

第3章 计算机系统

本章主要介绍计算机系统的构成,包括硬件系统和软件系统。首先介绍计算机系统结构,接着以冯·诺依曼体系结构为基础介绍计算机硬件系统的各个组成部件以及它们的基本工作原理,最后介绍软件系统的基础知识和常用软件。

3.1 概 述

一个完整的计算机系统包括硬件系统和软件系统两大部分,如图 3-1 所示。

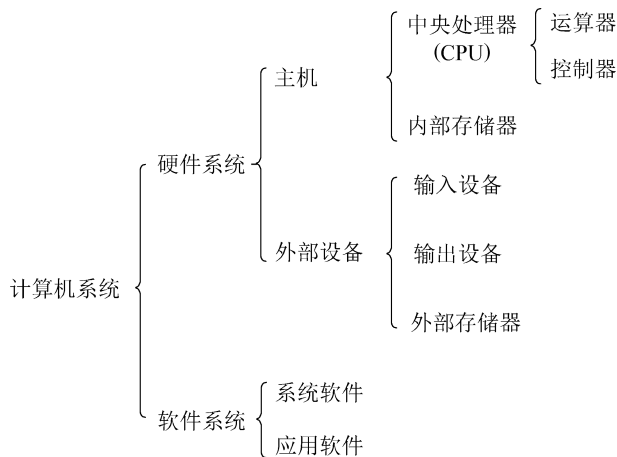


图 3-1 计算机系统的组成

硬件系统是指构成计算机的所有实体部件的集合,是由机械、光、电、磁等器件构成的具有计算、控制、存储、输入和输出功能的物理设备,它包括主机和外部设备。硬件系统是计算机系统运行的物理平台,是计算机完成任务的物质基础。

为了使计算机能正常工作,发挥其作用,除了这些看得见、摸得着的硬件设备之外,计算机系统中还需要有完成各项任务的程序以及支持这些程序运行的支撑平台,这就是计算机软件系统。

软件系统是计算机系统上可运行的软件的集合。软件是一系列按照特定顺序组织,并且能够完成特定功能的计算机数据和指令的集合。软件是用户和硬件之间的接口,主要解决如何管理和使用计算机的问题。通常,人们把不安装任何软件的计算机称为“裸

机”。裸机由于不配备任何软件,其本身不能完成任何功能,只有安装了一定的软件后才能发挥其功效。硬件是计算机的物质基础,而软件是计算机的逻辑载体。为了方便用户使用,使计算机系统具有较高的效用,在进行计算机系统的总体设计时,必须全局考虑硬件和软件之间的相互联系以及用户的实际需求。在计算机技术的发展进程中,计算机软件随着硬件技术的迅猛发展而发展;而软件的不断完善又促进了硬件的更新换代,两者的发展密切交织、缺一不可。

3.2 计算机硬件系统

3.2.1 冯·诺依曼体系结构

现代计算机虽在硬件系统结构上有诸多分类,但就其本质而言,多数都是基于美籍匈牙利科学家冯·诺依曼于1946年提出的冯·诺依曼体系结构。因此,这类计算机也称为冯·诺依曼型计算机。冯·诺依曼体系结构的核心思想是“存储程序”,把程序本身当作数据来对待,程序和该程序需要处理的数据以二进制方式进行存储,并确定了计算机的五大基本部件和基本工作原理。

根据此体系结构组成的计算机,必须具有以下功能。

- 把程序和程序所需的数据送至计算机中。
- 必须具有长期记忆程序、数据、中间结果以及最终运算结果的能力。
- 具有完成各种算术运算、逻辑运算和数据传送等数据加工处理的能力。
- 能够根据需要控制程序走向,并能根据指令控制机器的各部件协调操作。
- 能够按照要求将处理结果输出给用户。

为了完成上述功能,计算机必须具备五大基本部件,如图3-2所示。

- 输入设备:用来输入程序和数据的部件。
- 存储器:用来存放程序、数据和计算结果的部件。
- 运算器:用来进行数据加工的部件。
- 控制器:用来控制程序有条理执行的部件。
- 输出设备:用来输出结果的部件。

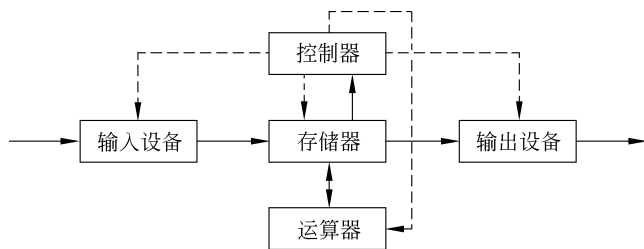


图 3-2 冯·诺依曼体系结构

注: ——— 数据/指令流 - - - - - 控制流

在冯·诺依曼体系结构形成之前,程序被看作控制器的一部分,而数据则存储在主存中,两者是区别对待的。而冯·诺依曼体系结构把程序当成数据来对待,程序和数据都是以相同的方式存储在主存中,这对于现代计算机的自动化和通用性起到了至关重要的作用。

冯·诺依曼型计算机的工作原理是将数据和预先编好的程序,输入并存储在计算机的主存储器中(即“存储程序”);计算机在工作时能够自动且高速地逐条取出指令,并加以执行(即“程序控制”)。这就是“存储程序”原理,是现代计算机的基本工作原理。

典型的计算机硬件组织结构是在冯·诺依曼体系结构上进行了细化,如图 3-3 所示。图中,冯·诺依曼体系结构中的控制器和运算器被集成在 CPU 中,主存对应存储器,磁盘、键盘、鼠标和打印机等各种设备分别对应输入设备和输出设备,各种总线(图中以空心箭头表示)对应于体系结构中的互连线,用于传输命令和数据。

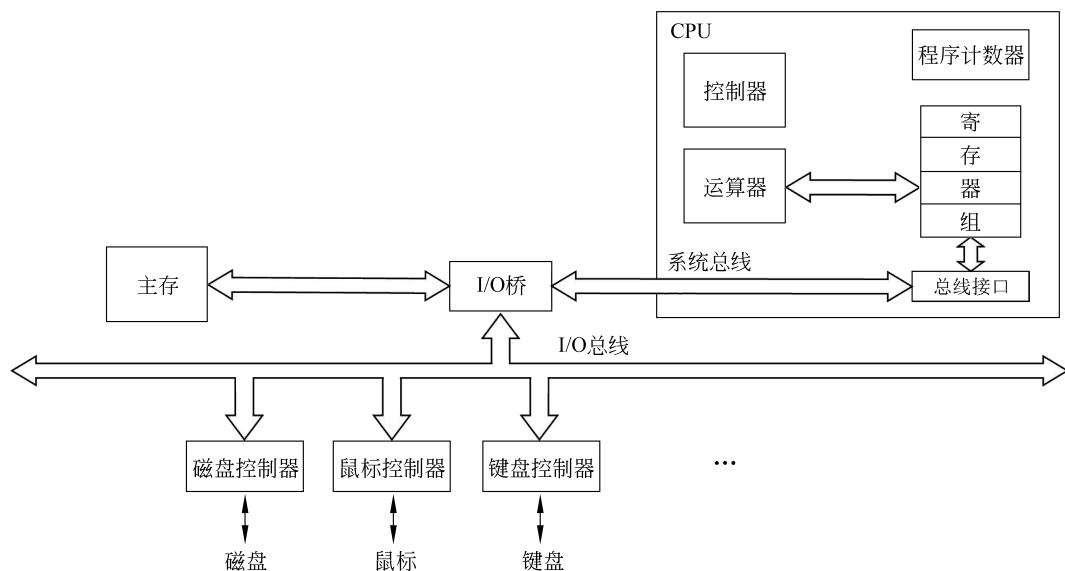


图 3-3 典型计算机硬件组织结构

3.2.2 中央处理器

中央处理器(Central Processing Unit,CPU)是计算机系统的“大脑”,是现代计算机系统的核心部件,承担着计算机指令的执行任务以及数据的处理任务,通过执行指令,控制各类硬件和软件协同完成任务。

1. CPU 的结构

CPU 是一块超大规模的集成电路,一般由算术逻辑运算器(Arithmetic Logic Unit, ALU,简称运算器)、控制器(Control Unit,CU)以及一些寄存器组通过 CPU 的内部总线连接成一个有机的整体,如图 3-4 所示。这里的算术逻辑运算器、控制器和寄存器并不是

某一个单独部件的名称,而是一组功能部件的统称。

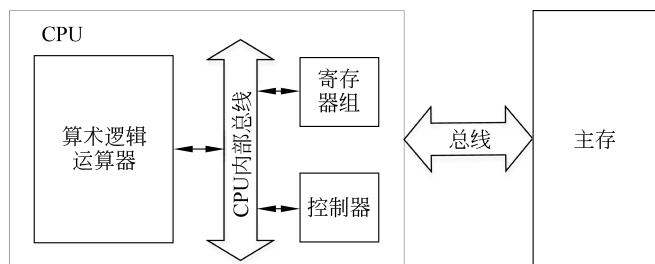


图 3-4 CPU 内部结构

控制器是整个 CPU 的指挥控制中心,是发布命令的“决策机构”,负责协调和指挥整个计算机系统有序工作。它的主要功能包括指令的分析、指令及操作数的传送、产生控制与协调整个 CPU 工作所需的时序逻辑等。一般由指令寄存器 (Instruction Register, IR)、指令译码器 (Instruction Decoder, ID)、操作控制器 (Operation Controller, OC) 和程序计数器 (Program Counter, PC) 等部件组成。

指令寄存器 (IR) 用来存放当前从主存储器取出的正在执行的一条指令。

指令译码器 (ID) 用来对当前存放在指令寄存器 (IR) 中的指令进行译码,分析这条指令的功能,以此来决定该指令操作的性质和方法。

操作控制器 (OC) 用来产生各种操作控制信号。

程序计数器 (PC) 用来存放下一条将要执行的指令所在主存单元的地址。

CPU 工作时,根据程序计数器 (PC) 里面存储的指令地址,操作控制器 (OC) 从主存中取出要执行的指令,存放在指令寄存器 (IR) 中;然后用指令译码器 (ID) 对指令进行译码,提取出指令的操作码、操作数等信息,操作码将被翻译 (译码) 成一组控制信号,用于控制运算器进行相应的运算、传输数据等操作;通过操作控制器 (OC),按照一定的时序,向相应的部件发出微操作控制信号,协调 CPU 其他部件工作;操作数经译码后分成地址或数据本身,根据控制信号取出所需的数据,送到运算器 (ALU) 中进行相应的操作;得出的结果根据控制信号保存到相应的寄存器中。

运算器是 CPU 中进行数据加工处理的部件,主要实现数据的算术运算 (如加、减、乘、除等) 和逻辑运算 (如与、或、非、异或等)。运算器是根据接收到的控制器的命令进行操作的,即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的。运算器输出的是运算的结果,一般会暂存在相应的寄存器中。此外,运算器还会根据运算的结果输出一些条件码到程序状态寄存器 (Program Status Word, PSW) 中,用于标识当前的运算结果状态和一些特殊情况,如进位、溢出等。

寄存器组由一组寄存器构成,分为专用寄存器组和通用寄存器组,用于临时保存数据,如操作数、运算结果、指令、地址和机器状态等。专用寄存器组保存的数据用于表征计算机当前的工作状态。例如,程序状态寄存器保存 CPU 当前状态的信息,如是否有进位、是否允许中断等。通用寄存器组保存的数据可以是参加运算的操作数或运算的结果,其用途广泛并且可由程序员规定其用途,通用寄存器的数目因微处理器不同而异。通常,

要对寄存器组中的寄存器进行编址,以标识访问哪个寄存器,编址一般从 0 开始,寄存器组中寄存器的数量是有限的。

指令和数据在 CPU 中的传输通道称为 CPU 的内部总线,总线实际上是一组导线,是各种公共信号线的集合,用作 CPU 中所有部件之间进行信息传输的共同通道。CPU 用它来传输地址、数据和控制信号。其中,用来传输 CPU 发出的地址信息的总线称为地址总线;用来传输数据信息的总线称为数据总线;用来传送控制信号、时序信号和状态信息等的总线称为控制总线。

2. 指令系统

机器指令(简称指令)是计算机唯一能够识别并执行的指令,是 CPU 执行的最小单位,它的表现形式是二进制编码。指令通常由操作码和操作数两部分组成,如图 3-5(a)所示,操作码指出该指令所要完成的操作,即指令的功能,操作数指出参与运算的对象或者运算结果所存放的位置等。一台计算机所能执行的全部指令的集合称为该计算机的指令系统。

指令的长度通常是一个或几个字长,长度可以是固定的,也可以是可变的。图 3-5(b)给出了某个型号 CPU 的加法指令的示例。该指令长度为 16 位(16 位机的一个字长),从左至右标识各位为 b15~b0。其中,b15~b12 代表的是操作码,为 0001,在该 CPU 的指令系统中表示加法操作;b11~b0 代表的是操作数,由于该指令需要 3 个操作数,b11~b0 被拆分成 3 段,分别对应两个相加数(源操作数)和一个求和结果(目的操作数)。b11~b9 对应保存目的操作数的寄存器地址,在该示例中为 010,表示寄存器 R2;b8~b6 与 b2~b0 分别对应保存源操作数的寄存器地址,分别为 R1 和 R2。因此,这条指令的功能是将寄存器 R2 和 R1 中保存的数值进行加法运算,所得到的结果存回寄存器 R2 中。b5~b3 用于扩展加法指令的操作,此处不做解释。

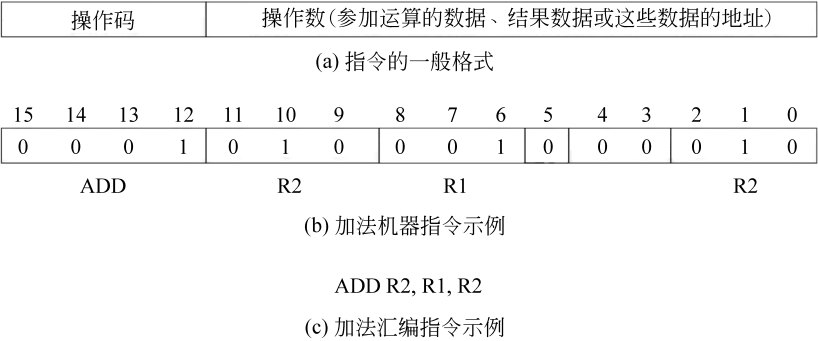


图 3-5 指令的一般格式及示例

指令是由 0 和 1 构成的,计算机易于阅读和理解,但并不适合人阅读和使用。因此,在指令中引入助记符来表示操作码和操作数,以帮助人理解和使用指令,这样的指令称为汇编指令。如图 3-5(c)所示,用 ADD 对应操作码 0001,用来标识该指令是加法指令;用 R1 和 R2 标识用到的寄存器。计算机不能直接执行汇编指令,要由编译器将其翻译成对

应的机器指令才可以执行。对图 3-5(c)的 ADD 指令,汇编器会将其翻译成图 3-5(b)中的指令的形式。

CPU 在设计时就由指令集体系结构规定了一系列与其硬件电路相配合的指令系统,不同型号的 CPU 其指令系统也不相同。在指令系统中,通常定义有以下的指令类型。

(1) 数据处理指令:用于对 CPU 中寄存器里存放的数据进行算术运算和逻辑运算,结果仍然存放在 CPU 的寄存器中。例如,算术运算、逻辑运算、比较指令等。

(2) 数据传送指令:用于在寄存器之间、寄存器与主存储器之间进行数据传输的指令。例如,将数据从主存移入寄存器的 LOAD 指令,将数据从寄存器送入主存的 STORE 指令,等等。

(3) 输入输出指令:用于在主机与外设之间进行数据传输的指令,包括各种外围设备的读和写指令等。有的计算机将输入输出指令包含在数据传送指令中。

(4) 程序控制指令:用于控制程序的流向,能改变指令执行顺序的指令。例如,条件跳转指令、无条件跳转指令、转子程序指令等,都是将程序计数器的值更改为一个非顺序的值,使得下一条指令从新的位置开始执行。

在使用计算机时,人们可以用该计算机所配置的 CPU 指令系统中的所有指令来编写程序。程序就是用于控制计算机行为,完成某项任务的指令序列。图 3-6 是一个程序示例,为了便于阅读,采用汇编指令来编写,分号后面是程序的注释以帮助人们阅读和理解程序,而计算机将忽略这些注释。

```
mov    #0,    R0    ;将寄存器 R0 置 0
mov    #1,    R1    ;将寄存器 R1 置 1
add     R1,    R0    ;将 R0 与 R1 相加,结果保存到 R0
loop:  add     #1,    R1    ;R1 自加 1
      cmp     R1,    #100 ;比较 R1 和 100 的大小
      ble     loop    ;如果 R1 小于或等于 100,从 loop 那条指令开始执行
      halt                    ;程序结束
```

图 3-6 汇编语言程序示例

这段程序的功能是计算 $1+2+\cdots+100$ 的和。程序开始时,首先将 R0 寄存器的值设为 0,R1 寄存器的值设为 1;然后将 R1 的值加到 R0 上,同时 R1 增 1;接着将 R1 的值与 100 进行比较,如果 $R1 \leq 100$,则重复执行将 R1 加到 R0 上以及 R1 增 1 的操作,再将 R1 和 100 进行比较。这种操作重复执行,一直到 $R1 > 100$ 时结束,同时程序结束。其中,add 是数据处理指令,cmp 是数据处理指令,ble 是程序控制指令。ble 与 cmp 是一起使用的,当 cmp 比较结果为小于或等于时,则该指令被执行,程序不再是顺序执行,而是跳转到 loop 所标识的那条指令。

通常,用机器指令和汇编指令来编写程序是非常困难的,程序员会把大量的精力和时间浪费在记忆指令格式、操作码和操作数等与实现程序功能无关的事情上。为此,人们又设计了更加贴近于自然语言和数学表达的高级语言,它的书写方式更接近人们的思维习惯,写出的程序更便于阅读和理解,也更易于纠错和修改,这给程序的调试带来了极大的便利,使得人们在编程时把精力和时间放在程序的功能实现上。图 3-6 所示的汇编语言

程序用高级语言 Python 编写后如图 3-7 所示。用高级语言编写的程序,经过编译器编译和连接,即可生成计算机能识别的机器指令构成的程序。

```
x = 1                //将 1 赋值给变量 x
sum = 0              //将 0 赋值给变量 sum
while x < 101:        //当 x<101 时,执行 while 循环体内程序;否则,跳出循环体
    sum = sum + x      //sum 与 x 相加,结果赋值给 sum
    x = x + 1          //x 自加 1
print("1+2+...+100 =",sum) //输出 sum 的值
```

图 3-7 Python 高级语言程序示例

3. CPU 的工作原理

冯·诺依曼体系机构中提到计算机的工作原理是将预先编好的程序和原始数据,输入并存储在计算机的主存储器中;计算机在工作时能够自动且高速地按照程序逐条取出指令,并加以执行,即计算机的工作过程实际上就是 CPU 自动循环执行一系列指令的过程。CPU 执行一条指令的时间称为指令周期,其中完成一个基本操作所需要的时间称为机器周期。不同型号的 CPU 可能执行指令的机器周期数不同,但是通常都可归为取指令、分析指令、执行指令和写结果 4 个步骤,如图 3-8 所示。



图 3-8 指令周期

(1) 取指令：指令通常存储在主存中,CPU 从程序计数器(PC)中获取将要执行的下一条指令的存储地址。根据这个地址,将指令从主存中读入 CPU,并保存在指令寄存器(IR)中。

(2) 分析指令：也称为译码,由指令译码器(ID)对存在指令寄存器(IR)中的指令进行译码,分析出指令的操作码以及操作数或操作数所存放的位置。

(3) 执行指令：将译码后的操作码分解成一组相关的控制信号序列,控制各部件相互协作完成指令动作,包括从寄存器读数据、输入到运算器进行算术或逻辑运算等。

(4) 写结果：将指令执行阶段产生的结果写回到寄存器/内存,并将产生的条件反馈给程序状态寄存器(PSW)。

以上的机器周期的划分是粗粒度的,事实上每个机器周期所包含的动作很难在一个时钟周期内完成,应进一步将每个机器周期进行细化,细化后的每个动作可以在一个时钟周期内完成,不能再细分。在这里,一个 CPU 时钟周期也称为一个节拍。例如,取指令可以再细分为:

- ① 将程序计数器(PC)的值装入主存的地址寄存器。
- ② 将地址寄存器所对应的主存单元的内容装入主存的数据寄存器。
- ③ 将主存数据寄存器的内容装入指令寄存器(IR),同时程序计数器(PC)内的地址自动“加 1”(注意,这里的“加 1”不是加一个主存单元的大小,而是加一条指令所占用的主

存空间的大小)。

由此可见,取指令这个动作要花费3个时钟周期。对现代计算机来说,每个时钟周期非常短。我们常说的CPU主频就是CPU的工作频率,即CPU一秒钟内执行的时钟周期的次数。因此,主频越高,一秒钟内执行的时钟周期的次数也就越多,CPU的执行速度也就越快。例如,对主频为3.3GHz的CPU而言,每秒将完成33亿个时钟周期,每个时钟周期的时间长度为0.303ns,而取指令需要花费3个时钟周期,即花费0.909ns。

下面以数据传送指令为例,详细介绍CPU是如何工作的。假设将要执行的是地址为0300H的主存单元中的指令,其指令码是1940H,其中高4位1H是操作码,表示当前指令是数据传送指令;低12位940H是操作数,表示操作数所在的主存单元的地址。该指令完成的操作是将操作数所指的主存单元的数据传送到累加器中。指令的执行过程如图3-9所示。

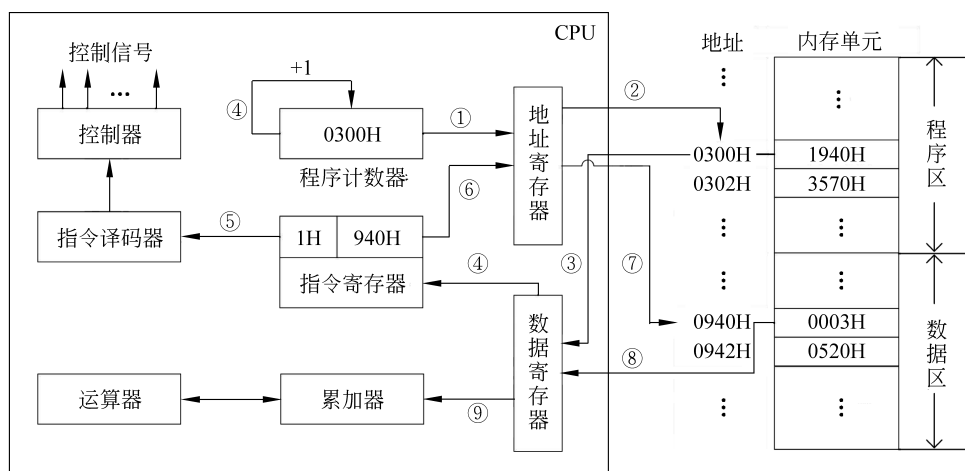


图 3-9 数据传送指令的执行过程

图3-9中指令的具体步骤如下。

- (1) 将程序计数器(PC)的值0300H装入地址寄存器。
- (2) 根据地址寄存器中的内容0300H找到相应的主存单元。
- (3) 将地址为0300H主存单元的内容1940H装入数据寄存器。
- (4) 将数据寄存器的内容装入指令寄存器(IR),同时程序计数器(PC)自动“加1”。
- (5) 指令译码器(ID)对指令寄存器(IR)中的指令进行译码,分析出是数据传送指令,以及需要传送的数据的地址是940H,并产生相应的控制信号。
- (6) 控制器控制将操作数地址940H装入地址寄存器。
- (7) 根据地址寄存器中的内容0940H找到相应的主存单元。
- (8) 将地址为0940H主存单元的内容0003H装入数据寄存器。
- (9) 将数据寄存器的内容传送到累加器中。

在一条指令的最后一个节拍完成后,控制器复位指令周期,从取指令节拍重新开始运行,此时程序计数器(PC)的内容已被自动修改,指向的是下一条指令所在的主存地址,所

以取到的就是下一条指令。数据处理指令、数据传送指令和输入输出指令的执行不会主动修改 PC 的值,PC 将会自动指向程序顺序上的下一条指令;而程序控制指令的执行会主动改变程序 PC 的值,使得程序的执行将不再是顺序的。

以图 3-6 所示的汇编语言程序为例,来理解 PC 的变化。假设这段程序在主存中的存储形式如图 3-10 所示。

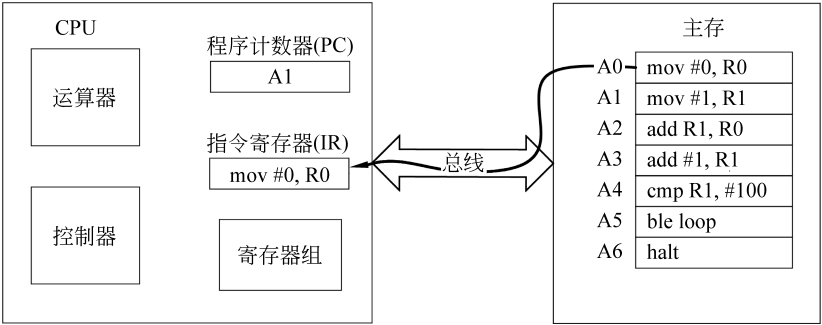


图 3-10 程序在主存中的存储形式

当这段代码被加载到主存中并将开始执行时,操作系统将程序计数器(PC)的值设为 A0,在取指令阶段将 A0 地址的指令“mov #0,R0”取出存入指令寄存器(IR),同时 PC “加 1”变成 A1。这条指令执行结束后,控制器复位指令周期,从取指令阶段重新执行,也就是根据 PC 的值 A1 取下一条指令“mov #1,R1”。该过程将一直执行到 ble 指令,该指令执行完后,将会对 PC 的值进行覆盖,将 loop 对应的指令地址写入 PC 中,使得下一条指令将不再是顺序执行的,而是跳转到 loop 对应的指令开始执行。当条件满足时,ble 指令的执行不修改 PC 的值,此时顺序执行下一条指令即 halt 指令。

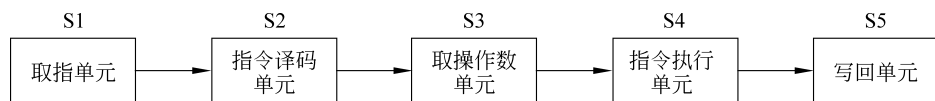
4. CPU 的性能

计算机系统性能的优劣很大程度上取决于 CPU 的性能。计算机设计者一直在努力提高计算机的性能,其中一种方法就是通过提高 CPU 的主频来使它运行得快一些,但每个新设计都要局限于当时的技术条件,主频的提高也是有极限的。因此,多数设计者都采取了并行处理(同时处理两件或两件以上事情)的方法,以便在给定主频下取得更好的性能。

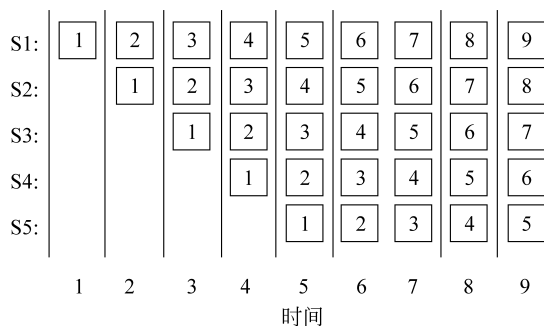
一般来说,并行可分为指令级并行和处理器级并行两种,前者是在指令之间采用并行,使计算机在单位时间内处理更多的指令;后者是指让多个 CPU 同时工作,解决同一个问题。

指令流水线是在指令执行周期分节拍的基础上,将指令执行分解成更细的步骤(如 10 个或更多),每个步骤由精心设计的硬件分别执行,使得同一时刻 CPU 能执行多条指令,实现指令级并行,如图 3-11 所示。

超标量体系结构在指令流水线的基础上,采用多条指令流水线提高 CPU 的性能。图 3-12 是双流水线的一种设计示例。该设计中两条流水线共用一个取指令部件,可以一次取两条指令,然后分别将指令送到各自的流水线执行。要实现这种设计,需要多个功能单元来同时执行指令,并且同时执行的指令不能有资源冲突。



(a) 5个节拍指令周期



(b) 随时间变化的指令流水执行过程

图 3-11 指令流水线

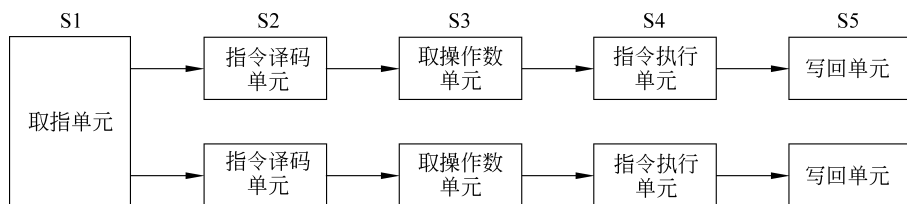


图 3-12 超标量体系结构双流水线设计示例

多核处理器也称为单芯片多处理器,其思想是将两个或多个完整的计算引擎(内核)集成到同一个芯片内,各个处理器并行执行不同的任务,相对于单 CPU 的指令流水线实现的指令级并行,多核处理器实现了处理器级并行。双核处理器结构如图 3-13 所示。

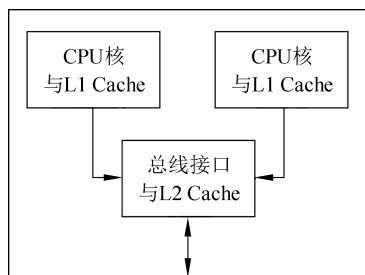


图 3-13 双核处理器结构

3.2.3 存储器系统

计算机系统存储器一般分为主存储器(简称主存,又称内存)和辅助存储器(又称

外存)。主存储器运行速度快,可与 CPU 直接进行信息交换,但主存储器容量相对较小且价格昂贵,在系统断电后,其保存的信息会丢失。辅助存储器属于外部设备的范畴,其存取速度比主存储器慢,且不能和 CPU 直接进行信息交换,但辅助存储器存储容量大且价格低廉,系统断电后其保存的信息不会丢失,存储的信息很稳定。不论是主存储器还是辅助存储器,其访问速度与 CPU 的运行速度都有很大的差距,通常是数量级上的差别(如微秒和纳秒的差别)。为了匹配 CPU 和主存储器之间的访问速度,通常会在 CPU 和主存储器之间插入一个比主存储器速度更快、容量更小的存储设备(如 Cache)。

事实上,为了实现容量、速度、成本的最高性价比,计算机系统通常采用层次结构的存储器系统,这是一个具有不同容量、访问时间和成本的存储设备的层次结构。由于计算机的主存储器不能同时满足存储容量大、存取速度快和成本低的要求,因此在计算机中必须要有速度由慢到快、容量由大到小的多级、多层次结构的存储器,以最优的控制调度算法和合理的成本,构成性能可接受的存储系统。如图 3-14 所示,最上层是 CPU 内部的寄存器,其存储速度非常快,CPU 可以在一个时钟周期内访问它,但其容量非常小,一般只有几百字节;下一层是高速缓冲存储器(简称高速缓存),其存储速度比访问主存快,CPU 可以在几个时钟周期访问它,一般容量是 32KB 到几兆字节;再往下一层是主存储器,CPU 可以在几十到几百个时钟周期内访问它;接下来一层是慢速但是容量很大的本地辅助存储器,用于保存永久存放的数据;最下面一层是用于后备存储的远程辅助存储器,如 Web 服务器、磁盘阵列等。

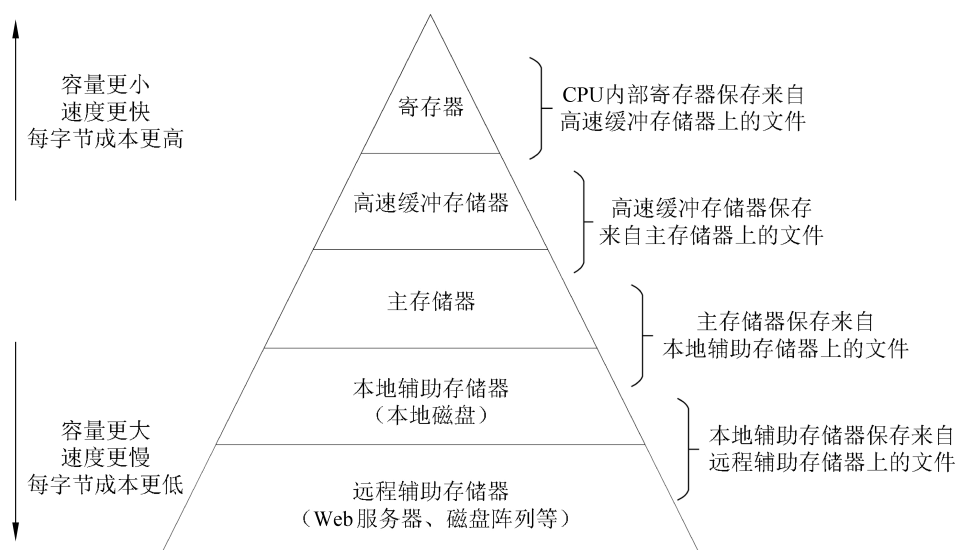


图 3-14 存储器系统层次结构

存储器层次结构的中心思想是：位于上一层的更快、更小的存储设备作为位于下一层的更慢、更大的存储设备的缓存。也就是说，层次结构中的每一层都缓存来自较低一层的数据信息。例如，本地磁盘作为从 Web 服务器上取出的文件(如 Web 页面)的缓存，主存储器作为本地磁盘上数据的缓存，以此类推，直到最小的缓存——CPU 内部寄存器。

基于缓存的存储器层次结构之所以能够行之有效,是因为较慢的存储设备比较快的存储设备容量大且价格便宜,还有程序的如下局部性原理的缘故。

- 时间局部性: 最近被访问的内容(指令或数据)很快还会被访问。
- 空间局部性: 访问了某个存储单元的内容后,其附近的存储单元的内容也将被访问。

在该层次结构中,自上而下的读写速度越来越慢、存储容量越来越大、成本/字节越来越低廉。首先,CPU 内部寄存器的访问时间是纳秒级的,一般是几个纳秒,高速缓存的访问时间接近 CPU 内部寄存器的访问时间,主存存储器的访问时间是几十个纳秒,而磁盘的访问时间至少是 10 毫秒以上;其次,CPU 中寄存器的大小一般为 128 字节或更多一点,高速缓存可以达到几兆字节,主存是几百兆字节到几千兆字节,而磁盘的容量是几吉字节或几千吉字节;再次,主存的价格一般为几美元/兆字节,而磁盘的价格则为几美分/吉字节。

1. 主存储器

主存储器是计算机中重要的部件之一,用于存放 CPU 运行时所需要用到的指令和数据,并能由 CPU 直接随机存取。因此,主存的性能对计算机的影响非常大。现代计算机为了提高性能,同时兼顾合理的造价,往往采用多级存储体系结构。

主存一般采用半导体存储器,与辅助存储器相比,主存具有容量小、读写速度快、价格高等特点。主存储器的结构如图 3-15 所示,包括用于存储数据的存储体和外围电路,外围电路用于数据交换和存储访问控制,与 CPU 或高速缓存连接。

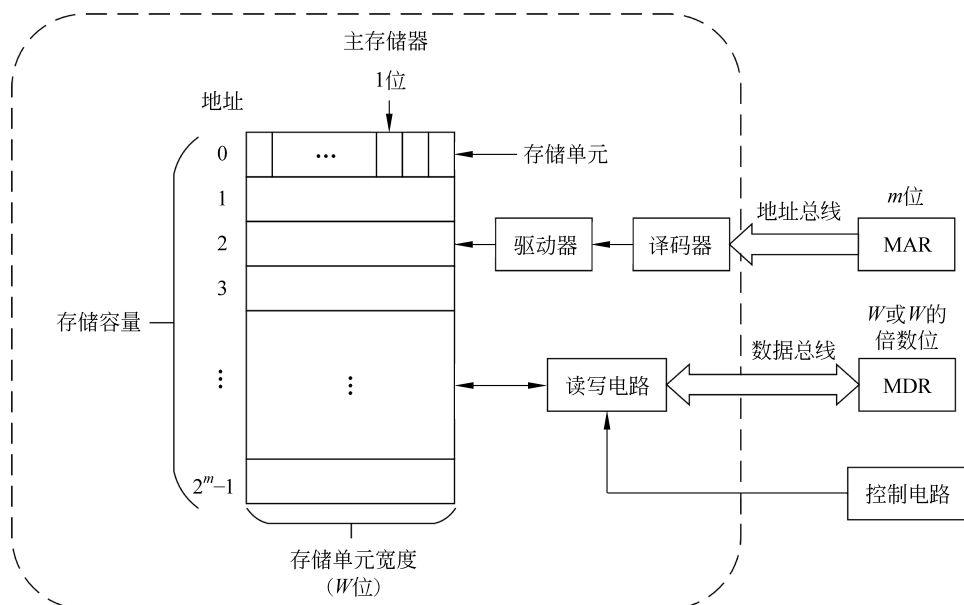


图 3-15 主存储器的结构

外围电路中有两个非常重要的寄存器——地址寄存器(Memory Address Register,

MAR)和数据寄存器(Memory Data Register,MDR),前者是用于临时保存 CPU 需要访问的主存单元的地址,后者用于临时保存从主存单元内读出或写入主存单元内的数据。CPU 需要访问主存时,首先将要访问的地址送入 MAR,经地址译码器、驱动器等电路,才能找到所需访问的主存单元。如果是读主存,则在控制电路控制下,将 MAR 指向的主存单元内的数据送入 MDR 中,然后发送到 CPU 或高速缓存;如果是写主存,则首先要将需要写入的数据送到 MDR 中,在控制电路控制下,将 MDR 中的数据写入 MAR 所指向的主存单元中。

主存中存储信息的地方称为存储体,它是主存的核心,是由若干个存储单元组成的。存储单元是主存中存储数据的最基本单元,它由一串二进制数组成,可以代表数字、字符等信息,存储单元的大小一般为 8 位二进制,即一个字节(Byte)。每个存储单元都有一个编号,这个编号就是主存的地址。主存地址也是用二进制数来表示的,如果计算机中的地址线有 m 根,则表示地址的二进制数就有 m 位,那么主存就有 2^m 个存储单元,存储容量就是 2^m 字节,主存地址编码的范围是 $0 \sim 2^m - 1$ (从 0 开始编码)。

CPU 是通过主存地址来对主存的存储单元存放的数据进行读写的,这种读写操作通常被称为访问主存。访问主存时可根据主存地址独立地对各存储单元进行读写,访问时间与被访问地址无关,因此主存又称为随机访问存储器(Random Access Memory, RAM)。

RAM 是构成主存的主要部分,其内容可以根据需要随时按地址读出或写入,以某种电触发器的状态存储,断电后信息无法保存,用于暂存数据。RAM 可分为动态随机存取存储器(DRAM)和静态随机存取存储器(SRAM)两种。DRAM 中存储的信息是电荷形式保存在集成电路的小电容中,由于电容漏电,因此数据容易丢失。为了保证数据不丢失,需要在很短的时间内不断地对 DRAM 充电(称为定时刷新)。由于需要周期性刷新,因此 DRAM 存取速度相对较慢。SRAM 是利用双稳态的触发器来存储 0 和 1,只要不断电,存储的信息就不会丢失。“静态”的意思就是指它无须像 DRAM 那样定时刷新,因此 SRAM 存取速度快,工作状态稳定,但其价格比 DRAM 要贵得多,因此主存采用相对便宜的 DRAM 来存储信息。

由于断电后 RAM 中所存储的信息会丢失,因此也称 RAM 为易失性存储器。主存中还有一类非易失性存储器,即使电源供应中断,其所存储的数据也并不会丢失,如只读存储器(Read-Only Memory,ROM)等。ROM 是一种一次写入、多次读出的存储器,它也可以随机访问,但只能随机读出信息,而不能重新写入新信息,其信息通常是厂家在制作芯片时在特定状态下写入的。因为 ROM 有断电不丢失信息的特点,计算机中常用 ROM 来存放重要的信息,例如系统的引导程序、自检程序、监控程序等。计算机开机时加载的第一个软件 BIOS(Basic Input Output System,基本输入输出系统)就是一组固化在计算机主板上一个 ROM 芯片中的程序,它保存着计算机最重要的基本输入输出程序、开机后自检程序和系统自启动程序。

下面是主存的主要性能指标。

(1) 主存容量:一个主存中容纳的存储单元总数通常称为该存储器的存储容量。存储容量这一概念反映了存储空间的大小。存储容量一般用字节数(B)来表示,如 512KB、

256MB、8GB、1TB。其中, $1\text{KB}=2^{10}\text{B}$, $1\text{MB}=2^{20}\text{B}$, $1\text{GB}=2^{30}\text{B}$, $1\text{TB}=2^{40}\text{B}$ 。

(2) 存储器访问时间: 又称读写时间, 是指从启动一次存储器读写操作到完成该操作所经历的时间。以读操作为例, 从 CPU 发出读操作命令给主存开始, 一直到 CPU 收到数据为止, 即为一个读操作的时间。目前, 主存访问速度总比 CPU 速度慢得多。一次访问时间大约为 $5\sim 10\text{ns}$, 比 CPU 的速度慢 10 倍以上。

(3) 存储周期: 是指连续启动两次独立的存储器操作(如连续两次读操作)所需间隔的最小时间。通常, 存储周期略大于存储器访问时间。

2. 高速缓冲存储器

早期的计算机系统的存储器层次结构只有 3 层: CPU 内部寄存器、主存储器和本地辅助存储器。随着 CPU 和主存访问速度之间的差距越来越大, CPU 发出访问主存的请求后, 往往要等待多个时钟周期才能获取到主存的内容, 主存越慢, 则 CPU 等待的时间就越长, 效率就越低下。系统设计者为了解决 CPU 和主存之间速度不匹配的问题, 提出了利用容量小但速度接近 CPU 的存储设备与大容量低速的主存组合使用的方法, 以适中的价格得到速度和 CPU 接近的大容量存储器。这种容量小但速度接近 CPU 的存储设备就是高速缓冲存储器(Cache, 简称高速缓存)。

Cache 是指存取速度比一般随机存取存储器来得快的一种 RAM, 一般而言它不像主存那样使用 DRAM 技术, 而使用昂贵但快速的 SRAM 技术, 它容量比较小但速度比主存高得多, 接近于 CPU 的速度。在计算机存储器系统的层次结构中, Cache 介于 CPU 和主存之间, 可以将其集成到 CPU 内部, 也可置于 CPU 之外。图 3-16 是一种典型的高速缓存配置方式, 位于 CPU 芯片上的 L1 高速缓存的容量可达几十 KB, 其访问速度几乎与 CPU 中的寄存器组访问速度一样快; 位于 CPU 外的 L2 高速缓存容量更大, 可到几 MB, 访问 L2 的时间要比访问 L1 的时间多几倍, 但仍比访问主存的时间快 10 倍。

通过对大量典型程序进行分析, 发现 CPU 从主存读取指令或数据时, 在一定时间内只对主存局部区域进行访问。这是因为某个程序的指令和数据在主存内都是连续存放的, 并且有些指令和数据往往会被多次调用(如循环程序), 这也说明指令和数据在主存的地址分布不是随机的, 而是相对的簇聚, 因此 CPU 在执行程序时, 访问主存具有相对的局部性, 这就是程序的局部性原理。根据这一原理, 很容易设想, 在 CPU 访问了主存的某个主存单元后, 将该单元及其相邻的(多个)单元的内容预先从主存读入 Cache, 这样当 CPU 下次访问主存时, 有很大的概率会访问上次访问过的主存单元相邻的单元, 由于已经预先将相邻(多个)单元的内容读入 Cache, 就不需要再去访问主存而是直接从 Cache 中读取数据即可, 从而缩

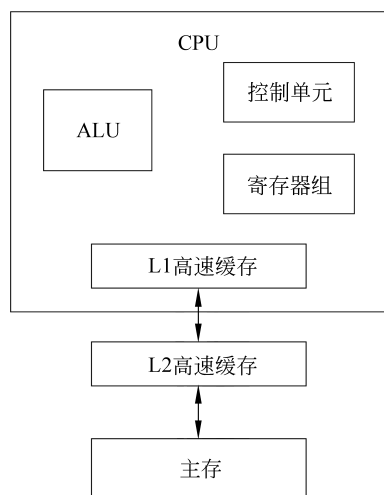


图 3-16 一种典型的高速缓存配置方式

短了 CPU 读取数据的时间。

Cache 工作原理非常简单：把当前和未来一段时间内使用频率最高的存储内容保存起来，也就是将 CPU 可能即将使用的信息暂存在 Cache 中。这样，当 CPU 需要从主存读取这些信息时，由于这些信息已经暂存在 Cache 中，因此不需要访问主存去获取，而是直接从 CPU 内部的 Cache 中获得，且 Cache 的访问速度与 CPU 的运行速度相当。在 Cache 的支持下，CPU 需要读入主存中的数据时，先在 Cache 中查找，只有当在 Cache 中找不到时才会去访问主存，这样大大提高了 CPU 获取数据的速度，提高了整个系统的效率。

高速缓冲存储器最重要的技术指标是它的命中率，指的是在程序某一部分执行期间命中主存引用的比率。较大容量的 Cache 会提高命中率，但会增加其访问时间。一般来说，高速缓存的容量越大，命中率就越高，性能也就越好，但是访问速度会变慢且成本越高。

随着科技的发展以及工艺水平的提高，CPU 的速度越来越快，虽然主存的访问速度也在不断提高，但仍然赶不上 CPU 速度的提高。因此，对于现代高性能 CPU 而言，Cache 的地位越来越重要。目前，在进行高速缓存的设计时，需要考虑 Cache 的速度、Cache 的容量、Cache 数量、指令与数据是否需要分开以及如何合理组织多级 Cache 等问题。

3. 辅助存储器

辅助存储器又称为外存储器，简称辅存或外存，是指除了 CPU 寄存器、高速缓存和主存以外的存储器。与主存相比，辅存容量大、速度慢、价格低，而且断电后仍能保存数据，是“非易失性”的存储介质。辅存属于外部设备的范畴，它们通过各种专门接口与计算机进行通信。常见的辅存有硬盘、光盘等。

1) 硬盘

硬盘具有存储容量大、数据存取方便、价格低廉等优点，目前已经成为保存用户数据最重要的外部存储设备。一般硬盘可分为机械硬盘 (Hard Disk Drive) 和固态硬盘 (Solid State Drive)。

(1) 机械硬盘：机械硬盘即是传统的普通硬盘，由一个或多个表面涂有磁性材料的铝质薄盘组成，内含一个或多个正好浮于磁盘表面的磁头，磁头含有一个引导线圈，磁头与盘面之间隔着一层薄薄的空气垫，盘片是以坚固耐用的材料为盘基，将磁粉附着在平滑的铝合金或玻璃圆盘基上，因此也称为硬磁盘，如图 3-17 所示。磁头可沿盘片的半径方向运动，加上盘片每分钟几千转的高速旋转，磁头就可以定位在盘片的指定位置上进行数据的读写操作。当磁头中有正或负电流通过时，会磁化磁头正下方的磁盘面，并且根据电流的正或负，使得磁性材料的颗粒向左或向右偏转，以此记录 0 或 1，这个过程是向磁盘写数据。当要读出磁盘数据时，磁头通过已经被磁化的区域，此时会感应出正或负电流，由此判断磁盘颗粒的朝向而获得记录的数据。硬盘作为精密设备，尘埃是其大敌，因此多采用密封结构。

硬磁盘的盘片上围绕着轴心的一系列同心圆称为磁道，用于记录数据。从外向内，对

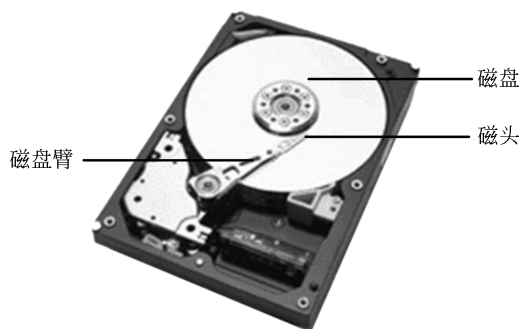


图 3-17 机械硬盘

磁道进行编址,最外圈磁道称为 0 磁道,如图 3-18(a)所示。每个磁盘有一个可伸缩的磁盘臂,它能从磁盘的轴心伸展到每个磁道,磁头就是附着在磁盘臂上进行数据读写的。硬磁盘一般由多个铝盘片叠起来构成,每个盘面都有自己的磁头和磁盘臂,所有的磁盘臂连在一起,如图 3-18(b)所示。在这种结构中,将位于同一半径的磁道合称为柱面。为了方便数据的记录,每个磁道又划分为若干的小区域,称为扇区。

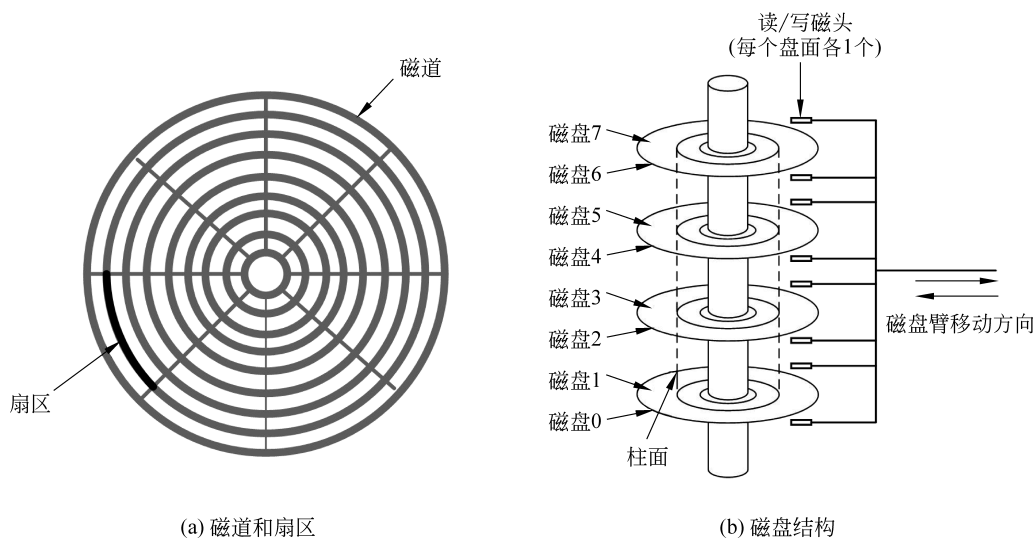


图 3-18 硬磁盘的结构

一般每个扇区可存放 512B 数据,数据区之前有用于读写前对磁头进行同步的前导区,数据区之后是纠错码,相邻的扇区之间是隔离带,一个扇区的结构如图 3-19 所示。因此,在存储数据时,磁道的存储区域不会被 100% 的使用,一般可用区域为 85%。

硬磁盘一旦通电,盘片就开始高速旋转。当接到读写命令时,在磁盘控制器的控制下,磁头根据给出的地址(由磁头号、柱面号和扇区号组成),首先按柱面号产生驱动信号进行定位,然后再通过盘片的转动找到具体的扇区,最后由磁面对应的磁头读取指定位置的信息,并传送到硬盘自带的高速缓存中。

根据磁盘的结构,由磁盘的参数可计算出按等量扇区划分方式的磁盘容量,公式

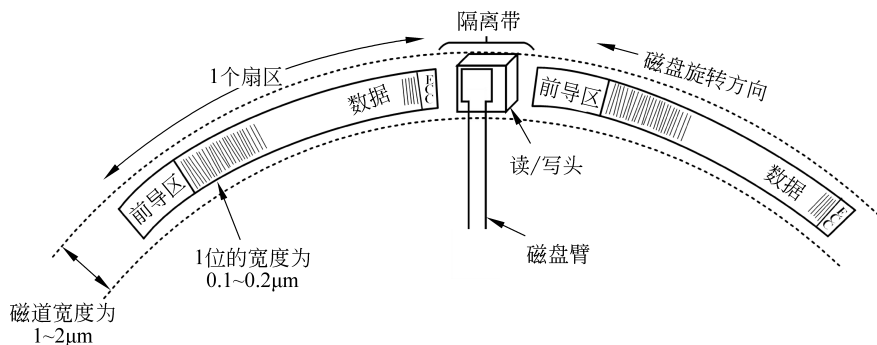


图 3-19 一个扇区的结构

如下。

磁盘容量(B)=磁面数×柱面数(磁道数)×扇区数×每扇区字节数(512B)

(2) 固态硬盘：固态硬盘是一种新式硬盘,是用固态电子存储芯片阵列制成的硬盘。其内部构造十分简单,固态硬盘的主体其实就是一块 PCB 板,而这块 PCB 板上最基本的配件就是控制芯片、缓存芯片(部分低端硬盘无缓存芯片)和用于存储数据的芯片。虽然它的技术和机械硬盘完全不同,但其在接口的规范和定义、功能及使用方法上与普通硬盘几乎相同,在产品外形和尺寸上也与普通硬盘完全一致。

固态硬盘按存储介质可分为两种：一种是采用闪存(FLASH 芯片)作为存储介质,另一种是采用动态随机存取存储器(DRAM)作为存储介质。基于闪存的固态硬盘,使用范围广,是固态硬盘的主要类别,它的外观可以被制作成多种模样,例如便携式硬盘、存储卡、U 盘等。这种固态硬盘最大的优点就是可以移动,而且数据保护不受电源控制,能适应于各种环境,适合于个人用户使用。一般它的擦写次数普遍为 3000 次左右,以 128GB 为例,可擦写的总数据量为 $128\text{GB} \times 3000 = 384\,000\text{GB}$ 。假设你每天下载 10GB 资料且当天删除的话,可用天数为 $384\,000/10 = 38\,400$ 天,也就是 $38\,400/365 = 105$ 年,即可不间断地用 105 年,也就是说理论上它可以无限读写。而基于 DRAM 的固态硬盘的应用范围较窄,属于非主流的设备,其应用方式可分为 SSD 硬盘和 SSD 硬盘阵列两种。它是一种高性能的存储器,使用寿命很长,美中不足的是需要独立的电源来保护数据安全。

与传统的机械硬盘相比,固态硬盘具有以下优点。

(1) 读写速度快：采用闪存或 DRAM 作为存储介质,读取速度相对机械硬盘更快。固态硬盘不用磁头,寻道时间几乎为 0。最常见的 7200 转机械硬盘的寻道时间一般为 12~14ms,而固态硬盘则可达到 0.1ms 甚至更快。

(2) 防震抗摔性：传统硬盘都是磁碟型的,数据储存在磁碟扇区里。而固态硬盘是使用闪存或 DRAM 芯片制作而成的,所以固态硬盘内部不存在任何机械部件,这样即使在高速移动甚至伴随翻转倾斜的情况下也不会影响到正常使用,而且在发生碰撞和震荡时能够将数据丢失的可能性降到最小。

(3) 低功耗：固态硬盘内部不存在任何机械活动部件,不会产生机械运动,因此能耗和发热量低、散热快。

(4) 无噪音：固态硬盘没有机械马达和风扇，工作时噪音值为 0db。

(5) 工作温度范围大：传统的机械硬盘只能在 $5^{\circ}\text{C}\sim 55^{\circ}\text{C}$ 内工作；而大多数固态硬盘可在 $-10^{\circ}\text{C}\sim 70^{\circ}\text{C}$ 内工作，有的工业级固态硬盘工作温度可达 $-40^{\circ}\text{C}\sim 75^{\circ}\text{C}$ 。

(6) 轻便：与常规 1.8in 机械硬盘相比，固态硬盘重量要轻 20~30g。

虽然和传统机械硬盘相比固态硬盘优点众多，但固态硬盘劣势也较为明显：价格仍较为昂贵，容量较低，一旦硬件损坏，数据较难恢复等；固态硬盘的耐用性（寿命）相对较短。因此，用户在选择硬盘时应按照实际需求和用途进行配置。

2) 光盘

光盘 (Compact Disk, CD) 是近代发展起来不同于完全磁性载体的光学存储介质，用聚焦的氢离子激光束处理记录介质的方法存储和再生信息。其容量大、价格低廉，已被广泛用于软件发布、电子书籍、音视频数据的存储，并作为硬盘的备份使用，其容量一般为 650~750MB。

光盘是在激光视频唱片和数字音频唱片基础上发展起来的。利用激光在某种介质上写入信息，然后再利用激光读出信息，这种技术称为光存储技术。如果光存储使用的介质是磁性材料，即利用激光在磁记录介质上存储信息，就称为磁光存储。通常把采用非磁性介质进行光存储的技术，称为第一代光存储技术，它不能把内容抹掉重写新内容。磁光存储技术是在光存储技术基础上发展起来的，称为第二代光存储技术，其主要特点是可擦除重写。根据光存储性能和用途的不同，光盘可分为以下 3 类。

(1) 只读光盘 (CD-ROM)：这种光盘的内容是由厂家事先写入的，使用时用户只能读出，不能修改或增加新的内容。CD-ROM 主要用于软件发布、电视唱片和数字音频唱片、计算机的辅助教学等。

(2) 只写一次光盘 (CD-R)：这种光盘允许用户写入信息，写入后可多次读出，但只能写入一次，且不能修改，故称它为“只写一次型”光盘。CD-R 主要用于计算机系统内的文件存档，或写入的信息不再需要修改的场合。

(3) 可擦写光盘 (CD-RW)：这种光盘类似于磁盘，可以重复读写。从原理上来看，目前仅有磁存储（热磁效应）和相变记录（晶态-非晶态转变）两种 CD-RW。

光盘主要由基板、记录层、反射层、保护层和印刷层组成，如图 3-20 所示。



图 3-20 光盘内部结构

(1) 基板：是光盘其他层的载体，更是整个光盘的物理外壳。它的材料是无色透明的聚碳酸酯，抗冲击韧性极好、使用温度范围大、尺寸稳定性好、耐候性好且无毒性。CD 光盘的基板厚度为 1.2mm、直径为 120mm，中间有孔，呈圆形。光盘之所以能够随意取放，主要取决于基板的硬度。

(2) 记录层：是刻录信息的地方。其主要的工作原理是在基板上涂抹上专用的有机

染料,以供激光记录信息。由于烧录前后的反射率不同,经由激光读取不同长度的信号时,通过反射率的变化形成 0 与 1 信号,借以读取信息。

对于可重复擦写的 CD-RW 而言,与 CD-R 有机染料层不同,CD-RW 盘片的记录层由银、铜、锑、碲合金构成。合金的记录层具有一个约 20% 发射率的多晶结构。CD-RW 驱动器的激光头有两种波长设置,分别为写(P-Write)和擦除(P-Erase),刻录时激光把记录层的物质加热到 $500\sim 700^{\circ}\text{C}$,使其熔化。在液态状态下,该物质的分子自由运动,多晶结构被改变,呈现一种非晶状状态,而在此状态下凝固的记录层物质的反射率只有 5%,这些反射率低的地方就相当于 CD-ROM 盘片上的“凹陷”。要擦除数据就必须让记录层的物质恢复到多晶结构,此时激光头采用低能量的擦除状态,它会把记录层的物质加热到 200°C ,不会使其熔化,但会使其软化。记录层的物质在软化并慢慢冷却时,其分子结构就会从 5% 反射率的非晶状结构转化为 20% 反射率的多晶结构,这样就恢复到 CD-RW 光盘的初始状态。

(3) 反射层:是反射激光光束的区域,借反射的激光光束读取光盘片中的资料。其材料为铝、银、金等金属。

(4) 保护层:是用来保护光盘中的反射层及记录层,防止信号被破坏。材料为光固化丙烯酸类物质。

(5) 印刷层:是用来印刷盘片的客户标识、容量等相关资讯的地方。它不仅可以标明信息,还可以起到一定的保护光盘的作用。

光盘是利用激光束在记录表面上存储信息,根据激光束和反射光的强弱不同实现信息的读写。在光盘中,凹坑是被激光烧录后反射弱的部分,平地是没有受激光烧录而仍然保持高反射率的部分。光盘是用激光束照射盘片后产生反射,然后根据反射的强度来判定数据信息是 0 还是 1。盘片中凹坑部分激光反射弱,平地部分激光反射强,光盘利用凹坑和平地边缘来记录 1,用凹坑和平地的平坦部分记录 0,凹坑和平地的长度代表 0 的个数。这里需要强调的是:凹坑和平地本身并不代表 0 和 1,而是凹坑和平地交界的地方表示 1。由于读数据的激光束的功率只有写数据的激光束的 $1/10$,因此不会在盘片烧出新的凹坑。

随着技术的发展,后来又出现了 DVD(Digital Video Disk),其基本设计与光盘相同,通过加大光盘的记录区域(记录区域从 CD 盘的 86cm^2 提高到 86.6cm^2)和使用波长较短的激光(CD 采用 780nm 红外光,DVD 采用 $635\sim 650\text{nm}$ 激光,光道间距和凹凸坑的长度和宽度做得更小)使得单层单面 DVD 容量可达到 4.7GB,如图 3-21 所示。

近年来,又出现了一种新技术 Blu-Ray,即使用蓝色激光,能更精确地定位,单面容量大约为 25GB,双面约为 50GB,预计 Blu-Ray 将会取代 CD 和 DVD。

光盘的主要性能指标如下。

- 容量:指一张光盘的数据存储量。例如,CD-ROM 存储容量一般为 $650\sim 750\text{MB}$ 左右;DVD 一般有 4 种规格,分别为 4.7GB(单面单层)、8.5GB(单面双层)、9.4GB(双面单层)和 17GB(双层双面)。
- 数据传输率:指光盘驱动器传输数据的速率,一般是 150KB/s 的倍数。例如, 2400KB/s 称为 16 倍数光驱,记为 16X。

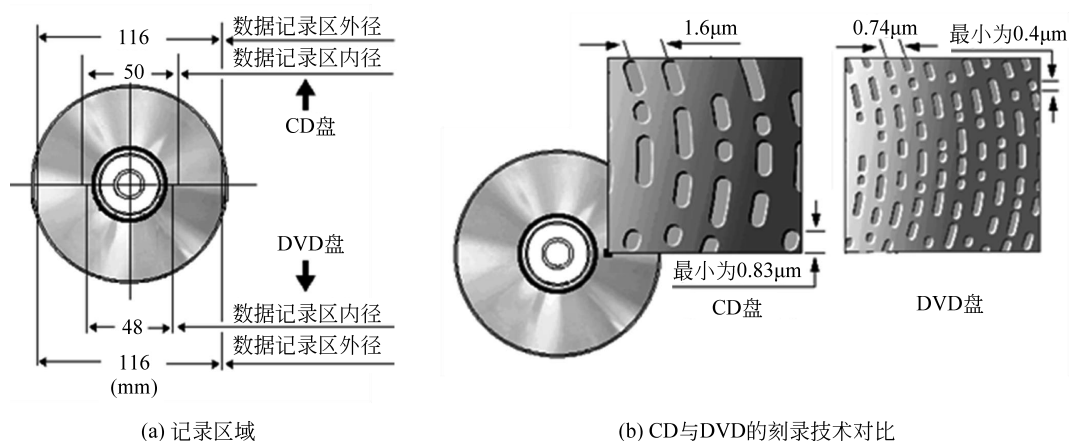


图 3-21 CD 与 DVD 刻录技术

- 读取时间：是指光盘驱动器接收到命令后，移动光头到指定位置，并将第一个数据读入高速缓存所花费的时间。

3.2.4 输入输出系统

输入输出系统(Input/Output System, I/O 系统)是计算机与外界联系的桥梁,实现了计算机与外界之间的通信,一个没有 I/O 系统的计算机系统是“与世隔绝”的废物,不能为人类提供任何服务。I/O 系统由外设和输入输出控制系统两部分组成,控制并实现信息的输入输出,是计算机系统的重要组成部分。

输入输出控制系统包括设备控制器和相应的输入输出软件,它的主要功能是组织和控制数据的输入输出传送,管理与检测外围设备的操作和状态。设备控制器的一端连接在计算机系统的 I/O 总线上,另一端通过接口与设备相连接,如图 3-22 所示。通过设备控制器使外设和主机进行连接并交换数据。例如,打开个人计算机的机箱,会看到主板上插了各式各样的板卡,有声卡、显卡、网卡等,而机箱后部可以看到板卡上带的各种插口。这些板卡就是设备控制器,而板卡上的插口就是接口,可用于连接键盘、鼠标、耳机、显示器等外设。

设备控制器的主要功能包括接收和识别来自 CPU 的控制信息,向 CPU 反馈外设当前工作的状态信息,实现 CPU 与设备控制器、设备控制器与外设之间的数据交换。因此,可以看出,CPU 与外设进行信息传输时,主要传输的是控制信息、状态信息和数据信息。这 3 种信息属于不同性质的信息,必须分别传输。但是,大部分微型计算机的 CPU 和外设进行信息交换时,只有 IN 和 OUT 两种指令,因此控制信息和状态信息都被视为一种数据信息在传输。

1. 输入输出控制方式

由于 CPU 和外设的速度严重不匹配,因此在实现 CPU 和外设数据交换时,要考虑