中等专业学校
电子信息类 规划教材

单片机及其应用

眭碧霞 主编

西安电子科技大学出版社

内 容 简 介

“单片机及其应用”课程是目前中专学校计算机应用专业及其相关专业的一门重要课程。本书是在多年的“单片机及其应用”课程教学、实验的基础上，结合目前单片机的广泛应用和新技术发展而编写的。

全书共分 12 章，全面阐述了 MCS－51 系列单片机的基本原理及其应用技术。本书主要介绍了 MCS－51 系列单片机的基本结构、组成、中断系统、存储器以及汇编语言程序设计。通过这些内容的学习，可使学生对 MCS－51 系列单片机有一个总体的概念和认识，并在掌握基本硬件的基础上用软件实现单片机的一些基本功能。在单片机的应用中详细介绍了并行口、串行口、定时器以及相应的扩展和应用，简单介绍了模拟量和数字量的相互转换以及转换器件与单片机的接口，同时综合前面所学内容，列举了单片机典型应用实例。最后，对 16 位单片机作了简单的介绍，使学生对 MCS－96 系列单片机原理及应用有所了解。

本书可作为中专计算机专业和相关专业的教材，也可供从事单片机应用的工程技术人员参考。

图书在版编目(CIP)数据

单片机及其应用/眭碧霞主编. －西安：西安电子科技大学出版社，2000.7
中等专业学校电子信息类规划教材
ISBN 7－5606－0877－9

Ⅰ. 单… Ⅱ. 眭… Ⅲ. 单片微型计算机-专业学校-教材 Ⅳ. TP368.1

中国版本图书馆 CIP 数据核字(2000)第 30058 号

责任编辑 马乐惠 杨宗周
出版发行 西安电子科技大学出版社(西安市太白南路 2 号)
电 话 (029)8227828 邮 编 710071
http：//www.xduph.com E-mail：xdupfxb@pub.xaonline.com
经 销 新华书店
印 刷 西安长青印刷厂
版 次 2000 年 7 月第 1 版 2001 年 3 月第 1 次印刷
开 本 787 毫米×1092 毫米 1/16 印张 17.5
字 数 409 千字
印 数 4 001～10 000 册
定 价 18.00 元
ISBN 7－5606－0877－9/TP・0461

前　言

“单片机及其应用”课程是目前中专学校计算机应用专业及计算机相关专业的一门重要课程。本教材是我们在多年的“单片机及其应用”课程教学基础上，结合目前单片机的新技术发展而编写的。

本教材参考教学时数为 90 学时，其中实验 20 学时。全书共 12 章。第 1、2 章主要介绍 MCS－51 系列单片机的基本结构、组成、中断系统和存储器及其扩展，使学生对 MCS－51 系列单片机有一个总体的概念和认识。第 3、4 章介绍指令系统和汇编语言程序设计，使学生在掌握基本硬件的基础上用软件实现单片机的一些基本功能。第 5、6、7、8 章介绍单片机的存储器扩展、中断组成与应用、定时器、串行口以及相应的扩展和应用。第 9 章介绍并行口的扩展和主要应用。第 10 章介绍模拟量和数字量的相互转换以及转换器件与单片机的接口。第 11 章综合前面所学内容，利用单片机来实现具体的应用。最后，对 MCS－96 系列单片机原理及应用作了简单介绍(这部分内容不作为本课程学时内的要求)。

本书特点之一：针对目前国内流行的单片机机型，着眼于应用，力求在较短的时间内，使学生掌握单片机的应用技术。

本书特点之二：在章节安排上，注意原理和应用并重。

本书特点之三：列举大量实用例子，使学生能借助基本内容，举一反三，灵活应用。

本书特点之四：考虑到非计算机专业学生的特点，循序渐进，通俗易懂。

本教材由常州无线电工业学校眭碧霞编写，福建电子工业学校陈国先高级讲师审稿。在编写过程中得到全国电子信息中专教材编审委员会和西安电子科技大学出版社的大力支持，在此表示感谢！

由于编者水平等问题，书中疏漏不妥之处，敬请读者指正。

编　者

2000 年 6 月

第 1 章　微型计算机基本知识

1.1　微型计算机的发展与应用

计算机是人类制造的用于信息处理的机器。这种机器在人的控制下(而且只能在人的控制下)，将输入的数据信息按照人们的要求进行存储、分类、整理、判断、计算、决策和处理等操作。

计算机俗称电脑，它是模仿人的大脑对信息进行加工的机器，但它与人脑截然不同。第一，它不是生物体，而是人制造的电子机器；第二，它不能离开人的操纵和控制；第三，外界信息不会自动进入电脑，必须在人的控制下，通过输入设备输入；第四，它处理信息的速度和精度都是人脑所无法比拟的。

1.1.1　微型计算机的发展史

1946 年，世界上第一台命名为“ENIAC”的电子计算机在美国诞生。50 多年来，作为高技术成果的电子计算机技术异常迅速地发展。通常人们习惯按电子元器件的工艺变化将电子计算机的发展划分为电子管时代、晶体管时代、集成电路时代和大规模、超大规模集成电路时代四个阶段。

1971 年～1973 年为第一代，即电子管时代。其典型产品是 Intel 4004 和 Intel 8008，字长分别为 4 位和 8 位，集成度约在 2000 个器件每片，时钟频率为 1 MHz。

1973 年～1975 年为第二代，即晶体管时代。其典型产品是 Intel 8080 和 M6800，字长为 8 位，集成度约在 5000 个器件每片，时钟频率为 2 MHz。

1975 年～1977 年为第三代，即集成电路时代。其典型产品是 Intel 8085、M6802 和 Z80，字长为 8 位，集成度约在 10 000 个器件每片，时钟频率为 2.5～5 MHz。

1978 年～1980 年为第四代，即大规模、超大规模集成电路时代。其典型产品是 Intel 8086、M68000 和 Z8000，字长为 16 位，集成度约在 30 000 个器件每片，时钟频率为 5 MHz。

1981 年以后产生了第五代微处理器，典型产品是 IAPX43201，字长为 32 位，集成度约在每片 11 万个器件以上，时钟频率可达 10 MHz。1985 年推出的 M6820 和 Intel 80386，集成度已达每片 27 万个器件，时钟频率为 16～25 MHz。由于集成度高，因此系统的速度和性能大为提高，可靠性增加，成本降低。

1989 年，Intel 公司将数字协处理器和高速缓存加到 386 芯片内，推出了 80486 微处理器。80486 与 80386 完全兼容，但速度要比 80386 快。80486 借用了 80386 指令流水线 RISC(精简指令系统的计算机)的设计思想，减少了大部分指令的时钟周期。80486 采用高集成度的超大规模集成电路，芯片内含有 120 万个晶体管，时钟频率可以达到 100 MHz。

1993 年 3 月，Intel 公司推出 64 位的 Pentium 微处理芯片，它采用比 486 更高集成度的超大规模集成电路。Pentium 与 386 或 486 完全兼容，采用 32 位地址线和 64 位数据线，寻址范围 4 GB(2^{32} B)，时钟频率可达 200 MHz，甚至更高。

1.1.2 微型计算机的特点

从工作原理和基本结构上看，微型计算机与大型、中型和小型计算机没有本质的区别。微型计算机具有计算机的基本特点，即运算速度快，计算精度高，具有“记忆”能力、逻辑判断能力，可自动连续工作等。此外，微型计算机还具有以下几个特点：

1. 体积小、质量轻、价格低和耗电量小

早期的计算机占地上百平方米，重量以吨计，价格昂贵，耗电量几百千瓦每小时。现在的微型计算机重量几公斤，耗电量 100 多瓦每小时，价格较低。

2. 可靠性高

广泛采用大规模和超大规模集成电路，微型计算机内部的器件数量减少，连线少，使微型计算机的工作可靠性提高，抗干扰能力增强。

3. 结构灵活

微型计算机采用总线结构，结构灵活，可以根据需要配置不同的计算机部件，极易组成各种系统来满足不同的需要。微型计算机可以单机使用，也可以非常方便地构成多机系统或计算机网络。

1.1.3 微型计算机的应用

微型计算机的应用已经渗透到工业、农业、商业、交通运输、教学科研以及人们的日常生活等各个领域中，成为现代社会生活的重要支柱，发挥着极其重要的作用，并取得了极为可观的效益。

目前，微型计算机主要应用在以下几个方面：数值计算、数据处理与信息加工、计算机辅助功能、人工智能、计算机通信、多媒体计算机、检测和过程控制等几类。微型计算机在检测和过程控制方面的应用具有简便、快捷、准确、可靠等优点，便于实现各种生产过程自动化。在改造传统产业、节约能源、提高产品质量和生产效率、改善生产者劳动条件等方面，具有十分重要的作用。微型计算机在检测和过程控制方面的应用主要有以下几个方面。

1. 微型计算机检测和数据处理系统

应用微型计算机能对生产过程的各种参数，例如温度、压力、流量、液位、角度、长度、电流、电压、频率、功率等参数进行检测，将检测结果存储、显示、打印、绘制成表格或曲线；对检测结果进行计算、比较和整理，当测量值超过一定范围时，发出声、光等报警信号。

由于微型计算机检测和处理数据的速度很快，因此一个微型计算机检测系统可以同时检测多个同类参数或不同参数，如温度、压力、流量、液位等，或者对某个系统中几百个参数进行巡回检测。

智能化仪表是带有微处理器和通信接口的新型仪表，它也是一种微型计算机检测和数

据处理系统。这种仪表利用计算机对传统仪表进行了革新，实现了数字化、微型化和智能化，例如，具有自动校正、消除误差、自动选择量程等“智能”性功能。

2. 微型计算机顺序控制系统

这类系统在微型计算机的控制下，使生产设备按照事先规定的某种顺序有规律地进行生产。例如，采用微型计算机控制的注塑机系统就是一个典型的顺序控制系统，其加工过程是一个按照预定顺序的周期性动作，一个加工周期包括以下几个步骤：合模→注射→保压→冷却→开模→取制品→合模。

顺序控制系统中，微型计算机并不随时检测生产设备每个步骤的状态来对系统进行校正，这是一种开环控制方式。

3. 微型计算机闭环控制系统

闭环控制系统在微型计算机检测到生产过程的状态参数并进行处理后发出控制信号，送有关执行机构去控制该生产过程，使之符合预定的要求。例如，用微机控制热处理炉的温度，其工作过程是通过温度传感器将炉温变成电信号，经接口电路输入微型计算机。微型计算机将此信号值与预定温度信号值进行比较，运算后得到调节量，由接口电路输出，控制加热元件增加或减少加热功率，从而保证炉温的恒定。类似的系统如电机转速控制系统、自动仓库系统等。

1.2 微型计算机系统的组成

一台完整的计算机应包括硬件部分和软件部分。只有硬件和软件的结合，才能使计算机正常运行，发挥作用。因此，对计算机的理解不能仅局限于硬件部分，应该将其看成一个系统，即计算机系统。计算机系统中，硬件和软件都有各自的组成体系，分别称为硬件系统和软件系统。

1.2.1 硬件

计算机系统是由硬件和软件组成的一个整体系统。系统中的电子、机械、光电元件等组成的计算机元部件和计算机设备，都是看得见摸得着的实体。这些元部件依照计算机系统结构的要求构成有机的整体，即为计算机系统的硬件。

1.2.2 软件

在一台计算机中，全部程序、数据和文档等信息的集合统称为这台计算机的软件系统。软件按其功能分为应用软件和系统软件两大类。

1. 系统软件

系统软件由计算机厂家作为系统资源提供给用户，主要用于实现计算机的管理、调度、监视和服务等功能。其目的是方便用户，提高计算机使用效率，扩充系统的功能。系统软件是使用和管理计算机的软件，是为其它软件服务的软件。它最靠近计算机硬件，其它软件都要通过它发挥作用。从用途上来看，系统软件可以分为三类：

(1) 面向计算机管理和操作的软件

此类软件主要用来实现对计算机系统的管理和调度，其包括操作系统、网络通讯系统、监控程序、故障处理程序等。

(2) 面向用户的软件

此类软件主要用来方便用户用计算机解决自己的问题，其包括语言处理程序，如汇编程序、编译程序和解释程序；辅助加工软件，如编辑程序、链接程序、各种通用和专用的计算程序、数据库和软件包；数据库管理程序等。

(3) 面向计算机维护人员的软件

此类软件主要用来对计算机系统进行调试和测试，如各种诊断调试程序、测试程序等。

2. 应用软件

所谓应用软件，是指用户在各自的领域中，为解决各类特定问题而编写的应用程序，如科学计算程序、自动控制程序、工程设计程序、数据处理程序、情报检索程序等。随着计算机的广泛应用，应用软件的种类和数量将越来越多，应用软件的规模也越来越庞大。

应用软件一般由用户编写。由于应用领域的不同，应用软件的差别很大。目前，应用软件已向标准化、模块化、商品化的方向发展，许多计算机厂家和软件公司研制出了具有通用性的软件，并把这些软件收入到系统库和软件库中，作为系统资源提供给用户。

系统软件和应用软件之间没有非常严格的界限。一些具有通用价值的应用程序，常常纳入厂家的系统程序中，作为系统的资源提供给用户。

1.2.3 程序设计语言

1. 机器语言

(1) 指令系统

计算机之所以能脱离人的干预，自动完成各种运算和输入/输出操作，是因为它可以执行人们事先编制好的各种命令。这些命令通常称为指令。在研制计算机时，设计出若干条指令，使设计出的硬件电路能唯一识别并执行这些指令。某类计算机所有指令的集合就是该计算机的指令系统。显然，指令系统的性能如何，决定了计算机的基本功能。因此，指令系统的设计是计算机系统设计的一个核心问题。从计算机组成的角度来看，指令是软件和硬件的接口。计算机设计者根据指令系统的要求设计硬件，软件工作者根据指令系统提供的功能编制程序。

(2) 源程序

通常，我们在使用计算机时要编制程序。编制程序的过程就是根据要解决问题的要求，从指令系统中选择若干条指令，依据一定的顺序和要求组织起来的过程，这些有序的指令集合就称为程序。为解决用户问题所编制的程序成为源程序，包括用高级语言编写的程序。把源程序翻译成机器码后，微机才能认识和执行，这种机器语言程序称为目标程序。

2. 汇编语言

计算机发展的初期，常常用指令的机器码直接编写用户的程序，这就是机器语言源程序。由于机器码是一串 0、1 字符，不好记忆也难于理解，编写程序十分烦琐和困难，极易

出错，因而，人们用一些助记符号来表示操作码。通常，这些助记符号是指令功能的英文缩写，例如，ADD 表示加法，MOV 表示数据传送等，这样理解和记忆也就方便多了，程序设计也变得简单。人们用助记符号编写的源程序，称为汇编语言源程序。关于汇编语言，将在第 4 章中详细介绍。

3. 高级语言

从计算机诞生起，人们就在寻找一种比较接近人的自然语言，并且能在各类计算机上使用的语言。20 世纪 50 年代，产生了第一种高级语言 FORTRAN，以后陆续产生并得到推广使用的高级语言有 BASIC、ALGOL、COBOL、PASCAL 和 C 语言等。

用高级语言编写的程序也称为源程序。由于它比较接近人的自然语言和数学语言，因而比较容易被人们所理解。例如，从某个存储单元中取得数据，加 1 后再送回去的操作，用 BASIC 语言来写，只要一条语句：

```
X=X+1
```

其中 X 是一个变量。

由于计算机不能识别高级语言编写的程序，因此需要一个翻译将高级语言程序“翻译”成机器可以接受的二进制代码。这种翻译是由一个程序来完成的，这种程序称为编译程序。

1.3 微型计算机系统硬件结构

1.3.1 微机的总线结构

微型计算机，简称微机，是由若干系统部件组成的，这些系统部件在一起工作才能形成一个完整的计算机系统。总线是连接计算机硬件的纽带，也是各功能模块之间传递信息的公共通路。

一个计算机系统中的总线大致分为三类：内部总线、系统总线和多机系统总线。内部总线是同一部件内部(如 CPU 内部)连接各寄存器及运算部件之间的总线；系统总线是同一台计算机系统的各部件(如 CPU、内存和各类 I/O 接口)之间相互连接的总线；多机总线是多台计算机之间相互连接的总线。

从计算机总线的功能看，可以将总线分为数据总线、地址总线和控制总线。计算机借助于三总线实现各个部件之间的数据、地址和控制信号的传送。

1. 数据总线

数据总线用于传送数据信息。所有需要传送数据的部件均挂到总线上，总线的条数(总线的宽度)多数由 CPU 的字长所决定(有些机器的总线宽度与字长不一致，如 Pentium CPU 字长为 32 位，但所有外部数据总线宽度为 64 位)。CPU 的字长是由它的内部结构决定的，一个 8 位(字长为 8)的 CPU 一次只能传输或运算 8 位二进制数据，它的寄存器也多为 8 位的，因此需要用 8 条线来传输数据信息，记为 DB_0～DB_7。一个 16 位(字长为 16)的 CPU 一次能传输或运算 16 位二进制数据，它的寄存器也多为 16 位的，因此需要用 16 条数据线，记为 DB_0～DB_{15}。CPU 通过数据总线向存储器或外部设备输出数据或者从存储器

或外部设备输入数据到 CPU，因此，数据总线是双向并行传输信息的。

2. 地址总线

用来传输地址信息的一组信号线称为地址总线。在单处理器系统中，地址信息由 CPU 发出，用来对存储单元或 I/O 端口进行寻址，因此它是单向并行传送的。地址线的条数与数据线的条数不一定相等，例如，8 位微机采用 16 位地址线，记为 $A_0 \sim A_{15}$，它能寻址的范围是 $2^{16}=65\ 536$，即为 64 KB。因此，访问 64 KB 范围内的存储器单元时，必须同时输出 16 位的地址信息。

3. 控制总线

控制总线用于传送各种控制信号，如存储器读写控制信号、I/O 读写控制信号等。不同类型的微机中的控制信号不同，所使用的控制线的条数也不同。

用总线把 CPU、存储器和外设的 I/O 接口连接起来就构成了一台微型计算机。图 1－1 所示是微型计算机的总线结构。

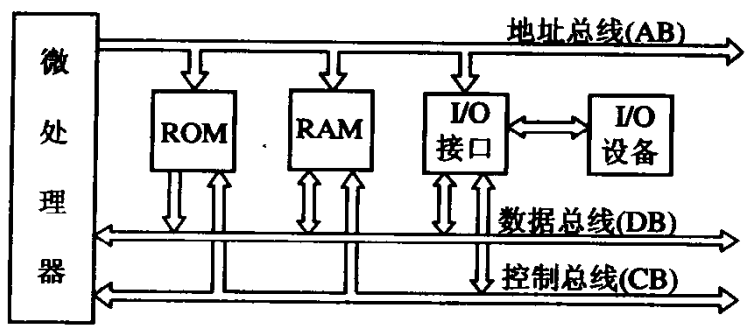

图 1－1　微机总线结构

1.3.2　微型计算机存储器组成

存储器是计算机系统用来存储数据、文件和程序的部件。存储器的分类方法很多，按存储器的工作方式可分为随机读写存储器(RAM)和只读存储器(ROM)。按存储器和 CPU 的关系可分为内部存储器和外部存储器。内部存储器又称为主存储器，简称内存或主存，外部存储器又称为辅助存储器，简称外存或辅存。

内存是计算机主机的一个主要组成部分，用来容纳当前正在使用的或者即将使用的程序和数据。内存一般采用半导体存储器，CPU 可以直接访问内存。目前，微机中通常将内存装在主机板上。

从计算机使用角度看，内存空间越大越好，但由于技术上的限制及价格因素的影响，内存空间是受到限制的。早期使用的 Z80 微型计算机最大内存空间是 64 KB；8086 微型机的最大内存空间为 1 MB；而 AST486 微型机的内存为 4 MB，可扩展到 16 MB，整机最高可扩展到 80 MB，目前使用的 586 计算机一般内存配置为 32 MB，为了满足某些软件的执行要求，其内存可以配置得更大。

外存一般不安装在主机板上，属于计算机的外部设备。它是为弥补计算机内存不足而配置的大容量存储器。它的特点是容量大，成本低，数据能长期保存，但存取速度慢，不能直接与 CPU 进行数据交换。CPU 要使用外存中的程序和数据时，必须通过专门的设备将信息首先成批传送到内存中。在微机中使用最广泛的外存是磁盘和光盘。有关外部存储器的知识将在第 5 章中详细介绍。

1.3.3 中央处理器 CPU 的结构

图 1－2 为一个 8 位微处理器的内部结构框图。微处理器主要由三个部分构成：寄存器阵列、运算器和控制器。

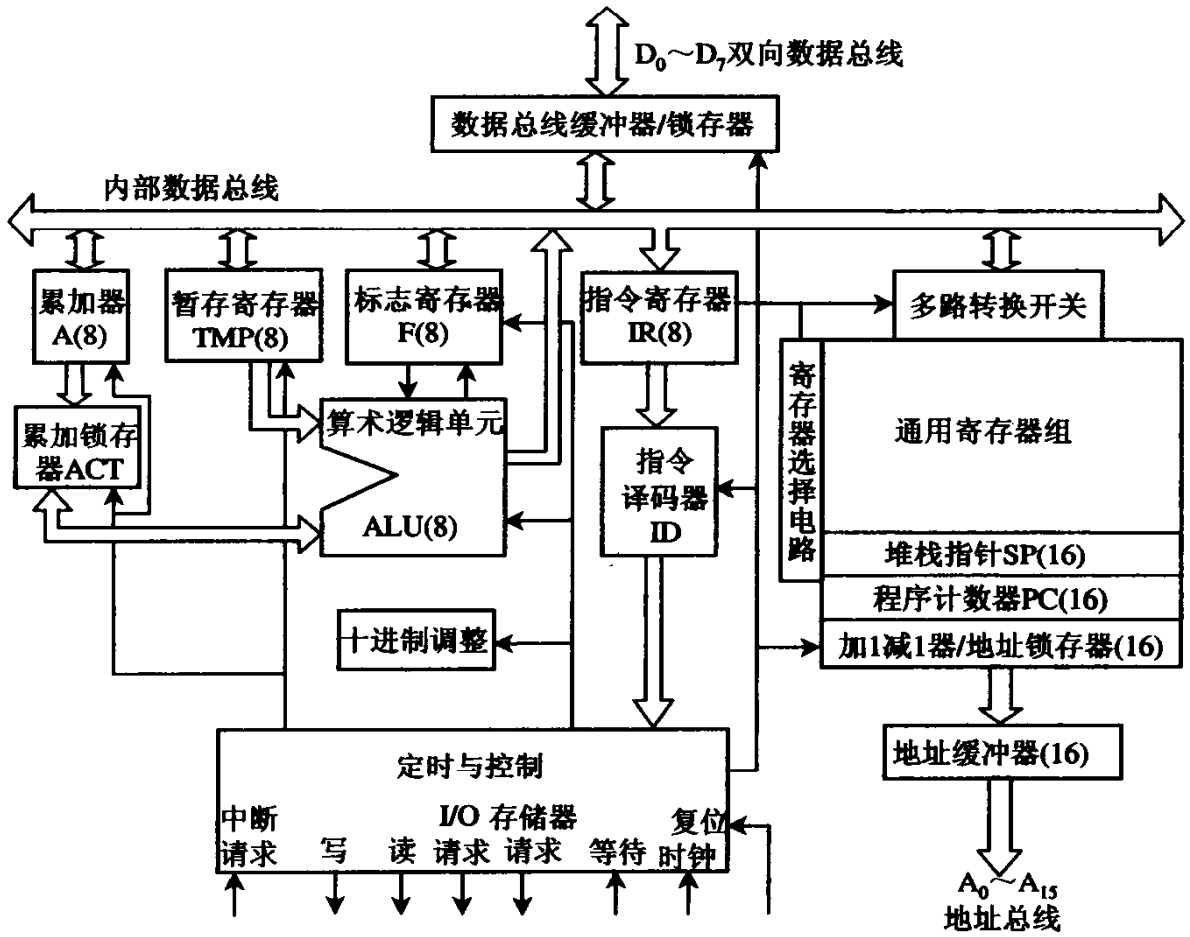

图 1－2　微处理器内部结构框图

1. 寄存器阵列

寄存器阵列是微处理器内部的临时存储单元，包括通用寄存器组和专用寄存器 SP、PC 等。

通用寄存器组用来临时存放数据和地址，减少 CPU 访问存储器的次数，从而提高运算速度。对于 8 位微处理器来说，每个寄存器可以存放 8 位二进制数，并与内部数据总线双向连接。由寄存器选择电路来确定哪个寄存器参与操作。

堆栈指针 SP 用来指示随机存储器中堆栈的栈顶地址。

程序计数器 PC 用来存放现行指令的 16 位地址。

起数据缓冲作用的寄存器有数据总线缓冲/锁存器、地址锁存器和地址缓冲器等。

数据总线缓冲/锁存器是三态双向的缓冲器/锁存器，在 CPU 内部数据总线和外部数据总线之间起数据缓冲作用。

地址锁存器用来存放 16 位地址码，并送往加 1、减 1 器或地址缓冲器。

地址缓冲器是单向的三态缓冲器，在 CPU 内部地址总线与外部地址总线之间起缓冲作用。

2. 运算器

运算器是进行算术运算和逻辑运算的部件，由累加器 A、累加锁存器 ACT、暂存寄存器 TMP、标志寄存器 F 和算术逻辑运算部件 ALU 等组成。

累加器用来存放参与运算的一个操作数以及运算后的结果。

累加锁存器用来锁存从累加器送来的操作数，并送往 ALU 进行运算。

暂存寄存器用来存放参与运算的另一个操作数。

标志寄存器用来保存 ALU 操作结果的条件标志，如进位标志位、零标志位、符号标志位、辅助进位标志位和奇偶标志位等。

ALU 由加法器和其它逻辑电路组成，其基本功能是进行加法和移位，由此实现各种算术和逻辑运算。

在进行运算操作时，操作数从累加锁存器、暂存寄存器和标志寄存器的进位标志位输入 ALU；运算完成后将结果送到数据总线或累加器，运算结果的状态标志送标志寄存器。

3. 控制器

控制器由程序计数器 PC、指令寄存器 IR、指令译码器 ID 和定时控制逻辑电路组成。它是分析和执行指令的部件，是统一指挥微机按一定时序协调工作的核心。

指令寄存器用来保存当前正在执行的一条指令。当执行一条指令时，先从存储器取出指令，并送至指令寄存器 IR，IR 的输出就是指令译码器 ID 的输入，指令由指令译码器译码后，即可向操作控制器发出具体操作的特定信号。

计算机时刻都在高速运行。指令译码后产生的微操作控制信号之间存在着严格的时序关系，因此，在它们输出之前必须对它们进行时间控制，即给予定时，按规定的时刻送到计算机各个部件。这个功能是由定时控制逻辑电路中的时序发生器完成的。时序发生器产生一组时序信号，送到微操作产生部件，对各种微操作控制信号进行定时。

1.4 数　　制

1.4.1 进位计数制

1. 十进制数

在日常生活中，人们已经非常熟悉和习惯使用十进制数，它的数值部分是用 10 个不同的数字符号 0～9 表示的。例如 123.45，小数点左边第一位数表示个位，数值为 3×1，小数点左边第二位数表示十位，数值为 2×10，小数点左边第三位数表示百位，数值为 1×100；小数点右边第一位数 4 表示 4/10，小数点右边第二位数 5 表示 5/100。因此，这个数可以写成 $123.45=1\times10^2+2\times10^1+3\times10^0+4\times10^{-1}+5\times10^{-2}$。一般可以将任意一个十进制数 N(正数)表示为

$$\begin{aligned} N &= K_n \times 10^n + K_{n-1} \times 10^{n-1} + \cdots + K_1 \times 10^1 + K_0 \times 10^0 + K_{-1} \times 10^{-1} \\ &\quad + \cdots + K_{-m} \times 10^{-m} \\ &= \sum_{i=-m}^{n} K_i \times 10^i \end{aligned} \tag{1.1}$$

其中，K_i 可以是 0～9 这 10 个数码中的任意一个，m、n 为正整数，10 称为计数制的基数，10^i 为该位的权。

所谓某进位制的基数，就是该进位制中可能用到的数码个数。若基数为 10，则每位计满 10 就向高位进位，即“逢十进位”。

所谓“权”，即为各位所具有的十进制的数值，上式就是按权展开的。

2. 二进制

在十进制的分析基础上，可以对二进制数进行定义。二进制数只有两个数符 0 和 1，而且是“逢二进一”。它的基数是 2，第 i 位的权为 2^i。例如：

$$(1010.1)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1} = (10.5)_{10}$$

任意一个二进制数可以写成

$$N_2 = K_n\times 2^n + K_{n-1}\times 2^{n-1} + \cdots + K_1\times 2^1 + K_0\times 2^0 + K_{-1}\times 2^{-1} + \cdots + K_{-m}\times 2^{-m}$$

$$= \sum_{i=-m}^{n} K_i\times 2^i \qquad (1.2)$$

其中，K_i 只能取 0 或 1；m、n 为正整数。

比较(1.1)和(1.2)式，除了基数不同，表示形式完全一样。一般来说，如果正整数 R 表示进位制基数，则一个 R 进制数 N_R 可以表示为

$$N_R = \sum_{i=-m}^{n} K_i\times R^i$$

其中，K_i 可以是 0、1、2…$(R-1)$中任一个数码；m、n 为正整数。$R=10$ 表示十进制数；$R=2$，8，16 分别表示二进制、八进制和十六进制数。

从以上分析可见，不同进位制有两个共同特点：

① 每个进位制都有一个固定的基数 R，它的每一个数位可能取 R 个不同的数码：0、1、…$(R-1)$。而且，“逢 R 进一”，即每位计满 R 向高位进一。

② 进位制数都可以写成 $N_R = \sum_{i=-m}^{n} K_i\times R^i$ 的展开式，它的每一位数码 K_i 对应一个固定的值 $K_i\times R^i$，R^i 称为 K_i 的权。

③ 对 R 进制小数而言，若小数点向左移一位，即等于原数缩小 R 倍，即乘了 $1/R$；若小数点向右移一位，则等于原数增大了 R 倍，即乘 R。

3. 二进制的特点

在计算机中，为什么要采用二进制？这是因为二进制较其它进位制有更特殊的优点。首先，二进制数码只要取 0 和 1 两个数码，使计算机只需要具有两个稳定状态的元件，因而制造它比制造具有较多稳定状态的元件要容易得多。例如，利用晶体管的截止、饱和，电位的高和低，灯的亮和灭，脉冲的有和无或正和负等。其次，二进制数计算简单。根据“逢二进一”的原则，只需简单掌握如下和与积的运算规则就行了。因而，实现运算的计算机电路简单、可靠，运行速度快。它们的基本运算法则是：

加法　　0+0=0

0+1=1+0=1

1+1=10

乘法　$0\times0=0$

$0\times1=1\times0=0$

$1\times1=1$

减法和除法分别是加法和乘法的逆运算。在做减法遇到不够减时，需向高位借位，借一作二。

根据上述规则，便可容易地进行二进制数的四则运算。例如：

```
  1011…被加数             10011…被减数
+1101…加数              － 1101…减数
─────────────           ─────────────
 11000…和                 110…差
```

```
        110011…被乘数                000111…商
×         1011…乘数         除数 101) 100100…被除数
──────────────────────                101
        110011                       ─────
       110011                         1000
      000000                           101
+    110011                          ─────
──────────────────────                 110
     1000110001…积                     101
                                     ─────
                                         1…余数
```

二进制虽然计算简单，但书写和阅读非常不便，所以在计算机中常用八进制数和十六进制数。

4. 八进制数和十六进制数

在八进制中，每位可选用的数码共八个，即 0～7，逢八进位，基数为 8；八进制整数，从低位起各位权依次是 8^0，8^1，8^2，…

在十六进制中，每位可选用的数码共 16 个，即 0～9，A～F(A～F 对应表示十六进制数中的 10～15)，逢十六进位，基数为 16；十六进制整数，从低位起各位权依次是 16^0，16^1，16^2，…

1.4.2　数制转换

人们习惯用十进制数，而计算机内部用的是二进制数。因此十进制数输入计算机时，必须将它变换成二进制数。若要求计算机把运算结果以十进制数形式输出时，就要用到从二进制数到十进制数的转换。下面简单介绍数制转换的一般方法。

1. 十进制数转换为二进制数

(1) 十进制整数转换为二进制整数

要把十进制整数转换为二进制整数，可以采用“逐次除 2 取余法”。将十进制整数除以 2，得到一个商和一个余数。再将商除以 2，又得到新的商与余数，如此继续下去，直到商等于零为止。将各次所得的余数按逆序排列(最后一个余数为最高位)，就可得到相应的二进制整数。

以十进制数 35 为例，逐次除 2 取余法如下：

商	余数
35/2＝17	1
17/2＝8	1
8/2＝4	0
4/2＝2	0
2/2＝1	0
1/2＝0	1

转换得到的二进制数为 100011。

(2) 十进制小数转换为二进制小数

要把十进制小数转换为二进制小数，则可以用“逐次乘 2 取整法”。用 2 乘十进制小数，然后去掉乘积中的整数部分，再用 2 乘剩下的小数部分，……如此继续下去，直到小数部分等于 0 或满足所要求的精度为止，以首次乘 2 所得乘积整数部分作为小数最高位，把各次乘 2 所得整数部分按正序排列，就可得到相应的二进制小数。以十进制小数 0.625 为例，逐次乘 2 取整法如下：

0.625×2＝1.25　取整数为 1
0.25×2＝0.5　取整数为 0
0.5×2＝1　取整数为 1

此时小数部分已为 0，所以转换的二进制小数等于 0.101。

需要指出的是，并非所有十进制小数都能用精确的二进制数表示。若将 0.132 6 转换为二进制小数时，读者会发现整个“乘 2 取整”的过程可无限制地进行下去，即乘积的小数部分总不为 0，此时应根据机器的精度，取一定的位数作为其近似值。

2. 十进制数转换为十六进制数

十进制数转换为十六进制数与十进制数转换为二进制数相似。即整数部分用“除 16 取余法；小数部分用乘 16 取整法”。例如将十进制数 1 192.359 375 转换为十六进制数的过程如下：

整数部分

商	余数
1 192/16＝74	8
74/16＝4	10
10/16＝0	4

小数部分

0.359 375×16＝5.75　整数为 5
0.75×16＝12.0　整数为 12

因此转换得到的十六进制数为 4A8.5C。

3. 二进制及其它进制转换为十进制

只要根据该数制的权展开式就可以得到对应的十进制数。

例如，$(1010.11)_2=1\times2^3+0\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+1\times2^{-2}=(10.75)_{10}$

例如，$(71.A)_{16}=7\times16^1+1\times16^0+10\times16^{-1}=113.625$

为了把二进制数或其它进制数表示为十进制数，必须熟练掌握二进制及其它进制的各位权值。

4. 二进制数转换为八进制数和十六进制数

二进制数转换为八进制数的方法是，从小数点开始向左(右)每三位划为一组读数，将三位二进制数用一位八进制数表示。

例如，$(\underline{11}\ \ \underline{011}.\underline{01001})_2=(33.21)_8$

二进制数转换为十六进制数的方法是从小数点开始向左(右)每四位划为一组读数，将四位二进制数用一位十六进制数表示。

例如，$(\underline{101}\ \ \underline{1011}.\ \underline{0110011})_2=(5B.62)_{16}$

要把八进制数(或十六进制数)转换为二进制数，可将每一位八进制数(或十六进制数)转换成对应的三位(或四位)二进制数。

例如，$(45.32)_8=(100\ 101.011\ 010)_2$

$(2A5.B2)_{16}=(1\ 010\ 100\ 101.101\ 100\ 10)_2$

1.4.3 数的表示方法

1. 机器数与真值

数学中用正负号表示数的正负，由于计算机不能识别正负号，因此计算机将正、负等符号数字化，以便运算时识别。通常，在数的前面加一位，作为符号位。符号位为 0 表示该数为正数，符号位为 1 表示该数为负数。一个带符号的数在计算机中可以有原码、反码和补码三种表示方法。

连同符号位一起表示的数称为机器数，机器数的数值称为真值。

例如，$X_1=\underline{0}\ \ \underline{1001101}$，真值为 $X_1=+1001101$

符号位　数值

$X_2=\underline{1}\ \ \underline{1011001}$，真值为 $X_2=-1011001$

符号位　数值

可见，在机器中数的符号被数字化了，符号和数值都是二进制数码。

2. 原码、补码、反码

(1) 原码

一个数的真值中的符号“+”用 0 表示，“−”用 1 表示的二进制数称为原码。例如：

$X_1=(+105)_{10}$，则$[X_1]_{原}=01101001$

$X_2=(-105)_{10}$，则$[X_1]_{原}=11101001$

原码的优点是它与真值的转换非常方便，只要将真值中的符号位数字化即可。一个字节所表示的原码数值的范围为−127～+127。但使用原码作两数相加时，计算机必须对两个数的符号是否相同作出判断。当两数符号相同时，进行加法运算，否则就要作减法运算。

而且对于减法运算又要比较两个数的绝对值大小，然后从绝对值大的数中减去绝对值小的数而得其差值，差值的符号取决于绝对值大的数的符号。为了完成这些操作，计算机的结构，特别是控制电路随之变得复杂，而且运算速度也变得较低。加法运算是计算机中最基本、最频繁的运算操作，因此在微机中一般不采用原码形式表示数。

(2) 反码

正数的反码表示与原码相同，而负数的反码表示与它相应的正数的原码连同符号位一并逐位求反。例如：

$$(+31)_{10}=[+31]_{原}=00011111\rightarrow[+31]_{反}=00011111$$
$$(+127)_{10}=[+127]_{原}=01111111\rightarrow[+127]_{反}=01111111$$

若要写出$(-31)_{10}$、$(-127)_{10}$的反码，则可按下列步骤完成，即

$$(-31)_{10}=[+31]_{原}=00011111 \xrightarrow{\text{连同符号位一起求反}}[-31]_{反}=11100000$$
$$(-127)_{10}=[+127]_{原}=01111111 \xrightarrow{\text{连同符号位一起求反}}[-127]_{反}=10000000$$

一个字节所表示的反码数值的范围为－127～＋127。对于正数，它相应的反码的符号位为 0，其余 7 位为数值；而当符号位为 1 时，则代表的是负数，其余 7 位并非为真实数值，而是数值的反码，为求其真值，则必须对反码再求反。例如，$[X]_{反}=10000000$，由符号位确定它为负数，则应将反码的其余 7 位求反得 1111111＝127，即真值为$(-127)_{10}$。

(3) 补码

正数的补码与原码相同，负数的补码为其反码加 1。例如：

$[+7]_{原}=00000111$　　$[-7]_{原}=10000111$
$[+7]_{反}=00000111$　　$[-7]_{反}=11111000$
$[+7]_{补}=00000111$　　$[-7]_{补}=11111001$
$[+0]_{原}=00000000$　　$[-0]_{原}=10000000$
$[+0]_{反}=00000000$　　$[-0]_{反}=11111111$
$[+0]_{补}=00000000$　　$[-0]_{补}=00000000$

8 位二进制补码所能表示的数值范围是－128～＋127。对于 8 位微型计算机，如果运算结果超过了它所能表示的数值范围，称为溢出。引入补码的好处是可以将减法运算化成加法运算，从而简化机器的控制线路，提高运算速度。

由补码求取反码的例子如下：

$$[X]_{原}=[[X]_{补}]_{补}$$

若　$[X]_{补}=11111111$

则　$[X]_{原}=[11111111]_{补}=10000001$

即　$X=-1$

1.4.4　数的运算方法

我们已经介绍了原码、补码和反码，数据在计算机中是以一定的编码方式表示的，不同的编码有不同的运算规则。原码在数值部分保留了原有的特征，在符号上与真值有区

别，因此用原码进行加减运算比较麻烦。大多数微机采用定点整数补码形式表示有符号数，补码运算比较简单，负数可以用相应的补码表示，减法可以转换为加法运算，这里主要讨论补码运算。

1. 补码加减法运算规则

两个数相加减不外乎以下几种情况：

$$(+A)+(+B)=(+A)-(-B)$$
$$(-A)+(+B)=(-A)-(-B)$$
$$(+A)+(-B)=(+A)-(+B)$$
$$(-A)+(-B)=(-A)-(+B)$$

式中，A、B 为数的绝对值。分析上面四种情况，不难发现减法都可以用加法代替，即等号右边的减法运算功能都可以用等号左边的加法运算实现。括号中的数据及其符号可以用补码表示。正是由于这个原因，一般计算机中只设置加法器，减法运算都是通过适当求补，然后相加来实现的。

计算机采用补码运算是指存储单元和运算寄存器中的数都采用补码表示，数据运算结果也采用补码表示。补码加法的运算规则很简单，即$[X]_{补}+[Y]_{补}=[X+Y]_{补}$。该式的含义是：两个数的补码之和等于两个数之和的补码。由$[X]_{补}+[Y]_{补}=[X+Y]_{补}$可推出$[X+(-Y)]_{补}=[X]_{补}+[-Y]_{补}=[X]_{补}-[Y]_{补}$。因此得出补码的减法运算规则是：$[X]_{补}-[Y]_{补}=[X-Y]_{补}$。那么，如何利用补码进行数的加减运算呢？下面分几种情况讨论。

2. 带符号数的加减运算与溢出判断

(1) 加法

设有带符号数 X、Y，用补码求 X+Y 的步骤如下：

① 将带符号数用补码表示，即将 X 变换为$[X]_{补}$，Y 变换为$[Y]_{补}$。

② 进行补码运算，$[X]_{补}+[Y]_{补}=[X+Y]_{补}$。

③ 求与补码对应的真值。

例 1：已知：$X_1=+0001010$　　$Y_1=+0000101$

$X_2=-0001010$　　$Y_2=-0000101$

计算 X_1+Y_1，X_2+Y_2。

解：

① 求补码：

$$[X_1]_{补}=00001010 \qquad [Y_1]_{补}=00000101$$
$$[X_2]_{补}=11110110 \qquad [Y_2]_{补}=11111011$$

② 补码相加：

$$\begin{array}{rl} 00001010 & [X_1]_{补} \\ +00000101 & [Y_1]_{补} \\ \hline 00001111 & [X_1]_{补}+[Y_1]_{补} \end{array} \qquad \begin{array}{rl} 11110110 & [X_2]_{补} \\ +\ 11111011 & [Y_2]_{补} \\ \hline 111110001 & [X_2]_{补}[Y_2]_{补} \end{array}$$

$[X_1+Y_1]_{补}=[X_1]_{补}+[Y_1]_{补}=00001111$　$[X_2+Y_2]_{补}=[X_2]_{补}+[Y_2]_{补}=11110001$