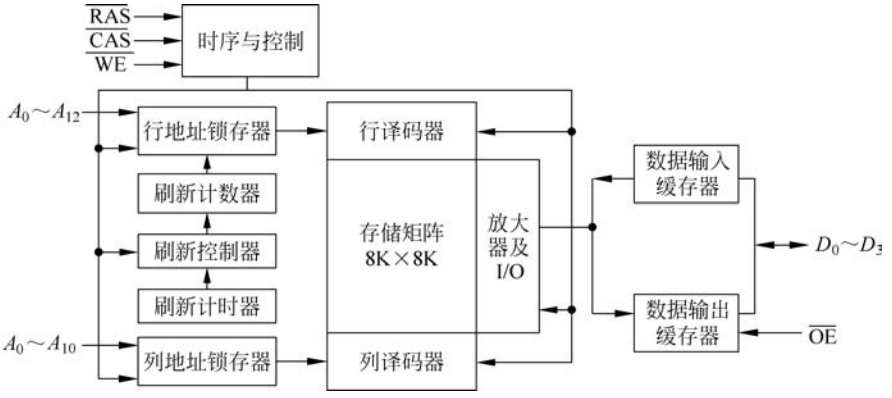


图 3-11　KM44C16000B DRAM 芯片的逻辑结构

550mW,最大刷新间隔为 64ms。图中刷新计时器、刷新控制器、刷新计数器等逻辑部件是为了实现芯片内部自刷新功能设置的,具体原理参见 3.2.7 节 DRAM 控制器部分。

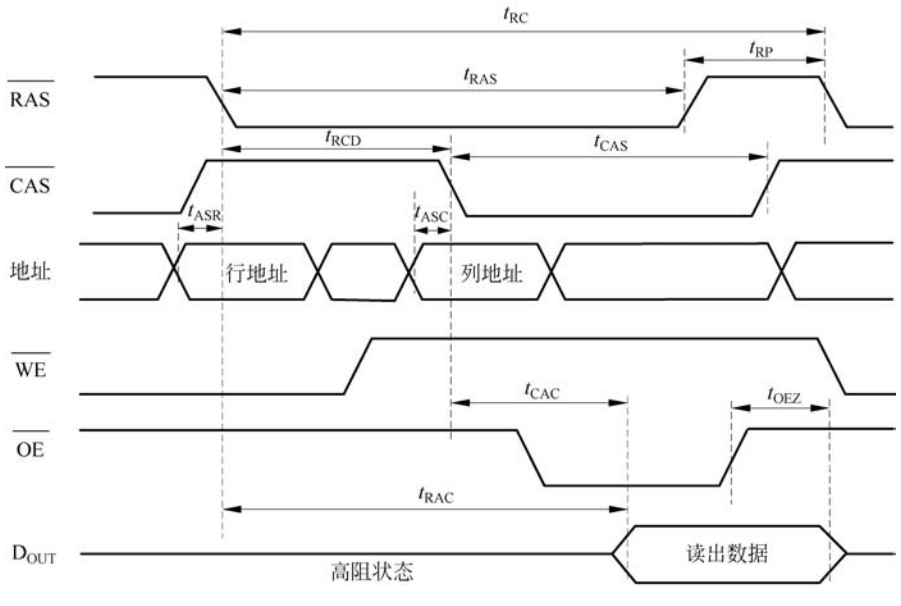
4. DRAM 的读写时序

下面以 KM44C16000B 为例说明 DRAM 的读、写和刷新周期时序。

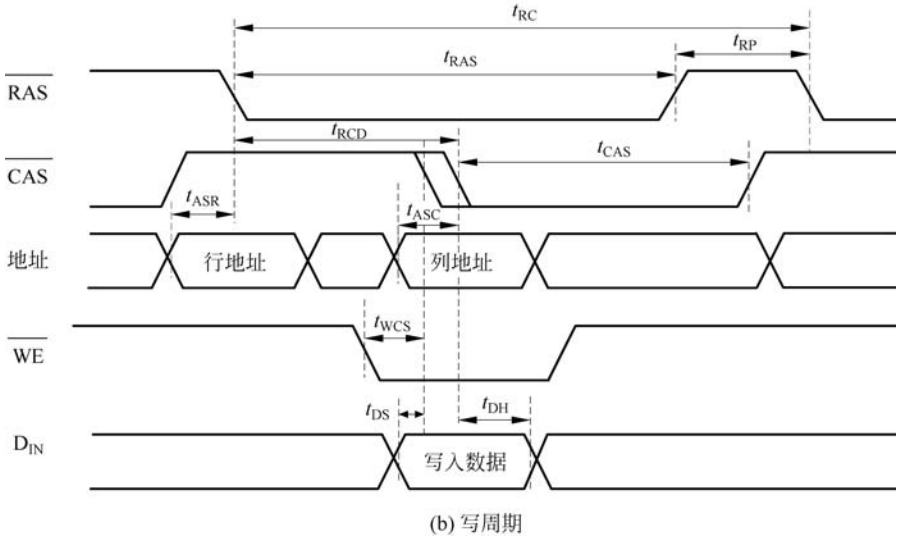
1) 读周期

图 3-12(a)给出了 KM44C16000B 读周期时序。为了有效锁存行、列地址,行地址应先于列地址有效,并保持一段时间。首先在 $A_{12} \sim A_0$ 引脚上提供行地址,在 t_{ASR} 时间后行地址选通信号 $\overline{RAS}$ 有效,行地址被锁存。然后在 $A_{10} \sim A_0$ 引脚上提供列地址,在 t_{ASC} 时间后列地址选通信号 $\overline{CAS}$ 有效,列地址被锁存。此时,行地址和列地址共同选中一个存储单元。输出使能信号 $\overline{OE}$ 有效,表示 CPU 开始读操作。在 $\overline{CAS}$ 有效 t_{CAC} 时间后,数据线 D_{OUT} 开

始输出数据。如果从读周期开始($\overline{\text{RAS}}$有效)计算,数据在t_{RAC}后输出。为了保证数据可靠读出,$\overline{\text{RAS}}$有效的时间t_{RAS}应满足一定的宽度。$\overline{\text{OE}}$无效后,数据还可能保持t_{OEZ}时间。t_{RC}两次连续存储器操作的间隔时间,即读/写周期。



(a) 读周期



(b) 写周期

图 3-12　KM44C16000B 的读/写周期时序

2) 写周期

图 3-12(b)给出了 KM44C16000B 的写周期时序。与读周期类似,首先在$A_{12}\sim A_0$引脚上提供行地址,并在t_{ASR}时间后行地址选通信号$\overline{\text{RAS}}$有效。然后,写信号$\overline{\text{WE}}$应在$\overline{\text{CAS}}$有效前t_{WCS}时间有效。随后,在$A_{10}\sim A_0$引脚上提供列地址,在t_{ASC}时间后列地址选通信号$\overline{\text{CAS}}$有效。写入数据信号D_{IN}在$\overline{\text{CAS}}$有效的下降沿被写入。要求写入数据信号应在

$\overline{CAS}$ 有效 t_{DS} 前有效,且在 $\overline{CAS}$ 有效后维持 t_{DH}。同样,为了保证数据可靠写入,$\overline{RAS}$ 有效的时间 t_{RAS} 应满足一定的宽度。

3) 刷新周期

KM44C16000B 提供了三种刷新方式,图 3-13 给出采用唯 $\overline{RAS}$ 有效时刷新周期时序,这是一种外部强制刷新方式。在进行刷新操作时,芯片只接收从地址总线上发来的行地址,由行地址在存储矩阵中选中一行所有存储元,将其中所保存的信息输出到读出放大器,经放大后再写回到原存储元,实现存储元的刷新操作。

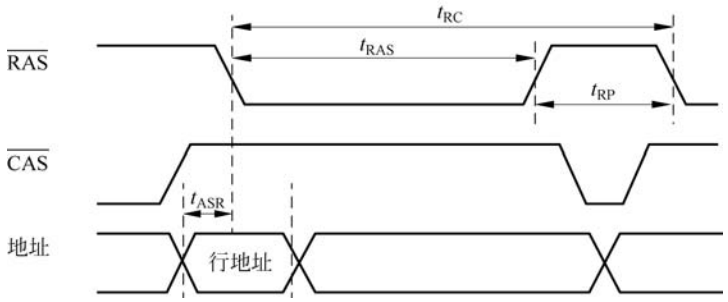


图 3-13 KM44C16000B 刷新周期时序

表 3-2 列出了 KM44C16000B 读周期、写周期和刷新周期的主要时间参数及其指标。

表 3-2 KM44C16000B 的读/写周期时序参数

参数	描 述	最小值	最小值	单位
t_{RC}	读/写周期时间	90	—	ns
t_{RAC}	从 $\overline{RAS}$ 有效到数据有效时间	—	50	ns
t_{CAC}	从 $\overline{CAS}$ 有效到数据有效时间	—	13	ns
t_{RP}	$\overline{RAS}$ 预充电时间	30	—	ns
t_{RAS}	$\overline{RAS}$ 脉冲宽度	50	10K	ns
t_{CAS}	$\overline{CAS}$ 脉冲宽度	13	10K	ns
t_{RCD}	从 $\overline{RAS}$ 有效到 $\overline{CAS}$ 有效的延迟时间	20	37	ns
t_{ASR}	行地址建立时间	0	—	ns
t_{ASC}	列地址建立时间	0	—	ns
t_{OEZ}	$\overline{OE}$ 无效后读出数据保持时间	0	13	ns
t_{DS}	数据建立时间	0	—	ns
t_{DH}	数据保持时间	10	—	ns
t_{WCS}	写命令建立时间	0	—	ns

3.2.5 新型 DRAM 存储器

目前,DRAM 存储器的应用比 SRAM 要广泛得多。其原因如下:

(1) 在同样大小的芯片中,DRAM 的集成度远高于 SRAM。例如,DRAM 的基本存储元电路为一个 MOS 晶体管,一般 SRAM 的基本存储元电路为 4～6 个 MOS 晶体管。

(2) DRAM 存储器芯片的行、列地址引脚分时复用,不仅减少了芯片引脚数量,封装尺寸也减少了。

(3) DRAM 芯片的功耗比 SRAM 小。

(4) 相同容量的 DRAM 芯片价格比 SRAM 便宜。当采用同一档次的实现技术时，DRAM 的容量是 SRAM 容量的 4～8 倍，SRAM 的存取周期比 DRAM 的存取周期快 8～16 倍，但价格也贵 8～16 倍。

随着 DRAM 容量不断扩大，速度不断提高，计算机中主存储器主要采用 DRAM 芯片。但是，DRAM 也有以下主要缺点：

(1) 由于使用动态元件(电容)，因此它的速度比 SRAM 低。

(2) 由于再生和刷新需配置再生电路，功率消耗比 SRAM 高。

近些年，出现了很多新型的 DRAM 存储器，这些存储器主要通过提高时钟频率和带宽等方法来缩短存储周期、提高存储速度。下面介绍几种常见的 DRAM 技术。

1. FPM DRAM(FAST PAGE MODE DRAM，快速页面模式 DRAM)

从前面的介绍我们知道，CPU 对传统 DRAM 中一个存储单元的访问，必须送出行地址和列地址各一次才能完成。FRM DRAM 以多个字节为单位进行突发式数据读写，但要求多字节必须位于同一行(也称为同一页)，这种访问方式被称为快速页面模式。即如果 CPU 需要访问若干存储单元，且它们的地址属于同一行，则在第一次输出行地址后连续输出列地址，而不必多次输出行地址。FPM DRAM 读操作基本时序如图 3-14 所示。

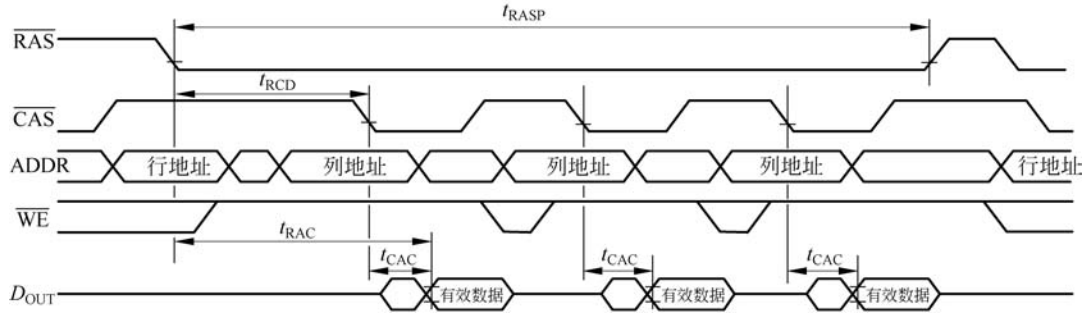


图 3-14　FPM DRAM 读操作时序

FPM DRAM 实现技术基于程序的局部性原理。由于大多数程序按照指令在主存中存放的顺序执行，并且数据在主存中的地址往往也是连续的。输出行地址后连续输出列地址就能得到所需要的后续指令或数据。因此，FPM DRAM 能提高主存的访问效率。

2. EDO DRAM(Extended Data Out DRAM，扩展数据输出 DRAM)

CPU 对 FPM DRAM 存储器访问时，上一次数据读取过程整体完毕后才能进行下一个数据的读取。EDO 存取模式与 FPM 基本类似，其主要区别在于存储器在数据线输出数据的同时，CPU 可以同时在地址线发送下次访问的地址，从而缩短列地址的间隔时间。即当前数据还有效时，下一个数据的列地址已经送到了地址总线上。这种按照流水线方式进行读操作的能力，可以提供比 FPM DRAM 更快的数据流，从而提高计算机工作效率。EDO DRAM 读操作基本时序如图 3-15 所示。

3. SDRAM(Synchronous DRAM，同步 DRAM)

传统的 DRAM 与 CPU 之间采用异步方式交换数据。“异步”是指由于存储器的速度较慢，存储器与 CPU 不采用同一个时钟信号进行同步。CPU 发出地址及控制信号后，需要

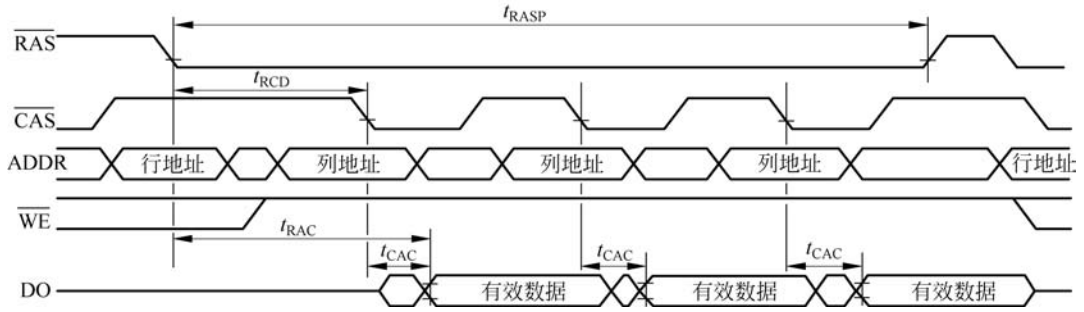


图 3-15　EDO DRAM 读操作时序

等待约定的延迟时间来确定存储器完成了读/写操作。在这等待时间内 CPU 处于空闲状态,效率较低。随着存储器技术的发展,出现了存储器操作与 CPU 时钟信号同步的 DRAM 存储器,即 SDRAM。“同步”是指存储器的操作时序与系统时钟相配合,由系统时钟来控制存储器的读写操作。SDRAM 在时钟上升沿接收读写命令和地址以及读写数据,即所有存储器操作均由时钟上升沿同步。

图 3-16 所示为 SDRAM 读写时序。CPU 在第 1 个时钟上升沿和第 4 个时钟上升沿分别发送行地址和列地址,经过两个时钟的延迟后,在第 6 个时钟上升沿就可以读取数据。其写数据过程类似。在时钟频率确定的情况下,CPU 访问 SDRAM 时只需要对时钟进行计数,而不需要精确控制每步操作的时间长度,可以极大地简化硬件设计。并且,CPU 在发送行地址和列地址后,可以读取或者写入连续 4 个单元的数据。这种访问方式称为猝发式(Burst)读/写,可以有效地提高 CPU 访存效率。

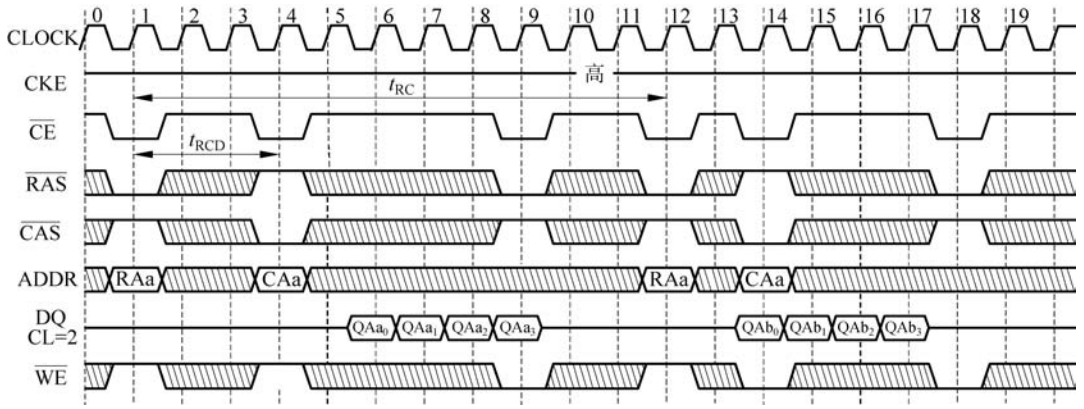


图 3-16　SDRAM 读/写时序

由于 DDR SDRAM(Double Data Rate SDRAM,双数据传输率同步 DRAM,简称 DDR)的出现,早期的 SDRAM 又被称为单数据速率 SDRAM。DDR 存储器中存储体的位宽是存储器外部位宽的 2 倍,称为 2 倍预存取。采用更先进的同步电路,使得存储器操作既由时钟上升沿同步,也由时钟下降沿同步。因此,DDR 本质上不需要提高存储器内部时钟频率,就能加倍提高 SDRAM 的访问速度。

图 3-17 为 DDR MT46V128M8 存储器芯片的内部逻辑结构。该芯片存储器容量为 128M×8 位,存储体由 4 个模块构成。为实现 2 倍预存取能力,每个模块位宽为 16 位。

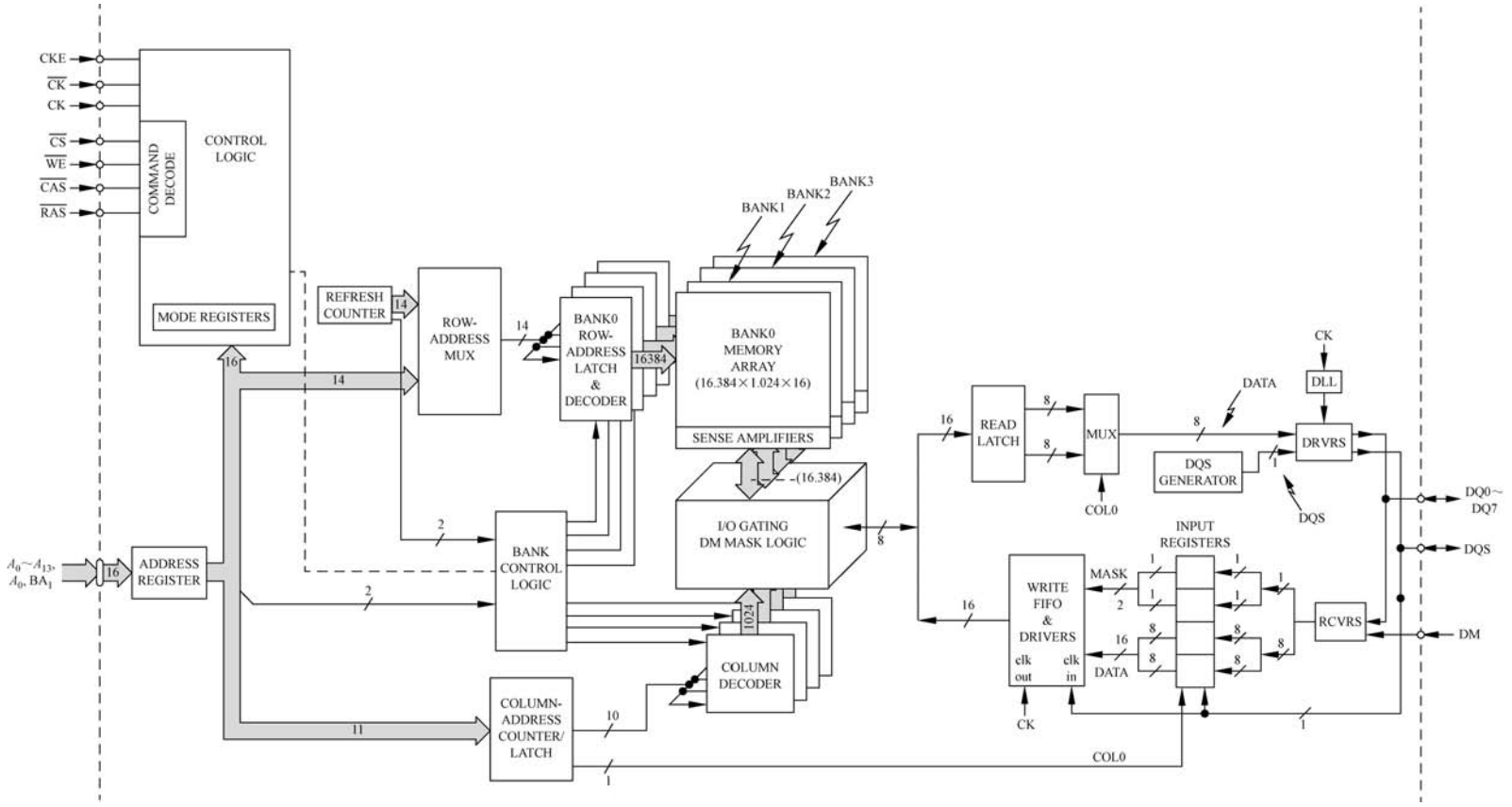


图 3-17　DDR SDRAM 内部逻辑结构

DDR2(Double Data Rate 2)SDRAM 与 DDR 同样由时钟上升/下降沿同步。但 DDR2 存储器却拥有两倍于 DDR 的预读写能力(4 倍)。

DDR3(Double Data Rate 3) SDRAM 在 DDR2 的基础上采用 8 倍预读写设计,以提供更高的外部数据传输率。

当前,主流的服务器、PC 以及手机等计算机主要使用 DDR3,并已经开始使用 DDR4。与 DDR3 比较,DDR4 采用了 16 倍预读写设计,工作电压更低,能耗更低。

3.2.6 只读存储器和闪速存储器

1. 只读存储器

顾名思义,只读存储器(Read-Only Memory,ROM)正常工作时只能读出不能写入。ROM 与前面介绍的 DRAM 和 SRAM 相比最大的特点是断电后存储的信息不会消失,即具有非电易失性。因此,常用于保存系统程序以及关键数据等,例如计算机启动时所使用的 BIOS 芯片。ROM 的主要用途就是实现软件的固化,相对于软件来说,把写到 ROM 中的程序及芯片称为固件。随着制造工艺的发展,又出现了 PROM、EPROM、EEPROM、闪存等可以单次或者多次写入的 ROM 芯片。

1) MROM

MROM(Mask ROM,掩膜式 ROM)存储的信息是由生产厂家根据用户的要求,在生产过程中采用掩膜工艺(即光刻图形技术)一次性直接写入的。掩膜式 ROM 的基本存储原理是用元件的“有”和“无”来表示二进制信息“0”和“1”。掩膜 ROM 一旦制成后,其内容不能再改写,因此它只适合于存储永久性保存的程序和数据。图 3-18 给出了一个 MOS 晶体管型 MROM 的逻辑结构。其存储容量为 16×1 位,4 行保存的信息分别为 1001、1010、0101 和 0101,即有晶体管的存储元的值为“1”,否则为“0”。很明显,出厂后 MROM 单元的值不能修改。

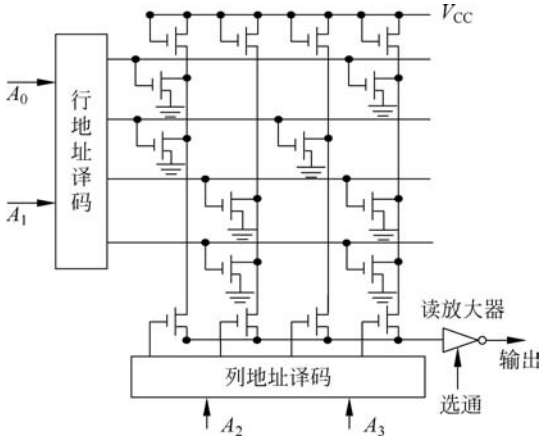


图 3-18 MOS 管型 MROM

2) PROM

PROM(Programmable ROM,可编程 ROM)芯片出厂后用户可以使用专门的 PROM 写入器进行写入操作,但只能写入一次。因此被称为一次编程型 ROM。

PROM 有两种工艺：熔丝式和反向二极管式。熔丝式 PROM 根据存储单元中熔丝的断开与接通来表示“1”和“0”。出厂时所有熔丝接通，表示内容全部为“0”，当需要写入“1”时，通过大电流将熔丝烧断。反向二极管型 PROM 根据存储单元中二极管的导通和断开来表示“1”和“0”。出厂时在每个存储位的行和列交点处有一个正向和一个反向二极管，由于反向二极管不导通，表示内容全部为“0”。当需要写入“1”时，在相应行和列加较高电压，将反向二极管永久击穿，由于行列交点只剩下一个正向二极管而导通，表示写入“1”。由于这两种 PROM 都是破坏性写入，因此写入后不能进行擦除和再次写入，即 PROM 只能写入一次。图 3-19 给出了一个双极型熔丝式 PROM 的单元电路。

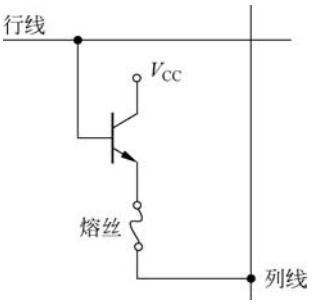


图 3-19　双极型熔丝式 PROM 的单元电路

3) EPROM

EPROM(Erasable Programmable ROM，擦除可编程 ROM)是一种可多次写入的 ROM。对 EPROM 写入信息要使用专用编程器来完成，它能进行多次擦除和再写入操作。图 3-20 给出了一个 N 型沟道浮动栅 MOS(FAMOS)型 EPROM 存储元示意图。

出厂时所有 FAMOS 晶体管浮栅都不带电荷，表示全存“1”；写“0”时，在源极 S 和漏极 D 间加正向高电压时，栅极获得电荷，使源极 S 和漏极 D 之间导通。当源漏极高压去除后，由于栅极被绝缘层包围，电荷无处泄漏，故 FAMOS 晶体管一直保持导通或者截止，使存入信息能够长期保持下去。

EPROM 芯片的上方有一个石英玻璃窗口，当需要改写时，将它放到紫外线灯光下照射 15～20min 使栅极放电，便可擦除信息。擦除信息后所有存储元恢复到初始状态(全“1”)，此时可通过编程器写入新的内容。

EPROM 采用 MOS 工艺，由于需要长时间紫外线照射进行擦除，因此速度比较慢。擦除后，芯片内的信息会全部丢失，因此使用不够灵活。

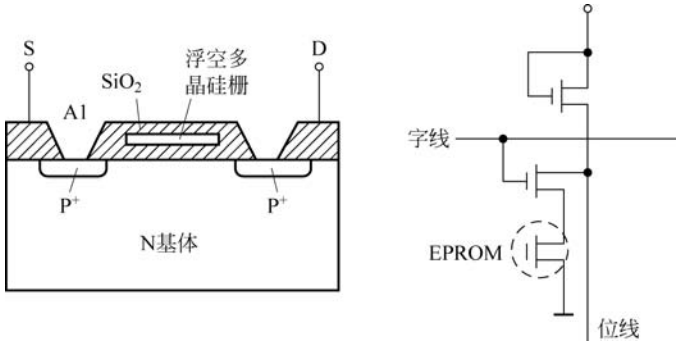


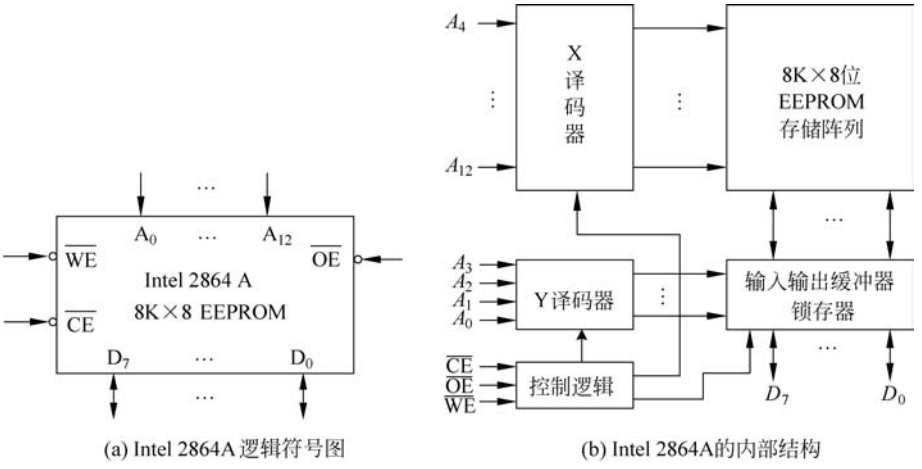
图 3-20　N 型沟道浮动栅 MOS 电路

4) EEPROM

EEPROM(Electrically Erasable Programmable ROM，电擦除可编程 ROM，也被称为 E^2PROM)是一种电可擦除可编程 ROM。它的编程原理与 EPROM 类似，但采用电擦除技术来实现数据的擦除。EEPROM 可以以字或数据块为单位进行擦除和改写操作，但可重写的次数通常是有限制的。

EEPROM的读写操作与SRAM类似,可进行按位或字节的随机读写,但又能在掉电时不丢失所保存的信息,并且EEPROM的改写不需要使用专用的编程器,只需在指定的引脚加上合适的电压(如+5V)即可进行在线擦除和改写,使用起来更加方便灵活。

下面以Intel 2864A EEPROM芯片为例介绍。2864A芯片的外特征如图3-21(a)所示,它是一个8K×8位的EEPROM,有13根地址线和8根数据线,$\overline{CE}$为片使能信号线,低电平时控制行列译码器输出;$\overline{OE}$为输出使能控制信号,低电平时表示读数据;$\overline{WE}$为写控制信号,低电平表示写数据。Intel 2864A采用单一+5V供电,最大工作电流为160mA,维持电流为60mA。读出时间最大为250ns,写入时间约为16ms。Intel 2864A的内部逻辑结构如图3-21(b)所示。



(a) Intel 2864A逻辑符号图　　(b) Intel 2864A的内部结构

图3-21　Intel 2864A存储芯片

Intel 2864A有3种常用工作方式见表3-3。

表3-3　2864A工作方式选择

方式	控制引脚			
	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	$I/O_0 \sim I/O_7$
读出	L	L	H	数据输出
写入	L	H	L	数据输入
不工作	H	X	X	高阻

EEPROM芯片的维持和读出操作与前面介绍的SRAM相同,下面主要介绍的2864A写入操作。2864A片内设有编程所需高压脉冲电路,因而无须外加编程电压和写入脉冲即可工作,并且2864A在写一个字节的数据之前,会自动将要写入单元进行擦除,因此无须专门的擦除操作。2864A的写操作有两种方式:字节写入和页写入。字节写入很少使用,这里主要介绍页写入方式。2864A内部有16字节的页缓存器,这样8K×8位的存储空间被分为512页,每页16字节。地址的高9位($A_4 \sim A_{12}$)用来确定页面,低4位($A_0 \sim A_3$)用来确定页中的字节。图3-22给出了Intel 2864A页写入操作的时序。

从图3-22中可以看出,2864A利用$\overline{WE}$脉冲进行写入控制,在它的下降沿时给出页内

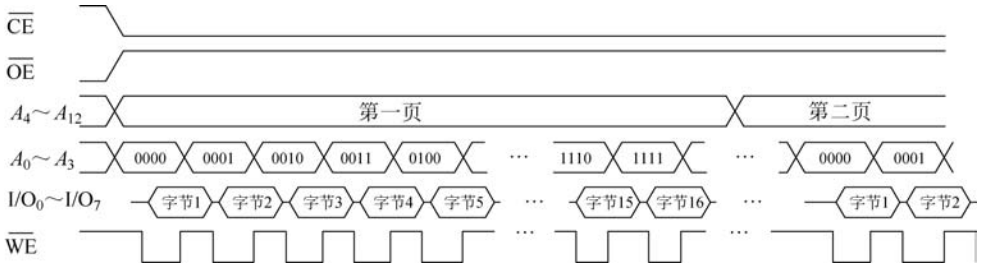


图 3-22　2864A 页面写入操作的时序

字节的地址，在它的上升沿时页缓存器锁存数据总线的内容。这个写入页缓存器的过程是重复地执行，直到写完一页 16 字节为止。在上述写入过程中，高 9 位地址 $A_4 \sim A_{12}$ 保持不变，以保证本次的页写入过程是对同一页进行的。

2. 闪速存储器

闪速存储器(Flash memory，简称 Flash 存储器或闪存)是近年来发展非常快的一种新型半导体存储器，与 EEPROM 类似，也是一种电擦写型 ROM。与 EEPROM 相比主要优点包括速度快、成本低、容量大、低功耗、可在联机状态下进行电擦除和改写。因此又被称为快擦型电可擦除可编程 ROM。目前被广泛应用于 U 盘、存储卡等移动存储设备中。

图 3-23 是一个闪速存储器的存储元，每个存储元由单个 FAMOS 晶体管组成。相对于 EPROM 存储元，FAMOS 中多了一个控制栅，用于控制浮栅充放电。

闪速存储器有三个基本操作：擦除操作、编程操作和读取操作。如图 3-23(a)所示，当控制栅加足够负电压时，浮栅中电荷泄漏，源极与漏极之间不导通，存储元存储二进制位“1”。如图 3-23(b)所示，当控制栅加足够正电压时，浮栅获得电荷，源极与漏极之间导通，存储元存储二进制位“0”。Flash 在擦除后，所有存储元的内容都为“1”，写入过程实际上就是对需要写入“0”的存储元浮栅充电。在控制栅不加电压的情况下，浮栅中电荷状态可以长期保持。读取操作根据源极和漏极是否导通，来判别读出的是“1”还是“0”，如图 3-23(c)所示。

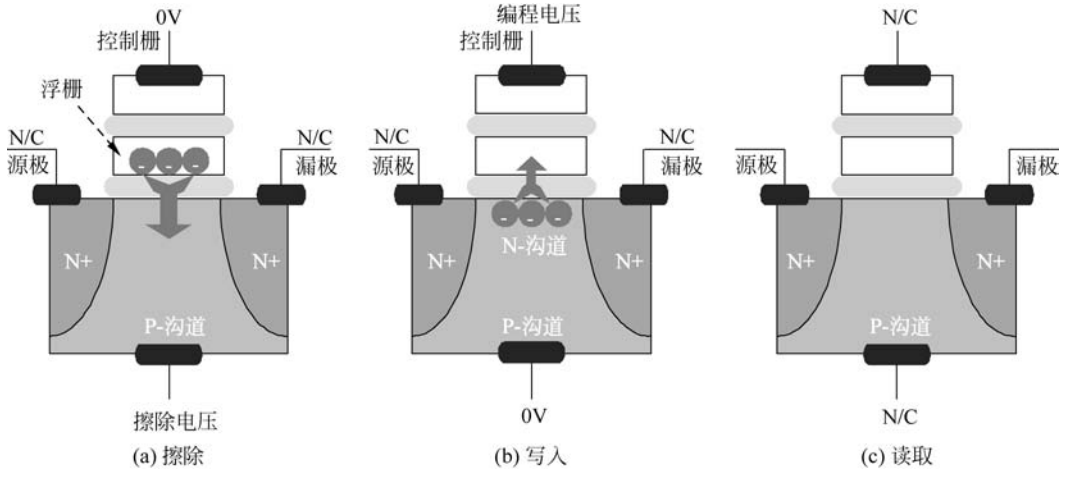


图 3-23　闪速存储器的存储元(N/C 表示不连接)

早期Flash每个存储元只能存储一位数据,随着Flash技术的日趋成熟,以及电子设备对Flash存储器容量的需求,上述单个存储元实际可以存储多位数据。如图3-24(a)所示为传统Flash存储元浮栅上电荷状态,存储元读写部件仅能区分这两种状态,即存储元只能存储一位数据,称为SLC(Single Level Cell)。如图3-24(b)所示,存储元读写部件可以区分4种不同状态,即存储元可以存储两位数据,称为MLC(Multiple Level Cell)。目前能够存储更多位的Flash存储器已经开始应用,例如存储三位数据的Flash技术TLC(Triple Level Cell)。

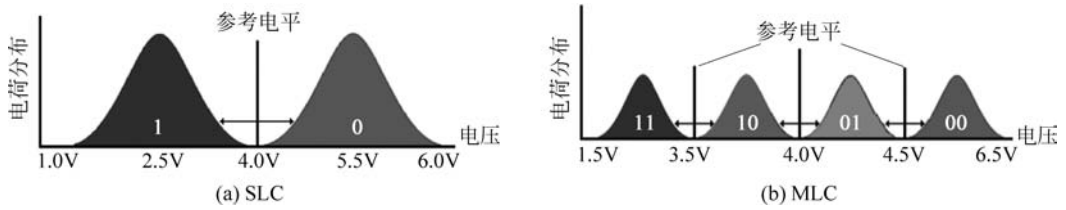


图3-24 Flash存储元浮栅电荷状态示意图

本章前几节介绍的各种半导体存储器根据其不同的物理特性,通常被用于计算机系统中的不同存储部件,表3-4给出了这些半导体存储器的一些典型应用。

表3-4 几种半导体存储器的典型应用

半导体存储器	典 型 应 用
SRAM	高速缓存存储器Cache
DRAM	主存中的用户程序
ROM	主存中的系统程序、微程序控制存储器
PROM	用户自编程序,用于工业控制机或电器中
EPROM	用户编写并可修改的程序或产品试制阶段的试统程序
EEPROM	IC卡上存储信息
Flash Memory	固态硬盘、各种存储卡、U盘

3.2.7 存储器容量扩展及其与CPU的连接

1. 存储容量的扩展

受集成度和功耗等因素的限制,单片存储芯片的容量是有限的,在字数和字长方面可能都很难满足实际应用的需要,因此必须将多个存储芯片组合起来构成满足实际容量需求的存储器。

存储容量的扩展,通常有位扩展、字扩展和字位扩展三种方式。不同类型的存储芯片由于其特性不同,SRAM、DRAM以及ROM芯片在进行扩展以及和CPU进行连接的时候,还各有特点。下面首先以SRAM为例说明容量扩展的三种方式。

1) 位扩展

位扩展是指将多片存储芯片连接起来以增加存储器的存储字长。进行位扩展后的存储器总字数与单个芯片的总字数一致,所需芯片的数量=存储器总字长/芯片字长。位扩展的基本方法是将各个存储器芯片的地址线、片使能线、读写控制线分别按同名端并连起来,每

个芯片的数据线分别引出作为整个存储器的数据线的一部分。图 3-25 给出了一个用 8 片 1K×1 位的 SRAM 芯片组成一个 1K×8 位存储器的例子。将 1K×1 位的 SRAM 芯片的地址线 $A_0 \sim A_9$、片使能线 $\overline{CE}$ 和读写控制线 $\overline{WE}$ 分别并联连在一起，每个 1K×1 位的芯片的数据线作为存储器数据线 $D_0 \sim D_7$ 中的 1 位。

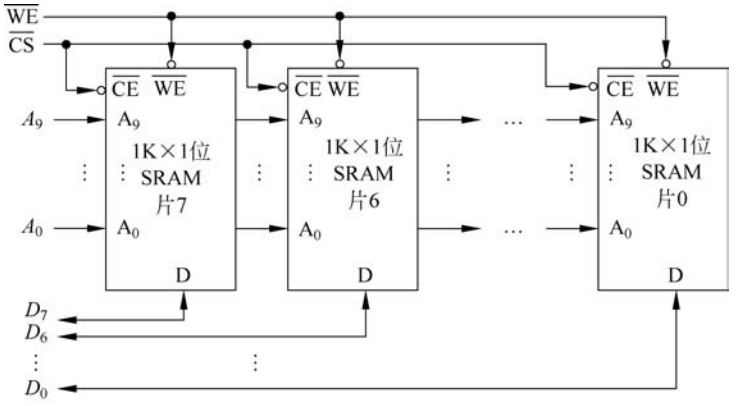


图 3-25　SRAM 存储器位扩展逻辑图

2）字扩展

字扩展是将多片存储芯片连在一起以扩充存储器的字数。进行字扩展后的存储器字长与单个芯片的字长一致，所需芯片的数量＝存储器总容量/芯片容量。字扩展方式通常是将各个存储器芯片的数据线、读写控制线分别并连起来；存储器地址线分为高位和低位两部分，低位地址线直接与各芯片的地址引脚相连，高位地址线输入到片选译码器产生各芯片的片使能信号线。图 3-26 给出了一个用 8 片 8K×8 位的 SRAM 芯片组成一个 64K×8 位的存储器的例子。将 8K×8 位的 SRAM 芯片的数据线 $D_0 \sim D_7$ 和读写控制线 $\overline{WE}$ 分别并连在一起；地址线 $A_0 \sim A_{12}$ 与各芯片的地址引脚对应相连，地址线 $A_{13} \sim A_{15}$ 经过 3-8 译码器产生的片使能信号与每个芯片的片使能引脚相连。

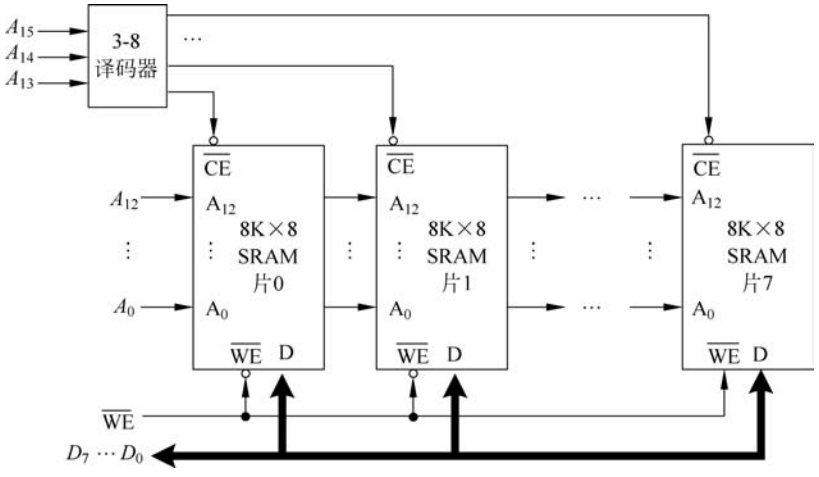


图 3-26　SRAM 存储器字扩展逻辑图

3) 字位扩展

字位扩展是指既增加存储字数,又增加存储字长。所需芯片的数量=存储器总容量×存储器总字长÷芯片容量÷芯片字长。字扩展方式通常是将各个存储器芯片的读写控制线并连在一起;数据线沿位方向同名端并连引出;存储器地址线分为高位和低位两部分,低位地址线直接与各芯片的地址引脚相连,高位地址线输入到片选译码器产生片选信号,每根片选信号连接字方向所有芯片的片使能端。

图 3-27 给出了一个用 8 片 8K×8 位的 SRAM 芯片组成 32K×16 位存储器的例子。每两片 8K×8 的 SRAM 芯片进行位扩展构成了 8K×16 位存储器,这样四组 8K×16 位存储器再进行字扩展构成了 32K×16 位存储器。地址线 A_{14} 和 A_{13} 经过片选译码器产生的四个片选信号分别连接到四组每组两片 8K×8 位芯片的片使能端。

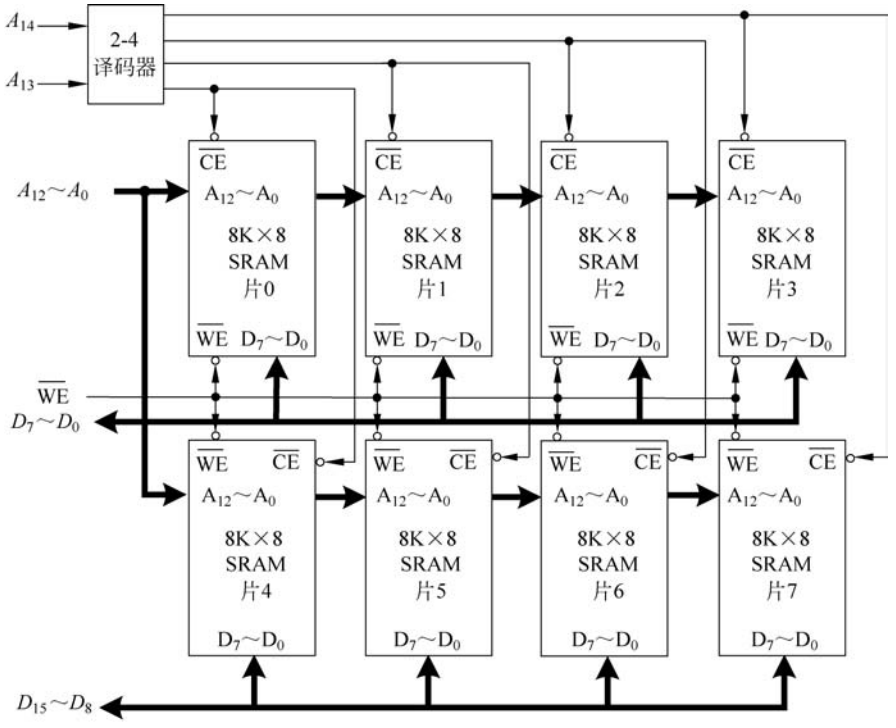


图 3-27 SRAM 存储器字位扩展逻辑图

在设计计算机主存时,除了需要考虑存储芯片的数量要满足存储单元位数和主存总容量的要求外,还需要考虑存储芯片的类型。在主存中通常选用只读存储器 ROM 芯片来存放系统程序、标准子程序和各类常数等;而选用随机存储器 RAM 芯片来存放用户的程序和数据。

上面介绍的存储器扩展,主要是针对 SRAM 芯片的。对于 ROM 芯片,它通常没有 $\overline{WE}$ 引脚(EEPROM 除外),其他引脚的连接方法与 SRAM 基本一致。具体连接方法可参见图 3-29 中 ROM 芯片的连接。

对于 DRAM 芯片容量扩展的基本方法与 SRAM 一致,但是地址线、片使能信号线等的连接还需要进行特殊处理,主要区别如下。

(1) DRAM 芯片地址引脚通常采用多路复用技术，片地址分行地址和列地址分时输入。这种技术的采用要求 DRAM 地址线通过地址多路选择器与总线连通，地址多路选择器一般由动态存储控制器(DRAMC)提供。DRAMC 的组成及工作原理将在下面详细介绍。

(2) DRAM 为减少引脚数所采取的另一个常用措施是不设片使能引脚。当采用字扩展技术时，通常将片使能信号与行地址选通信号复合后通过 $\overline{RAS}$ 引脚输入芯片，因为 DRAM 芯片是由 $\overline{RAS}$ 信号启动工作的，$\overline{RAS}$ 信号无效时，芯片内部既不会产生行时钟，也不会产生列时钟。DRAM 片选译码原理与 SRAM 基本相同，但 DRAMC 芯片通常提供一定的片选功能，此种情况下可不再专设片选译码器。

(3) DRAM 需要在刷新地址计数、刷新定时等外围电路的支持下，才能正确完成刷新操作，这些外围功能通常也由 DRAMC 芯片提供(具有自动刷新功能的 DRAM 芯片中含有刷新地址计数器)。

图 3-28 给出了 16 个 256K×8 位 DRAM 芯片(内部采用 8 体结构)构成 1M×32 位存储器的芯片扩展示意图。图中地址多路选择器负责将片地址分行地址和列地址两部分分时输入，并且还负责在刷新时提供刷新的行地址。片选译码器负责产生 $\overline{RAS_0}$ 到 $\overline{RAS_3}$ 四组片选信号，选择 256K×32 位组中的一组芯片。

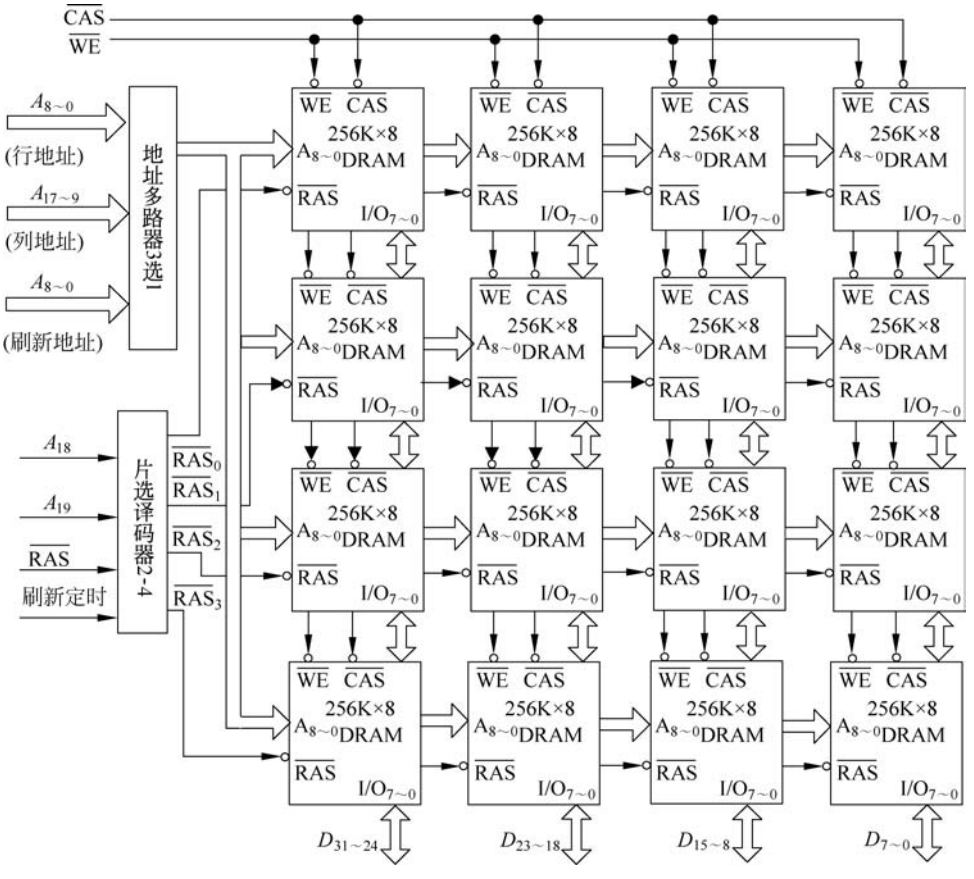


图 3-28　DRAM 容量扩展示意图

2. 存储器与系统总线的连接

存储芯片与系统总线在进行连接时,主要涉及地址线、数据线和控制线的连接。

(1) 数据线的连接。存储器的数据线位数必须和系统总线的数据线位数一致,并进行逐位连接。通过位扩展形成的存储器,扩展后的位数也必须与系统总线数据线的位数一致。

(2) 地址线的连接。单片存储器的地址线与系统总线的地址线一一对应。对于经过字扩展形成的存储器,系统总线地址线数比单个存储芯片的地址线数多,这时系统总线地址线的低位与存储芯片的地址线相连,而高位通常用于形成存储芯片的片使能信号,也可直接作为片使能信号等。

(3) 控制线的连接。与存储器相关的控制线主要是读写控制线,通常总线的读写控制线直接与存储芯片的读写控制端相连,一般情况下高电平为读,低电平为写。也有些系统总线的读写控制线是分开的。

在 CPU 与存储芯片相连时,CPU 发出的地址的高位通常被送到译码器以产生片选信号;地址线的低位送到存储芯片用于进行地址选择。CPU 的访存控制信号 $\overline{\text{MREQ}}$ 信号有效时译码器才能产生有效的片使能信号,CPU 使用 $\overline{\text{MREQ}}$ 信号来区分访存和访问 I/O 操作。

图 3-29 给出了一个存储器通过系统总线与 CPU 连接的例子。由两个 1K×8 位的 ROM 芯片和 6 个 1K×8 位的 RAM 芯片共同构成了一个 8K×8 位的存储器。ROM 一般用于存放系统程序,通常占用低位地址空间,RAM 一般用于存放用户程序,通常占用高位地址空间。

图 3-29 中用到了 74138 译码器(3-8 译码器),其中 A、B、C 是三个变量输入端,Y_0 到 Y_7 是 8 个变量输出端;$\overline{G_{2A}}$、$\overline{G_{2B}}$、G_1 是三个控制端,当 $\overline{G_{2A}}$ 和 $\overline{G_{2B}}$ 为低电平并且 G_1 为高电平时译码器才能工作。图中将 CPU 发出的访存控制信号 MREQ 信号与 $\overline{G_{2A}}$ 和 $\overline{G_{2B}}$ 相连,这样只有当 CPU 需要访问存储器时译码器才能工作。

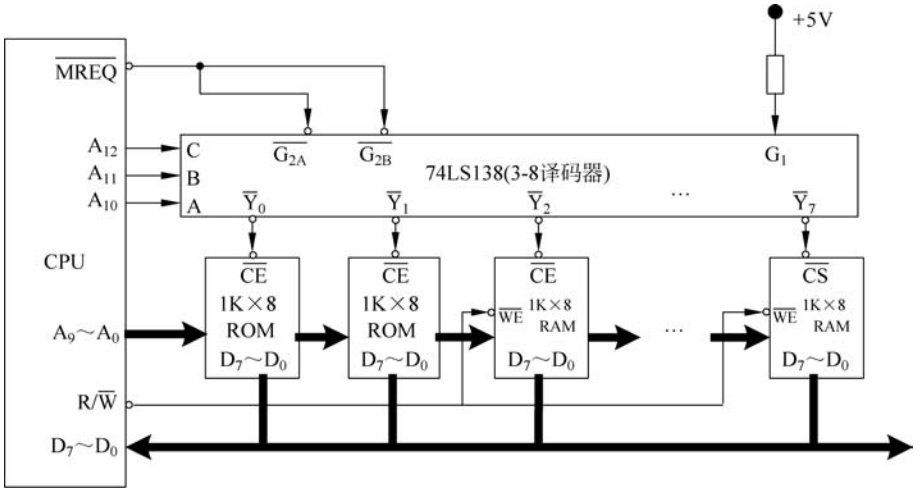


图 3-29　SRAM 和 ROM 与 CPU 的连接

由于 DRAM 需要一些特殊的外围控制电路和特殊的控制信号,因此 DRAM 一般不直接与 CPU 相连,而是通过 DRAM 控制器(简称 DRAMC)与 CPU 相连,如图 3-30 所示。

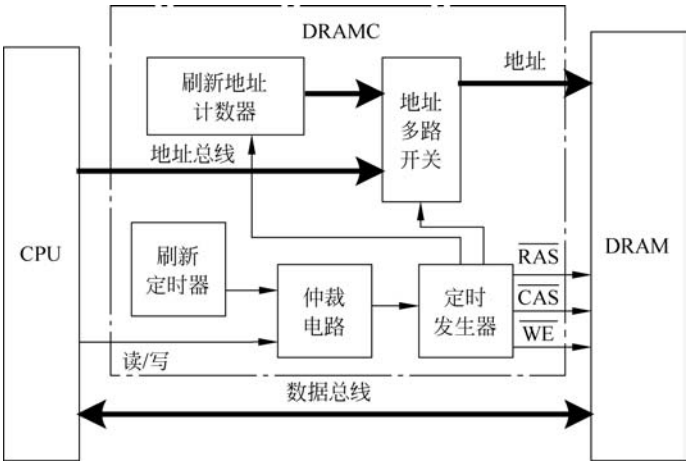


图 3-30　DRAM 与 CPU 的连接

DRAMC 在 CPU 和 DRAM 之间用于完成地址转换、产生控制信号、控制刷新等操作。图 3-30 中的虚线框内为 DRAMC 的基本逻辑框图。从图可以看出，DRAMC 主要包括以下五大组成部分。

(1) 刷新地址计数器：刷新时向 DRAM 提供刷新行地址，并自动顺序计数到下一行地址。

(2) 地址多路选择器：在行、列、刷新地址三者间选一路送给 DRAM。

(3) 刷新定时器：根据所选用的刷新定时方式控制刷新时间，当需要刷新时及时发出刷新请求信号。

(4) 仲裁电路：在 CPU(I/O)访存请求、刷新请求中裁决出一种执行访存操作。

(5) 定时发生器：产生一系列 DRAM 工作时所需的时间控制信号($\overline{RAS}$、$\overline{CAS}$、$\overline{WE}$ 等)。

另外，在将存储器芯片连接到 CPU 时，除了要考虑存储器芯片的类型(ROM、RAM 等)和数量。还需要考虑时序的配合问题、速度匹配问题、负载能力等问题。CPU 与存储器芯片的时序配合问题是两者在进行连接时的难点问题之一。

【例 3.1】 某 16 位微型计算机地址码为 20 位，若使用 8K×4 位的 DRAM 芯片组成模块板结构的存储器，问：

(1) 该机所允许的最大主存空间是多少？

(2) 若每个模块板为 64K×8 位，共需几个模块板？每个模块板内共有几片 RAM 芯片？

(3) CPU 如何选择各模块板？

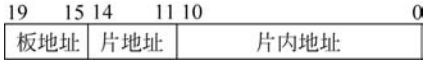
解：

(1) 2^{20}=1M，则该机所允许的最大主存空间是 1M×16 位。

(2) 模块板总数=1M×16/64K×8=32 块；

板内片数=64K×8 位/8K×4 位=8×2=16 片。

(3) CPU 通过最高 5 位地址译码选板，次高 4 位地址译码选片。地址格式分配如下：



【例 3.2】 设 CPU 共有 16 根地址线,8 根数据线,并用 $\overline{\text{MREQ}}$(低电平有效)做访存控制信号,R/$\overline{\text{W}}$ 作读写命令信号(高电平为读,低电平为写)。现有下列存储芯片:ROM(2K×8 位,4K×4 位,8K×8 位),RAM(1K×4 位,2K×8 位,4K×8 位),及 74138 译码器和其他门电路(门电路自定)。试从上述规格中选用合适芯片,画出 CPU 和存储芯片的连接图。要求:

(1) 最小 4K 地址为系统程序区,4096～16383 地址范围为用户程序区。

(2) 指出选用的存储芯片类型及数量。

(3) 详细画出片选逻辑。

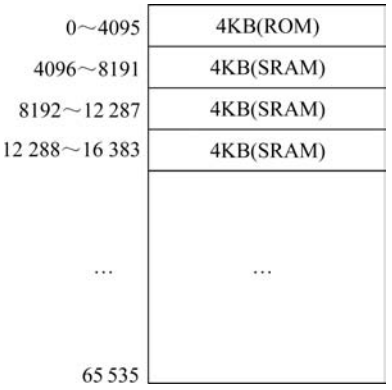


图 3-31 地址空间分配

解:

(1) 地址空间分配如图 3-31 所示。

(2) 在进行芯片选择时,应注意以下问题。

- 当采用字扩展和位扩展所用芯片一样多时,选位扩展。这是因为:字扩展需设计片选译码,而位扩展只需将数据线按位引出即可。例如,本题如果选用 2K×8 ROM,片选要采用二级译码,实现比较麻烦。
- 当需要 RAM、ROM 等多种芯片混用时,应尽量选容量等外特性较为一致的芯片,以便于简化连线。
- 应尽可能地避免使用二级译码,以使设计简练。但要注意,在需要二级译码时如果不使用,会使选片产生二义性。

基于以上原则,需选择 2 片 4K×4 位 ROM 和 3 片 4K×8 位 RAM 芯片。

(3) CPU 和存储器连接逻辑及片选逻辑如图 3-32 所示。

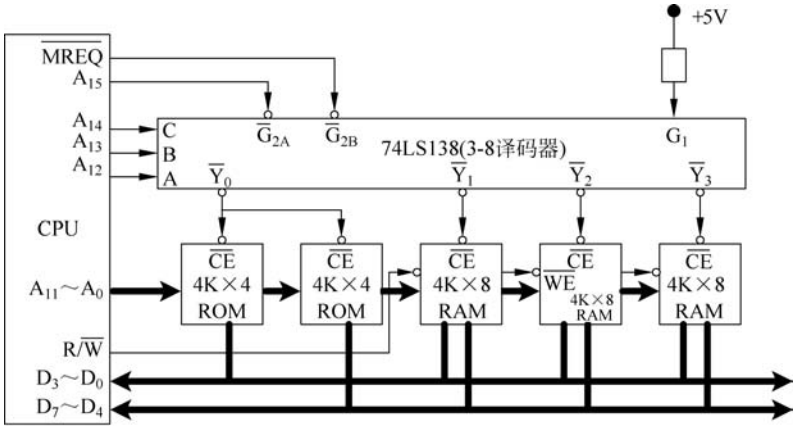


图 3-32 CPU 和存储器连接逻辑及片选逻辑

3.2.8 微处理器与存储器连接举例

为了能与存储器相连并保证其正常工作,微处理器都会提供一些控制信号来控制存储

器的工作。对于存储器来说，一方面需要对 CPU 通过地址总线送来的地址进行译码；另一方面，也需要将 CPU 发出的控制信号与存储芯片的控制引脚相连。本节将举例示意性说明微处理器与存储器芯片的连接。

16 位的微处理器 8086(80286)的地址总线宽度为 20 位，数据总线的宽度为 16 位，采用按字节编址的方式。8086 CPU 的存储器采用两个存储体结构，即把 1MB 的主存空间分成两个 512KB 的存储体，如图 3-33 所示。在图中只画出了主要的信号线。两个存储体中一个存储体的地址全部为偶数，被称为偶(低字节)存储体；另一个存储体的地址全部为奇数，被称为奇(高字节)存储体。偶存储体与低 8 位数据总线($D_7 \sim D_0$)相连；奇存储体与高 8 位数据总线相连($D_{15} \sim D_8$)。通过地址线($A_{19} \sim A_1$)和存储体选择信号线(BHE 和 A_0)可以灵活地实现各种寻址方式下的数据传输，见表 3-5 所示。

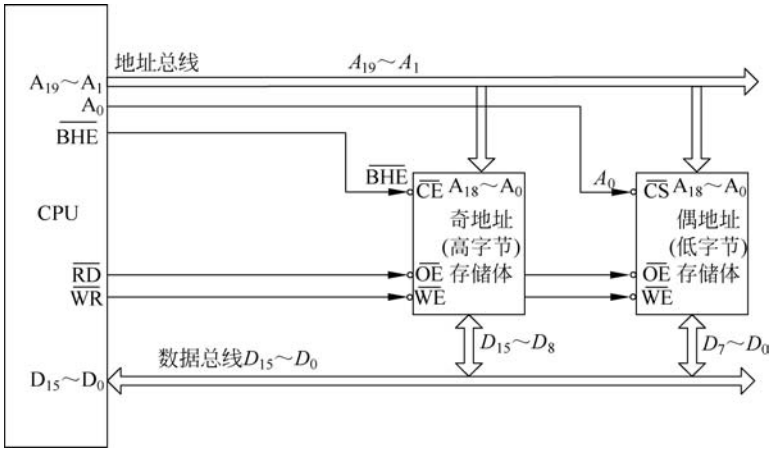


图 3-33　16 位(8086)存储器的组织

表 3-5　8086 存储体的选择

$\overline{BHE}$	A_0	含　义
0	0	全字传输(两个存储体同时被选中)
0	1	在数据总线的高 8 位上进行字节传输(选中奇地址存储体)
1	0	在数据总线的低 8 位上进行字节传输(造中偶地址存储体)
1	1	备用

例如，当 $\overline{BHE}$ 和 A_0 同时为 0 时(读操作)，地址总线 $A_{19} \sim A_1$ 给出的 19 位地址同时选中奇、偶存储体中的两个存储单元，偶存储体中存储单元的数据送到数据总线 $D_7 \sim D_0$，奇存储体中存储单元的数据送到数据总线 $D_{15} \sim D_8$，此时共有两字节的数据送出。而当 $\overline{BHE}$ 为 0，A_0 为 1 时(读操作)，地址总线 $A_{19} \sim A_1$ 给出的 19 位地址仅选中奇存储体中的一个存储单元，该存储单元的数据送到数据总线 $D_{15} \sim D_8$，此时仅有一字节的数据送出。

在 8086 中，对于字及双字的存储，会把低字节放到低位地址，高字节依次放到后续字节中；并且最小的字节地址是该字或者双字的地址。对于 8086，如果一个字是边界对齐的，即其低字节在偶存储体，高字节在奇存储体，则在一个存储周期可完成该字的存取。如果一个字的边界不对齐，即其高字节在偶存储体，低字节在偶存储体，则需要两个存储周期才能完

成该字的存取,并且此时字中两个字节的顺序是颠倒的,CPU会自动完成两个字节的对换。

图3-34给出了针对不同存储体的寻址情况。图3-34(a)和(b)是CPU按字节读取存储器的示意图,CPU可以用偶地址或奇地址访问存储器中的字节单元。图3-34(c)是CPU用偶地址读取存储器的字单元,这种情况下仅需要一次存储器读操作和一次数据总线传送就可以完成一个字数据的访问。图3-34(d)是CPU用奇地址读取存储器的字单元,这是字边界不对齐的情况,所以需要两次存储器读操作和两次数据总线传送才能完成一个字数据的访问。

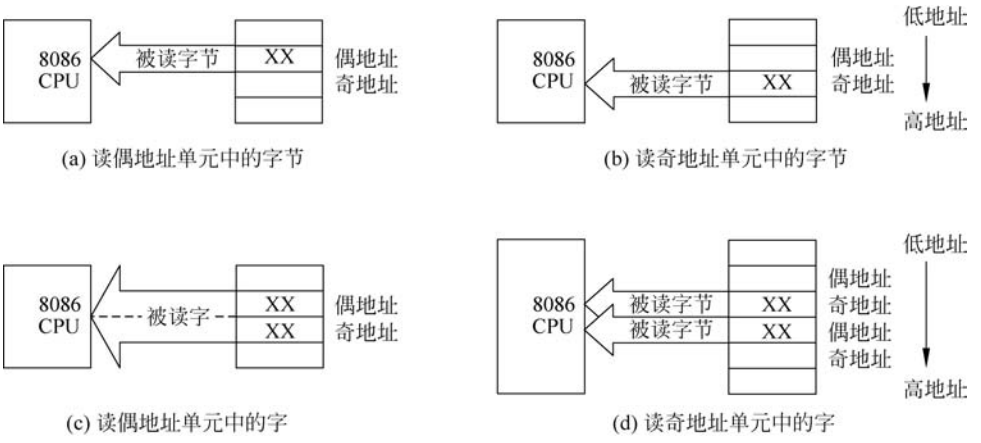


图3-34　8086 CPU从存储器读字节和字的情况

下面给出了一个8086微处理器的小型存储系统。图3-35所示的存储系统的存储容量为64K×16位,由16个8K×8位的SRAM芯片组成,构成的存储器的地址范围为:00000H～1FFFFH。这16个SRAM芯片分成两组,用于构成偶存储体和奇存储体。A_0用

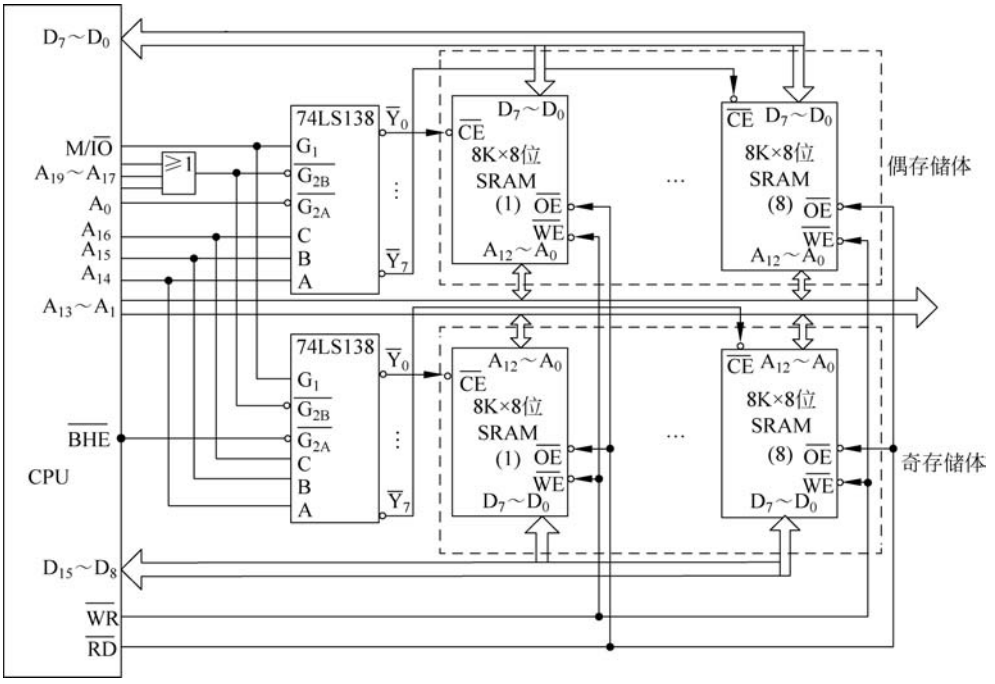


图3-35　8086微处理器与存储器的连接

于选择偶存储体，$\overline{\text{BHE}}$ 用于选择奇存储体，偶存储体的数据线与地址总线的 $D_0 \sim D_7$ 相连，奇存储体的数据线与地址总线的 $D_8 \sim D_{15}$ 相连。地址总线中的 $A_1 \sim A_{13}$ 与每个芯片的地址线相连，地址线 $A_{14} \sim A_{19}$ 与两个 74LS138 译码器的控制端相连。CPU 发出的读信号 $\overline{\text{RD}}$ 和写信号 $\overline{\text{WR}}$ 分别与芯片的 $\overline{\text{CE}}$ 与 $\overline{\text{WE}}$ 相连。

3.3 相联存储器

3.3.1 相联访问的思想

上节介绍的 ROM 和 RAM 都是按地址访问的，而相联存储器（Associative Memory，AM）是按照存储单元中存放的全部或部分内容进行访问的存储器，因此也被称为按内容访问存储器（Content Addressed Memory，CAM）。

相联存储器在进行写操作时，采用按地址访问的方式。在进行读操作时，除了可以按地址进行访问外，还可以按内容访问，即将 CPU 给出的关键字（存储单元信息的全部或部分内容）和存储器中所有单元中的相应信息进行比较，定位到与关键字匹配的存储单元后，将此单元中的所有信息读出。

假设存储器中存储了表 3-6 所示的学生信息表，该表由五条记录组成，每条记录包含四个字段：学号、姓名、出生年月和成绩。当对表 3-6 所示的表格执行如“学号为 050701 学生的成绩是多少?”查询时，采用传统的随机存储器，必须给出存储单元的物理地址才能找到相应的存储单元。而物理地址与学号和成绩并没有逻辑上的关联关系，因此查询程序的复杂性较高。但是，如果选择表 3-6 中的一个字段作为关键字来访问存储器时，显然会提高查询的效率。例如，选择“学号”作为关键字来访问存储器，则很快能查询到该学号对应学生的成绩等信息。

表 3-6 存放在存储器中的一张表格

物理地址	学号	姓名	出生年月	成绩
n	050701	张 **	1982.12	82
$n+1$	050702	李 **	1983.1	90
$n+2$	050703	王 **	1982.5	75
$n+3$	050704	赵 **	1983.6	62
$n+4$	050705	吴 **	1983.9	92

相联存储器就是基于上述思想产生的。简单地说，相联存储器就是用存储单元中的某一个存储项的内容来对存储器进行寻址。这个用来定位存储单元的字段被称为关键字，简称为键（key）。这样，相联存储器中每个存储单元中存储的信息都由关键字和数据两部分组成。

3.3.2 相联存储器的结构及工作原理

相联存储器由存储体、检索寄存器、屏蔽寄存器、匹配寄存器、数据寄存器、比较电路和译码电路等组成，如图 3-36 所示。

由于相联存储器的写入过程采用前面介绍的按地址进行访问,这里不再赘述。下面以相联存储器的读出过程为例,介绍图 3-36 中相联存储器主要功能部件的工作原理。

在进行读操作时,检索寄存器中存放用于查询的检索字,其位数与相联存储器存储单元的字长相等。为了能确定检索寄存器中检索字的哪些位是关键字,需要使用屏蔽寄存器中存放的屏蔽码。屏蔽码的位数与检索寄存器位数相同,并且屏蔽码中关键字对应的位被置“1”,其余位各位被清“0”。这样,当检索字与屏蔽码进行“与”操作后,结果只保留了关键字的值。在图 3-36 中,检索字中的第 8 到第 11 位作为关键字(位序从右到左编排,最右边的位是第 0 位),这样屏蔽寄存器中的屏蔽码的值为“0000 1111 0000 0000”,即只有第 8 到第 11 位的值为“1”。

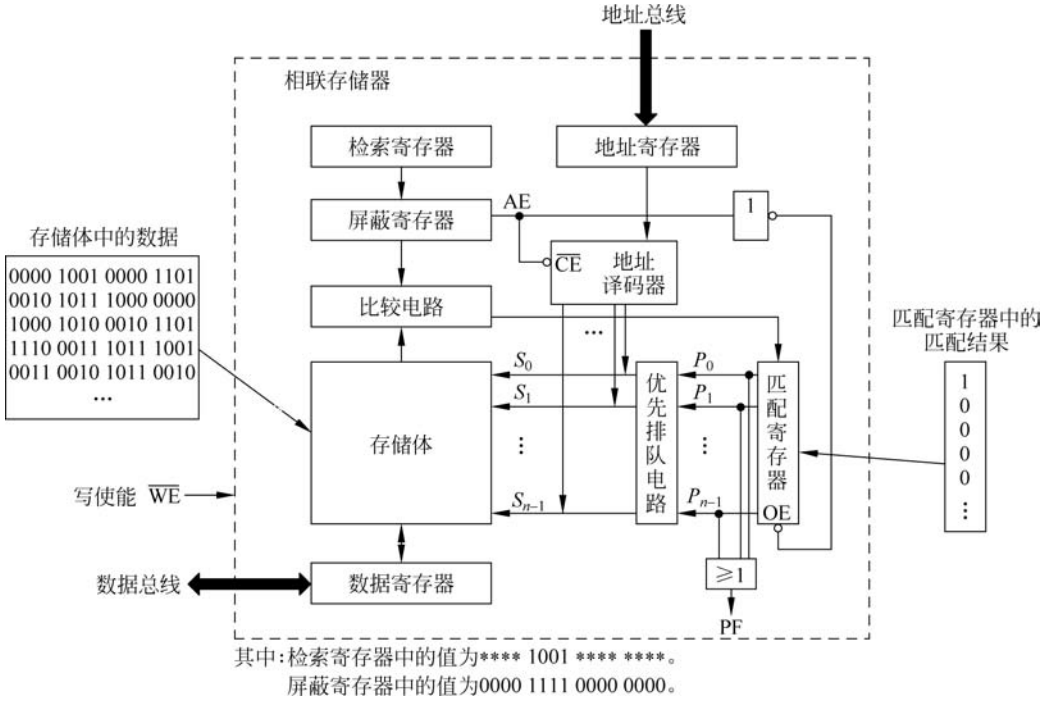


图 3-36　相联存储器的组成

检索寄存器中的检索字与屏蔽寄存器中的屏蔽码进行“与”操作的结果(即关键字)将通过比较电路与所有存储单元的对应位进行比较。若某个存储单元的内容与关键字相符,则置该存储单元在匹配寄存器中相应位的值为“1”。匹配寄存器的位数等于相联存储器的字数。当所有存储单元内容都比较完后,匹配寄存器中所有值为“1”的位对应的存储单元,就是与检索字匹配成功的存储单元。在图 3-36 的例子中,只有第一个存储单元匹配成功。

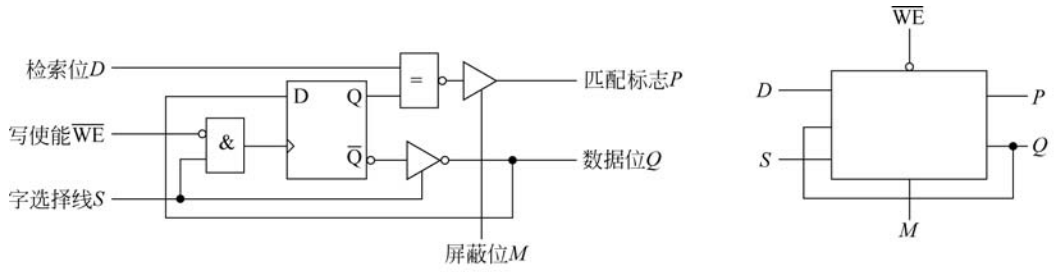
需要注意的是,由于与检索字匹配的存储字可能有多个,即匹配寄存器中有多位为“1”,因此还需要使用优先排队电路对匹配寄存器的输出进行排队。常用的方法是只保留匹配寄存器中位置最靠前的“1”,后面的全部清“0”。这样就只有一个匹配的存储字输出到数据寄存器。

图 3-36 中的匹配标志位 PF 是匹配寄存器中所有位进行“或”操作的结果。当 PF=1 时,数据寄存器中保存的是匹配存储单元的内容;当 PF=0 时,表示没有检索到匹配的存储

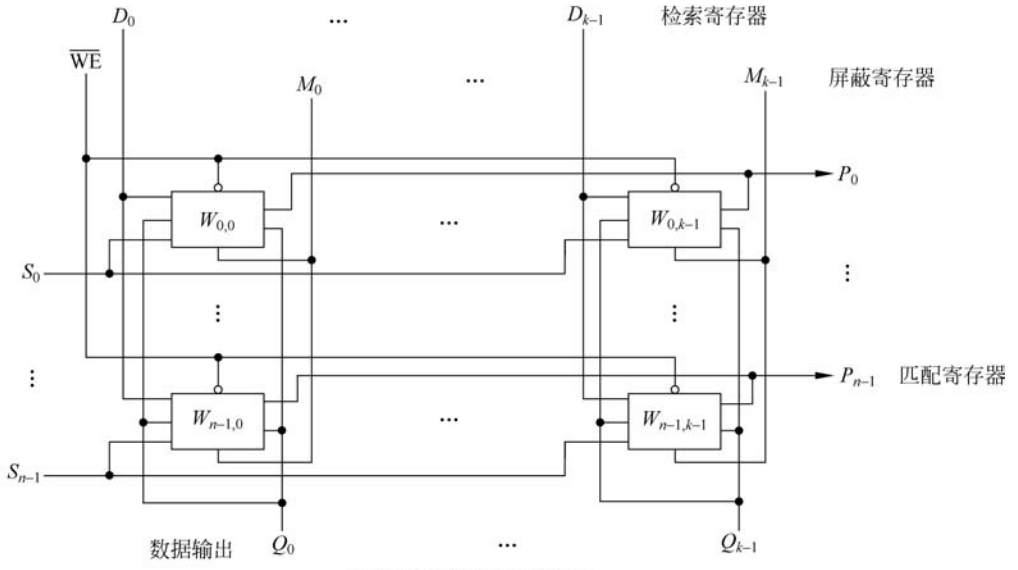
单元。

此外，相联存储器也可以按地址访问。当屏蔽寄存器中屏蔽码的值为全“0”时，表示检索寄存器被屏蔽，输出信号 AE 为低。此时，存储器将不按照匹配寄存器的内容进行读出，而是采用传统的地址译码器进行译码，按照地址总线送来的地址访问存储器。

相联存储器的存储体如图 3-37 所示。其中，图 3-37(a)为相联存储器的存储元，图 3-37(b)为相联存储器的存储矩阵。相联存储器的存储元主要由一个 D 触发器组成。当存储元未被屏蔽($M=1$)时，检索位与存储位进行同或操作后经三态门输出到 P 端，即匹配寄存器中的对应位。当 $P=1$ 时表示检索位与存储位相同。当存储元被屏蔽($M=0$)时，P 端输出为高阻。字选线 S 是从优先排队电路或地址译码器中送来的，若 $S=1$，则该存储元数据经 Q 端输出到数据寄存器的对应位；若 $S=0$，则既不能读也不能写。当写使能信号 $\overline{\text{WE}}$ 为负脉冲时，则可以进行写入操作。



(a) 相联存储器的存储元结构



(b) 相联存储器的存储矩阵

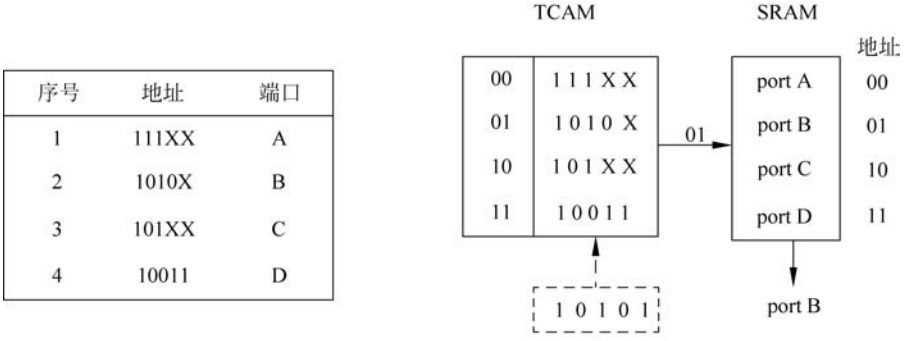
图 3-37　相联存储器的结构

多个这样的存储元就构成了如图 3-37(b)所示的 $n\times k$ 位的存储矩阵。图中 M_0 到 M_{k-1} 的值来自屏蔽寄存器，D_0 到 D_{k-1} 的值来自检索寄存器，Q_0 到 Q_{k-1} 与数据寄存器相连，S_0 到 S_{n-1} 的值来自排队电路。按照图 3-37 中给出的例子，屏蔽寄存器的值 $M_0\cdots M_{15}=$

0000 1111 0000 0000,则每个存储字的第 15 到 12 位、第 7 到第 0 位被屏蔽,对应的存储元 P 端输出为高阻,由于检索寄存器中关键字为“**** 1001 **** ****”,则最终只有 P_0 端的输出为 1,且 $W_{0,0}$ 到 $W_{0,k-1}$ 的数据通过 Q_0 到 Q_{k-1} 输出。

由于传统的比较电路只能进行 1∶1 比较,而相联存储器能够进行 1∶N 比较,即一次检索操作是将需检索内容与全部被检索数据同时进行比较,这样无论被检索的数据有多少,都只需进行一次检索操作即可得到结果。在计算机的存储系统中,相联存储器主要应用于需要快速检索的场合。例如,在高速缓冲存储器 Cache 中存放目录表和块表;在虚拟存储器中存放分段表、页表和快表。

目前广泛使用的一种 CAM 叫作 TCAM(Ternary Content Addressable Memory,三态内容访问存储器)。一般的 CAM 存储器中每位存储信息的状态只有两个:“0”或“1”,而 TCAM 中每位存储信息有三种状态,除了“0”和“1”外,还有一个“don't care(无关)”状态,所以称为“三态”。TCAM 的第三种状态特征使其既能进行精确匹配查找,又能进行模糊匹配查找。由于 CAM 没有第三种状态,所以只能进行精确匹配查找。TCAM 在高端网络路由器等设备中广泛使用,图 3-38 是网络路由器中 TCAM 的应用实例。路由器中一般包含一个 TCAM 和一个 SRAM,TCAM 用于 IP 地址高速查找(TCAM 通过配置可以输出存储单元的地址),其查找结果作为访问 SRAM 的地址,获取与 IP 地址对应的网络端口号。



序号	地址	端口
1	111XX	A
2	1010X	B
3	101XX	C
4	10011	D

(a) 路由表逻辑结构　　(b) 路由表物理实现及查表过程

图 3-38　TCAM 在网络路由器中的应用

根据图 3-38(a)给出的路由表,路由器有 4 个网络端口(A、B、C、D),分别可以连接 4 个子网,每个子网对应不同的 IP 地址前缀。为表示简化起见,假设网络 IP 地址长度为 5 位。第一个网络端口连接的子网段 IP 地址为 111XX,即地址前三位为 111 的 IP 地址属于 A 端口连接的子网。第二个网络端口(B)连接子网段 IP 地址前缀为 1010,第三个网络端口(C)连接的子网段 IP 地址前缀为 101,第四个网络端口(D)连接的子网段 IP 地址为 10011。依据 IP 地址最长前缀匹配原则,前缀为 1010 的 IP 地址虽然同时满足第二个和第三个网络端口对应子网的 IP 地址前缀,但优先匹配第二个网络端口对应的子网。

TCAM 和 SRAM 中保存的内容如图 3-38(b)所示。在路由器接收到 IP 地址为 10101 的数据包后,首先在 TCAM 中进行匹配,根据 TCAM 模糊查找特点及判优逻辑,输出第二个存储字的地址。该地址再作为访问 SRAM 的地址读取 SRAM 中对应单元的内容,得到目标子网对应的端口号。

3.4 高速缓冲存储器 Cache

3.4.1 Cache 的工作原理

与主存储器相比，高速缓冲存储器 Cache 的存取速度快、容量小、位价格高。它位于 CPU 与主存之间，能高速地向 CPU 提供指令和数据（读写），以加快程序的执行速度，是解决 CPU 和主存速度不匹配问题的重要技术。

由于 Cache 的容量比主存小得多，因此 Cache 中保存的只能是主存内容的一个子集。Cache 保存的内容一方面要与主存保持一致；另一方面还要使 CPU 需要访问的指令和数据尽可能在 Cache 中找到，即在 Cache 中能够命中。显然，Cache 的命中率越高，CPU 的访存速度就越接近于 Cache 的存取速度。

Cache 与 CPU 和主存的互连结构如图 3-39 所示。CPU 访问存储器时的基本数据单位是字或字节。基于程序运行的局部性原理，Cache 和主存的内容都按块进行逻辑组织，它们之间以块为单位进行内容交换。一个块由若干字组成，一般是定长的，并且主存和 Cache 块的大小相同。

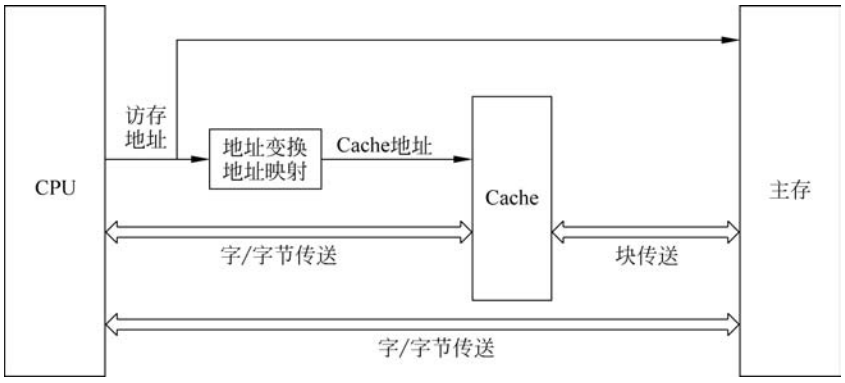


图 3-39　Cache 基本原理示意图

CPU 对存储器的读操作相对于写操作来说更加频繁，因为程序执行过程中 CPU 要不断地从主存中读取指令。写操作仅仅涉及对数据的存储操作。所以，我们先针对存储器读操作讲解有关 Cache 的工作原理和设计方法，写操作相关内容将在下一节中专门介绍。

CPU 在访问存储器时面对的是主存空间，所以 CPU 送出的地址是主存地址，但这个地址同时送给了 Cache 和主存。Cache 接收到地址后，首先要进行地址变换操作，即将主存地址变换成 Cache 地址。同时，通过地址变换机制还能判断出本次要访问的存储单元内容是否已经装入 Cache 中（即是否命中）。如果 CPU 需要的指令或数据在 Cache 中命中，则指令或数据由 Cache 直接送到 CPU；否则指令或数据仍然需要从主存中读取，并且还要通过地址映射机制将该指令或数据所在的块装入 Cache 中的指定位置。如果当前 Cache 已满，还要使用某种替换策略将 Cache 中的一个旧块替换出去。增设 Cache 后 CPU 访存的处理流程如图 3-40 所示。

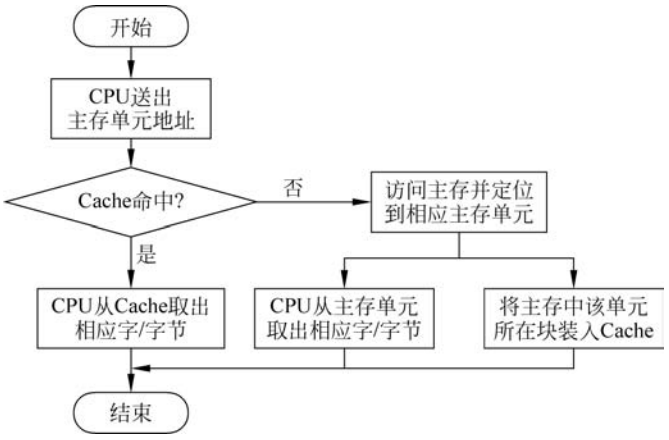


图 3-40　CPU 访存的处理流程

随着半导体器件集成度的不断提高,大多数 Cache 已经被集成到 CPU 中,这样 Cache 的工作速度更接近于 CPU 的速度。通常,Cache 还被设计为两级或者两级以上的结构。

程序执行的局部性原理保证了 CPU 对存储器进行指令或数据请求时,在 Cache 中有一定的命中率,但是并不能保证所有请求的指令或数据都在 Cache 中。显然,Cache 的命中率越高,CPU 的平均访存时间就越短。因此,提高 CPU 访存效率首先需要考虑的就是提高 Cache 命中率。

一般来说,Cache 的存储容量比主存要小得多,但 Cache 的容量也不能太小,太小会导致命中率太低。同样,Cache 容量也不能太大,太大不仅会增加成本,而且当 Cache 容量大到一定程度时,命中率不会随着容量的增加而明显地增大。只要配置合理,即 Cache 容量与主存容量在一定范围内保持适当比例的映射关系,Cache 命中率可以相当高。下面介绍评价 Cache 性能的主要指标。

Cache 的命中率 H 是指 CPU 访存时在 Cache 中找到所需要信息的概率。假设 CPU 总共访问了 N 次存储器,其中在 Cache 中命中了 N_1 次,未命中 N_2 次,则 Cache 的命中率 H 为:

$$H = N_1/N = N_1/(N_1 + N_2) \tag{3-1}$$

命中率是衡量 Cache 性能的主要指标,命中率 H 越接近 1 越好,这意味着 CPU 平均访存速度越接近于 Cache 的存取速度。命中率主要受以下几个因素的影响:程序执行过程中地址流分布情况、Cache 替换算法、Cache 容量、Cache 块大小以及 Cache 预取算法等。

此外,评价 Cache 性能经常还有以下几个主要指标。

(1) 失效率 F:指 CPU 访存时在 Cache 中未找到所需要信息的概率。

$$F = 1 - H \tag{3-2}$$

(2) 平均访存时间 T_a:指 CPU 单次访存所需要的平均时间。

$$T_a = HT_c + (1 - H)T_m \tag{3-3}$$

其中,T_c 为 Cache 的存取时间(也被称为命中时间),T_m 为主存的存取时间。

(3) 加速比 S_P:指主存的存取时间与平均访存时间的比值。

$$S_p = \frac{T_m}{T_a} = \frac{T_m}{HT_c + (1 - H)T_m} = \frac{1}{(1 - H) + H\dfrac{T_c}{T_m}} \tag{3-4}$$

(4) 访问效率 e：指 Cache 的存取时间与平均访存时间的比值。

$$e=\frac{T_c}{T_a}=\frac{T_c}{HT_c+(1-H)T_m}=\frac{1}{H+(1-H)\dfrac{T_m}{T_c}} \tag{3-5}$$

【例 3.3】 CPU 执行一段程序过程中，共有 1000 次访存操作，其中在 Cache 命中 950 次。已知主存的存取时间为 100ns，Cache 的存取时间为 10ns。Cache 的命中率是多少？CPU 平均访存时间以及访存加速比各为多少？

解：

命中率 $H=950/1000=95\%$

平均访存时间 $T_a=95\%\times10\text{ns}+(1-95\%)\times100\text{ns}=14.5\text{ns}$

加速比 $S_P=100\text{ns}/14.5\text{ns}\approx6.9$

3.4.2 Cache 的设计要素

本节将逐一讨论 3.1.2 节中提出的关于多层次存储系统的几个问题在 Cache 存储器中的解决策略。

1. 地址映射与地址变换

在 Cache 中，地址映射机制就是解决如何将主存地址空间映射到 Cache 地址空间的问题。简单地说，就是把主存中的内容按照某种规则装入 Cache 中，并建立主存地址与 Cache 地址的映射关系。主存内容按照这种映射关系装入 Cache 后，在执行程序时，应首先将主存地址变换成 Cache 地址，即执行地址变换。地址映射和地址变换是密切相关的两个过程，采用什么样的地址映射方法，就必然采用与这种映射方法相对应的地址变换方法。

为了方便进行地址映射，将主存与 Cache 都分成若干块(也经常被称为“行”)，每块由若干字组成且大小相同。这样，主存地址就可以分为两个部分：块地址(块号)和块内地址。同样，Cache 地址也分为块地址(块号)和块内地址两部分。主存和 Cache 之间以块为单位进行数据的调入、调出，如图 3-41 所示。

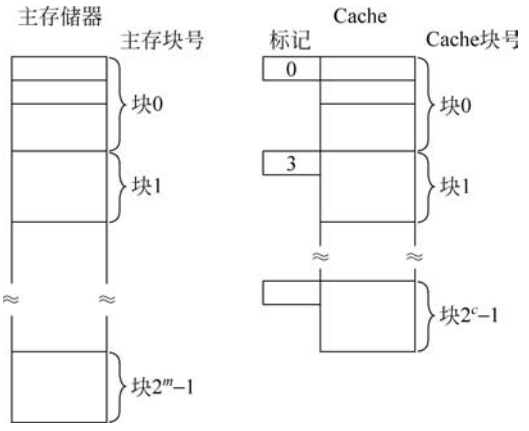


图 3-41　主存与 Cache 间的映射关系

在图 3-41 中，主存分 2^m 块，Cache 分 2^c 块。主存和 Cache 的块大小都是 2^b 字。Cache 中每一块都增加了一个标记字段，用于说明该块是主存中哪一块的副本。在图 3-41 的例子

中,Cache 中块 0 保存的是主存中块 0 的副本,而 Cache 中块 1 保存的是主存中块 3 的副本。在 CPU 发出访存地址时,需要将所访问的主存块号与 Cache 中的标记进行“比较”,才能判断出该地址是否在 Cache 中命中。通常,所有块的标记字段被集中存放,例如下面介绍的几种地址映射方法中的目录表、区表及块表。

常用的 Cache 地址映射方式有全相联映射、直接映射和组相联映射。下面介绍这三种 Cache 地址映射及其地址变换方法。

1) 全相联映射及其地址变换

全相联地址映射是指主存中的每一块都可以映射到 Cache 中的任意块,如图 3-42 所示。这种映射方法是最灵活的,也是 Cache 利用率最高的一种方式,但同时也是成本最高的一种方式。

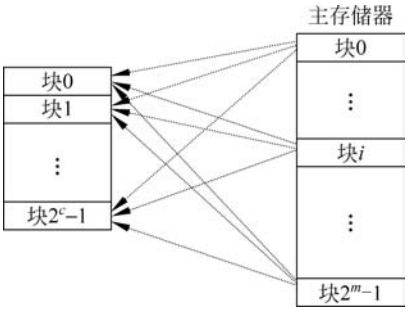


图 3-42　全相联映射方式

在全相联映射方式下,主存地址被分为两个部分:高 m 位表示主存块号(用符号 B 表示),低 n 位表示块内地址(用符号 W 表示)。同样,Cache 地址也分为两个部分:高 c 位表示 Cache 块号(用符号 b 表示),低 n 位表示块内地址(用符号 w 表示)。通常采用目录表记录主存块与 Cache 块之间的映射关系,并将目录表存放在一个相联存储器中。目录表中的每个存储字主要包括三个部分:主存块号、Cache 块号和有效位。有效位表示目录表中主存块号和 Cache 块号建立的映射关系是否有效。目录表共有 2^c 个存储字,即 Cache 中每个块对应目录表相联存储器中一个存储字。

当一个主存块调入 Cache 时,同时将主存块号和 Cache 块号存入目录表中,并将其有效位置“1”。这样,当 CPU 发来一个访存地址时,地址变换就是根据主存地址中的块号查询相联存储器目录表的过程。如果在目录表中找到该主存块号,并且其对应的有效位为“1”,则表示该主存块在 Cache 中命中;否则,未命中,如图 3-43 所示。若命中,则按照目录表中的 Cache 块号和主存块内地址(与 Cache 块内地址相同)作为 Cache 地址访问 Cache。

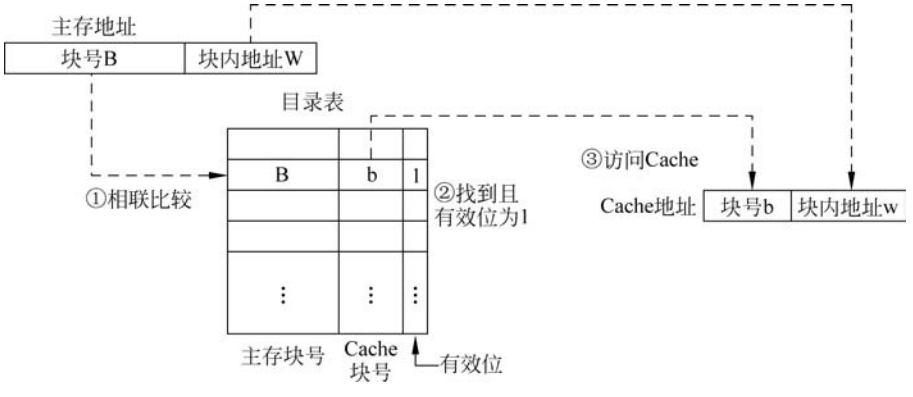


图 3-43　全相联映射的地址变换过程

2) 直接映射及其地址变换

直接地址映射是指主存中的块只能映射到 Cache 中某个固定的块中,主存和 Cache 块

的对应关系可用如下公式表示：

$$b = B \text{ Mod } 2^c \tag{3-6}$$

其中，b 为数据在 Cache 中的块号，B 为数据在主存中的块号。在这种映射方式中，主存第 0 块、第 2^c 块、第 2^{2c} 块、……，只能映射到 Cache 第 0 块，而主存第 1 块、第 2^c+1 块、第 $2^{2c}+1$ 块、……，只能映射到 Cache 第 1 块，以此类推，如图 3-44 所示。

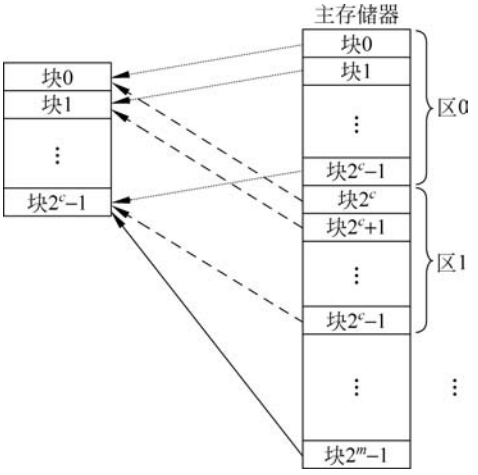


图 3-44　直接地址映射方式

在直接地址映射方式下，主存地址由三部分组成：区号 E（t 位）、区内块号 B（c 位）和块内地址 W（n 位）。通常用区表来保存主存块与 Cache 块的映射关系。区表中的每个存储字主要包括两个部分：主存区号和有效位。有效位表示区表中的主存块是否已经装入 Cache 中。区表中共有 2^c 个存储字。区表通常存放在一个小容量高速存储器中，按地址进行访问。

CPU 发出的访存地址被分解为：主存区号 E、区内块号 B 和块内地址 W 三个部分。当一个主存块调入 Cache 时，将该块在主存的区号 E 存入区表中，并将有效位置“1”。直接相联映射方式的地址变换过程如图 3-45 所示。当 CPU 送来一个访存地址时，首先以主存地址中的 B 字段作为地址访问区表，若区表中对应存储字的有效位为“1”，则将区表中的区号与主存地址的 E 字段进行比较。若相符表示命中；否则未命中。当命中时，由主存地址中的 B 字段确定数据在 Cache 中的块号，由 W 字段确定数据在 Cache 中的块内地址。

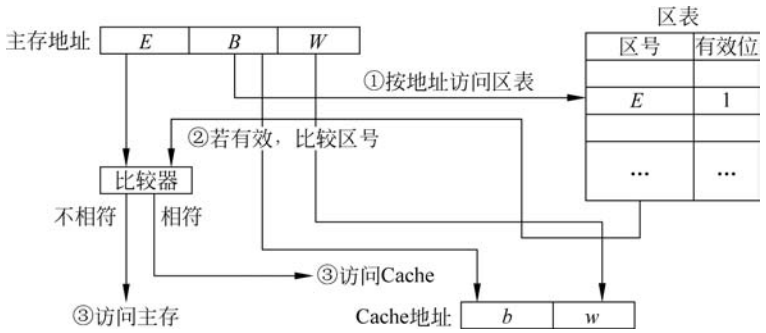


图 3-45　直接地址变换过程

直接地址映射方式的优点是实现简单,地址变换速度快,区表采用 SRAM,价格低。由于主存中的块只能唯一地对应 Cache 中的某个块,因此该机制不够灵活,也使得 Cache 存储空间得不到充分利用,Cache 命中率低。

3) 组相联映射及其地址变换

组相联地址映射是全相联映射和直接相联映射的折中方案,如图 3-46 所示。主存和 Cache 的块都先进行分组,并且主存和 Cache 每组包含的块数相同。在地址映射时,组间采用直接相联映射,组内采用全相联映射,即主存组与 Cache 组之间具有固定的映射关系,但主存组内的块可以自由映射到对应 Cache 组中的任何一块。

在图 3-46 中,Cache 被分为 2^u 组,每组 2^v 块;主存中共有 2^s 区,每个区有 2^u 组,即总共有 2^{s+u} 组。主存中每个区的第 i 组都只能映射到 Cache 中第 i 组,在组内块采用全相联方式,即每个块可以映射到 Cache 第 i 组的任一块。

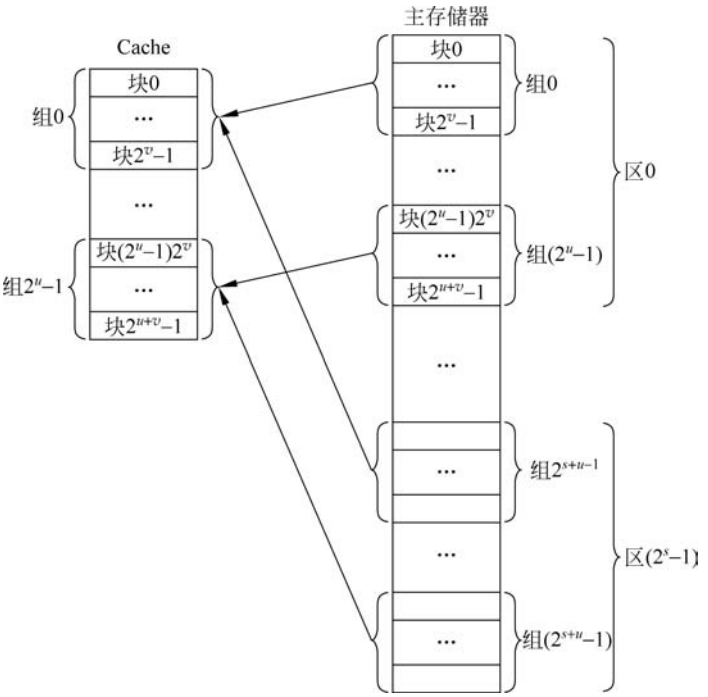


图 3-46 组相联映射方式

在组相联映射方式下,由块表记录从主存地址到 Cache 地址的映射关系。块表由 2^u 个相联存储器(简称组表)构成,每个相联存储器拥有 2^v 个存储字,每个存储字主要包含主存区号 E、主存块号 B、Cache 块号 b 和有效位字段。

CPU 发出的访存地址被分解为:区号 E、组号 G、组内块号 B 和块内地址 W 四个部分;而 Cache 的地址可分解为:组号 g、组内块号 b 和块内地址 w 三个部分。图 3-47 给出了组相联的地址变换过程。

当 CPU 访存时,首先用主存地址中的组号 G 字段作为地址选择组表,然后将主存地址的 E 字段和 B 字段与该组表中所有块标记字段(区号+组内块号)进行相联比较。若命中,则将组表中该项的 b 字段与主存地址 G、W 字段形成访问 Cache 的地址。

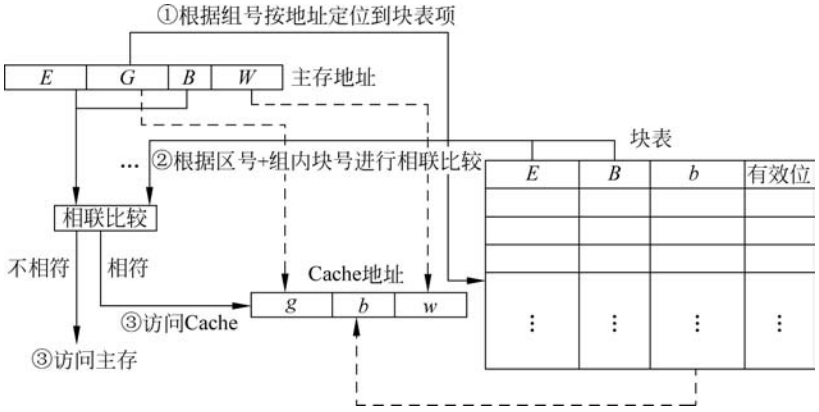


图 3-47　组相联地址变换过程

组相联映射方式可以避免全相联方式下大容量相联存储器的昂贵价格，又可以提高直接映射方式下的 Cache 命中率，因此被广泛采用。

通常，把一个主存块能映射到 Cache 块的数量称为关联度。直接映射方式下的关联度为 1，即每个主存块只能有一个固定的 Cache 块存放；全相联方式下的关联度为 Cache 中块的数量，即每个主存块可以存放在 Cache 中的任意位置；n 路组相联方式（每组 n 块）的关联度为 n，即每个主存块可存放在固定组内的任意块中。当 Cache 大小和主存块大小一定时，关联度越低，则命中率越低。

【例 3.4】 若主存地址为 32 位，块大小为 16 字节，Cache 总共有 4K 个块，那么在以下映射方式下，标记字段的位数是多少？

（1）直接映射方式；

（2）2 路组相联映射方式；

（3）4 路组相联映射方式；

（4）全相联射方式。

解：标记字段的作用可参见图 3-47 中所示的标记字段，简单地说就是为了记录 Cache 块存放的是主存中的哪个块而增加的额外信息。具体地说，就是全相联映射方式中的目录表的主存块号字段，直接映射方式中的区表的区号字段，组相联映射方式中的块表的区号＋组内块号字段。

（1）直接相联映射方式下，Cache 块数为 $4K=2^{12}$，每块大小为 $16=2^4$ 字节，因此标记位所占位数为 $32-4-12=16$ 位。

（2）2 路组相联映射方式下，分了 $2K=2^{11}$ 组，每组 2 块，每块大小为 $16=2^4$ 字节，因此标记位所占位数为 $32-4-11=17$ 位。

（3）4 路组相联映射方式下，分了 $1K=2^{10}$ 组，每组 4 块，每块大小为 $16=2^4$ 字节，因此标记位所占位数为 $32-4-10=18$ 位。

（4）全相联映射方式下，每块大小为 $16=2^4$ 字节，因此标记位所占位数为 $32-4=28$ 位。

【例 3.5】 容量为 64 块的 Cache 采用组相联映射方式，块大小为 128 字节，每 4 块为一组，若主容量为 4096 块。主存地址中分为哪几个字段，各占多少位？

解：采用组相联的主存地址构成为区号、组号、组内块号和块内地址四个部分。

4096/64=64=2^6,因此需要6位来表示区号;每4块为一组,故共有组数=64/4=16=2^4,因此需要4位表示组号。每组4=2^2块,因此组内块号需要2位;每块128=2^7字节,因此块内地址需要7位。

主存地址的位数=6+4+2+7=19位。

2. 替换算法

替换算法主要用于解决3.1.2节中的第四个问题,即当有新的块需要从主存调入Cache,而Cache中没有空闲块时,可以将哪一块数据替换出Cache。在直接映射的Cache中,由于主存中的块只能调入Cache中的某一个特定位置,因此替换策略很简单。在组相联和全相联映射的Cache中,由于主存中的块可以调入Cache中的多个位置,因此就需要设计合理的替换算法,来保证Cache具有较高的命中率,即替换出去的块近期不会被访问。常用的替换算法主要包括以下几种。

1) 随机算法(RAND法)

这种算法不考虑Cache中各块的使用情况,随机地选择一个块作为替换对象。随机算法在硬件上容易实现,且速度快;缺点是在替换时既没有利用程序的局部性特点,也没有考虑历史上块地址流的分布情况,随意替换出的数据很有可能马上又要使用,从而降低了命中率和Cache的工作效率。

2) 先进先出算法(First-In-First-Out,FIFO)

这种算法也不考虑各块的使用情况,根据每块调入Cache的时间,当需要替换时,将最先调入Cache的主存块替换出去,由于该算法不需要记录各块的使用情况,因此算法硬件实现较容易,系统开销较小。例如,Cache每块都设置一个计数器,当某块被装入或者替换时,该块的计数器值清0,其他块的计数器值加1,当需要进行替换操作时,将计数器值最大的块替换出去。这种方法不考虑主存块在Cache中的使用规律,有可能造成马上需要使用的块被调出,从而影响Cache的命中率。对于线性程序来说,FIFO算法有较高的Cache命中率。

3) 近期最少使用算法(Least Recently Used,LRU)

这种算法以Cache中每块的历史使用情况为依据,在需要替换时将近期最少使用的块替换出去。这种算法比较好地反映了程序的局部性原则,特别对于循环程序有较高的命中率。由于LRU算法需要记录Cache中各块的使用情况,因此实现相对复杂、开销较大。通常采用计数器的方式记录每个块被使用的情况。

4) 最不经常使用算法(Least Frequently Used,LFU)

这种算法的基本思想是将访问次数最少的块替换出去。这种算法与LRU算法类似,也需要为Cache中每个块设置一个计数器,以记录每个Cache块的访问次数。当需要执行替换时,将计数器值最小的块替换出去。同时将所有块的计数器值清0。这种方法的缺点是不能反映最近的访问情况,只能反映两次替换时间间隔内的使用情况。

【例3.6】 假设程序在主存中有5块(P1到P5),在Cache中有3块。CPU执行程序的顺序为P1,P2,P1,P4,P5,P4,P1,P2,P3,P4,P1,P2。请分别计算当采用FIFO算法和LRU算法时的命中率。本题中地址映射采用全相联映射方式。

解:采用FIFO算法的Cache中存放程序的变化情况如下所示(带"√"表示命中):

程序执行顺序	1	2	3	4	5	6	7	8	9	10	11	12
Cache 块 1	P1	P1	P1√	P1	P5	P5	P5	P5	P3	P3	P3	P2
Cache 块 2		P2	P2	P2	P2	P2	P1	P1	P1	P4	P4	P4
Cache 块 3				P4	P4	P4√	P4	P2	P2	P2	P1	P1

由此可看出总共命中 2 次，因此采用 FIFO 算法的命中率为(2/12)×100%=17%。

采用 LRU 算法的 Cache 中存放程序的变化情况如下所示：

程序执行顺序	1	2	3	4	5	6	7	8	9	10	11	12
Cache 块 1	P1	P1	P1√	P1	P1	P1	P1√	P1	P1	P4	P4√	P4
Cache 块 2		P2	P2	P2	P5	P5	P5	P2	P2	P2	P1	P1
Cache 块 3				P4	P4	P4√	P4	P4	P3	P3√	P3	P2

由此可看出总共命中 5 次，因此采用 LRU 算法的命中率为(5/12)×100%=42%。

3. Cache 的写操作及一致性

由于 Cache 中的内容只是主存内容的副本，必须保证 Cache 与主存内容的一致性，显然一致性问题主要涉及的是写操作，即更新主存内容的算法。

造成 Cache 与主存内容不一致的原因主要包括以下两种情况：

(1) CPU 对 Cache 执行写操作，但没有立即写主存。

(2) I/O 设备或 I/O 处理机写主存，但没有同时写 Cache。

下面以 CPU 访存操作为例，介绍常见的写策略。

1) 写直达法(Write Through)

写直达法是指 CPU 在执行写操作时，将数据同时写入主存和 Cache。这样，当 Cache 中某个块需要被替换出 Cache 的时候，不需要执行写主存的操作。这种方式的优点是硬件实现简单，并且 Cache 和主存的内容总是一致的，缺点是频繁访问主存将会降低平均访存速度。

2) 写回法(Write Back)

写回法是指 CPU 在执行写操作时，数据只写入 Cache，不同时写入主存。只有当 Cache 中的某个块需要替换出 Cache 时，才把修改过的 Cache 块写回主存。

写回法又可细分为以下两种方法。

(1) 简单写回法：不管块是否被更新，都进行写回操作。

(2) 采用标志位写回法：只在块被更新过时，才进行写回操作。

写回法的优点是可以减少访问主存的次数，有利于提高平均访存速度。缺点是增加了 Cache 的复杂性，并且存在主存与 Cache 内容不一致的问题，影响系统的可靠性。

下面按照 CPU 执行写操作时在 Cache 中是否命中，来讨论上述写策略在执行写操作时的具体实现策略。

(1) 若命中：可直接对 Cache 块进行写操作，然后根据所采用的更新主存的算法，决定何时对主存块的内容进行更新。

(2) 若不命中：无论是写回法还是写直达法都存在“写时是否取”的问题。这个问题的解决，主要有以下两种方法：

- 不按写分配法：当 Cache 写不命中时，只执行对主存的写入操作。
- 按写分配法：当 Cache 写不命中时，首先执行写主存操作，然后将该主存中的块从主存调入 Cache。

写回法一般采用按写分配法，写直达法一般采用不按写分配法。

4. 多级 Cache

早期的计算机系统中通常只有一级 Cache。近年来，为了提高系统的性能，越来越多的系统中采用多级 Cache。在系统中设置多级 Cache 时，一般是按照 Cache 的位置和保存的内容不同进行设置，下面介绍两种最常见的多级 Cache。

1) 片内和片外 Cache

随着大规模集成电路技术的发展，Cache 通常被集成在 CPU 芯片中，称为片内 Cache。片内 Cache 的最大优点是，如果 CPU 要访问的数据在片内 Cache 中命中，则 CPU 的访问将在 CPU 内部进行，不需要通过系统总线来进行数据传输，因此存取速度非常快。并且，由于不使用系统总线，还可以减轻总线负载，降低总线数据传送的冲突。

但是，由于成本和制造工艺等的限制，片内 Cache 的容量通常不会很大。为了增加系统中 Cache 的容量以提高命中率，通常在 CPU 芯片外再增加一个容量较大的 Cache。这样，CPU 芯片内部的 Cache 被称为第一级 Cache(L1 Cache)，CPU 外部的 Cache 被称为第二级 Cache(L2 Cache)。在这种两级 Cache 结构中，第二级 Cache 容量比第一级要大得多，在第一级 Cache 中保存的信息也一定保存在第二级 Cache 中(多层次存储器系统中的包含性原则)。当 CPU 访问第一级 Cache 未命中时，才去访问第二级 Cache。

随着芯片技术的发展，目前大多数处理器将 L2 Cache 也集成到了 CPU 芯片中，并且在 CPU 芯片外设置第三级 Cache(L3 Cache)。当然，技术的进步使得有些 CPU 将 L3 Cache 也集成到了 CPU 内部。依此思路，系统也可以再增加更多级的 Cache。这种趋势导致 Cache 的层次将越来越多，Cache 的设计将变得更加复杂。

在有两级 Cache 的系统中，CPU 的平均访存时间为

$$T_a = h_1 C_1 + (1-h_1)h_2 C_2 + (1-h_1)(1-h_2)M \tag{3-7}$$

式中，h_1、h_2 分别为一级和二级 Cache 的命中率，C_1、C_2 分别为一级和二级 Cache 的存取时间，M 为主存的存取时间。

2) 数据 Cache 和指令 Cache

存储单元中存放的内容可以是指令或者数据。根据 3.1.2 节中介绍的程序的局部性原理，指令的存放呈现出较强的局部性特征，即下一条将执行的指令很可能与当前正在执行的指令存放在相邻的存储单元中。并且大多数的循环子程序的循环范围很小，通常可以将整个循环代码都放入一个 Cache 块中。

但是，CPU 对于数据的访问通常呈现较强的随机性，因此若将指令和数据混合存放在同一个 Cache 中，将影响 Cache 的命中率，进而影响系统的效率。因此，很多计算机系统在 Cache 设计时采用了数据和指令相分离的策略，分别为指令 Cache(I-Cache)和数据 Cache(D-Cache)。

3.5 辅助存储器

3.5.1 辅助存储器概述

主存储器由于价格昂贵通常容量不会很大，并且早期的 SRAM 和 DRAM 都具有易失性，无法在断电后保存信息。因此，在计算机系统中引入了辅助存储器，以扩大整个存储系统的容量，并提供长期保存信息的能力。常见的辅助存储器包括磁盘、磁带、光盘、U 盘等。硬磁盘（简称硬盘）通常被作为外存来使用，它和主存一起构成虚拟存储器。软磁盘（简称软盘）、光盘、磁带等通常用作离线存储器。

下面首先以最常用的磁表面存储器为例说明辅助存储器的基本工作原理。

1. 磁表面存储器的基本工作原理

顾名思义，磁表面存储器的存储介质是某种涂有磁性材料的载体，载体的形状可以是各式各样的。例如，磁盘采用圆盘状载体，磁带则采用带状载体。

磁表面存储器通过磁头来完成数据的读写，磁头能实现电磁信号的变换。按照读写时磁头是否与磁记录介质接触，可将磁头分为接触式磁头和浮动式磁头两种。

接触式磁头的结构简单，但会因为磨损而降低磁头和介质的使用寿命。软盘和磁带由于采用软性介质，只能采用接触式磁头。浮动式磁头和介质间存在一定的间隙，读写时不会磨损磁头和介质。硬盘主要采用浮动式磁头。浮动式磁头对磁介质进行读写的过程如图 3-48 所示。

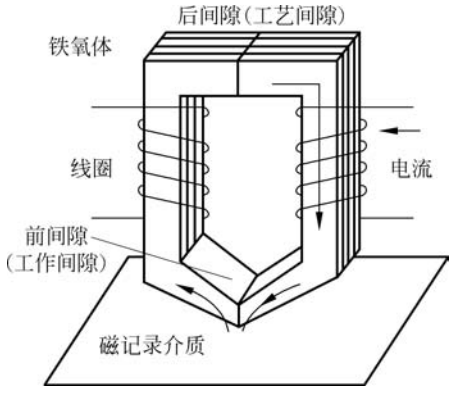


图 3-48　磁头对磁介质读写示意图

磁头是在一个很小的软磁体（如铁氧体）上绕上线圈而制成的一个具有磁隙的装置。在执行写操作时，线圈中通过一定方向的电流，磁芯内就产生了一定方向的磁通。由于磁头和介质间的间隙非常小，磁力线穿过磁介质的表面将磁头下方的区域磁化，该区域非常小，被称为磁化单元。电流的方向不同，磁化单元被磁化的极性就不同，这样便可以记录（写入）二进制位的“0”和“1”。

在执行读操作时，当磁头经过介质上的磁化单元时，由于磁头是良好的导磁材料，磁化单元的磁力线很容易通过磁头而形成闭合磁通回路，产生感应电动势 e，其极性与磁通变化的极性相反。磁化单元上的磁化状态不同，感应电动势也不同，这样就可以区别出磁化单元上存储的是“0”还是“1”。当磁化单元被磁化后，可以多次读出而不被破坏。

早期的磁头包括亚铁盐类磁头、隙含金属（Metal In GAP，MIG）磁头和薄膜磁头，这些传统的磁头是用线圈缠绕在磁芯上制成的，通过电流方向变化来区分“1”和“0”，并且采用读写合一的方式。由于硬盘在进行数据传输时，读操作比写操作快得多，这种方式造成了硬盘设计的局限性。为解决这个问题，1991 年 IBM 发明了磁阻磁头（Magnetoresistive heads，MR 磁头），这种磁头采用读写分离式的结构，写入磁头仍采用传统的磁感应磁头，读取磁头则采用新型的磁阻磁头，这就是所谓的感应写、磁阻读。这种读写分离的方式增加了硬盘设

计的灵活性,可以大大提升读写性能。并且由于MR磁头是通过阻值变化区分“1”和“0”,因此对信号的变化非常敏感,读取数据的准确性较高;由于读取的信号幅度与磁道宽度无关,因此磁道较窄,盘片密度较高。MR磁头已得到广泛应用,而采用多层薄膜结构和磁阻效应更好的材料制作的巨磁阻磁头(Giant Magnetoresistive heads,GMR)也逐渐普及。

2. 磁表面存储器的记录方式

磁表面存储器的可靠性、记录密度等性能不仅和磁介质及磁头的物理特性有关系,还取决于所采用的磁记录方式(也被称为编码方式)。在磁表面存储器中,磁编码方式就是按照一定的规则将二进制位串变换为记录介质上相应磁通翻转形式。不同编码方式的存储性能不同,编码方式设计的目标主要有:更高的编码效率、更高的自同步能力和更高的读写可靠性等。

编码效率是指位密度与磁化翻转次数之比,例如下面介绍的不归零制的编码效率为100%,调相制编码和调频制的编码效率为50%。自同步能力指单个磁道读出信息提取同步脉冲的难易程度。

下面介绍几种常见的编码方式,它们的电流波形如图3-49所示。

(1) 归零制(RZ)。磁介质在写入前处于未磁化状态,磁头线圈中通正向电流脉冲时写“1”,通负向电流脉冲时写“0”,每写一位,电流都要归零。特点:记录密度低,抗干扰能力差,具有自同步能力。

(2) 不归零制(NRZ)。磁头线圈中通正向电流时写“1”,通反向电流时写“0”,每写一位,电流不需归零。特点:相对于归零制,抗干扰性能好,记录密度提高,但可能造成传播误码,无自同步能力。

(3) “见1就翻”的不归零制(NRZ1)。写“1”时,电流反向一次;写“0”时,电流保持不变。特点:与不归零制类似,但不传播误码。

(4) 调相制(PM)。写“1”时,电流相位在周期1/2处由负变正(或者正变负);写“0”时,电流相位在周期1/2处由正变负(或者负变正);连续写相同代码时,周期起始处电流变化一次。特点:不传播误码,具有自同步能力,抗干扰能力强。

(5) 调频制(FM)。写“1”时,电流在周期起始和中心处各反向一次;写“0”时,电流在周期起始处反向一次。特点:记录密度高,具有自同步能力,可靠性高。

(6) 改进调频制(MFM)。写“1”时,电流只在周期中心处反向一次;写单个“0”时,电流不变;连续写“0”时,电流在周期起始处翻转一次。特点:记录密度比调频制提高一倍,具有自同步能力。

3. 校验码

磁表面存储器由于磁介质的缺陷、灰尘等原因,使其很容易出现差错。为此,在磁表面存储器中通常使用校验码来发现和纠正在存储和传输过程中的差错。所谓校验码就是为了检测差错而在数据后面添加上的冗余码。常见的校验码有奇偶校验码、海明校验码、循环冗余检验等,这些校验码在冗余码的长度、检错和纠错能力等都不尽相同。

循环冗余检验(Cyclic Redundancy Check,CRC)是磁表面存储器中最常使用的校验码。CRC校验使用多项式码,也称为CRC码。CRC码的检错原理在《数字逻辑》《计算机网络原理》等教材中均有讲述,在此不再展开讨论。

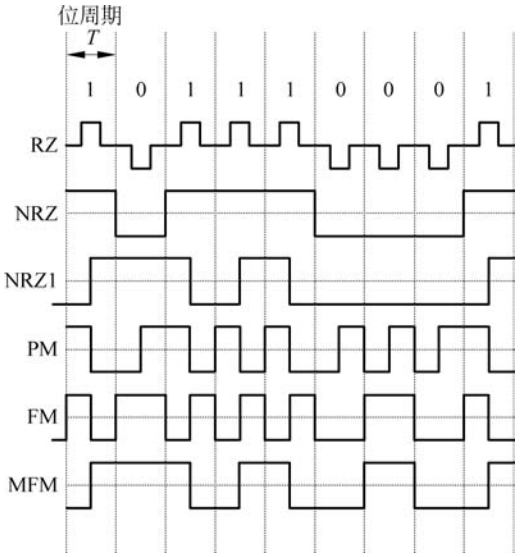


图 3-49　磁记录方式写入电流波形

3.5.2　硬磁盘存储器

硬磁盘存储器简称硬盘，是一种利用磁记录技术在涂有磁介质的旋转圆盘上进行数据存储的辅助存储器，是一种典型的磁表面存储器。它具有存储容量大、数据传输率高、存储数据可长期保存等特点，是计算机系统中最常用的联机辅助存储器，通常与主存储器构成虚拟存储器以扩充主存的容量。

1. 硬盘的结构及工作原理

图 3-50 给出了一个硬盘的内部结构。其中，图 3-50(a)为硬盘的实物解剖图，图 3-50(b)为硬盘逻辑结构示意图。

从图 3-50 中可以看出，硬盘中包含了外壳、密封罩、滤尘器等密封防尘装置，这些保护装置避免灰尘的进入，保证磁盘组能得到高质量的磁表面和空气。硬盘中最主要的控制部件是磁盘驱动器和磁盘控制器。

磁盘驱动器是磁盘设备的主体，其主要功能是将磁头定位到需要进行读/写数据所在的位置，并进行读/写操作。磁盘驱动器由主轴系统、定位驱动系统、读/写控制系统三部分组成。

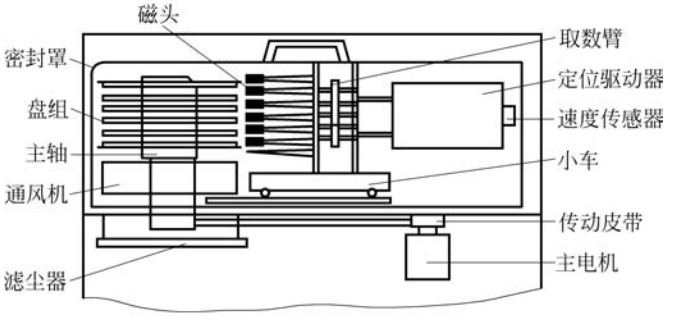
(1) 主轴系统：负责带动盘片组进行旋转，由盘组、主轴、主电机、传动皮带等组成。其中，主电机是带动盘片组进行旋转的主要部件，是决定磁盘读/写速度的关键因素。

(2) 定位驱动系统：负责驱动磁头沿径向运动，由取数臂、小车、速度传感器、定位驱动等部件组成。

(3) 读/写控制系统：负责控制数据的读/写操作。由磁头、磁头选择(译码)电路、读/写电路等部分组成。盘组有一组磁盘片组成，每个盘片的上下两面都能记录信息，因此每个盘面都有一个磁头，所有这些磁头都固定在取数臂上。

磁盘控制器是主机和磁盘驱动器间的接口部件，其主要作用是控制主机与硬盘间数据的交换方式，通常以接口板的形式插在主机板总线插槽上。磁盘控制器包括以下两个接口：

(a) 硬盘实物解剖图



(b) 硬盘结构示意图

图 3-50　硬盘组织结构

设备级接口,即与磁盘驱动器相连的接口,如图 3-51 中 A 点位置;系统级接口,即与主机相连的接口,由系统总线界定,如图 3-51 中 B 点位置。通常,硬盘接口就是指系统级接口,常见的硬盘接口有 IDE 接口、SATA 接口、SCSI 接口和光纤通道等。

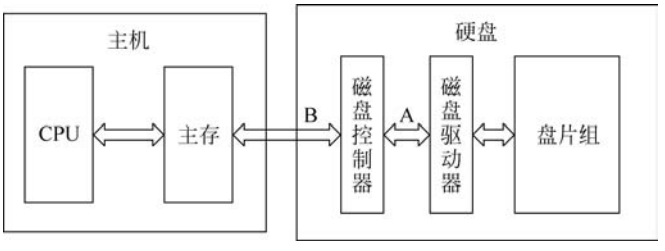


图 3-51　硬盘与主存的连接

另外,在硬盘中通常也会设置一定数量的缓存,其主要目的是解决硬盘与主存读/写速度不匹配的问题。

由于硬盘磁表面的纯度和空气质量的好坏,将直接影响到磁道上信息的存储密度,因此大多数磁盘在出厂时已经进行了密封,以防止灰尘的进入。这种硬磁盘被称为“温彻斯特磁盘”(简称为温盘)。温盘的基本组成包括磁头、盘片组及其控制部件,并且把它们密封起来。

磁盘片固定并能高速旋转，磁头沿盘片径向移动完成定位操作。磁头对盘片接触式启停，但工作时不与盘片直接接触，即采用悬浮动式磁头。

2. 硬盘中数据的存放

一块硬盘通常由安装在一个轴上的一个或多个磁盘片组成，每个磁盘片的表面（单面或双面）涂有磁性材料作为记录介质，在进行数据读/写操作时，磁盘片会随轴高速旋转来进行定位，并通过磁头在磁层上进行读/写操作。

硬盘通常由一组盘片组成，称为硬盘的盘片组。盘片组固定在一个轴上（主轴），所以盘面中心是空心的，不能记录信息。每个盘面的有效记录区又被划分为数目相等、由内向外排列的同心圆，这些同心圆的间距相等，被称为磁道。磁头靠近主轴接触的表面，即线速度最小的地方，是一个特殊的区域，它不存放任何数据，称为启停区或着陆区，启停区外是数据区。通常从最外侧的磁道开始编号，起始磁道号为 0。在硬盘中通过 0 磁道检测器来完成硬盘磁道的初始定位。

这样，不同盘面上具有相同磁道号的磁道就形成了一个圆柱，称之为硬盘的圆柱面或柱面（Cylinder）。显然，磁盘的柱面数与一个盘面上的磁道数是相等的。每个柱面上的磁头（Head）从上到下从 0 开始编号。在对文件进行写入操作时通常是按柱面进行，即磁头读/写数据时首先在同一柱面内从 0 磁头开始进行操作，依次向下在同一柱面的不同盘面即磁头上进行操作，只在同一柱面所有的磁头全部读/写完毕后磁头才移动到下一柱面。这是设计的目的是提高读/写效率，因为选取不同的磁头是通过电子切换完成，而要选取不同的柱面则必须进行机械动作来完成磁头的移动。

为了方便读写，一个磁道通常被分成若干段的圆弧线，称之为扇区（Sector）。其实将扇区称为扇段更加合适，因为它并不是整个扇形区域，而是磁道上的一段弧线。一个磁道上的所有扇区是等长的，早期的扇区通常可存放 512B 的数据块，硬盘以扇区为单位进行存取。因此，硬盘可寻址的最小单位为扇区，而不是像主存那样使用字节或字作为单位。

扇区的定位和划分有硬件（称为硬分区）和软件（称为软分区）两种方式。无论是哪种方式，为标识一个磁道信息的起始，都在磁道的起始处打一个索引孔，通过光电检测的方法获得一个电脉冲，称为“0 索引”。这样，随着磁盘的旋转，就很容易识别扇区的扇区号。硬分区通常使用定长记录格式，即每个扇区中存放的数据块大小固定。这种记录格式中，虽然外圈磁道的周长大于内圈磁道的周长，但是却存放着相同的字节，因此这种方式简单但记录区的空间利用率不高。软分区可实用定长和不定长的记录格式。不定长记录格式是指扇区（通常被称为记录块）中存放的数据块大小可变；在这种记录格式中，外圈磁道可以比内圈磁道存储更多的字节，因此灵活性好、空间利用率也较高。

在对硬盘上存储的信息进行定位时，常用如下的地址格式：

驱动器号	柱面号	盘面号	扇区号

其中，驱动器号用于系统中有多台硬盘时，指出数据所在的硬盘编号；柱面号即磁道号，说明数据所在的磁道；盘面号也可以用磁头号表示，用于说明当前读/写操作所使用的磁头；扇区号用于说明当前是对哪个扇区进行读/写操作。磁盘在进行读/写操作时，磁盘驱动器首先要进行寻道操作，即确定需要操作的柱面。这时，定位驱动系统带动磁头做径向

运动。完成定位操作后,主轴系统带动磁盘组进行旋转,将需要进行读/写的扇区移动到磁头下(上)方。最后,读/写控制系统控制磁头进行读/写操作。

早期硬盘在进行数据读写时地址是按照CHS(Cylinder,Head,Sector)进行组织的。但随着对硬盘容量需求的逐步增大,如何有效利用磁盘盘面是增加磁盘容量的有效方式,然而CHS方式导致外圈磁道记录密度过低。为了提高外圈磁道的记录密度,现代硬盘采用ZBR(Zone bit Recording)记录方式。在该方式中,外圈磁道被划分成更多的扇区,这样可以有效提高外圈磁道的记录密度,从而提高磁盘的整体记录容量。图3-52示意了采用ZBR记录方式的盘面信息分布。

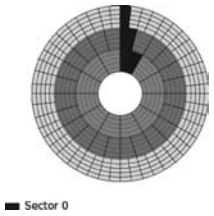


图3-52 硬磁盘的ZBR记录方式

在ZBR记录方式下,磁盘外圈磁道的道容量大于内圈的道容量,在硬盘转速不变的前提下,外圈的读/写速率高于内圈的读/写速率。通常,磁盘的0磁道位于最外圈,在磁盘安装操作系统时一般位于0磁道附近。随着写入数据的增多,逐步开始使用内圈磁道,这就是新安装操作系统后计算机运行速度较快的原因之一。

3. 硬盘的主要技术参数

与主存一样,硬盘的性能也主要从容量和速度两个方面进行评价。

1) 容量

作为辅助存储器中用于扩大整个系统存储容量的设备,容量是硬盘最主要的性能参数。一台硬盘的总容量是指所有盘面上全部磁道中存储信息的总和。

硬盘容量与其记录密度是密切相关的,所谓记录密度是指单位面积或单位长度上可以存储的二进制位数,通常使用道密度和位密度来衡量。

道密度是指沿着盘面径向上单位长度内的磁道数量,常用单位为TPI(track per inch,磁道数/英寸)。一个盘片的磁道总数=道密度×盘片有效半径。

位密度是指盘面磁道上单位长度所能记录的二进制位数,常用单位为bpi(bit per inch,位/英寸)。一个磁道的总容量=位密度×磁道的周长。

下面以定长记录格式为例说明硬盘容量的计算方法。CHS硬盘的总容量可以使用以下公式来计算:

$$\begin{aligned}\text{硬盘容量} &= \text{硬盘个数} \times \text{磁盘记录面数} \times \text{磁道数} \times \text{磁道容量}\\ &= \text{硬盘个数} \times \text{磁盘记录面数} \times \text{道密度} \times \text{盘片有效半径} \times\\ &\quad \text{位密度} \times \text{磁道的周长}\end{aligned}$$

上面计算出的硬盘容量被称为硬盘的非格式化容量,也就是硬盘在物理上可以记录的二进制位总数。但是,磁盘(包括硬盘和软盘)必须格式化后才能使用,磁盘的格式化分为物理格式化和逻辑格式化。物理格式化又被称为低级格式化,是对磁盘的物理表面进行处理,在磁盘上建立标准的磁盘记录格式,包括划分磁道和扇区、为每个扇区标注地址和扇区头标志等;逻辑格式化又被称为高级格式化,是在磁盘上建立一个系统存储区域,包括引导记录区、文件目录区、文件分配表等,这个工作由操作系统的文件管理系统完成,例如FAT32、NTFS、ext4等。

由于增加了很多管理位,因此格式化后的容量小于非格式化容量,称为格式化容量。格式化容量的计算公式如下(仍以定长记录为例):

格式化磁盘容量＝硬盘个数×磁盘记录面数×磁道数×每道扇区数×扇区容量

【例 3.7】 某磁盘组共有 2 个磁盘，每个磁盘有 5 个盘片，盘片采用双面记录，并且盘片组的最上及最下两个面不用。盘片存储区域的内径为 22cm，外径为 33cm，道密度为 40 道/cm，内层位密度为 400 位/cm，磁盘组采用定长记录方式。问磁盘组总的存储容量是多少？

解：由于采用定长方式，每个磁道存储的位数相等，因此

每个磁道存放的二进制位数＝位密度×磁道的周长＝$400\times2\times\pi\times(22/2)$b

＝27 632b＝3454B

磁道数＝道密度×盘片有效半径＝$40\times(33-22)/2=220$

每个盘面容量＝磁道数×磁道容量＝220×3454B＝759 880B

总容量＝磁盘个数×盘面数×每个盘面容量

＝$2\times(5\times2-2)\times759\,880$B＝12 158 080B

≈11.59MB

2）传输速率

从前面介绍的硬盘读/写过程可以知道，硬盘的传输速率主要取决于以下因素：磁头定位到磁道的时间、需要访问的扇区旋转到磁头下（上）的时间以及磁头进行读/写操作所需的时间。这些因素又主要由平均寻址时间和数据传输率来衡量。

（1）平均寻址时间。

平均寻址时间是指磁头从起始位置到达目标磁道位置，并且定位到目标磁道上的目标扇区所需的平均时间。平均寻址时间由平均寻道时间和平均等待时间组成，即

平均寻址时间＝平均寻道时间＋平均等待时间

平均寻道时间是指磁头到达目标数据所在磁道的平均时间。磁头平均寻道时间主要决定于磁头动力臂的运行速度，单位为毫秒（ms）。目前，主流 SCSI 和 SATA 硬盘的平均寻道时间都在 5ms 以下。

平均等待时间是指磁头到达目标磁道后，等待所要访问的扇区旋转至磁头下（上）方所需的平均时间。平均等待时间一般取盘片旋转一周所需时间的一半。单位也为毫秒（ms），即平均等待时间＝1/转速/2。这样转速为 7200r/min 的磁盘的平均等待时间＝$(60/7200)/2$s≈4.2ms。

硬盘转速是指单位时间内硬盘盘片旋转的圈数，单位为 r/min（revolutions per minute，转数/分钟）。显然，硬盘转速越高，平均等待时间越短。常见的硬盘转速有 7200r/min、10000r/min、15000r/min 等。

（2）数据传输率。

硬盘数据传输率是指在读/写数据时单位时间内的数据传输量，通常单位用字节/秒（B/s）。硬盘数据传输率又可分为内部数据传输率和外部数据传输率。

内部传输率也称为持续传输率，它反映了硬盘缓冲区未用时的性能。内部数据传输率主要取决于硬盘的转速，即硬盘的内部数据传输率＝磁道容量×转速。显然，硬盘转速越快，数据传输速率就越高。

外部传输率也称为突发数据传输率或接口传输率，是指系统总线与硬盘缓冲区之间的数据传输率。外部数据传输率与硬盘接口类型和硬盘缓存的大小有关。

4. 硬盘数据的存储格式

在软盘和小容量硬盘中大都采用定长记录格式；在大容量硬盘中，可采用定长或不定长记录格式。

早期的定长记录格式中，多采用512B的扇区格式，如图3-53所示。每个扇区包含间隙(Gap)、同步(Sync)、地址标记、数据和纠错码(ECC)几个部分。其中，数据部分的长度为512字节，控制信息的总长度为65字节。在控制信息中，前三部分的总长度为15字节，被称为前导区，用于在读/写之前对磁头进行同步。间隙部分用于分隔扇区；同步部分用于标志扇区开始并提供计时对齐；地址标记部分包含可识别扇区号和位置的信息，还可提供扇区本身的状态。数据部分后面紧接着是纠错码ECC部分，ECC部分长度为50字节，包含用于复原受损数据的纠错代码。从图3-53可以看出，在连续的两个扇区之间设置了固定宽度的间隙，间隙不仅可以隔离不同的扇区，更重要的是为了保证磁头能准确地定位到每个扇区的开始位置。

图3-53　512字节扇区的记录格式

随着区域密度的增加，512B扇区在硬盘表面上占用的空间比率越来越小，同样大小的介质缺陷对总体数据负载损害的百分比就增加了，这使得错误纠正变得更加困难。为此，国际硬盘设备与材料协会(International Disk Drive Equipment and Materials Association，IDEMA)于2009年提出了一种称为“高级格式化”的磁盘标准，它将硬盘扇区大小从512字节增加到4096(4K)字节。IDEMA还宣布，自2011年1月1日起，所有的SATA接口的新硬盘产品，都将支持高级格式技术。高级格式化标准的4K字节扇区也包含间隙、同步、地址标记和纠错码部分，除了将纠错码字段增加到100字节外，其他三个控制信息部分的长度不变，控制信息的总长度由原来的65字节增加到了115字节。这样，扇区格式化效率从89%(512/(512+65))提高到了97%(4096/(4096+115))。

以希捷科技在2010年12月发布的首款基于高级格式4K扇区硬盘为例，示意性说明这类硬盘的相关性能参数。该硬盘型号为Barracuda Green系列，容量为2TB，采用3盘片6磁头设计，5900r/min转速，64MB缓存，数据传输率为144MB/s，采用SATA3.0(6Gb/s)技术。

采用不定长记录格式时，一个磁道内的信息由若干个记录组成，图3-54给出了IBM 2311盘的不定长记录格式。

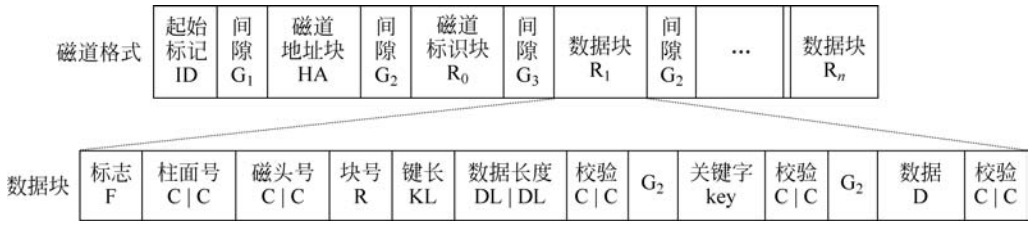


图3-54　IBM 2311盘的不定长度磁道记录格式

其中，起始标志 ID，也被称为索引标志，用于标识磁道的起点。间隙 G1，用于将连续的磁道划分为不同的区，以便进行定位和同步。磁道地址块 HA 用来标识磁道的状态，包括磁道是否完好，磁道的物理地址（柱面地址和磁头号），柱面逻辑地址，磁头逻辑地址和校验码。间隙 G2，用于将连续的磁道划分为不同的区，以便进行定位和同步。零号记录 R0，它是用户用来说明本磁道的有关状态，如果磁道发生故障，将依据 R0 提供的信息实现磁道替换。间隙 G3，包含一个地址标志，用于说明后面紧跟着的是数据记录块。数据记录块 Ri：由计数区、关键字区和数据区三部分组成。其中，计数区包含本区瑕疵、物理地址、标志、ID 识别区、记录号、本记录的关键字长度、本记录内数据长度、校验码。ID 识别区又包含磁道状况、好的原始道、坏的原始道、好的替补道、坏的替补道等地址。关键字区是数据的识别信息，如顺序号、密码及校验码。数据区用来记录用户数据，允许使用不同长度，最后是校验码。在进行磁盘格式化时，可以发现坏道并将其替换。

【例 3.8】 某磁盘存储器转速为 3000r/min，共有 4 个有效记录面，每道记录信息为 12 288B。最小磁道直径为 230mm，每毫米 5 道，共有 275 道，扇区大小为 512B。问：

(1) 磁盘存储器的存储容量是多少？

(2) 最高位密度与最低位密度是多少？

(3) 磁盘数据传输率是多少？

(4) 平均等待时间是多少？

(5) 给出一个磁盘地址格式方案。

解：磁盘存储器的存储容量与其结构密切相关，因此，

(1) 磁盘存储容量 ＝总盘面数×总柱面数×道容量

＝4 面×275 道×12 288B＝13 516 800B≈12.89MB

(2) 最高位密度既最内道的位密度，因此，

最高位密度＝道容量/最内道周长

＝12 288B/230mm×π≈17B/mm≈136 位/mm

最低位密度既最外道的位密度，由于磁盘的外直径未知，因此需先算出外径。即

最大磁道直径＝最内径＋有效记录区宽度×2

＝230mm＋275 道/5 道×2＝230＋110＝340mm

最低位密度＝道容量/最外道周长

＝12 288B/340mm×π≈11B/mm≈92 位/mm

(3) 数据传输率＝道容量×转速

＝12 288B×3000/60＝614 400B/s＝614.4KB/s

(4) 平均等待时间既磁盘转半圈的时间，因此，

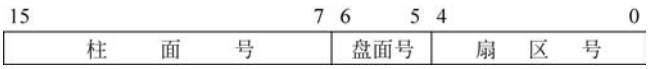
平均等待时间＝1/转速/2＝60/3000/2＝10ms

(5) 磁盘地址格式与信息在盘面上的分布方式有关，由于数据是按柱面、盘面、扇区划分的，因此地址中应含有这些编号。首先确定未知的扇区个数，

扇区数＝12 288B/512B＝24 区，则扇区号取 5 位地址。

同理，4 个有效记录面需 2 位地址，275 道需 9 位地址，则共需 9＋2＋5＝16 位地址。

磁盘地址格式安排如下：



3.5.3 磁盘阵列

由硬盘构建的虚拟存储器层次大大扩充了计算机系统可访问的存储空间,但是硬盘在存取速度上远远跟不上CPU速度的发展,在容量、可靠性等方面也难以满足计算机系统对存储系统的要求。

为了进一步提高硬盘的容量、速度及可靠性,RAID(Redundant Array of Independent Disk,独立冗余磁盘阵列)技术应运而生。冗余磁盘阵列技术诞生于1987年,由美国加州大学伯克利分校提出。其基本思想是将 n 台硬盘组合起来作为一个逻辑硬盘来使用,这些硬盘在RAID控制器的统一控制下协同工作,从而提供比单个硬盘更高的存储性能并提供数据备份等技术。这些硬盘和RAID控制器及一些电源等附件通常被集成在一个硬盘机柜中,通过标准的硬盘接口(如SCSI、SATA等)和计算机连接。CPU把通过RAID组和起来的磁盘组当作一块硬盘来使用。

磁盘阵列中针对不同的应用设计了不同的技术标准,被称为RAID级别。常用的级别包括RAID 0到RAID 7等,如图3-55所示。需要注意的是,不同的RAID级别虽然对应于不同的性能、容量和可靠性,但级别的高低并不代表性能的好坏,只是反映了关键参数的不同组合。同时,RAID级别可以相互组合或通过扩展形成诸如RAID10、50和60等组合RAID级别。

1. RAID 0:无差错控制的带区组磁盘结构

RAID 0采用多磁盘体交叉存储,将磁盘的有效存储区域划分为带区(strip),带区可以是一个物理块或者是一个扇区等。RAID 0把要存储的文件划分成带区大小的数据块,然后将连续数据分散存储到不同磁盘上。例如,图3-55(a)中将奇数数据块存放在Disk 0上,而偶数数据块存放在Disk 1上。由于将数据分布在不同磁盘上可以交叉重叠访问,因此数据吞吐率大大提高,磁盘的负载也比较均衡。但RAID 0中没有对存储数据进行校验,实现容易但可靠性较差。

2. RAID 1:带镜像的磁盘结构

RAID 1是对RAID 0的扩展,也采用多磁盘体交叉存储,但RAID 1采用镜像结构,通过冗余来提高可靠性。如图3-55(b)所示,两个磁盘的内容完全一致,镜像硬盘相当于一个备份盘。RAID控制器能够同时对两个磁盘进行读操作和写操作。通过镜像的方式,RAID 1提高了系统的容错能力,并且比较容易设计和实现。但是RAID 1每次只能访问一个数据块,即数据块的传送速率与单个磁盘的读取速率相同,并且RAID 1中硬盘容量的利用率很低,只有50%,是所有RAID级别中最低的。

3. RAID 2:带海明码校验的磁盘结构

RAID 2采用磁盘体的位交叉存储技术,即将数据条块化地分布在不同硬盘上,条块的单位为位或字节,如图3-55(c)中使用的条块单位为4位,这样数据需要使用4块硬盘,每个硬盘存放条块中的一位。并且,RAID 2使用特定的编码技术(如海明码)来提供错误的检查及恢复,在图3-55(c)中使用了3个磁盘来存放校验位。这种编码技术需要多个磁盘存放检查及恢复信息,使得RAID 2技术实施更加复杂。因此,在商业环境中很少使用。

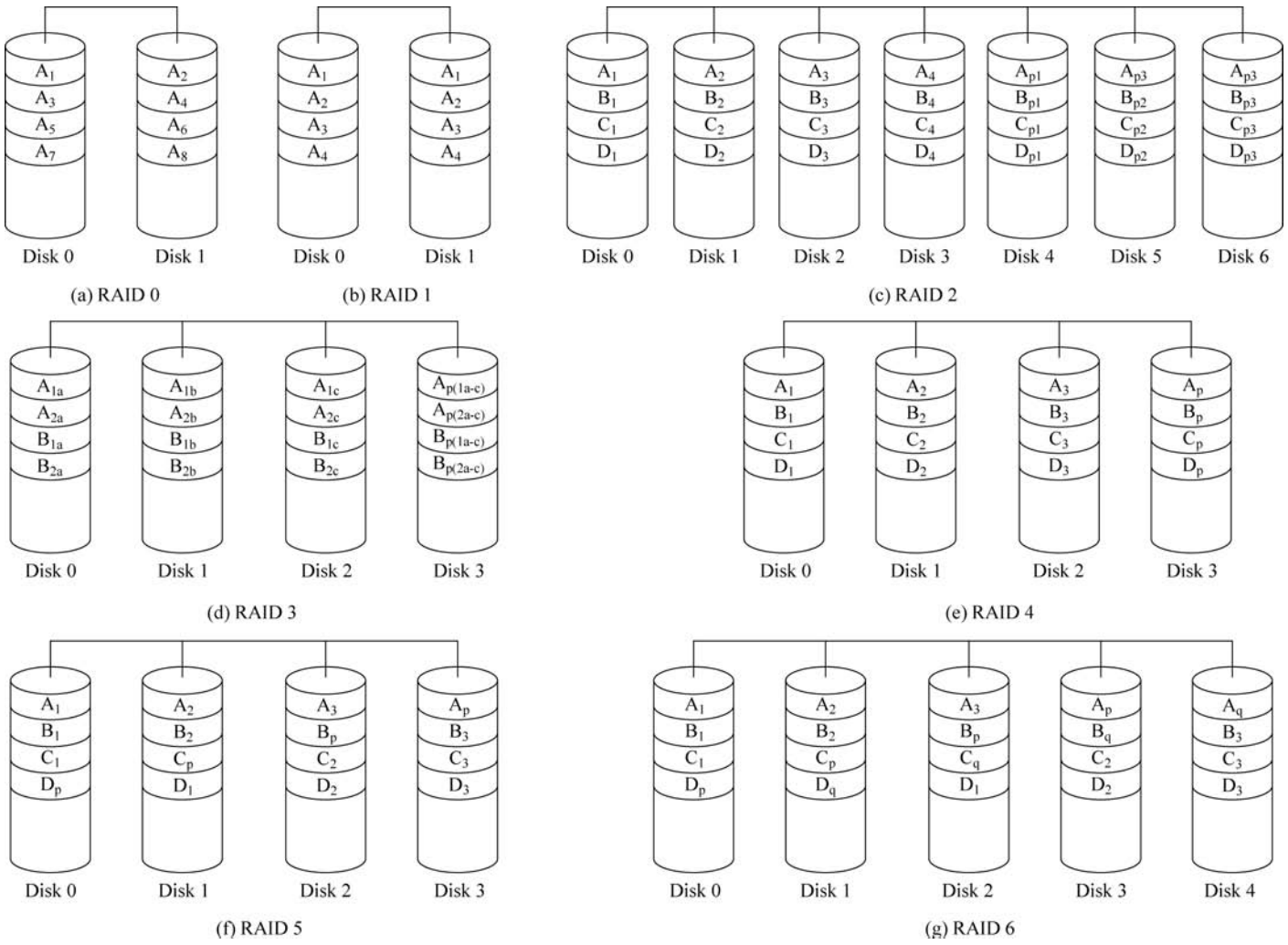


图 3-55　常用的 RAID 级别中磁盘的组织

4. RAID 3：带奇偶校验码的并行传送磁盘结构

如图 3-55(d)所示，RAID 3 与 RAID 2 类似也采用磁盘体的位交叉存储技术。不同的是 RAID 3 采用奇偶校验码，只需要增加一块磁盘存放奇偶校验信息。如果一块磁盘失效，奇偶校验盘及其他数据盘可以重新产生数据。RAID 3 对于大量的连续数据可提供很好的传输率，但对于随机数据，奇偶盘会成为写操作的瓶颈。

5. RAID 4：带奇偶校验码的独立磁盘结构

如图 3-55(e)所示，RAID 4 与 RAID 3 类似，不同的是，RAID 4 采用较大的条块(通常为一个扇区)，并且增加一块专门用来存放奇偶校验条块的冗余磁盘。RAID 4 对数据的访问是以数据块为单位进行的，即按磁盘进行的，每次访问仅涉及一个磁盘。

6. RAID 5：分布式奇偶校验的独立磁盘结构

为了克服 RAID 4 对校验盘访问的瓶颈问题，对 RAID 4 改进后形成了 RAID 5。从图 3-55(f)中可以看到，它的奇偶校验码，如图 3-55(f)中的 A_p 到 D_p，由 RAID 4 的集中存放改为分别存放在所有的数据磁盘上，图 3-55(f)中的 A_p 代表第 A 带区的奇偶校验值。由于 RAID 5 的奇偶校验码存放在不同的磁盘上，因此提高了可靠性，但是并行能力较差。

7. RAID 6：带有两种分布存储的奇偶校验码的独立磁盘结构

RAID 6 是对 RAID 5 的扩展，除了采用 RAID 5 中对带区的奇偶校验外，还在每块磁盘上增加了数据块的奇偶校验，如图 3-55(g)中的 A_q 到 D_q。但是，由于引入了第二种奇偶校验值，磁盘的数量增加了，同时对控制器的设计变得十分复杂，写入速度较慢，验证数据正确性所花费的代价也较大。

8. RAID 7：优化的高速数据传送磁盘结构

RAID 7 与前面的 RAID 级别具有明显的区别，RAID 7 可以理解为一个独立存储计算机，它带有操作系统和管理工具，可以独立运行，如图 3-56(a)所示。RAID 7 所有的 I/O 传送均是同步进行的，可以分别控制，这样通过并行性提高系统访问数据的速度。

9. RAID 10：高可靠性与高效磁盘的结构

这种结构是 RAID 0 和 RAID 1 的组合，是一种带区加镜像的结构，结合了 RAID 0 和 RAID 1 的优点，如图 3-56(b)所示。数据除分布在多个盘上外，每个盘都有其物理镜像盘，提供全冗余能力，允许一个磁盘故障，而不影响数据的可用性，并具有快速读/写能力。一般要求至少 4 个硬盘才能做成 RAID 10。

实际应用中最常见的是 RAID 0、RAID 1、RAID 5 和 RAID 10。由于在大多数场合，RAID 5 涵盖了 RAID 2 到 RAID 4 的优点，所以 RAID 2 到 RAID 4 目前已经很少使用。

从前面的介绍中可以看出，使用 RAID 的主要优点是通过在多个磁盘上同时存储和读取数据大幅提高存储系统的数据吞吐量，以及通过数据校验和镜像技术提供容错功能等。

3.5.4 其他辅助存储器

1. 磁带

磁带也是一种磁表面存储器，是一种以磁带为存储介质，由磁带机及其控制器组成的存储设备，是计算机中常用的一种备份级的离线存储器。磁带与磁盘最大的区别是磁盘属于直接存取存储器，而磁带属于顺序存取存储器。

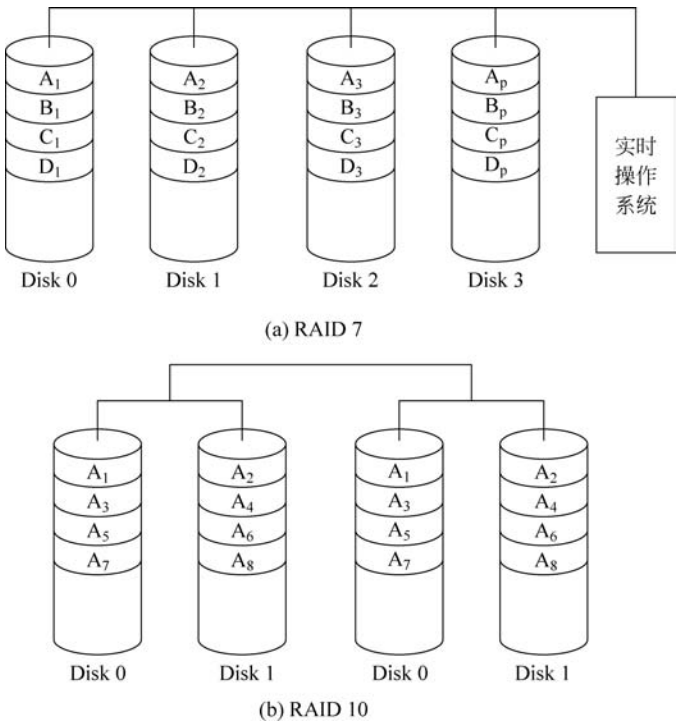


图 3-56　RAID 7 和 RAID 10 磁盘的组织

磁带存储器由磁带和磁带机两部分组成。磁带是通过在一条柔韧的聚酯薄膜带上涂上一层磁性材料来记录信息的，常见的盒式磁带通常被卷在两个可以来回转动的转轴上，并被封装在一个外壳中。不同类型磁带的宽度和长度并不相等。磁带上的磁道是沿磁带运动方向平行排列的，常见的磁带系统沿带宽方向有多个磁道，并且每个磁道上各有一个磁头，因此一次可以读取多个位的数据。

与磁盘不同，磁带机的磁头是固定不动的，通过磁带在磁头下的移动来完成定位操作。通常情况下，磁带被放在盘盒中，由放带盘送出、收带盘卷绕回收，放带盘和收带盘都有自己的驱动控制系统，如图 3-57 所示。磁带机是一种顺序存取的存储器，数据的位置直接影响到定位操作所需要的时间。在定位时，磁带可以正向或反向移动，并且读/写完毕后需要将磁头停在两个记录区之间。因此要求磁带驱动器不仅能保证磁带以一定的速度平稳地运动，并且能快速地启停。

按照磁带的记录格式不同可将磁带分为启停式和数据流式两种。

1）启停式磁带机

启停式磁带机在数据块与数据块之间需要启动和停止，因此被称为启停式磁带机。启停式磁带机主要由走带机构、磁带缓冲机构、带盘驱动机构、磁头等部分组成。走带机构的作用是带动磁带运动，以完成读/写操作；缓冲机构的作用是减小磁带运动中的惯性，以便使磁带机能够快速的启停；带盘驱动机构的作用是控制带盘电动机的方向和速度，以控制磁带盘的正转或反转以及旋转的速度。磁带机磁头的工作原理和硬盘磁头的工作原理完全一样，但为了能将多个磁道的数据同时读/写，将多个磁头组装在一起构成组合磁头。在进行读/写操作时，磁头不动，磁带随走带机构转动到磁头下。

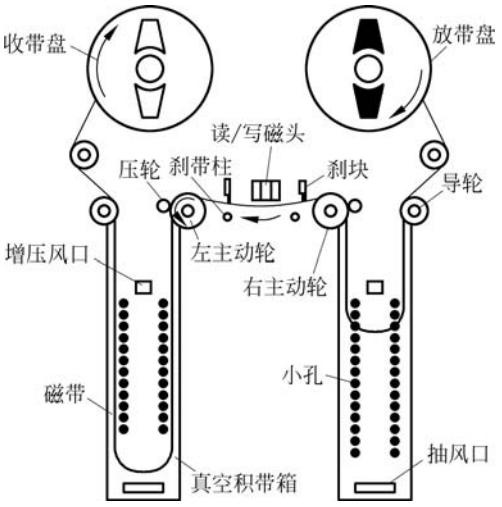


图 3-57　磁带机工作原理图

2) 数据流磁带机

数据流磁带机中数据被连续地写在磁带上,每个数据块间插入记录间隙,使磁带机在数据块间不需要启停。数据流磁带机简化了启停机构,用电子控制代替机械控制,不仅降低了成本,还提高了可靠性。

图 3-58 给出了一个 9 磁道的 1/4 英寸盒式数据流磁带的记录格式。

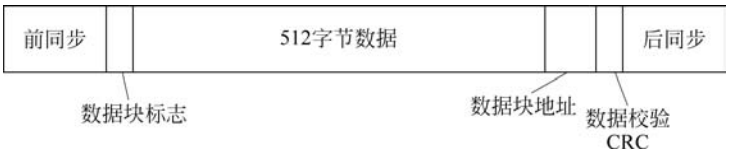


图 3-58　9 磁道 1/4 英寸磁带的记录格式

2. 光盘

光盘是利用光存储技术进行信息读/写的存储介质,如图 3-59 所示。光盘通常作为离线存储器来使用,具有存储容量大、存取方便、位价格低、易于保存等特点。

从图 3-59(a)可以看出光盘与磁盘一样,也采用圆形盘片作为记录面,并且盘片中心有一个圆孔,直径为 15mm,放入光盘驱动器(简称为光驱)后,主轴将带动盘片进行旋转从而进行光盘的定位。

光盘与硬盘有许多相同点,包括采用非接触式读写头、与主机采用 IDE、SCSI 等接口连接、以扇区作为最小读写单位等。但是,除了读写和记录的技术不同外,光盘和硬盘在信息的存放方式上也不相同。

与磁盘采用同心圆型磁道不同,光盘的记录区域是由内到外的螺旋形光道,如图 3-59(b)所示,信息以扇区为单位记录在光道上。由于光盘以恒定的线速度匀速转动,因此光盘的扇区是按照时间为单位来进行编址的,其地址格式为:分、秒、扇区号。例如,某扇区的地址为 10 分 20 秒 30 号扇区。

光盘上存储的各种类型的信息经过数字化处理变成“0”与“1”,其所对应的就是光盘上的

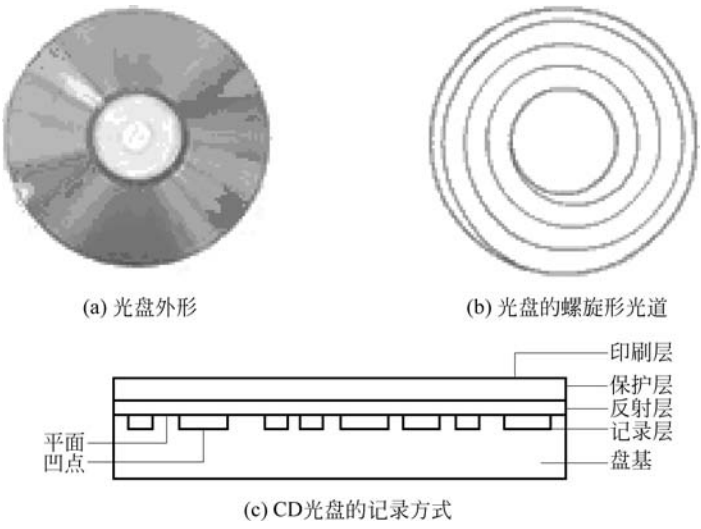


图 3-59　光盘

凹点和平面。当激光映射到盘片上时，如果是照在平面上，那么就会有 70%到 80%激光被反射回；如果照在凹点上，就无法反射回激光。根据反射和无反射的情况，光盘驱动器就可以解读“0”或“1”的数字编码了。

图 3-59(c)给出的一个 CD 光盘的横截面，从下到上分别是：

(1) 盘基(衬底)：采用透明的聚碳酸酯(PC)材料，是光盘的物理载体。具有冲击韧性好、使用温度范围大等优点。

(2) 记录层：通过在盘基上涂抹有机染料而构成，因此记录层也被称为有机染料层，不同类型的光盘最本质的区别就是染料层所涂覆的有机染料是不同的。

(3) 反射层：由喷镀的金属膜构成，在光驱进行数据读/写时用来反射激光光束。

(4) 保护层：用来保护光盘中的反射层和记录层，防止信号被破坏。

(5) 印刷层(标签)：一方面可以用于用户标识盘片内容等信息，另一方面也可以起到对光盘的保护作用。

光盘按照其所用的激光类型不同可以分为以下几类。

1) CD 光盘

CD 光盘(Compact Disk，高密度盘，)使用激光波长为 780nm，光斑直径为 1.74μm，光道间距为 1.6μm，凹坑宽度为 0.6μm，最小凹凸坑长度为 0.83μm。CD 光盘的容量一般为 650MB。

2) DVD 光盘

DVD 光盘(Digital Versatile Disc，数字多功能光盘)使用激光波长为 635～650nm，光斑直径为 1.08μm，光道间距为 0.74μm，凹坑宽度为 0.4μm，最小凹凸坑长度为 0.4μm。DVD 光盘有单层单面、单层双面、双层单面和双层双面。单层单面 DVD 的容量可达 4.7GB。

3) BD 光盘

BD 光盘(Blu-ray Disc，蓝光光盘)是最新的、容量较大的光盘。采用波长 405nm 的蓝

色激光光束进行读/写,一个单层的蓝光光盘的容量可以达到25GB或是27GB。主要是因为采用了以下写入模式:缩小激光光点以缩短轨距(0.32μm)增加容量,蓝光光盘构成"0"和"1"数字数据的凹坑(0.15μm)变得更小,因为读取凹坑用的蓝光激光波长比红光激光小;利用不同反射率达到多层写入效果;沟轨并写方式,增加了记录空间。

光盘按照其读写能力不同又可以分为以下几类。

1) 只读型光盘

只读型光盘内的数据或程序是出厂前写好的,并且出厂后用户只能读不能写。常见的只读型光盘有CD-ROM、DVD-ROM、BD-ROM等。

CD-ROM型光盘记录信息的原理是,大功率的激光束照射记录层时使记录层发生形变,在记录层表面形成小的凹坑(被称为凹区),没有经过照射的部分被称为凸区。凹区部分代表记录二进制"0",凸区部分代表记录二进制"1"。在读出时,光驱上的激光器发出的激光束经过透镜聚焦后照射在光盘上,从光盘的反射层反射回来的激光束沿原光路返回激光分离器后被分离出来,由于凹区反射光比凸区的弱,因此可以通过反射光的强弱来识别"0"和"1"。这样,反射光经反射到达光电检测器后被转变为电信号,在经过数字电路的处理,最终被还原为二进制数据。

2) 写一次型光盘

顾名思义,写一次型光盘在出厂后可以并且只可以进行一次写入操作,写入后可以多次读出。常见的写一次型光盘有CD-R、DVD-R、DVD+R、BD-R。

与只读型光盘被激光照射后产生物理凹坑不同,写一次型光盘的凹区和凸区是通过不同的反射光来模拟的。例如,CD-R型光盘出厂时,记录层是透明的。在进行写操作时,使用8～16mW的激光照射记录层的记录点,激光照射产生的能量使得该点发生化学反应,改变了燃料的分子结构,从而产生一个黑点。在读出时,光接收器可以分辨出黑点和未经照射的透明区域,从而识别"0"和"1"。

3) 可擦写型光盘

这种光盘与磁盘类似,可以进行多次的读/写操作。目前常用的记录方式有:光磁记录和相变记录。常见的可擦写型光盘有CD-RW、DVD-RW、DVD+RW、RD-RW等。

光磁记录型光盘(又被称为磁光盘)利用磁膜矫顽力随温度变化性质或者铁磁-顺磁转变的性质来进行写入。磁光盘的存储介质为稀土过渡金属GdGo(钴化钆),其物理特性是在常温下为铁磁体,磁畴取向稳定,当温度升高至居里温度时变为顺磁体,使得磁畴具有相同的磁化方向。在进行写入时,用激光向需要存储"1"的单元区域加热,使其温度超过居里点,失去磁性;同时,在盘的另一面的电磁线圈上施加一个外磁场,使被照单元反向磁化。这样该单元区域磁化方向与其他未照射单元方向相反,从而写入二进制"1",而其他未经照射单元相当于存储信息"0"。信息擦除过程与写过程刚好相反,即恢复原来的磁化方向。在读出时,采用偏振光来检测磁化方向,利用磁光效应将磁化方向的变化变换成光线强度的变化,从而识别出"0"和"1"。

相变记录型光盘(又被称为相变光盘)是利用记录介质的两个稳态之间的互逆相结构的变化来实现信息的记录和擦除。相变光盘的存储介质为Re-Tm晶态合金,这种材料有两种稳态,分别是反射率高的晶态和反射率低的非晶态(玻璃态)。在一定条件下,可完成晶态和非晶态的变换。这样可以用晶态和非晶态来分别表示二进制"1"和"0"。写入"0"时,利用高

功率窄脉冲的激光进行照射，是介质温度上升到熔点，再进行聚冷，从而使记录的单元区域变为非晶态，从而达到写入“0”的目的。擦除时，利用低功率宽脉冲激光照射，使得介质缓慢加热至低于熔点并且高于非晶态的变换温度，使记录的单元区域还原为晶态，这样也就相当于写入了“1”。在读出时，通过晶态和非晶态不同的对光线的反射率来识别“1”和“0”。

3. U盘

U盘也被称为闪存盘，采用3.2.6节介绍的闪速存储器作为存储介质。U盘是目前最常使用的移动存储器。U盘的组成比较简单，主要包括闪存芯片、设备控制器、石英振荡器和USB插头等几个部分，如图3-60所示。

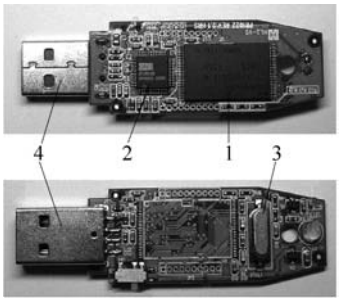


图3-60　U盘结构

(1) 闪存芯片：用以存储数据。

(2) USB存储设备控制器：提供USB设备控制器及与闪存的接口。

(3) 石英振荡器：用于提供U盘工作所需的时序信号。

(4) USB插头：提供连接到主机的接口。

4. 固态硬盘

固态硬盘(Solid State Disk，SSD)也被称为电子硬盘或者固态电子盘，是一种近年来出现的新型硬盘。这种硬盘并不属于磁表面存储器，而属于半导体存储器。它是由固态存储单元和控制单元组成的。目前绝大多数的固态硬盘采用DRAM和NAND型闪存作为存储介质。

固态硬盘中的控制单元负责对存储单元的读/写操作，由于没有常规硬盘中的机械部件，因此固态硬盘启动速度快，内部传输速率远高于常规硬盘。同时，固态硬盘属于随机访问的存储器，其寻址时间与数据的存储位置无关。

与常规硬盘相比，固态硬盘还有低功耗、无噪声、抗振动、低热量、体积小等优点。但是又存在着成本高、容量小、寿命短、可靠性低、写入速度慢等缺点。

思考题与习题

1. 存储系统的层次结构的意义何在？这种结构给计算机系统带来了哪些新的问题？

2. 在存储系统的层次结构中，设计高速缓冲器和虚拟存储器的目的各是什么？对这两个存储层次的管理有何异同点？

3. 解释存储元、存储单元、存储体的概念及它们之间的关系。

4. 说明主存器的容量、最大寻址空间、CPU交换数据的单位各由哪些因素决定？

5. 回答下列问题：

(1) 说明存取周期和存取时间的区别；

(2) 什么是存储器的带宽？若存储器的数据总线宽度为32位，存取周期为200ns，则存储器的带宽是多少？

6. 某机字长32位，其主存存储容量为64KB，问：

(1) 若按字编址它的寻址范围是多少？其存储容量应如何描述？

(2) 若按字节编址,试画图示意主存字地址和字节地址的分配情况;

(3) 试比较按字编址与按字节编址的优缺点。

7. 设有一个具有20位地址和32位字长的存储器,问:

(1) 该存储器能存储多少字节的信息?

(2) 如果此存储器由512K×8位SRAM芯片组成,需要多少芯片?

(3) 需要多少位地址作为芯片选择?

8. 对于SRAM芯片,如果片使能信号始终是有效的,问:

(1) 若读命令有效后,地址仍在变化,或数据线仍被其他信号占用,则对读出的正确性有什么影响?还有什么其他问题存在?

(2) 若写命令有效后,地址仍在变化,或写入数据仍不稳定,会对写入有什么影响?

9. 在DRAM存储器中为何将地址分为行地址和列地址?采用这种双向地址译码后,需要增加哪些器件?会给DRAM存储器的性能带来哪些方面的影响?

10. 某DRAM每1ms必须刷新64次,一个存储周期需要250ns,刷新周期与存储周期相同。则刷新时间占存储器总操作时间的百分比是多少?

11. 若用1M×1位的DRAM芯片构成1M×16位的主存储器,芯片内部存储元排列成正方形阵列,其刷新最大间隔时间为4ms。则采用异步刷新时,两次刷新操作应相隔多长时间?4ms时间内共需多少个刷新周期?

12. 某32位机主存地址码为32位,使用64M×4位的DRAM芯片组成,设芯片内部由4个8K×8K存储体结构组成,4个体可同时刷新,存储周期为0.1μs。若采用异步刷新方式,设存储元刷新最大时间间隔不超过8ms,则刷新定时信号的周期时间是多少?对整个存储器刷新一遍需要多少个刷新周期?

13. 用16K×8位的DRAM芯片构成64K×32位存储器,要求:

(1) 计算该存储器的芯片用量;

(2) 画出该存储器的原理性组成逻辑图;

(3) 采用异步刷新方式,设芯片内部矩阵为128×128×8结构,如存储元刷新最大间隔不超过8ms,则刷新定时信号周期是多少?对整个存储器刷新一遍需要多少个刷新周期?

(4) 若改用分散刷新方式,设存储周期为0.5μs,则在8ms时间内可对整个存储器刷新多少遍?有多少遍是多余的?

(5) 若改用集中刷新方式,CPU访存的死时间是多少?

14. 某机存储器的ROM区域所占的地址空间为0000H～3FFFH,由8K×8位EPROM芯片组成。RAM区域的大小为40K×16位,起始地址为6000H,采用的SRAM芯片容量仍为8K×8位。假设SRAM芯片有$\overline{CE}$和$\overline{WE}$信号控制端,CPU的地址总线为A_{15}～A_0,数据总线为D_{15}～D_0,控制总线给出的控制信号有R/$\overline{W}$(读/写)和$\overline{MREQ}$(访存)。要求:

(1) 画出地址空间分配图,并在图中标出译码方案;

(2) 画出该存储器的原理性组成逻辑图,并与CPU总线相连。

15. 设CPU有16根地址线,8根数据线,并用$\overline{MREQ}$作访存控制信号(低电平有效),用-WR作读/写控制信号(高电平为读,低电评为写)。现有下列存储芯片:1K×4位SRAM;4K×8位SRAM;8K×8位SRAM;2K×8位EPROM;4K×8位EPROM;

8K×8 位 EPROM 及 74LS138 移码器和各种门电路，请画出 CPU 与存储器的连接图。要求：

(1) 主存地址空间分配如下：6000H～67FFH 为系统程序区；
6800H～6BFFH 为用户程序区；
6C00H～6FFFH 为系统程序工作区。

请画出主存地址空间分配图，并标出译码分配方案。

(2) 合理选用上述存储芯片，请说明芯片的选择方案。

(3) 详细画出存储芯片的片使能逻辑图。

16. 某机主存地址码为 32 位，使用 64M×4 位的 SRAM 芯片组成，并采用存储条(模块板)结构，请问：

(1) 若该主存采用按字节编址方式，其寻址范围可达多少？

(2) 若每个存储条容量为 512MB，共需几块存储条才能构成支持上述寻址范围的主存？

(3) 每个存储条内需要多少 SRAM 芯片？该主存共需多少 SRAM 芯片？

(4) 画出该存储器的地址格式分配图。

(5) 若采用 74LS138 译码器芯片，画出存储条(模块板)的逻辑组成图。

(6) 设 CPU 采用 $\overline{\text{MERQ}}$(访存请求，低有效)信号、R/$\overline{\text{W}}$(读/写控制信号，高为读令，低为写令)信号与主存联络，画出该存储器的逻辑组成框图并与 CPU 连接。

17. 用 2 片 1M×4 位的 SRAM 芯片和若干片 256K×8 位的 SRAM 芯片构成 1M×16 位的主存储器，设 CPU 的地址总线为 A19～A0，数据总线为 D15～D0，控制信号为 R/$\overline{\text{W}}$(高电平表示读，低电平表示写)，$\overline{\text{MREQ}}$(低电平表示访存)，试问：

(1) 除 2 片 1M×4 位 SRAM 芯片外，还需多少片 256K×8 位 SRAM 芯片？

(2) 画出该存储器的组成逻辑图，并与 CPU 连接。

18. 某 64 位机主存地址码为 26 位，使用 256K×16 位的 DRAM 芯片组成，并采用模块板结构，问：

(1) 若每个模块板容量为 1M×64 位，共需几块模块板？

(2) 每个模块板内有多少 DRAM 芯片？

(3) 主存共需多少 DRAM 芯片？

(4) 试述该存储器的地址译码方案。

19. 现有两个 IA-32 汇编程序，其中分别定义了一个数据段，定义方式如下：

(1)
```
data segment
msg db "Hello"
align 4
dw 100,200,300
data ends
```

(2)
```
data segment
msg db "Hello"
dw 200,300,400
data ends
```

若这两个程序运行前，数据段加载到主存中的起始地址为 00B04010H，请分别画出两

个程序中数据段在主存中放置的示意图,并标出每个字节单元的地址及内容。

20. 相联存储器是如何按内容寻址的?这种寻址方式有何优势,适用于哪些场合?

21. 图3-3中,如果检索寄存器的值为"**** 1011 **** ****",屏蔽寄存器的值是什么?检索完成后,匹配寄存器中的值又是什么?

22. Cache与CPU和主存间进行数据交换的单位是什么?虚拟存储器中,辅助存储器与主存间进行数据交换的单位是什么?

23. 在计算机系统中设计多级Cache增加了系统的复杂性,能不能只设计一级Cache,并将其容量扩展到足够大?

24. 将数据Cache和指令Cache分开有什么好处?

25. 某计算机系统的内存由Cache和主存构成,Cache的存取周期为40ns,主存的存取周期为200ns。已知在一段给定的时间内,CPU共访问内存4500次,其中300次访问主存,求:

(1) Cache的命中率是多少?

(2) CPU访问内存的平均访问时间是多少?

26. 32位计算机的Cache容量为64KB,Cache块的大小为16B,若采用直接映射方式,则主存地址为123456F8(十六进制)的存储单元装入到Cache中的Cache地址是多少?

27. 给定以下程序段:

```
int array[1000][1000];
  for ( i = 0; i < 1000; i ++)
       for ( j = 0; j < 1000; j ++)
           sum += a[i][j];
return sum;
```

试比较按行优先和按列优先两种方式存储方式哪一种有更高的执行效率。

28. 某计算机的Cache采用组相联映射方式,每块大小为128B,Cache容量为64块,按4块分组,主存容量为4096块,问:主存地址共需多少位?主存地址字段中主存块标记,组地址和块地址各需多少位?

29. 假设主存与Cache间采用2路组相联映射,块大小为256字。Cache容量为8K字,主存地址空间为2M字。问:主存地址该如何划分?说明主存地址和Cache地址是如何映射的?

30. 假设某计算机的Cache的容量为8块,块大小为一个字,开始时Cache所有块为空。如果采用直接映射方式及LRU替换算法,按字编址。CPU按照如下地址顺序执行:1,2,4,8,16,21,56,45,2,4,6,7,5,4,7。问:

(1) 说明每次访问时命中还是缺失,并计算上述访问的命中率是多少?

(2) 假设数据区的容量为4块,重新计算(1)。

(3) 如果采用两路组相联映射方式,其他条件不变,重新计算(1)。

31. 考虑以下简单的程序:

```
for i = 1 to 100
    x = x + 1;
```

分别说明采用写回法和写直达法时,分别需要对主存执行多少次写操作?

32. 某磁盘组有 6 片盘片，每片有两个记录面，最上最下两个面不用。存储区域内直径 22cm，外直径 33cm，道密度为 40 道/cm，内层位密度 500 位/cm，转速 5400r/min。问：

(1) 共有多少柱面？

(2) 盘组总存储容量是多少？

(3) 数据传输率是多少？

(4) 采用定长数据块记录格式，直接寻址的最小单位是什么？若一个扇区的大小为 512B，系统中带有两台盘驱，则寻址命令中如何表示磁盘地址？

(5) 如果某文件长度超过一个磁道的容量，应将它记录在同一个存储面上，还是记录在同一个柱面上？为什么？

33. 假设磁盘存储器共有 5 个盘片，最外两侧盘面不能记录，每面有 200 个磁道，每条磁道有 12 个扇区，采用定长记录格式，每个扇区可保存 512 字节。磁盘机以 7200r/min 速度旋转，平均定位时间为 8ms。问：

(1) 这个磁盘存储器的存储容量是多少？

(2) 磁盘存储器的平均寻址时间是多少？

(3) 数据传输率是多少？

34. 某磁盘存储器的转速为 7200r/min，共有 6 个记录面，每毫米磁道数为 10 道。采用定长记录格式，每个磁道可以记录信息 12 288 字节，最小磁道直径为 22cm，共有 200 道，问：

(1) 这个磁盘存储器的存储容量是多少？

(2) 最外层磁道和最内层磁道的位密度分别是多少？

(3) 数据传输率分别是多少？

35. 对于一个有多个盘面构成的磁盘存储器，当需要存储的文件长度超过一个磁道的容量时，应该将超出部分记录在同一个盘面的不同磁道，还是不同盘面的同一个磁道？

36. 已知某磁盘存储器转速为 2400r/min，每个记录面道数为 200 道，平均查找时间为 60ms，每道存储容量为 96Kb，求磁盘的存取时间与数据传输率。

37. 分别画出 FM 和 MFM 记录 01100010 的写入电流波形。