高等学校教材

微型计算机原理与应用

易先清　莫松海　喻晓峰　等编著

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书共分为 7 章。第一章介绍微型计算机基本功能与技术。第二章介绍微处理器的物理结构、信号引脚功能以及总线操作周期的时序分析。第三章介绍微处理器的应用结构，详细说明现代微机编程用的系统数据结构以及以保护模式为重点的三种工作模式。第四章介绍半导体存储器的内部结构、工作原理以及简单的应用设计。第五章介绍现代微机总线技术，重点分析其系统架构的组成核心——PCI 总线。第六章介绍现代微机的系统组成结构。第七章介绍现代微机外设接口控制技术。

该书可作为高校理工科非计算机专业的高年级本科生或研究生的教材，同时，还特别适用于过去学习过微机知识现在又想进一步系统了解现代微机工作原理及应用技术的读者，也可作为从事现代微机硬件设计与高级编程开发人员的参考用书。

图书在版编目(CIP)数据

微型计算机原理与应用/易先清等编著．—北京：电子工业出版社，2001.1
高等学校教材
ISBN 7-5053-6258-5

Ⅰ．微…　Ⅱ．易…　Ⅲ．微型计算机-高等学校-教材　Ⅳ．TP36

中国版本图书馆 CIP 数据核字(2000)第 81696 号

丛 书 名：高等学校教材
书　　名：微型计算机原理与应用
编　　著：易先清　莫松海　喻晓峰　等
策划编辑：秦　梅
责任编辑：张燕虹
排版制作：电子工业出版社计算机排版室
印 刷 者：
装 订 者：
出版发行：电子工业出版社　URL：http://www.phei.com.cn
　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销：各地新华书店
开　　本：787×1■092　1/16　印张：31.25　字数：800 千字
版　　次：2001 年 1 月第 1 版　2001 年 1 月第 1 次印刷
书　　号：ISBN 7-5053-6258-5 / TP・3370
印　　数：6■000 册　　定价：38.00 元

前　言

本书是根据作者多年从事高校微型计算机(以下简称微机)原理及其应用技术的教学实践与应用开发的实际经验和体会而编写的。该书可作为高校理工科非计算机专业的高年级本科生或研究生学习有关现代微型计算机(以下简称现代微机)原理与应用方面课程的教材。同时,也特别适用于从事现代微机应用与开发人员的参考用书。

自 1981 年第一代通用微型计算机 IBM PC/XT 在 IBM 公司诞生以来,一直以令人目不暇接的态势飞速发展,新技术、新结构、新思想等层出不穷,日新月异。现代微型计算机所表现出的高性能已大大超过过去的大型计算机,其内部结构与工作原理也日趋复杂。仅以构成微机的核心部件 CPU 的集成度为例:8086 内含 2.9 万个晶体管,而目前的第七代 CPU 则内含 2000 多万个晶体管,增长了 1000 倍左右,这对微机系统性能的提高固然很好,但也相对地增加了掌握现代微机应用与开发的难度。值得欣慰的是,随着微机的应用普及,现今,读者在学习微机内部原理之前已经接触过,甚至使用过微机,具有了一定的感性认识,这与过去的读者是不同的。本书正是基于这种现状,专为需要深入学习和系统掌握现代微机工作原理和应用技术的读者而编写。

本书共分为 7 章。第一章介绍微机基本功能与技术,包括顺序执行、中断执行、流水线技术、超序执行技术、推测执行技术、高速缓存技术等。同时,对现代微机的复杂模型进行分析。第二章介绍微机核心部件 CPU 的物理结构,以其信号引脚功能和时序分析为线索详细解释现代微机的工作原理。第三章介绍微机核心部件 CPU 的应用结构,主要从应用角度详细陈述供现代微机编程使用的系统数据结构以及以保护模式应用知识为重点的三种工作模式。第四章介绍现代微机的重要组成部件半导体存储器的内部结构、工作原理以及简单的应用设计。第五章介绍现代微机总线技术,重点分析其系统架构的组成核心——PCI 总线。第六章介绍现代微机的系统组成结构及其核心控制逻辑,并从系统结构角度描述现代微机的工作原理。第七章介绍现代微机外设接口控制技术,如 DMA、中断、IDE、USB 等,并通过系统接口控制逻辑加以说明。

本书各章主要由易先清编写,并负责全书内容的修改和最终定稿;莫松海编写了第三章的部分内容,并负责全书的校对;喻晓峰编写了第七章的部分内容;李劲松、徐伟和谭树人也参加了本书的编写工作。

在本书编写过程中,得到了姚庭宝教授的悉心指导,同时,还得到了严国进、戴长华、莫立宇等同志的帮助,谨在此一并表示深切的谢意。

由于编者的水平有限,加之时间比较仓促,书中一定存在错误或不妥之处,请读者不吝指正。

国防科技大学　易先清

目　　录

第一章　微型计算机基础

本章从介绍微型计算机的初级模型结构入手，介绍微型计算机的一些基础知识：微型计算机的产生与发展、基本组成结构、采用的一些计算机技术以及复杂的现代微型计算机模型，最后，以一个现代微型计算机的简单分析实例总结全章。

1.1　概述

计算机是一类复杂的智能处理系统，由多种不同功能的部件组成，在计算机中发生的处理过程也非常复杂。本节先从计算机的一些基础知识入手进行引入性介绍：包括计算机中处理和表示的数值、数值的基本类型、计算机组成以及计算机指令系统。

(1)　　1.1.1　计算机中的数值与编码系统

下面对计算机中的数值与编码系统分别进行介绍。

1.　　1.1.1.1　计算机中的数值

计算机最早作为一种计算工具出现，最基本的功能是对数进行加工和处理。被加工和处理的数在计算机中以物理器件的不同状态来表示，一个具有两种不同的稳定状态且这两种状态能够相互转换的器件就可以用来表示一位二进制数。以二进制表示数不仅形式最简单，而且，物理上最可靠，同时，二进制数的运算规则也最简单。因此，在计算机中，几乎都用二进制来表示数值。

1．二进制数

一个二进制数具有以下两个基本特点:具有两个不同的数字符号，即 0 和 1；逢二进一。

由于二进制数逢二进位，所以，同一个数字符号在不同的数位所表示的值是不同的。例如，对于二进制数 111.11，小数点左边第一位的“1”代表的值就是它本身；小数点左边第二位的“1”是由第一位逢二进上来的，所以，它的值为 1×2^{1}，则左边的第三位的“1”的值为 1×2^{2}；小数点右边的第一位的“1”的值为 1×2^{-1}；右边的第二位的“1”的值为 1×2^{-2}……这与十进制数类似。

可见，每一个数位有一个基值与之相对应，这个基值就称为权。对于一个二进制数的权，小数点左边的是 2 的正次幂；小数点右边的是 2 的负次幂。

一个二进制数的值可以用它的按权展开式来表示，即：

$$(101.11)_2=1\times2^{2}+1\times2^{0}+1\times2^{-1}+1\times2^{-2}=(5.75)_{10}$$

$$(1010.101)_2=1\times2^{3}+1\times2^{1}+1\times2^{-1}+1\times2^{-3}=(10.625)_{10}$$

于是，一个任意的二进制数可以表示为：

$$\begin{aligned}(B)_2=&B_{n-1}\times2^{n-1}+B_{n-2}\times2^{n-2}+\cdots+B_1\times2^{1}+B_0\times2^{0}\\&+B_{-1}\times2^{-1}+B_{-2}\times2^{-2}+\cdots+B_{-m}\times2^{-m}\\=&\sum B_i\times2^{i}\end{aligned}$$

其中，n 为整数部分的位数，m 为小数部分的位数；B_i 的值为 0 或 1 取决于一个具体的数。

2．十六进制数

目前，因为大部分微型机的字长是 4 的整数倍，所以，广泛地采用十六进制数来表示。一个十六进制数的特点为：具有 16 个数字符号，采用 0~9 和 A~F，这 16 个数字符号与十进制数和二进制数之间的关系如表 1.1 所示；逢 16 进位。

由于是逢 16 进位，所以，同一个数字符号在不同的数位所代表的值是不同的，即每一个数位有一个权与之相对应。小数点左边的权是 16 的正次幂，小数点右边的权是 16 的负次幂。一个 16 进制数的值可以用它的按权展开式来表示：

$(23)_{16}=2\times16^{1}+3\times16^{0}=(35)_{10}$

$(FF)_{16}=15\times16^{1}+15\times16^{0}=(255)_{10}$

$(2AD.0F)_{16}=2\times16^{2}+10\times16^{1}+13\times16^{0}+0\times16^{-1}+15\times16^{-2}=(685.05859375)_{10}$

于是，一个任意的十六进制数 D 可以表示为：

$$(D)_{16}=D_{n-1}\times16^{n-1}+D_{n-2}\times16^{n-2}+\cdots+D_{1}\times16^{1}+D_{0}\times16^{0}+D_{-1}\times16^{-1}+D_{-2}\times16^{-2}+\cdots+D_{-m}\times16^{-m}=\sum D_{i}\times16^{i}$$

其中，n 为整数部分的位数，m 为小数部分的位数；D_i 的值在范围 0~9 和 A~F 中。

但在机器中，数并不是用十六进制表示的，由于一开始提到的理由，在机器中数仍是用二进制表示的。二进制和十六进制之间存在着一种特殊的关系，即 $2^4=16$。

于是，一位十六进制数可以用 4 位二进制数表示，它们之间存在着直接而又是惟一的对应关系，如表 1.1 所示。

表 1.1　十进制、十六进制、二进制对应关系

十进制	十六进制	二进制	十进制	十六进制	二进制
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

因此，二进制数和十六进制数之间的转换是十分简捷且又方便的。

(1) 十六进制转换为二进制

不论是十六进制的整数还是小数，只要把每一位十六进制的数用相应的 4 位二进制数代替，就可以转换为二进制数。

例如，$(3DA.12FB)_{16}$ 可转换为：

```
3       D       A   .   1       2       F       B
|       |       |       |       |       |       |
0011    1101    1010    0001    0010    1111    1011
```

所以，$(3DA.12FB)_{16}=(0011\ 1101\ 1010.0001\ 0010\ 1111\ 1011)_2$

(2) 二进制转换为十六进制

二进制的整数部分由小数点向左，每 4 位一分，最前面不足四位的则在前面补 0；小数

部分由小数点向右，每 4 位一分，最后不足 4 位的则在后面补 0。然后，把每 4 位二进制数用相应的十六进制数代替，即可转换为十六进制数。

例如，$(1\ 1100\ 1110.1001\ 0000\ 1)_2$ 可转换为：

0001	1100	1110.	1001	0000	1000
1	C	E	9	0	8

所以，$(1\ 1100\ 1110.1001\ 0000\ 1)_2=(1CE.908)_{16}$

总之，数在机器中是用二进制表示的，但是，一个二进制数书写起来太长且容易出错。而且，目前大部分微机的字长是 4 位、8 位、16 位、32 位或 64 位的，都是 4 的整数倍，我们在书写时用十六进制来表示。一个字节（8 位）就可以用 2 位十六进制数来表示，2 个字节（16 位）可以用 4 位十六进制数来表示。书写起来方便且不容易出错。

2. 1.1.1.2 计算机中信息的编码表示

如上所述，在计算机中，数是用二进制表示的。下面还会介绍，进入到 CPU 中的数据和在 CPU 中处理的数据以及经 CPU 处理后输出的数据都是以二进制的形式出现的，这些二进制数就是我们要处理的信息。而在现实世界中，我们要处理的信息有数据、文字符号和图形等，而数据又有各种进制的数据，文字符号有不同语种的文字、各种各样符号，图形的变化形式更是复杂多样。要对现实世界中这些不同表现形式的信息通过计算机进行处理，就必须将这些表现不同形式的信息（即不同进制的数据、不同语种的文字符号和各种各样的图形等）与计算机中二进制数发生联系，即计算机中的二进制数何时、如何解释成为不同进制的数据，何时、如何解释成为不同语种的文字符号，何时、如何解释成为各种各样的图形等。这种解释必须遵从某种公共的约定，这样，在不同的地点、不同的时间才能得到一致的解释，这种公共的约定就是计算机的二进制编码。下面介绍 3 种约定：

① 用于二进制数向十进制数解释的二进制编码的十进制数约定。

② 用于二进制数向字母、字符解释的字母与字符的编码约定。

③ 用于二进制数向汉字解释的汉字编码约定。

当然，还可以扩展这些约定，如二进制数向其他语种解释的编码约定以及其他领域的编码约定等，这些都是计算机针对外界的内部二进制数的解释约定。另一种类型的解释约定就是计算机针对内部指令执行的内部二进制数的解释约定。在 CPU 执行指令时，在 CPU 内部流动的二进制数可能解释为指令的操作码，也可能解释为指令的操作数，这时遵从的约定便是计算机的指令系统。

下面首先介绍计算机针对外界的内部二进制数的解释约定。

由于计算机中的基本物理器件是具有两个状态的器件，所以，不同的信息只能用若干位的二进制码的组合来表示，这种组合的二进制码称为二进制编码。

1．二进制编码的十进制数

因为二进制数容易实现、可靠，二进制的运算规律十分简单。所以，在计算机中采用二进制。但是，二进制数不直观，于是，在计算机的输入和输出时通常还是用十进制数表示。不过，这样的十进制数要用二进制编码来表示。

一位十进制数用 4 位二进制编码的表示方法很多，较常用的是 8421 BCD 码，表 1.2 列出了一部分编码关系。

表 1.2　BCD 编码表

十进制	8421 BCD 码	十进制	8421 BCD 码
0	0000	8	1000
1	0001	9	1001
2	0010	10	0001　0000
3	0011	11	0001　0001
4	0100	12	0001　0010
5	0101	13	0001　0011
6	0110	14	0001　0100
7	0111	15	0001　0101

8421 BCD 码有 10 个不同的数字符号，且它是逢“十”进位的，因此，它是十进制数。由于它的每一位是用 4 位二进制编码表示的，因此，称为二进制编码的十进制数 BCD（Binary Coded Decimal）。

BCD 码是比较直观的。例如，可以很方便地将（0100 1001 0111 1000.0001 0100 1001）$_{BCD}$ 认出为：4978.149。只要熟悉 BCD 的十个编码，就能很容易地实现十进制与 BCD 码之间的转换。

但是，BCD 码与二进制之间的转换是不直接的，要经过十进制。即 BCD 码先转换为十进制码后再转换为二进制，反之亦然。

2．字母与字符的编码

如上所述，字母和各种字符也必须按特定的规则用二进制编码才能在机中表示。编码也可以有各种方式（即规定）。目前，在微机中最普遍地采用 ASCII（美国标准信息交换码）码。

ASCII 码为 7 位二进制编码，故可表示 128 个字符，其中包括数码（0~9）、英文字母等可打印的字符。数码 0~9 分别用 0110000~0111001 来表示。因微型机通常字长为 8 位，通常最高位的位 7 用作奇偶校验位，但在机器中表示时，常认它为零，故用一个字长（即一个字节）来表示一个 ASCII 字符。于是，0~9 的 ASCII 码为 00110000~00111001，十六进制表示为 30H~39H，大写字母 A~Z 的 ASCII 码的十六进制表示为 41H~5AH。

3．汉字的编码

当计算机在我国运用时，特别是把计算机用于管理等事务处理领域时，就要求计算机能够输入、处理和输出汉字。显然，汉字在计算机中也只能用若干位的二进制编码来表示。但是，用 8 位编码来表示汉字是远远不够的，那么，要用多少位来表示汉字呢？这首先取决于一个计算机中能输入、存储、处理的汉字的个数。

1981 年，国家根据汉字的常用程序定出了一级和二级汉字字符集，并规定了编码，这就是中华人民共和国国家标准信息交换用汉字编码，即 GB2312－80 国标码。该标准编码字符集共收录汉字和图形符号 7445 个，由三部分组成。

第一部分是字母、数字和各种符号，包括下列拉丁字母、俄文、日文平假名与片假名、希腊字母、汉语拼音等 682 个。

① 一般符号 202 个。包括间隔符、标点、运算符、单位符号和制表符等。

② 序号 60 个。它们是 1.~20.（20 个），（1）~（20）（20 个），①~⑩（10 个），（一）~（十）（10 个）。

③ 数字 22 个。它们是 0~9 和 I~XII。

④ 英文字母 52 个。大、小写各 26 个。
⑤ 日文平假名 83 个。
⑥ 日文片假名 86 个。
⑦ 希腊字母 48 个。其中大、小写各 24 个。
⑧ 俄文字母 66 个。其中大、小写各 33 个。
⑨ 汉语拼音符号 26 个。
⑩ 汉语注音字母 37 个。

第二部分为一级常用汉字，共 3755 个，按汉语拼音排列。第三部分为二级常用字，共 3008 个，因不太常用，所以，按偏旁部首排列。

GB2312 国标字符集呈二维表，分成 94 行×94 列，行号称为区号，列号称为位号。由于每一个汉字或符号在码表中都有各自的位置，因此，各有一个惟一的位置编码，该编码就是字符所在的区号（行号）及位号（列号）的二进制代码（7 位区号在左、7 位位号在右，共 14 位），这也就是汉字的区位码。因此，字符集中的任何一个图形、符号及汉字都用惟一的区位码表示。在数据通信应用中，一些表示低数值的二进制码已规定用作通信流程的控制，如 00000110（06H）用于通信的肯定应答（用 ACK 表示），为了不影响已经约定好的数据通信规程，将区位码的区号和位号都加 32（即 100000），变换成对应的国标码，如处于 19 区、3 位的汉字“常”的区位码为 0010011 0000011（13H 03H），其国标码则为 0110011 0100011（33H 23H）。

为了使汉字的编码与常用的 ASCII 码相区别，在机器中，汉字又是以内码形式存储和传输的，一种机器常有若干种汉字输入方式（输入码），但其内码是统一的。而汉字的内码是将汉字国标码的区号和位号都扩展成 8 位（即由两个字节组成），将再它们的最高位 b_7 都置 1 构成的，如上述汉字“常”的国标码为 0110011 0100011（33H 23H），扩展为两个字节后为 00110011 00100011（33H 23H），再置两字节的 b_7 置 1 后即得到汉字‘常’的机器内码 10110011 10100011（B3H A3H）。

(2) 1.1.2 基本数据类型

在微型计算机中，对微处理器处理的数据对象进行如下分类：

1．位 b（bit）

位是计算机所能表示的最基本、最小的数据单元。因为在计算机中广泛采用二进制数，所以，位就是一个二进制位，它只能有两种状态：“0”和“1”，通常用小写的“b”表示二进制位，如 10101011b。由若干个二进制位的组合就可以表示各种数据、字符等。

2．字节 B（Byte）、字（Word）、双字（DoubleWord）、四字（QuadWord）

字节、字、双字和四字的表示如下：

- 把相邻的 8 位二进制数位称为一个字节（1 字节＝8 个二进制位），1 个字节为 8 个二进制位，通常用大写的“B”表示字节，如 4B 即为 4 个字节，共 32 个二进制位。
- 一个字包含相邻的 2 个字节或 16 个二进制位（1 字＝2 字节＝16 个二进制位）。通常用大写的“W”表示字，如 4W 即为 8 个字节，共 64 个二进制位。
- 一个双字包含相邻的 2 个字或 4 个字节或 32 个二进制位（1 双字＝2 字＝4 字节＝32 个二进制位）。通常用大写的“DW”表示双字。

- **一个四字包含相邻的 2 个双字或 4 个字或 8 个字节或 64 个二进制位（1 四字＝2 双字＝4 字＝8 字节＝64 个二进制位）。通常用大写的“ QW ”表示四字。**

图 1.1 示出 X86 体系结构处理器中所采用的基本数据结构。

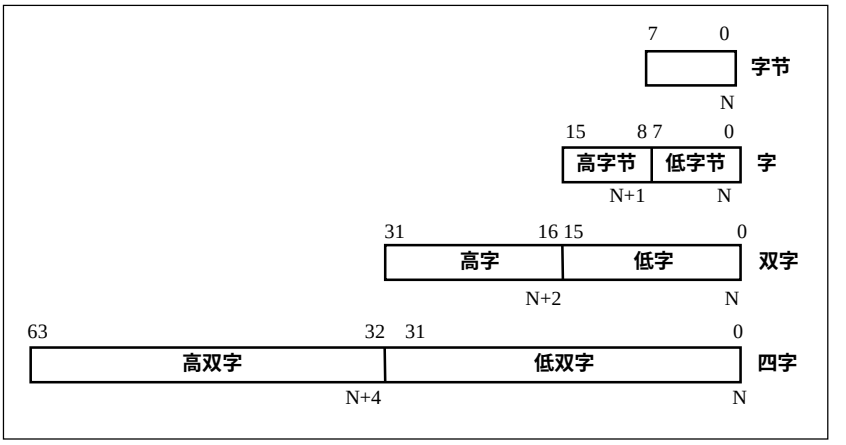

图 1.1 X86 体系结构处理器的基本数据结构

图 1.2 示出了 X86 体系微型计算机中每种基本数据类型在存储器中的存放顺序，每种数据类型的低字节存放在较低地址处。

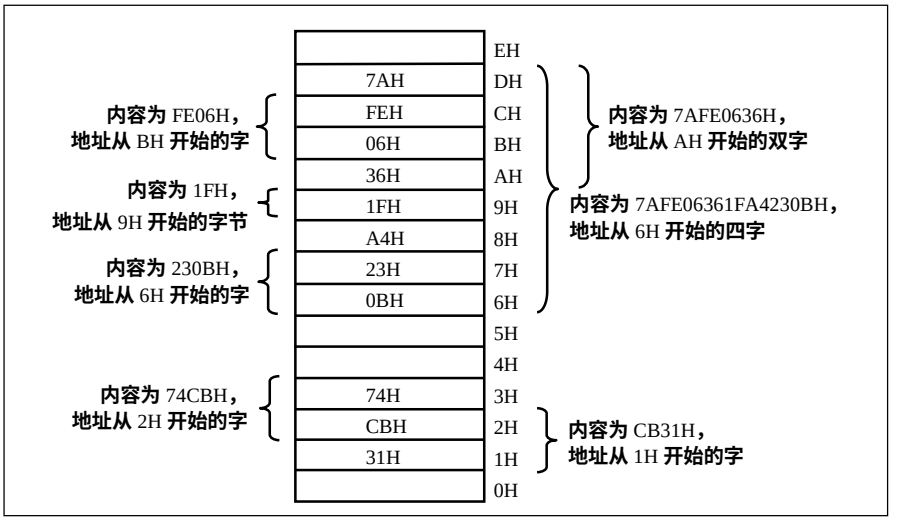

图 1.2 存储器中字节、字、双字和四字的存放示意图

(3) 1.1.3 计算机的基本结构

电子数字计算机一开始是作为一个计算工具而出现的。若要计算机能够脱离人的直接干预、自动地完成计算，它应该具备哪些主要部件呢？

如图 1.3 所示，一台计算机的基本结构是以运算器为中心，由运算器、存储器、控制器、输入设备、输出设备等组成的。通过运算器进行各种运算操作，通过存储器保存指令和数据，控制器则根据不同的指令向计算机各部件发相应的操作命令，输入部件用于向计算机输入待处理的各种数据信息，输出部件用于从计算机输出经过处理的各种数据信息，这样，就构成了一个基本的计算机系统。

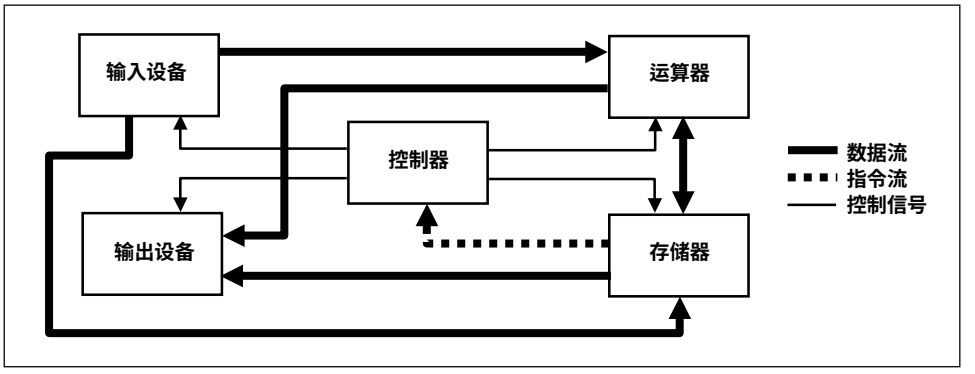

图 1.3　计算机的基本结构图

在计算机中，主要有两种类型的信息流在流动，一种为数据，即各种原始数据、中间结果、程序等。在输入过程中，它们由输入设备输入至运算器，再存于存储器，或直接从输入设备存于存储器；在运算处理过程中，它们从存储器读入运算器，由运算器处理得到结果，这种结果当作中间结果时再次存入存储器，或当作最终结果时经输出设备输出。人给计算机的各种命令（以编写的程序表现）开始时也以数据形式由存储器送入控制器，由控制器经过译码后变为各种控制信号。这些控制信号也就是另一种类型的信息流，由控制器控制计算机中各部件发生的动作，包括控制输入设备的启动与停止，控制运算器按规定一步步地进行各种运算与处理，控制存储器的读和写操作，控制输出设备输出结果等等。

存储器通常分成内部存储器和外部辅助存储器两部分，它们分别简称为内存和外存，内存容量相对较小（通常为 32~128MB），但存取速度较快，它们主要为半导体存储器。这些半导体存储器根据其性质与功能的不同又分成只读存储器和读写存储器两大类，如果再细分，还可分成速度极高的寄存器组、速度很高的高速缓存 Cache 和高速的系统主存，它们在容量上则依次递增。另一部分是大容量外存，但存取速度较慢，主要有磁盘（软、硬盘）、磁带、磁光盘 MO 以及各种诸如 CD-ROM、DVD 等激光存储设备。

输入设备常用的有鼠标、键盘、触摸屏、音频输入设备、视频输入设备、网络等。输出设备常用的有 CRT 显示器、各种类型的打印机、绘图仪、网络、控制信号输出设备等。

图 1.1 中的各部分构成了计算机的硬件系统，运算器、存储器和控制器组合在一起称为计算机的主机，其中，运算器与控制器一起组成中央处理单元（简称为 CPU），各种输入输出设备统称为计算机的外围设备（简称为外设）。

(4)　　1.1.4　指令程序与指令系统

上一节提到了组成计算机的几个主要部件，这些部件构成了计算机的硬件基础。但是，仅有这样的硬件，还只是具有了计算的可能，要进行什么样的计算或去计算什么，还必须具有相应软件的配合，首先是进行什么样的计算，这由通常称之为程序（Program）的指令集合给出。

计算机之所以能够脱离人的直接干预，自动地进行各种计算，是因为人把实现计算的一步步操作用命令的形式——即一条条指令（Instruction）预先输入到存储器中，在执行时，机器把这些指令一条条地取出来，加以翻译和执行。

首先，以两数相加这一最简单的运算为例进行说明。执行时，需要以下 4 步（假定要运

算的数已在存储器中）：

① 把第一个数从它所在的存储单元（Location）中取出来，送至运算器。

② 把第二个数从它所在的存储单元中取出来，送至运算器。

③ 相加。

④ 将加完的结果送至存储器中指定的单元。

所有这些取数、送数、相加、存数等等都是一种操作，把要求计算机执行的各种操作用命令的形式写下来，这就是指令。通常一条指令对应着一种基本操作，但是，计算机怎么能辨别和执行这些操作呢？一个计算机能执行什么样的操作，能做多少种操作，是由设计计算机时所设计的指令系统决定的。一条指令对应着一种基本操作，计算机所能执行的全部指令就是计算机的指令集（Instruction Set），这是计算机所固有的，编程人员只能遵循，无法更改。

在使用计算机时，必须把我们要解决的问题以一条指令接一条指令的形式给出。但是，这些指令必须是所用计算机能识别和执行的，也即每一条指令必须是对其运行的计算机的指令系统中所具有的指令，而不能随意捏造，所用到的这些指令的集合就称为程序，用户为解决自己的问题所编写的程序称为源程序（Source Program）。

指令通常包含操作码（Opcode，即 operationcode）和操作数（Operand）两大部分。操作码表示计算机执行什么操作；操作数指明参加操作的数或操作数所在的地址。

因为计算机中的数据信息是以二进制数表示的，所以，计算机指令系统中的所有指令都必须以二进制编码的形式来表示。例如，在 X86 系列微处理器 8088 中，从存储区取数（以 SI 变址寻址）送至累加器 AL 指令的操作编码为 8A04（2 字节指令），一种加法指令的操作编码为 02C3，向存储器存数（一种串操作指令）的操作编码为 AA（1 字节指令）等等。这就是指令的机器码（Machine Code）。在 8 位的微型机中，因为字长较短，用一个字节不能充分表示各种操作码和操作数。所以，有 1 字节指令，有 2 字节指令，也有多字节指令（如 4 字节指令），这种指令操作码长度不等。指令一般称为复杂指令集 CIS，采用 CIS 的计算机称为复杂指令集计算机 CISC。

计算机发展的初期，就是用指令的机器码直接来编制用户的源程序，这就是机器语言阶段。但是，因为机器码是由一连串的 0 和 1 组成的，没有明显的特征，不便记忆，不易理解，容易出错，所以，编程成为一种十分困难和十分繁琐的工作。于是，人们就用一些助记符（Mnemonic）——通常是指令功能的英文词缩写来代替操作码。如在 8088 中，传数指令用助记符 MOV（MOVE 的缩写），加法用 ADD，转移用 JMP 表示等等。这样，每条指令就有明显的特征，易于理解和记忆，也不易出错，这就前进一大步，此时称为汇编语言阶段，用户用汇编语言（操作码用助记符代替，操作数也用一些符号——Symbol 来表示）来编写源程序。

要求机器能自动执行这些程序，就必须把这些程序预先存放到存储器的某个区域。由于程序通常是顺序执行的，所以，程序中的指令也是一条条顺序存放的。计算机在执行时要能把这些指令一条条取出来加以执行，此时必须有一个电路能追踪指令所在的地址，这就是程序计数器 PC（Program Counter）。在开始执行时，给 PC 赋以程序中第一条指令所在的地址，然后，每取出一条指令（确切地说是每取出一个指令字节），PC 中的内容便自动加 1，指向下一条指令的地址（Address），以保证指令的顺序指向。只有当程序中遇到转移指令、调用子程序指令或遇到中断时，PC 才转到所需要的地方去。