CHAPTER 3

第3章

存储系统

本章要点

存储系统部分不但是计算机组成原理课程学习的重点，也是实际运行的计算机系统中最重要的组成部分，任何数据在运算、输入和输出的过程中都必须经过存储系统的中转。因此数据的访问方式是本章的核心部分。围绕访问方式展开的要点主要包括：便于访问的存储结构设计、访问的物理连接方式、访问方式的优化。而评价访问最重要的指标是访问速度和准确率。本章最后介绍的Cache部分则是存储系统在计算机体系结构中应用的重要实例。

3.1 存储系统概述

存储系统是一系列具有记忆功能单元的集合。计算机需要处理和已经处理过的数据都必须放置在存储系统中，因此存储系统是计算机组成中不可或缺的部分。同时，存储系统的功能定位也决定了对其进行衡量的重要指标是容量与速度。考虑到实际生产应用的需要，存储系统还应当具有另一个重要的评价因素——价格。存储系统的发展和设计应用均是围绕着上述三个因素的考虑来展开的。

3.1.1 基本概念

存储系统的作用是存储数据(既包括各种待处理的数据，也包括各种程序和指令；根据冯·诺依曼特征，程序可以被视为一种特殊的数据)，即以某种状态的变化来存储和表示二进制化的数据。一般而言，存储系统的最小组成单位是存储元件或者记忆元件。存储元件可以呈现出两种稳定状态，并能在这两种稳定状态间进行切换和转换，每一个存储元件可以表示二进制数的一位，因此为了强调存储元件所代表位的概念时，也将其称为存储位元。

在存取的过程中，数据写入或者读出的数量都是较大的，如果每次都是按位进行操作，显然对存取速度会有较大的影响。因此在实际工作环境中，数据都是批量进行存取的。计算机每次能够进行存取的最小单元就是一个存储单元，存储单元所包含的位数即包含的存储位元的个数被称为存储字长。

大量的存储单元在一起组成的集合称为存储体，计算机中的数据主要存储在存储体中。

既然存储体中包含大量的存储单元，如果想对其进行存取就势必要涉及寻址问题，即对其中某些特定的存储单元进行存取。一般为了区分存储体中的各个存储单元，会给每一个存储单元分配一个编号，这个编号被称为该存储单元的地址。地址是存储单元位置的标记，并不是存储单元内容的表示。图 3.1 中描述了一个 64KB 大小的存储体，并在其中表示出了存储位元、存储单元和存储体间的关系。

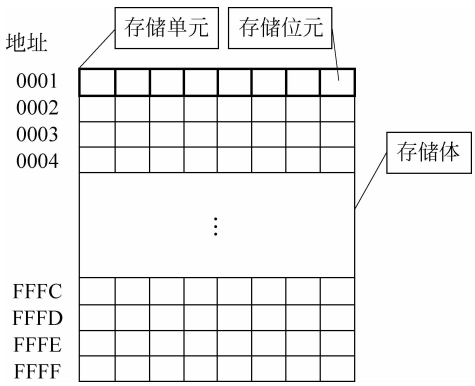

图 3.1 存储位元、存储单元、存储体关系图

通常地址的编制是以字节为单位进行的，即每一个地址会对应到每一个字节，这种安排方式称为字节编址。有时也会将地址与字（双字节）相对应编址，这种方式被称为字编址。现代计算机主要使用字节编址的方式，这种方式也可以支持字寻址。

在字节编址方式中，将一个双字节字放入到存储体时会出现一个问题，即是将高字节放入地址较低的单元还是将低字节放入（按照 Intel 模式，权重较高的字节称为高字节，权重较低的字节称为低字节。例如 0x1234 中 0x12 为高字节，0x34 为低字节。在 Motorola 模式中称呼方式相反，本书中的描述采用 Intel 方式进行）。

进一步讲，程序设计语言中的字符串是指连续的一串字符，通常占用主存中连续的多个字节，每个字节存一个字符。同样，多字节数据通常也连续存放在主存中，占用多个连续的字节存储单元。不管是字符串还是数字串，它们既可以从高位字节向低位字节顺序存放，也可以从低位字节到高位字节的方式存放。

（1）小端方式（Little Endian）：指低字节数据存放在低地址存储单元，高字节数据存放在高地址存储单元。例如，在早期苹果计算机中应用的 Motorola 680x0 系统。

（2）大端方式（Big Endian）：指低字节数据存放在高地址存储单元，高字节数据存放在低地址存储单元。例如，Intel 的 x86 系统。

如图 3.2 所示为数据 0x12345678 在内存中的存放情况，图 3.2(a)所示为小端方式，图 3.2(b)所示为大端方式。

存储体只是单纯的数据存储器件，必须配合相应的控制器、寄存器、寻址器件等诸多部件一起工作才能完成支持数据存取的工作。以存储体为中心，包含控制器、寄存器、寻址器件等诸多部件，可以完成数据存取功能的系统被称为存储系统。存储器则是存储系统概念实例化的表现形式。

通常而言，每个存储器都具有一个用来进行寻址的寄存器（Memory Address Register，MAR）和一个用来存储临时数据的寄存器（Memory Data Register，MDR）。在进行数据存取的过程中，需要访问的地址被放置在 MAR 中，而需要处理的数据

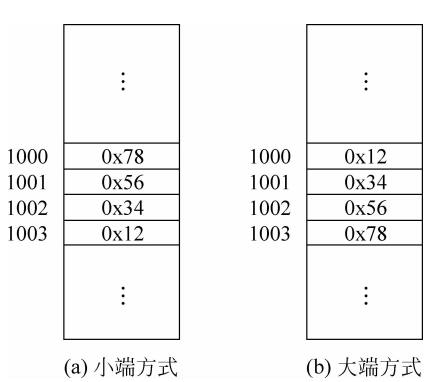

图 3.2 数据存放方式

会被放置在 MDR 中。即读取过程中，激活读取信号并将欲访问单元的地址从地址总线上读入到 MAR 中后，欲访问单元的内容就被取出放置在 MDR 中并输出到数据总线上；在存储过程中，激活写入信号，欲访问单元的地址从地址总线上读入到 MAR 中，欲写入的内容被 MDR 从数据总线上读入后，数据就被存储到存储器中了。换言之，外部对存储器中存储体的访问操作都是通过对 MAR、MDR 和功能控制信号的交互完成的。

3.1.2 评价指标

存储器的技术衡量指标大都是围绕着速度、容量和价格三个因素展开的，总体而言有以下几项。

1. 存取速度

存储器主要支持读取和存储两种操作，但是一般在存储器速度指标的衡量上，并不将其分割来进行评判，而是将之视为统一的操作进行考量。存取速度指标主要通过存取时间和存取周期来进行表征。

存储器存取时间(Memory Access Time，MAT)指的是从启动存储器操作到完成该次操作所经历的时间。例如：从存储器接收到读命令直至 MDR 中的数据被稳定地输送到数据总线的时间间隔。MCT 也被直译为存储器访问时间。

存储器周期时间(Memory Cycle Time，MCT)指的是连续启动两次独立不相关的存储器操作之间所需的时间间隔。例如：连续进行两次读取操作时，第二次操作启动时间和第一次操作启动时间之间的时间差值。

一般而言，MCT 略大于 MAT。使用 MAT 往往用以衡量存储器本身的性能，而用 MCT 衡量计算机系统对于存储器的控制和管理性能。

2. 存储容量

存储容量通常使用存储器内可以容纳的二进制数据的数据量来加以表征。一般使用存储地址寄存器 MAR 的编址数量与存储单元位数的乘积表示。例如：某存储芯片的 MAR 为 8 位，即可寻址 2^8 个单元，每个单元大小为 1 字节，则该存储芯片容量为 $2^8 \times 1 = 256$ 字节。

当前的 PC 系统往往具有较大的内存寻址能力，大都能支持 16GB 以上的内存容量，服务器级别的系统甚至可以达到 64GB 以上。然而实际部署时还要考虑具体的应用需求，所以家用 PC 往往只配置 2～4GB 的内存。由此可见，实际装机内存容量往往小于系统的最大支持能力。

3. 可靠性

计算机系统是对精度要求较高的系统，无论作为数据存储使用，还是作为程序存储使用的存储器都要具有较高的可靠性。存储器的可靠性往往使用平均无故障时间(Mean Time Between Failures，MTBF)来进行表征，即两次故障之间的时间间隔。显然 MTBF 值越大系统越可靠，主存储器往往使用校验码技术来增加 MTBF 值，其他存储器也有相应的方法来增强其可靠性。

4. 性价比

性价比是一个较为抽象的指标，没有具体的参数和表征量来进行描述。它的出现是为了表示对于存储的应用购置应当针对具体的需求来考虑，而不是一味地追求低价或者高性能。例如：现在的内存价格已跌破 0.1 元/MB 的瓶颈，而早期的 EDO 内存却是千金难求。如果单纯从价格/容量的角度衡量，16MB 左右的 EDO 内存完全没有购置的必要。但在一些工业控制系统中，由于硬件实施的时代特征必须使用 EDO 类的内存，如果将系统改制成向 DDR2、DDR3 兼容的模式，资金投入可能需要以数十万计。这种情况下，即使以 10 元/MB 甚至 100 元/MB 的价格去购置 EDO 内存都会有更好的性价比。

3.1.3 存储器分类

既然存储器有着不同的评价指标和特点，在选用存储器和设计存储器时就需要综合考虑对于存储器的不同需求，在不同环境、不同系统、不同架构中选择功能恰当、性价比合理的存储器。因此对于存储器的分类研究就成为一个十分重要的问题。在计算机组成中常用的分类方法如图 3.3 所示。

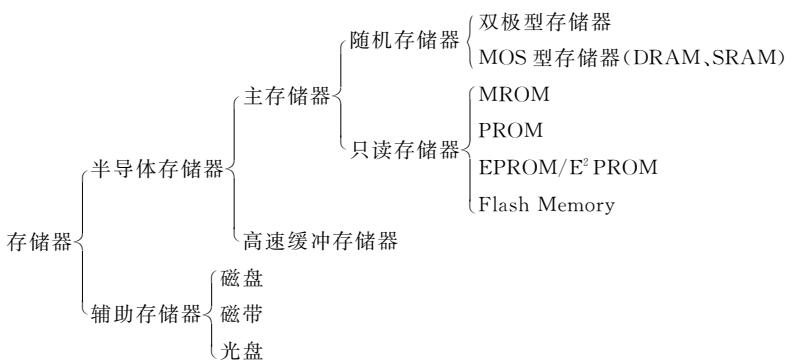

图 3.3 存储器分类示意图

1. 按存储介质分类

存储器中的存储体负责以存储元件的状态变化来存储二进制数据，存储元件、存储体采用的物理材料一般被称为存储介质。当前根据存储介质主要将存储器分为半导体存储器、磁存储器和光存储器三种。

半导体存储器主要分为双极型半导体存储器(TTL 型)和 MOS 型半导体存储器。TTL 型存储器虽然存取速度较快，但是由于功耗较大且集成度低导致成本较高，在现代的计算机组成中往往只用其来制作高速缓存使用；而 MOS 型存储器功耗小、集成度高，虽然存取略低但成本也相应较低，现在的主存储器(内存)大多使用 MOS 型存储器。半导体存储器的具体存储原理在脉冲与数字逻辑的相关课程中有详尽的描述，本书中不再赘述。

磁存储器主要分为磁芯存储器和磁表面存储器。前者由于制作工艺复杂、成本较高已经基本被半导体存储器所取代。磁表面存储器是将磁性材料涂于载体介质表面作为存储介质的存储器，通过电磁原理进行二进制数据的记录。常见的磁盘、磁带都属于磁表面存

储器。

光存储器是指利用光学原理进行二进制数据存取的存储器，CD、DVD 等光盘形式的存储器都属于光存储器。总体而言现代计算机还是电子计算机，由于光也可利用“通”、“断”两种状态来表示二进制数据，很多研究机构和科研人员正在着力进行光计算机的研究。

2. 按功能分类

在上一章介绍运算器的部分中，运算器运算的结果会立即输出到外部，以便下一次运算的进行；而运算器接收的输入大都是从外部接收而来的。存储器就是用以存储运算结果和运算参数等数据的。用以和 CPU 进行交互的存储器被称为主存储器(即 PC 系统中的内存)。主存是计算机组成中的重要部件，主存的容量和存储速度都是需要关注的技术指标。CPU 可以直接调用指令来访问主存，读取主存中的数据进行运算或者读取主存中的指令加以执行。

由于主存往往是易失性存储器，即无源情况无法保存信息，加之主存成本略高。在计算机组成中便出现了辅助存储器。辅助存储器往往使用成本较低的磁存储器或者光存储器来担当。辅助存储器的容量往往远大于主存的容量，而速度也远远低于主存储器。因此辅助存储器主要用于存储暂时不需要使用的数据或程序，当这部分数据和程序需要使用时，再将之从辅存加载到主存中使用。一般情况下，CPU 不能直接与辅存通信。

还有一类高速缓冲处理器将在 3.3 节中详细介绍。在有的分类中，CPU 内的寄存器也被视为一种特殊的存储器。

3. 按访问方式分类

对存储器的访问实质上就是对存储单元的访问，根据寻找访问单元的方式可以将存储器分为两类——随机访问存储器和顺序访问存储器。

随机访问存储器(Random Access Memory，RAM)是应用较多的存储器，大多半导体存储器都属于随机访问存储器。这里所指的随机指的是被访问单元的位置是随机的、独立的。换言之，RAM 中的每一个单元都可以独立地直接访问。每两次连续访问之间的关系是独立的，两次访问单元的位置与访问时间无关(这里不考虑访问的局部性原理等统计策略级的问题)。在计算机系统中，RAM 主要用作主存和高速缓冲存储器。

与 RAM 相对应的是顺序访问存储器(Sequential Access Memory，SAM)，SAM 中的单元一般不能被独立访问。当有存取操作时，SAM 先根据欲访问单元地址定位至一个储存块(存储单元集合)，再按顺序寻找到欲访问的单元进行存取操作。最为极限的情况下，SAM 会从存储体的第一个单元开始按顺序逐个查找，直至找到欲访问单元。因此相比 RAM 而言，SAM 往往具有较低的访问速度，但在价格/容量的指标上优势比较明显。常见的磁带机、磁盘等辅助存储器都属于 SAM。

只读存储器(Read Only Memory，ROM)是一种特殊类型的存储器。当 ROM 内的数据被写入后，对其的访问只能进行读取操作。ROM 主要用于存储一些不需要改动的重要参数或者程序，这些数据存储在 ROM 中可以有效地防止用户的误操作，或者其他原因导致的破坏。例如，许多嵌入式设备的系统程序就是存储在 ROM 中的。早期的 ROM 只能进行一次写入，编程后的 ROM 是无法进行修改的。现代的 ROM 在特殊情况下(例如特殊电

压、特殊管脚信号等),可以被多次修改。

4. 按信息保存情况分类

根据信息放入存储器后是否需要电源来维持数据存储的情况,存储器可以被分为易失性存储器和非易失性存储器两种。

一般作为主存的存储器都是易失性的,即在无电源供电的情况下无法保存任何数据。而 DRAM 类存储器则要求不断地进行刷新来维持数据的保存状态,否则即使有源状态也无法保存数据。而无源状态下仍可将数据维持保存状态的存储器被称为非易失性存储器,大多数辅存都属于非易失性存储器。例如,磁带机就属于非易失性的顺序存储器。

图 3.4 磁带设备

磁带机(Tape Drive)通常由磁带驱动器和磁带构成,是一种经济、可靠、大容量的储存设备,如图 3.4 所示。通常采用高纠错能力编码技术和写后即读通道技术,以提高数据备份的可靠性。根据装载磁带方式的不同,一般分为手动装带磁带机和自动装带磁带机即自动加载磁带机。自动加载磁带机实际上是由磁带和磁带机组合而成的。它可以从装有多盘磁带的磁带匣中拾取磁带并放入驱动器中,或执行相反的过程。自动加载磁带机能够支持例行备份过程,自动为每日的备份工作装载新的磁带,并完成相应的数据备份工作。磁带机的最大劣势在于其属于顺序存储设备而且运转速度较慢,所以无法作为在线设备使用。但其最大的优势在于价格十分低廉,因此作为不需要频繁存取的备份系统而言,磁带机是一个非常理想的选择。

3.1.4 存储系统的层次结构

根据上一小节中对于存储器的分类研究,结合容量、速度和价格指标,可以看出现代计算机存储器具有如图 3.5 所示的指标层次图。其中的容量指的是同样成本条件下,可支持的最大容量大小。

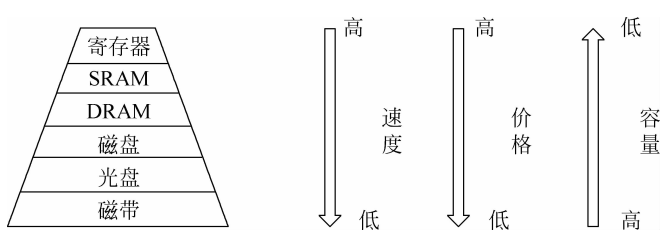

图 3.5 存储器指标层次图

既然存储器是计算机系统中不可或缺的重要部分,那么显然存储器应当是访问速度越快、容量越大越好。理想情况下计算机系统应当采用图 3.6 中的结构,即 CPU 与最高速度的超大容量存储器直接连接,所有的程序以及数据都放置在该存储器中。根据目前的存储

器发展状况来看，该存储器还应当具有不间断电源加以供电。

然而实际上这是难以实现的，尤其是在计算机进行普及化的过程中成本和能耗是首要考虑的问题。图 3.6 中的理想模型无论从成本角度还是能耗角度都是不能接受的。但是如果直接把主存完全用辅存代替的话，存取速度又成了一个无法回避的问题。

经过对大量程序运行情况分析后发现：在较短的时间内程序对于存储空间的访问中90％的访问会集中在存储空间 10％的区域内。冯·诺依曼体系中的程序和数据都是连续存储的，即连续的指令或者数据会被放置在连续的存储单元内。在程序运行过程中，充斥着大量的循环结构；结构化组织的数据也会被放置在连续的存储单元内。因此在较短时间内对于存储空间的访问确实会集中在比较少的区域内。这个规律被称为访问局部性规律，也称为访问局部性原理。访问局部性原理是计算机组成乃至计算机体系结构中一项非常重要的原理，许多结构性的设计都是基于这项原理衍生出来的。

根据访问局部性原理，可以对于图 3.6 所示的理想计算机存储体系进行改造，即将计算机的存储体系进行分层次的分级管理，如图 3.7 所示为二级存储体系示意图。

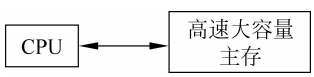

图 3.6　理想计算机存储体系

图 3.7　二级计算机存储体系

价格低廉、容量较大、访问速度较低的辅存中存放不常使用的程序和数据，而将访问频繁的程序和数据加载到价格较贵、容量较小、访问速度较快的主存中。例如，计算机上有“扫雷”、“计算器”、“浏览器”三个程序，这三个程序平时都被存储在辅存中，当需要运行某个程序如“扫雷”时，“扫雷”程序被装载到主存中，与 CPU 直接交互运行，而不需要运行的其他程序仍然被放置在辅存中。这样便构成了具有主存、辅存两级存储系统的计算机存储体系。

二级计算机存储体系出现的一个重要原因在于辅存的速度远远低于主存和 CPU 的访问速度。随着计算机系统的发展，当 CPU 的访问速度又远远超过主存的速度的时候，如图 3.8 所示的三级计算机存储体系便出现了。

基于与二级计算机存储体系同样的解决思路，即在速度不匹配的两端之间，增设一个高速存储模块。这个高速存储模块被称为高速缓冲存储器(Cache)，有时也被称为缓存。通常情况下，Cache 被放置在 CPU 内部直接连入 CPU 内总线以保证高速访问。在二级计算机存储体系中 CPU 不能直接访问主存内容，所有的数据必须经由主存传递给 CPU。而在三级计算机存储体系中 CPU 不但可以直接访问 Cache，也可直接访问主存。

通过理想存储体系、二级计算机存储体系到三级计算机存储体系的发展不难看出，计算机存储体系发展的推动力是为了解决由于功耗和价格原因导致的速度不匹配问题。解决方法都是利用访问局部性原理配置缓冲存储器以实现性能和成本间的折中。如图 3.9 所示，当今的多级存储体系正是继续沿用这个思路的产物。

一方面，由于 CPU 内运算器的速度不断提升导致运算器与 Cache 间的速度差日益增加，为了解决 CPU 内部的速度不匹配问题，多级 Cache 方案被设计出来，即在 Cache 之间再增设作为缓冲的存储部件。当前的主流 CPU 大都已经整合了三级 Cache。另一方面，辅存与主存间的速度差也在不断地扩大，为此在辅存与主存间也出现了相应的 Cache 来解决速

度不匹配等问题。不同于 CPU 与内存间的 Cache，这种辅存 Cache 一般被放置在低速的辅存设备中。值得一提的是，辅存中的缓存 Cache 与缓冲 Buffer 不是一个概念，具体的细节将在本书的后续章节中说明。

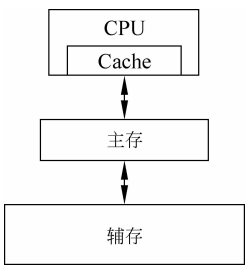

图 3.8 三级计算机存储体系

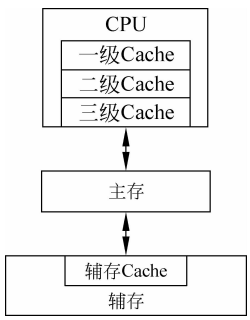

图 3.9 多级计算机存储体系

3.2 主存储器

主存储器简称为主存，在讨论微型计算机等具体的结构应用环境时又往往称其为内存。在以存储器为中心结构大行其道的今天，主存储器更是成为计算机组成中的核心部件。如上节所述，主存储器不是单纯的存储体，而是要包含一系列的周边组件才能成为一个完整的部件。这些部件的组成和连接方式构成了主存储器的基本结构。而不同的存储体选择会使主存储器有不同的特征和性能，本节中将其分为静态随机存储器、动态随机存储器和只读存储器来介绍。

存储器的访问是通过电子线路访问存储体完成的，由于外部的环境干扰和自身的电气特性，在访问过程中很容易出现写入或者读出的数据发生错误，这时就需要使用编码的方式来对数据进行校验。具体的校验方式和编码方式将在存储器校验部分中进行介绍。每一款存储器芯片都有固定的参数指标，例如寻址空间大小、存储单元容量等。但是计算机对于主存储器的需求往往是多种多样的，寻找到一款恰好满足计算机需求的存储芯片是比较困难的。在这种情况下，往往需要将几款存储器芯片进行组合连接以满足计算机对于主存储器的需求，这就是存储器与 CPU 的连接问题。为了进一步提升主存储器的性能，除单纯依靠并联提高总线位宽的方式外，在多体交叉存储器部分中将介绍高位交叉、低位交叉等其他存储器的组合方式。

3.2.1 主存储器的基本结构

计算机的主存储器主要由随机存储器构成。如在 3.1 节中所述，主存储器最主要的组成部分是用状态翻转来记录数据变化情况的存储体。但存储体只是起到数据保存的作用，要想建立存储器还需要相应的其他部件。主存储器的基本结构如图 3.10 所示。

存储体中包含着大量的存储单元，通常情况下这些存储单元是按照矩阵方式排列的，以方便集成化和扩展。要想对其中某个单元进行访问，首先需要获得该单元的位置信息，即该单元的地址。计算机的地址一般是通过地址总线进行传递的，显然存储器需要一个与地址总线

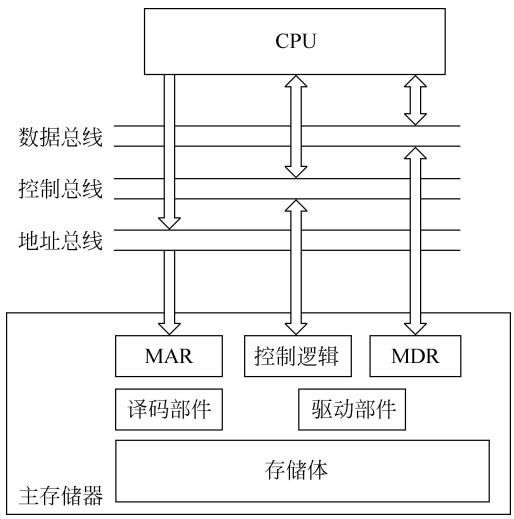

图 3.10　主存储器基本结构图

兼容的接口部件以便从地址总线上获取地址。同时，也需要 MAR 来将当前获取的地址进行存储以便进行访问。当获得欲访问单元地址后，需要相应的地址译码部件来选择该单元。

无论是对存储器进行写入操作还是读取操作，存储器都需要和数据总线交换数据；和控制总线交换“读命令”、“写命令”等控制信号。这就要求存储器应当具备相应的总线驱动部件。如同获取地址一样，与存储器交换的数据也需要放置在相应的寄存器 MDR 中。同时还需要一系列的逻辑控制部件和驱动部件来控制相应的时序逻辑与操作流程。

主存储器的结构并不复杂，尤其在本课程的学习范围内只需要重点关注 MAR、MDR、总线连接方式，译码部件中的扩展方式与逻辑控制中的读命令、写命令、使能信号即可。

3.2.2　静态随机存取存储器

1. 静态随机存储器原理

静态随机存储器(Static Random Access Memory，SRAM)是一种重要的随机存储器，也是当前应用最广的一种随机存储器。其名称的“静态”主要是相对于动态随机存储器的“动态”而言的，即该种存储器只要保持有源状态，其中所储存的数据就可以一直保持而不丢失。而动态随机存取内存(DRAM)里面所储存的数据即使在有源状态下也只能保持较短的时间，需要不断地进行刷新来维持数据的存储状态。

当前的 SRAM 主要使用价格低廉、功耗较小的 MOS 管制作。如图 3.11 所示为 SRAM 存储位元的一种六管存储位元电路基本结构，其中的 T_1 和 T_2 管构成双稳态正反馈电路，T_3 和 T_4 管作为负载管，T_5 和 T_6 管作

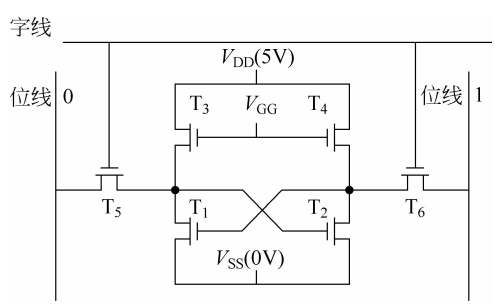

图 3.11　SRAM 存储位元结构

为控制字线和位线的门控管。在没有访问操作时，字线处于低电平状态，两根位线上呈高电平状态。由于门控管 T_5 和 T_6 截止，T_1 和 T_2 将保持原有的稳定状态。

进行写入访问时，字线上需加载高电平。欲写 0 时，在位线 0 上加载低电平；欲写 1 时，在位线 1 上加载低电平。当位线 0 上加载低电平时，T_2 管被截止 T_1 管导通，这种状态被记为 0。反之，T_1 管被截止 T_2 管导通，这种状态被记为 1。

进行读取访问时，字线上也需加载高电平，T_5 和 T_6 管被打开。当存储状态为 0 的时候，电流将通过 T_5 和 T_1 管流入地线。当存储状态为 1 的时候，电流将流到位线 1 上。这样的电流流动不会改变 T_1 和 T_2 管的状态，因此被称为非破坏性读出，即数据读出后对存储单元的数据存储状态没有影响。

2. 静态随机存储器芯片结构

图 3.12 所示为 SRAM 的芯片结构，比之图 3.11 中的一般结构显得更为具体。大量的存储单元按照类似于图 3.1 的方式组成存储单元阵列，即存储体，以便存储信息。当存储单元过多的时候往往将存储单元按照矩阵的方进行排列，要选择某一个存储单元时就需要使用行地址和列地址来进行标记。因此，SRAM 中也具备对于支持使用行地址和列地址进行访问的译码器和驱动器，以及相应的数据缓冲器和各种电路。

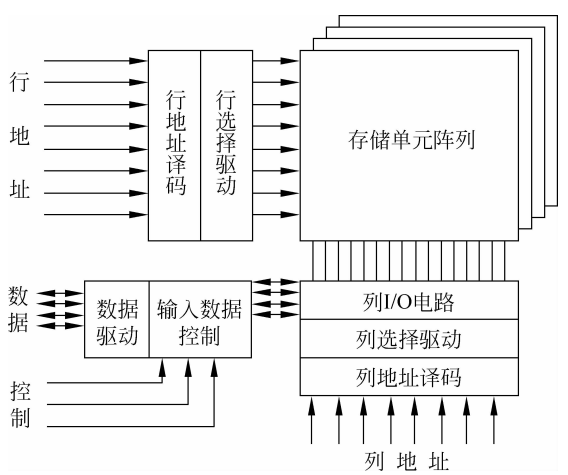

图 3.12 SRAM 芯片结构

结合上文中的存储位元与芯片结构，可以得知使用 SRAM 芯片进行访问操作的一般控制流程。首先，SRAM 芯片往往提供片选信号 CS，访问功能控制信号 RD/$\overline{WR}$等基本控制信号线与数据总线、地址总线接口。当欲访问 SRAM 芯片的时候，首先将地址信号送上地址总线，然后送出片选信号和访问功能控制信号，最后再与数据总线进行数据的交换。数据总线上的数据往往需要保持较长的时间，以增加数据访问的可靠性，即防止数据总线上的数据没有被及时接收。数据总线上出现有效数据后功能控制信号才能恢复成非有效状态，地址总线上的地址信号也才允许变更。地址总线上的有效信号持续时间被称为访问周期，又称为存取周期。读取操作中地址总线上的有效信号持续时间被称为读周期时间；存储操作中地址总线上的有效信号持续时间被称为写周期时间。通常情况下，令读周期时间等于写

周期时间，即两者都等于存取周期。

SRAM 的特点就在于上文所说的非破坏性读取，在有源的条件下所存储的数据不会丢失，结构也比较简单，具有较高的可靠性，访问速度也比较快，功耗较小。但是构成一个存储单元往往需要多个 MOS 管才能完成，这就使得 SRAM 的成本相对较高，集成度也较低。

3.2.3 动态随机存取存储器

1. 动态随机存储器原理

SRAM 成本高的主要原因在于使用了较多的 MOS 管来设计存储位元，那么如果降低组成存储位元的 MOS 管数量，显而易见可以降低存储器的成本。SRAM 中利用 MOS 管的翻转状态来表示二进制的状态，即数据线上是否有电流流动来表示被存储的数据是“0”还是“1”。电容作为储能器件也可以实现类似的功能，即利用内部是否存储有电荷来表示二进制的状态。使用电容代替 MOS 管来做状态存储器件显然可以极大地降低成本，这就是动态随机存储器的设计初衷。

动态随机存取存储器（Dynamic Random Access Memory，DRAM）也是一种半导体存储器，主要的作用原理是利用电容内是否储存有电荷来代表一个二进制位（bit）是 1 还是 0。常见的 DRAM 存储位元结构有四管 MOS 型、三管 MOS 型和单管 MOS 型。如图 3.13 所示，本书主要对单管 MOS 型进行介绍。

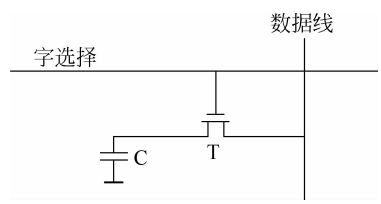

图 3.13 动态存储单元结构

当存储单元被选中后，字选择线加载高电平，使得控制管 T 被打开，电流在数据线和存储电容 C 之间流动。写入 1 时数据线呈高电平状态，电流通过 T 流入 C 中；写入 0 时，数据线呈低电平状态，将 C 中的电流导出，使其内部不存有正电荷。读出时，如果 C 中有正电荷，将有电流流过 T 管，拉升数据线上的电平状态；否则数据线仍保持低电平状态。DRAM 的读取是破坏性的读取，一旦进行读取操作将可能导致电容中失去正电荷呈现无电荷状态，因此必须在读出后进行重写工作，即还原读取前电容的存储状态，即“刷新”。

电容在实际工作中会有漏电的现象，从而导致内部存储正电荷不足而丢失存储的状态。虽然读出后的重写工作可以为电容补充电荷，但是并不是每一个电容在电荷泄漏前都可以被访问到。因此为了使其能正常工作，即使没有读取操作也要进行周期性的重写工作，否则无法长期保持存储状态。也正是由于需要周期性的定时刷新，因此这种利用电容存储电荷性质制成的存储器被称为“动态”存储器。

2. 动态随机存储器的刷新

刷新是一项周期性的工作，刷新周期指的是刷新工作的间隔时间，它主要取决于电容可以保持状态的时间。通常刷新周期被设定为 2～16ms，也有一些特殊器件的刷新周期可以延长到 100ms 以上。而且刷新一般由 DRAM 自行进行控制，不受 CPU 的干预控制，因此刷新对于 CPU 是透明的。刷新的执行方式多种多样，本书只介绍最基本的三种方式：集中刷新、分散刷新和异步刷新（透明刷新）。为了方便描述，使用 T_A 表示存储器的访问周期，