

第 1 章

微型计算机基础

本章要点

- 了解微型计算机的基本概念
- 掌握微型计算机的主要组件
- 掌握计算机信息编码

1.1 概述

1946年美国宾夕法尼亚大学研制出世界上第一台数字电子计算机 ENIAC(Electronic Numerical Integrator And Calculator) 以来,计算机的发展突飞猛进,日新月异。短短 50 多年中,已经历了电子管计算机、晶体管计算机、集成电路计算机和大规模 / 超大规模集成电路计算机等四代的发展历程。20 世纪 80 年代中期起,开始了以模拟人的大脑神经网络功能为基础的第五代计算机的研究。各代计算机的更替主要表现在组成计算机的电子元器件更新换代,还包括计算机系统结构和计算机软件技术的改进。

作为第四代计算机的一个重要分支,微型计算机于 20 世纪 70 年代初诞生了。微型计算机(Microcomputer)与其他大、中、小型计算机的区别在于其中央处理器 CPU(Central Processing Unit)采用了大规模、超大规模集成电路技术,其他类型计算机的 CPU 则是由相当多的分立元件电路或集成电路所组成。为了将这两种 CPU 相区别,把微型计算机的 CPU 芯片称为微处理器。微处理器(Microprocessor)又称 MPU(Micro Processing Unit)。

目前主要有两大系列微处理器: Intel 80X86 和 Motorola 68X0, 而其中又以 Intel 80X86 系列产品独领风骚,在各种通用微机、专用微机和工作站中应用最为广泛。

微处理器和微型计算机正在向着更微型化、更高速、更廉价和图形、超媒体、更强功能的方向发展。其结果是,一方面各种便携式微机(笔记本式微机、膝上微机、掌上微机)大量涌现,另一方面将超级微型计算机和巨型计算机技术紧密结合、融为一体的“微巨机”也将不断问世。

计算机的应用,归纳起来主要有这样几个方面:

(1) 科学计算与数据处理。这是最原始、也是占比重最大的计算机应用领域。在科学研究、工程设计和社会经济规划管理中存在大量复杂的数学计算问题,如卫星轨道的计算、大型水坝的设计、航天测控数据的处理、中长期天气预报、地质勘探与地震预测、社会经济发展规划的制定等,常常需要涉及大量数值计算,利用计算机可快速得到较理想的结

果。

(2) 生产与试验过程控制。在工业、国防、交通等领域,利用计算机对生产和试验过程进行自动实时监测、控制和管理,可提高效率、提高质量、降低成本、缩短周期。

(3) 自动化仪器、仪表及装置。在仪器、仪表装置中使用微处理器或微型计算机,可明显增强功能、提高性能、减小重量和体积。

(4) 信息管理与办公自动化。现代企事业单位和政府、军队各部门需要管理的内容很多,如财务管理、人事档案管理、情报资料管理、仓库材料管理、生产计划管理、信贷业务管理、购销合同管理等。采用计算机和目前迅猛发展的计算机网络技术,可实现信息管理自动化和办公自动化。

(5) 计算机辅助设计。在航空航天器结构设计、建筑工程设计、机械产品设计和大规模集成电路设计等复杂设计活动中,为了提高质量、缩短周期、提高自动化水平,目前普遍借助计算机进行设计,即计算机辅助设计 CAD(Computer Aided Design)。CAD 技术发展迅速,应用范围不断拓宽,目前又派生出计算机辅助测试 CAT(Computer Aided Test)、计算机辅助制造 CAM(Computer Aided Manufacture)和将设计、测试、制造融为一体的计算机集成制造系统等新的技术分支。

(6) 计算机仿真。在对一些复杂的工程问题和复杂的工艺过程、运动过程、控制行为等进行研究时,在建立数学模型的基础上,用计算机仿真的方法对相关的理论、方法、算法和设计方案进行综合、分析和评估,可以节省大量的人力、物力和时间。

(7) 人工智能。人工智能是用计算机系统模拟人类某些智能行为的新兴学科技术,它包括声音、图像、文字等模式识别、自然语言理解、问题求解、定理证明、程序设计自动化和机器翻译、专家系统等。

(8) 文化、教育、娱乐和日用家电。计算机辅助教学 CAI(Computer Aided Instruction) 已成为一种重要的教学手段。今天,电影、电视片的设计、制作、多媒体组合音像设备的推出,许多全自动、半自动“家电”产品的出现,以至许多智能型儿童玩具,无一不是微型计算机在发挥着作用,显示出奇功。

1.1.1 电子计算机的主要组成及功能

电子计算机是能够自动、高速、精确地进行逻辑控制和信息处理的现代化电子设备。计算机系统由硬件系统和软件系统两大部分组成。硬件是指计算机物理实体部分,由各种集成电路以及机械部件组成,是计算机存储和运算的基础。一台只有硬件的计算机叫做裸机,它不能为人们所用,必须为它配上相应软件,才能完成人们规定的工作。一个包括硬件和软件的计算机称为计算机系统。

1. 电子计算机基本结构

计算机硬件结构基本上采用计算机经典结构—冯·诺依曼结构。这种结构的主要特点是:

(1) 由运算器、控制器、存储器、输入设备和输出设备五大部分组成。其中运算器和控制器合称为中央处理单元,简称 CPU(Central Processing Unit)。它是计算机硬件的核心,完成控制和运算任务。存储器用来存放程序与数据。CPU 与存储器合称为主机,主机以外

的设备称为外部设备。

(2) 数据和程序以二进制码形式存放在存储器中。

(3) 控制器是根据存放在存储器中的指令序列即程序来控制计算机的工作过程。控制器具有判断能力,可选择不同工作流程。

2. 计算机硬件功能

计算机硬件的五大组成部分如图 1.1 所示。

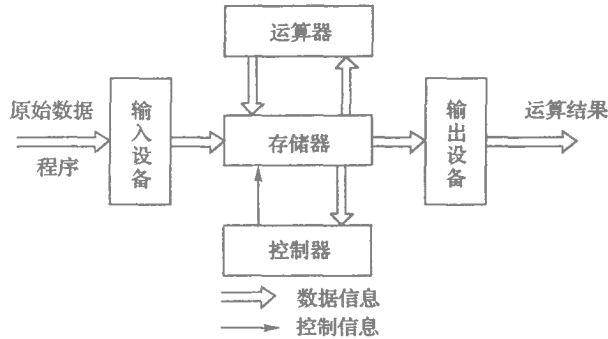


图 1.1 电子计算机的主要组成

(1) 运算器

运算器是计算机中进行运算的部件。运算分为两类:一类是算术运算(即加、减、乘、除四则运算);另一类是逻辑运算(如与、或、非等运算)。运算器是由加法器、移位寄存器和一些控制电路组成。此外,电子计算机需要实现数码的传送和移位等操作,也是通过运算器完成的。

(2) 控制器

控制器是计算机的指挥中心,它严格按时间要求发出对各个部件动作进行控制的信号,使整个计算机各部件能够有条不紊地协同工作。

控制器主要由三大部分组成:指令部件、时序部件和微操作控制部件。控制器的工作实质就是执行程序,它每次从存储器中读取一条指令,经过分析译码,产生一串操作控制命令,控制各部件动作,从而执行这条指令。控制器产生操作命令依据以下三个方面:一是指令,它放在指令寄存器中,指令是计算机操作的主要依据。二是各部件的状态触发器,其中存放反映机器运行状态的有关信息。三是时序电路,使控制器执行命令被有序地发送出去,控制打开机器的控制门,完成指令所规定的动作。

(3) 存储器

存储器的主要功能是存放程序和数据。程序是计算机操作的依据,数据是计算机操作的对象,不论程序还是数据,在存储器中均以二进制形式表示,统称为信息。为了实现自动计算,人们对数据处理过程编制程序,把这些程序存入计算机中。计算机运行时逐条地从存储器中取出指令和数据,进行规定的运算。

(4) 输入设备与接口

输入设备的作用是将程序和各種信息转换成二进制形式存入计算机中，例如键盘、鼠标器、硬盘、光盘驱动器等。由于各种输入设备的信号形式及输入速度不同，为了使计算机正常接收这些数据，就需要在输入设备与主机之间加一个适配装置，称为接口，它使输入数据能够正确无误地存入计算机。

(5) 输出设备与接口

输出设备将计算机处理结果转换成我们所需要形式输出，如打印机、显示器等。有的输出设备将二进制信息变换一种格式存入磁盘，也有的通过通信线路传给远方计算机。输出设备与主机之间协调工作，也需要相应接口设备。

1.1.2 微处理机和微型计算机的概念

微型计算机是采用高集成度的器件和部件，它的核心部分是微处理器。微处理器（微处理机）是指一片或几片大规模集成电路组成具有运算器和控制器功能的中央处理器，称 CPU，微型计算机在系统结构和基本原理上与其他计算机基本相同。典型微处理器结构如图 1.2 所示。

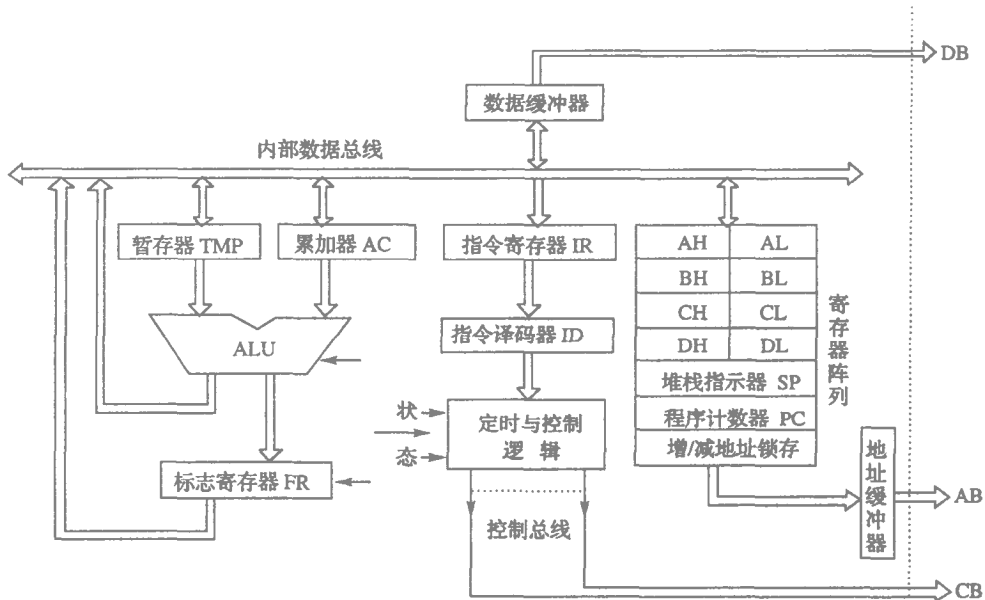


图 1.2 典型微处理器结构

以微处理机为核心，配备上存储器及可编程接口，再配上外部设备（包括键盘、显示器、打印机、磁盘机等）及电源组成了微型计算机。微型计算机与系统软件结合构成微型计算机系统。如图 1.3 所示。单纯的处理器不是计算机；单纯的微型计算机也不是完整的计算机；只有微型计算机系统才是完整的计算机系统，才能正常工作。

微型计算机的分类方法有多种，按微处理器处理位数，可分为 1 位机、4 位机、8 位

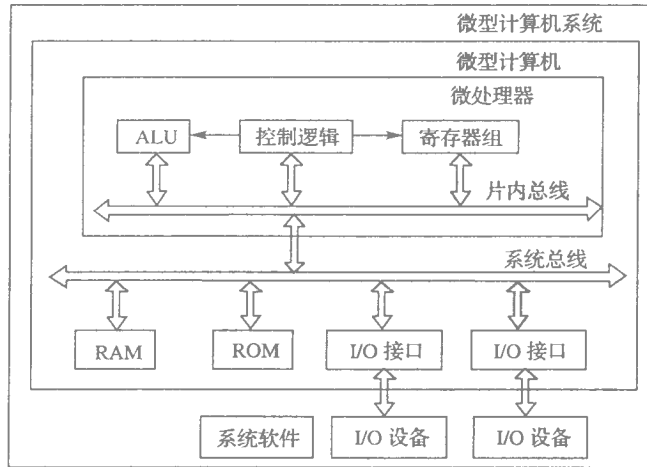


图 1.3 微型计算机组成框图

机、16位机、32位机和64位机等；按结构可分为单片机和多片机；按外形和使用特点，可分为台式微机和笔记本式微机等。

微型计算机系统和普通计算机系统一样，衡量其性能技术指标主要有：

1. 字长

计算机一次所能够处理的最多位数的二进制信息的整体称作一个字。一个字所具有的二进制位数叫字长。字长越长，一个字所表达的信息量越大，所能表示的数据精度越高；在完成同样精度运算时，则数据处理速度越快。在计算机中，8位连续的二进制数称作一个字节。

2. 存储器容量

存储器中能够存放的二进制信息的多少称为存储容量，它是衡量计算机存储能力的一个重要指标。微型计算机中一般以字节 B(byte) 为单位表示存储容量，将 1024B 简称为 1KB, 1024KB 简称为 1MB(兆字节), 1024MB 简称为 1GB(吉字节), 1024GB 简称为 1TB(太字节)。

3. 运算速度

计算机的运算速度一般用每秒能执行的指令数表示。由于不同类型的指令所需时间长度不同，因而运算速度的计算方法也不同。常用计算方法有：

(1) 根据不同类型指令出现的频率，乘以不同系数，求平均值，得到平均运算速度。这时常采用百万条指令/秒做单位。

(2) 以执行时间最短指令为标准来估算速度。

(3) 直接给出 CPU 的主频和每条指令的执行所需的时钟周期。主频一般以 MHz 为单位。

4. 外设扩展能力

主要指计算机系统配接各种外部设备的可能性、灵活性和适应性。计算机允许配接

多少外部设备，对于系统接口和软件都有重大影响。

5. 软件配置情况

软件是计算机系统必不可少的重要组成部分，它配置是否齐全，直接关系到计算机性能和应用效率的高低。

1.2 计算机中信息编码

计算机作为一种计算工具，它的基本功能是对信息进行加工和处理。计算机中存储信息 数值、字符、汉字、指令、状态等 都是用二进制数形式表示的 这样计算机才能识别。

1.2.1 带符号数的表示法

1. 无符号数和带符号数

在计算机中有无符号数和带符号数之分。所谓无符号数，就是计算机的全部有效位数用来表示数的大小，相当于绝对值。

实际中大量用到的还是带符号的数。由于计算机无法识别符号“+”或“-”因此需要把符号数码化，通常约定二进制数的最高位为符号位，“0”表示该数为正，“1”表示为负。这种在计算机中表示的数形式称为机器数。带符号数有原码、补码、反码等不同的编码。

2. 原码表示法

原码表示法是保持机器数的原型不变，符号位为 0 表示该数为正 符号位为 1 表示该数为负，其有效数值部分用二进制数的绝对值表示。一个数 x 的原码记作 $[x]_{\text{原}}$

$$\begin{aligned} \text{例 } x_1 &= +0110101 & [x_1]_{\text{原}} &= 00110101 \\ x_2 &= -0110100 & [x_2]_{\text{原}} &= 10110100 \end{aligned}$$

3. 反码表示法

正数的反码与原码相同，即符号位为 0 数值位不变。负数的符号位为 1 数值位变反 这是反码表示法。一个数 x 的反码记作 $[x]_{\text{反}}$ 。

$$\begin{aligned} \text{例 } [x_1]_{\text{原}} &= 01100101 & [x_1]_{\text{反}} &= 01100101 \\ [x_2]_{\text{原}} &= 10001010 & [x_2]_{\text{反}} &= 11110101 \end{aligned}$$

4. 补码表示法

正数的补码与原码相同 即符号位为 0 数值位原样不变。负数的符号位为 1 数值位取反 且末位加 1 这是补码表示法。一个数 x 的补码记作 $[x]_{\text{补}}$ 。

$$\begin{aligned} \text{例 } [x_1]_{\text{原}} &= 00110101 & [x_1]_{\text{补}} &= 00110101 \\ [x_2]_{\text{原}} &= 11010100 & [x_2]_{\text{补}} &= 10101100 \end{aligned}$$

在微型计算机中，带符号数通常是用补码表示的。读者可以验证，一个数 x 它的 $[[x]_{\text{补}}]_{\text{补}} = [x]_{\text{原}}$ 。一个 8 位的二进制数表示的补码数的范围是 $-128 \sim +127$ (见表 1.1)，同理，1 个 16 位的二进制数表示的补码数的范围是 $-32768 \sim +32767$ 。

表 1.1 8 位二进制数的原码、反码、补码对照表

二进制表数码表示	原码	补码	反码
0000000	+0	+0	+0
0000001	+1	+1	+1
0000010	+2	+2	+2
⋮	⋮	⋮	⋮
0111100	+124	+124	+124
0111101	+125	+125	+125
0111110	+126	+126	+126
0111111	+127	+127	+127
1000000	-0	-128	-127
1000001	-1	-127	-126
1000010	-2	-126	-125
⋮	⋮	⋮	⋮
1111100	-124	-4	-3
1111101	-125	-3	-2
1111110	-126	-2	-1
1111111	-127	-1	-0

引入补码概念之后，符号位参加运算，能将减法运算转换成加法运算，从而简化了机器的硬件结构。举例如下：

$$\begin{array}{r} 89 \\ - 53 \\ \hline 36 \end{array}$$

在计算机中是利用 +89 和 -53 的补码相加运算的（以 8 位二进制补码为例）

$$\begin{array}{r} 01011001 \quad \cdots \cdots + 89 \text{ 的补码} \\ + 11001011 \quad \cdots \cdots - 53 \text{ 的补码} \\ \hline 00100100 \quad \cdots \cdots + 36 \text{ 的补码} \end{array}$$

$$\begin{array}{r} 53 \\ \text{又如：} \quad - 89 \\ \hline - 36 \end{array}$$

在计算机中是利用 +53 和 -89 的补码相加运算的。

$$\begin{array}{r} 00110101 \quad \cdots \cdots + 53 \text{ 的补码} \\ + 10100111 \quad \cdots \cdots - 89 \text{ 的补码} \\ \hline 11011100 \quad \cdots \cdots - 36 \text{ 的补码} \end{array}$$

1.2.2 定点数和浮点数

计算机常用的数据表示格式有定点数和浮点数两种。

1. 定点表示法

所谓定点表示法即事先约定机器中，所有的数据中的小数点的位置是固定不变的，且通常将数据表示为纯整数（规定小数点固定在最低数值位之后）或纯小数（规定小数点固

定在最高数值位之前)。定点法表示的数据称作定点数。

如果用 n 位二进制数表示一个纯整数 (不包括符号位) x 的范围是:

$$0 \leq |x| \leq 2^n - 1$$

它能表示数的最大绝对值为 $2^n - 1$ 最小绝对值为 1。

如果用 n 位的二进制数表示一个纯小数 (不包括符号位) x 的范围是:

$$0 \leq |x| \leq 1 - 2^{-n}$$

它能表示的数最大绝对值为 $1 - 2^{-n}$ 最小绝对值为 2^{-n} 。

图 1.4 给出了定点数的两种表示法。



图 1.4 定点数的两种表示法

2. 浮点表示法

浮点表示法是指小数点的位置不固定,是浮动的。

任意一个二进制数 x 总可以写成下面的形式:

$$x = \pm d \times 2^{\pm P}$$

其中 d 称为尾数,是二进制纯小数,指明数的全部有效数字。前面的符号称作数符,表示数的正负,用尾数之前的一位表示,0表示正号,1表示负号; P 称为阶码,它前面的符号称作阶符,用阶码前的一位表示,阶符为正时用0表示,阶符为负时用1表示。由此可知,将尾数 d 的小数点往右(当为 $+P$ 时)或往左(当为 $-P$ 时)移动 P 位,即得出数 x 的值。所以阶符和阶码指明小数点的位置。小数点随着 P 的符号和大小而浮动。这种数称为浮点数。浮点数的编码格式如图 1.5 所示。



图 1.5 图浮点的编码格式

设阶码的位数为 m 位,尾数的位数为 n 位,则浮点数的取值范围为(最小绝对值到最大绝对值):

$$2^{-n} \cdot 2^{-(2^m - 1)} \leq |x| \leq (1 - 2^{-n}) \cdot 2^{(2^m - 1)}$$

浮点数表示的数值范围大,这是它的主要可取之处。对浮点数进行运算之前,需要使小数点对位。若两个数的阶码不同,则在运算前首先要“对阶”,且按大的阶去对阶,否则可能产生误差或出错。一般浮点数都以“规格化”方式表示,即尾数的最高位是一个有效数字,而不是“0”,这样可以保留最多的有效数字,提高运算精度。

1.2.3 十进制数编码

计算机与人交互时,经常采用十进制表示法,而计算机内部只有二进制数,因此人们规定用二进制数表示十进制数,称为二—十进制数,简称 BCD 码。

这种编码规定用四位二进制编码代表一位十进制数,见表 1.2。当有多位十进制数时,按十进制数的次序将各位的 BCD 码排列起来。

表 1.2 8421BCD 码

十进制	0	1	2	3	4	5	6	7	8	9
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

例如：

十进制数 41.2 用 BCD 码表示是 $(01000001.0010)_{BCD}$

十进制数 524 用 BCD 码表示是 $(010100100100)_{BCD}$

1.2.4 计算机非数值数据的表示

非数值数据是一类不具有“值”大小的数据。比如，计算机作文字处理的处理对象——“文字”就是典型例子。

字符编码

因为计算机中只能存储二进制码，所以非数值字符也只能用二进制编码形式表示，计算机才能接受。ASCII 码是 American Standard Code for Information Interchange（美国信息交换标准代码）的缩写。见附录一。ASCII 码表中用七位二进制码进行编码，用以代表各种不同的可见不可见字符 包括英文字母、数字、标点符号、特殊符号以及一些控制符号等。

在计算机中，汉字也是用二进制代码表示的，每个汉字用二个字节表示。

1.3 微型计算机的主要组件

微型计算机发展是以微处理器的升级换代为标志，而微处理器的发展是按字长和功能为主要指标。

微处理器发展是以 Intel 公司 80X86 微处理器为主流。

1.3.1 微处理器（机）

计算机由五大部分组成 即运算器、控制器、存储器、输入设备和输出设备 其中运算器和控制器合称为中央处理器（CPU），如果把其中运算器和控制器制做到一块集成电路上 这样构成的中央处理机被称为微处理机（或微处理器）微处理机不但体积小 减少功耗，而且还由于减少复杂外部连线，从而大大地提高处理器的可靠性。

1. 8086/8088 型微处理器

8086 型微处理器的内外数据均为 16 位 地址总线 20 位 因此称为标准 16 位微处理器。它直接寻址空间达 1MB。8086 主要由执行部件 EU 和总线接口部件 BIU 组成，工作主频为 4~8MHz 运算速度约为 2.5MI/S(1MB) 指令周期为 0.4 μ S 外形为 40 引脚芯片。

8088 是准 16 位机处理器 内部数据总线为 16 位 外部数据总线为 8 位。

2. 80286 型微处理器

80286 型微处理器是高于 8086 的 16 位微处理器 地址总线为 24 位 直接寻址空间为 16MB(字节) 工作频率为 8~10MHz 运算速度约为 0.4MI/S。主要部件由执行部件、地址总线、总线部件、指令部件组成。80286 工作方式采用流水线方式，执行速度大大提高，

80286 提高了存储器管理功能。

3. 80386 型微处理器

80386 型有 SX、SL、SLC、EX、和 DX 等多种产品，其内外部数据总线和地址总线均为 32 位，寻址能力达 4GB，工作频率为 16 ~ 50MHz。80386 内部由 6 部分组成：执行部件、总线接口部件、指令译码部件、预取部件、分段部件、分页部件。所以可实现段式、页式或段页式管理，这是 80386 的突出特点。

4. 80486 型微处理器

80486 型是标准 32 位微处理器，时钟频率 33 ~ 100MHz，80486 型微处理器由 8 个基本部件组成，分别是总线接口部件、指令预取部件、指令译码部件、执行部件、控制部件、存储部件、高速缓存部件、浮点处理部件。

5. Pentium 微处理器

Pentium 的内部数据总线为 32 位，外部数据总线为 64 位，工作频率 66 ~ 200MHz。CPU 内部具有 8KB 的代码高速缓存，还有 8KB 的数据高速缓存。

6. Pentium Pro/mmx 高能奔腾和多能奔腾

Pentium Pro/mmx CPU 由两块芯片构成：一个是 CPU，另一个是 512KB 的二级高速缓存 (L2 Cache)，通过一条 64 位宽并且与 CPU 相等时钟频率的专用总线，实现其与 CPU 之间通信，从而提高 CPU 的性能。

7. Pentium III 奔腾 3 代

Pentium III 是 Intel 公司第一款为提高用户互联网性能而设计的微处理器，目前有 450MHz、500MHz、550MHz、600MHz 等几种。Pentium III 的一级代码高速缓存和数据高速缓存均为 16KB，二级高速缓存是 512KB，使用 100MHz 的内存总线速度；其数据总线和地址总线宽度分别达到 64 位和 32 位，最大寻址范围可达 64GB。它可以构成两个处理器共同工作系统。这种处理器采用 0.25 μm 的半导体工艺，晶体管数量达到 950 万个。

8. Pentium Xeon(至强处理器)

Pentium II Xeon 主要用于工作站和服务器，它能扩充到 4 个处理器，极大提高了 Windows NT 服务器的性能。Pentium III Xeon CPU 一级代码高速缓存达到 16KB，二级高速缓存为 512KB，使用 100MHz 的内存总线速度，它的数据总线和地址总线分别达到 64 位和 32 位，最大寻址空间为 64GB。

1.3.2 存储器

若没加特殊声明，存储器是指计算机内部存储器，又叫内存或主存，是计算机的存储和记忆部件，用以存放数据（包括原始数据、中间结果和最终结果）和程序。往存储器中存入信息（又称写入）或从存储器中取出信息（也称读出）的过程叫存取操作（或读写操作），有时也称作对内存的“访问”。

计算机内存是由一个个内存单元组成，每个单元一般存放一个字节（8 位二进制数）的二进制信息，内存单元的总数称为内存总量。

内存器主要分成两类：随机存储器 RAM(Random Access Memory) 和只读存储器 ROM(Read only Memory)。RAM 可以被 CPU 随机地读和写，所以又称为读写存储器，这种存储

器用于存放用户的程序、数据及部分系统信息。当计算机断电后，保存的信息丢失。ROM 中的信息只能被 CPU 随机读取，不能任意写入。机器断电后，信息不丢失。所以这种存储器主要用来存放那些固定不变、不需要修改的程序和数据，如基本输入和输出程序等标准程序和有关计算机硬件的数据。ROM 的内容是由生产厂家或用户用专用设备写入固化的。

1.3.3 输入和输出设备的接口电路

输入和输出接口电路是微型计算机系统的重要组成部分，微型计算机通过它与外部设备交换信息。常用输入设备有硬盘、鼠标器、扫描仪等；常用输出设备有显示器、打印机、绘图仪等。磁带、硬盘、光盘、触摸屏等既是输入设备，又是输出设备。

输入输出设备种类繁多，结构、原理各异，因此微型计算机与输入输出设备之间连接与信息交换不能直接进行，而必须设计一个“接口电路”作为两者之间的桥梁。这种输入输出接口电路又叫“输入输出适配器”。

1.3.4 总线

总线是一组各种公共信号线的总称，是微型计算机中各功能部件间传递信息的公共“通道”。

微型计算机有三种总线，它们是：

1. 数据总线 (DB)

数据总线用来传输数据信息，是双向总线，CPU 既可通过数据总线从内存或输入设备读入信息，又可通过数据总线将内部数据传送到内存或输出设备。

2. 地址总线 (AB)

地址总线用于传送 CPU 发出的地址信息，是单向总线。主要指与 CPU 交换信息的内存单元或输入输出设备的地址。

3. 控制总线 (CB)

控制总线用来传送控制信号、时序信号和状态信息等。其中有的是 CPU 向内存和外设发出信息，有的则是内存或外设向 CPU 发出信息，控制总线均以双向线表示。

习 题

1.1 解释下列概念

(1) 微处理器、微型计算机、微型计算机系统

(2) I/O 设备、I/O 接口

1.2 试述微型机中字节、字、字长的含义。

1.3 微处理器一般应具有哪些基本功能？在内部结构上主要由哪些部分组成？

1.4 微型计算机由哪些基本功能部件组成？各部分的功能是什么？

1.5 什么是总线？总线分哪几种？各自作用是什么？

第 2 章

8088 微处理机

本章要点

- 掌握 8088 CPU 内部基本结构及外部引脚
- 掌握 8088 的存储器组织及相关概念
- 了解 8088 支持芯片及其在系统中的作用
- 掌握总线控制器

2.1 8088 CPU 结构及存储器组织

CPU 是微型计算机的核心部件，其性能和结构特点决定了微型计算机的性能。因此，了解 CPU 的组织结构、引脚功能等是学习微机原理、进行微机应用系统开发的基础。

2.1.1 Intel 8088 CPU 主要特点及技术参数

8086/8088CPU 是 Intel 系列微处理器中具有代表性的高性能 16 位微处理器，它保持同以后推出的各种 Intel 系列微处理器的兼容性。因此，深入理解 8086/8088 CPU 也是进一步掌握 Intel 其他高档微处理器的基础。

Intel 8086/8088 CPU 采用 HMOS 工艺制造，外形封装为双列直插式，共有 40 个引脚。主时钟频率为 4 ~ 8MHz。8086/8088 内部采用 16 位数据传送和并行流水线结构，从而允许在总线空闲状态预取指令，使取指令和执行指令的操作并行进行，提高了 CPU 处理速度。8088 对外提供 8 位数据总线，8086 提供 16 位和 20 位地址总线，直接寻址空间为 1MB。

8086/8088CPU 具有格式灵活、功能完善的指令系统，可对多种数据类型进行处理。

8086/8088 支持多处理器系统，可以方便地与数值协处理器 8087 或输入/输出协处理器 8089 相连，构成多处理器系统，以提高系统数据处理能力和输入/输出能力。

8086 和 8088 两个 CPU 内部结构基本相同，都是 16 位内部结构，但对外数据总线宽度不同，8086 是 16 位，而 8088 只有 8 位，所以 8088 称为准 16 位 CPU。下面讨论主要针对 8088，但也适用于 8086。

2.1.2 8088 内部结构

为了说明 8088CPU 内部结构，如图 2.1(b) 的特点，我们从 CPU 执行指令的过程看其体系结构。

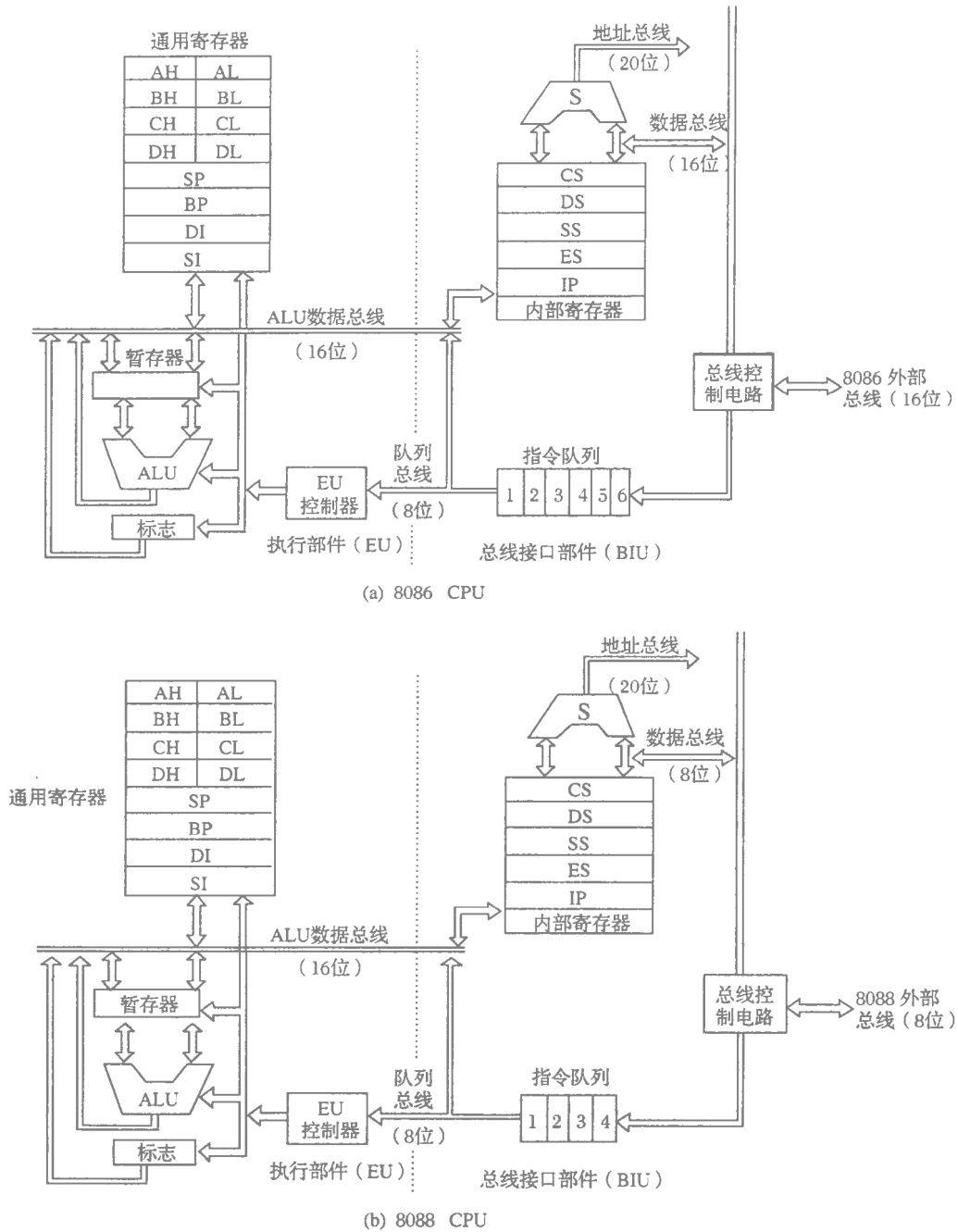


图 2.1 8086/8088CPU 内部结构框图

8088 CPU 采用并行流水线处理方式，即通过反复循环下列步骤执行指令的：

- (1)从内存取出一条指令（总线忙）
- (2)读操作数 如果有 0 总线忙）
- (3) 执行指令（算术逻辑单元忙）
- (4)存结果 如果指令要求 0 总线忙）

8088CPU 把上述步骤分配给内部两个独立部分：总线接口部件（BIU）和指令执行部件（EU）去执行。总线接口部件 BIU 负责取指令、取操作数和存结果，即所有的外部操作均由 BIU 完成。指令执行部件 EU 专门负责执行指令 它从 BIU 的指令队列中取出指令 并执行指令，不必访问存储器或 I/O 端口，节省了取指令时间，加快了程序执行速度。若要访问存储器或 I/O 端口 也是由 EU 向 BIU 发出请求 并提供访问所需的有效地址 在 BIU 中形成物理地址，然后由 BIU 访问存储器或 I/O 端口 取得操作数送 EU 或将运算结果送到指定的内存单元或 I/O 端口。BIU 和 EU 并行操作，在大多数情况下都能使取指令和执行指令的操作并行进行，使 ALU 和总线的空闲时间减少，从而加快程序的执行速度。

1. 总线接口部件 BIU(Bus Interface Unit)

总线接口部件是 8086/8088CPU 与外部存储器和 I/O 的接口。它对外提供 8 位 8086 为 16 位 双向数据总线和 20 位地址总线。BIU 功能是负责与存储器、I/O 接口进行数据传送，所有外部总线操作均由它完成。BIU 具体工作：

(1)总线接口部件 BIU 要从内存取指令送到指令队列。取指令时，将代码段寄存器 CS 中 16 位段基地址的最低位后补 4 个 0 再与指令指示器 IP 中的 16 位偏移地址在地址加法器中相加形成 20 位物理地址，通过总线控制逻辑发出存储器读命令 \overline{RD} 启动存储器，按 20 位物理地址从存储器中取出指令，并送入指令队列供 EU 执行。

(2)8088CPU 内 BIU 中有一个 4 字节指令队列，允许预取 4 个字节指令代码。当指令队列有 1 个字节空余时，BIU 将自动取指令到指令队列，EU 直接从 BIU 的指令队列中取指令执行，在大多数情况下，指令队列中至少应有一个字节的指令，这样 EU 就不必等待 BIU 去取指令。

(3)配合 EU 执行指令，存取操作数或运算结果。EU 可源源不断地直接从 BIU 中的指令队列取指令并执行，当 EU 执行指令时需要取操作数或存结果，先向 BIU 发出请求，并提供操作数的有效地址 BIU 将根据 EU 的请求和提供的有效地址 形成 20 位物理地址并执行一个总线周期去访问存储器或 I/O 接口 从指定单元或 I/O 端口取出操作数送交 EU 使用或将结果存入指定单元或 I/O 端口。需要指出，当 EU 向 BIU 申请从存储器或 I/O 端口读写操作数时，如果 BIU 此时已准备好取指令 则 BIU 先完成取指令的操作，然后进行操作数读写。

(4)如果指令队列已满 且 EU 又没有向 BIU 提出访问总线请求 则 BIU 不执行任何总线操作，处于空闲状态。

(5)当 EU 执行转移指令时，存放在指令队列中的预取出指令就无用了。这时 BIU 应清除指令队列 按 EU 提供新地址取指令。BIU 新取得的第一条指令，将直接送到 EU 中

去执行,然后,BIU 将随后取得的指令重新填入指令队列。

(6)BIU 可负责产生总线控制信号,如访问存储器或 I/O 端口的读写控制信号 \overline{RD} 、 \overline{WR} 等。

为了完成上述功能,BIU 由以下几部分组成:

4 个 16 位段地址寄存器

CS—代码段寄存器。存放当前程序所在段的段首地址(段基址)。

DS—数据段寄存器。存放当前程序所用数据段的段基址。

ES—附加数据段寄存器。存放辅助数据所在段的段基址。

SS—堆栈段寄存器。存放当前程序所用堆栈段的段基址。

16 位指令指示器 IP:存放下一条要执行指令的有效地址(EA 偏移地址)。IP 的内容由 BIU 自动修改,通常进行加 1 修改,当执行转移指令、调用指令时,BIU 装入 IP 中的是转移的目的地址。

20 位物理地址加法器:将 16 位逻辑地址变换为读/写存储器所需的 20 位物理地址,即完成地址加法操作。

指令队列:8088CPU 的 BIU 中有 4 个字节的指令队列,允许预取 4 个字节的指令代码。指令队列相当于一个先进先出栈。

总线控制逻辑:总线控制逻辑的作用是产生并发出总线控制信号,以实现存储器 and I/O 端口的读/写控制。它将 8088CPU 的内部总线与 8 位外部总线相连,是 CPU 读/写操作必不可少的路径。

2. 指令执行部件 EU(Execution Unit)

执行部件 EU 的作用主要是负责执行指令。

由图 2.1 可知,EU 由以下几部分组成:

(1)算术逻辑单元 ALU:完成 8 位/16 位的二进制算术、逻辑运算,绝大部分指令执行由 ALU 完成,16 位的暂存器可暂存操作数。运算结果可通过内部总线送入通用寄存器或由 BIU 存入存储器。

(2)标志寄存器 FR:FR 为 16 位,实际仅用 9 位存放指令执行结果的特征和处理器状态。

(3)寄存器组:8 个 16 位寄存器,按功能分为两组:4 个通用寄存器即 AX、BX、CX、DX,其中 AX 又称是累加器。4 个专用寄存器,即基址指示器 BP(BP 中存放的是堆栈段中某一单元偏移地址)、堆栈指示器 SP(存放的是栈顶地址)、源变址寄存器 SI 和目的变址寄存器 DI。

(4)EU 控制逻辑/EU 控制器:作用是从 BIU 中的指令队列取指令并执行,根据指令要求向 EU 内各功能部件发送相应的控制命令,以完成每条指令所规定的操作。

EU 的工作就是执行指令。它不与外界发生联系,直接从 BIU 中的指令队列中不断地取指令并执行指令,省去了访问内存取指令的时间,加快了程序运行速度。当执行转移指令、调用指令和返回指令时,EU 要等待 BIU 清除指令队列,重新按照转移目的地址取指令。

供 EU 执行 这时 BIU 取出的第一条指令直接送入 EU 去执行, 随后取出的指令填入指令队列。当 EU 在执行过程中需要访问存储器或 I/O 端口存取操作数时, EU 首先向 BIU 发出请求 并提供操作数的有效地址 由 BIU 形成 20 位物理地址并访问内存或 I/O 端口 读取操作数送给 EU 使用。EU 和 BIU 间既配合又相对独立, 称并行流水线方式, 大大提高 CPU 执行程序速度。

3 寄存器结构。

在 CPU 中采用通用寄存器暂时存放操作数, 可以提高程序执行的速度。一般来说, CPU 中包含的通用寄存器越多, 使用就越灵活, CPU 执行程序的速度也就越快。

由图 2.2 可见, 8086/8088 CPU 内部含有 14 个 16 位寄存器, 根据它们的使用特点和功能的不同, 可分为通用寄存器组、指示器、变址寄存器和段寄存器组, IP 和 FR 属于控制寄存器类。

现在介绍如下:

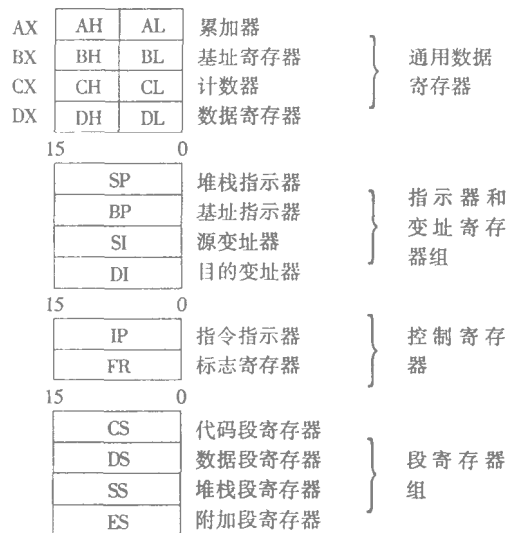


图 2.2 8086/8088 CPU 的内部寄存器

(1) 通用寄存器组

通用寄存器组包括 AX、BX、CX、DX 四个 16 位寄存器 主要用来保存算术、逻辑运算的操作数、中间结果和地址。它们既可以作为 16 位寄存器使用 也可以作为两个 8 位寄存器使用, 这时每个寄存器高字节和低字节的名称分别是: 低字节时用 AL、BL、CL、DL 表示 高字节时用 AH、BH、CH、DH 表示。这些寄存器的双重性使得 8086/8088 CPU 能很容易地处理字节和字数据。通用寄存器具有良好的通用性, 对任何指令它们都具有相同的功能。但是为了缩短指令代码的长度, 某些通用寄存器又规定了专门的用途。例如, 在字符串处理指令和移位指令中约定必须用 CX 作为计数器名、存放串的长度和移位次数。这样在指令中就不必给出 CX 寄存器名 缩短了指令长度 简化了指令的书写形式 这种使用方式 称为“隐含寻址”。隐含寻址实际上就是给某些寄存器规定一些特殊用法 事先

约定好，不直接在指令中表现出来，程序设计者编程时必须遵循这些规则。同样道理，AX、BX、DX 寄存器又分别称为累加器、基址寄存器和数据寄存器。表 2.1 列出了 8086/8088 CPU 中通用寄存器的特殊用途和隐含性质。

表 2.1 8086/8088 CPU 中通用寄存器的特殊用途和隐含性质

寄存器名	特殊用途	隐含性质
AX, AL	在输入输出指令中作数据寄存器用	不能隐含
	在乘法指令中存放被乘数或乘积, 在除法指令中存放被除数或商	隐含
AH	在 LAHF 指令中, 作目标寄存器用	隐含
AL	在十进制运算指令中作累加器用	隐含
	在 XLAT 指令中作累加器用	隐含
BX	在间接寻址中作基址寄存器用	不能隐含
	在 XLAT 指令中作基址寄存器用	隐含
CX	在串操作指令和 LOOP 指令中作计数器用	隐含
CL	在移位/循环移位指令中作移位次数计数器使用	不能隐含
DX	在字乘法/除法指令中存放乘积高位或被除数高位或余数	隐含
	在间接寻址的输入输出指令中作地址寄存器用	不能隐含
SI	在字符串运算指令中作源变址寄存器用	隐含
	在间接寻址中作变址寄存器用	不能隐含
DI	在字符串运算指令中作目标变址寄存器用	隐含
	在间接寻址中作变址寄存器用	不能隐含
BP	在间接寻址中作基址指针用	不能隐含
SP	在堆栈操作中作堆栈指针用	隐含

(2) 指示器和变址寄存器组

指示器和变址寄存器包括 SP、BP、SI、DI 四个 16 位寄存器。它们主要是用来存放（指示）操作数的偏移地址，其中：

SP 为堆栈指示器。SP 中存放的是当前堆栈段中栈顶单元的偏移地址。堆栈操作指令 PUSH 和 POP 就是从 SP 中得到操作数的段内偏移地址的。

在 RAM 区内按‘后进先出’的原则来存取数据的一组连续的存储单元称作堆栈。往堆栈中存入数据（又称压入或推入数据）时，新数据依次地堆放在老的数据之上，见图 2.3(a)。

往堆栈中压入的数据 2B、F0、31、4E 等是依次一个个地往上堆放的，最后压入的数据是 7C，它所在的单元地址是 2003。我们把最上面的数据所在的存储单元这个位置叫做堆栈的栈顶。栈顶单元的地址实际上是堆栈段存储器的段内偏移地址。存放在堆栈指针 SP 中的图中的栈顶单元地址 2003 即存于 SP 之中。换句话说 SP 指向栈顶，现在的栈顶是 2003 单元。如果继续往堆栈中压入数据，每压入一个字节数据，栈顶的位置就会向上移动一个单元，SP 中的地址会自动减 1。推栈指针 SP 始终指向栈顶。从堆栈中取出数据（又