

21世纪高等理工科重点课程辅导丛书

微型计算机原理 及应用学习指导

赵邦信 林 嵘 编著



化学工业出版社

21 世纪高等理工科重点课程辅导丛书

微型计算机原理及应用学习指导

赵邦信 林 嵘 编著



化学工业出版社

· 北 京 ·

本书是普通高等教育“十一五”国家级规划教材《微型计算机原理及应用（第二版）》（侯晓霞、王建宇、戴跃伟编著）的配套辅导教材。

全书共分为10章，前9章与主教材的前9章对应，每章的内容均按基本要求、内容体系、重点及难点分析、典型例题解析的形式组织，第10章为综合应用及考研试卷与解答，给出了综合应用例题及南京理工大学近年硕士研究生入学试卷及分析、解答。

本书既可作为高等院校微型计算机原理及应用、微机原理与接口技术等课程的辅导教材，也可以作为硕士研究生入学考试相关课程的参考书，还可作为从事相关科研工作的工程技术人员的参考材料。

图书在版编目（CIP）数据

微型计算机原理及应用学习指导/赵邦信，林嵘
编著. —北京：化学工业出版社，2010.7
（21世纪高等理工科重点课程辅导丛书）
ISBN 978-7-122-08724-9

I. 微… II. ①赵…②林… III. 微型计算机-高等学校-教学参考资料 IV. TP36

中国版本图书馆 CIP 数据核字（2010）第 102211 号

责任编辑：郝英华

责任校对：周梦华

装帧设计：杨 北

出版发行：化学工业出版社（北京市东城区青年湖南街13号 邮政编码100011）

印 刷：北京永鑫印刷有限责任公司

装 订：三河市万龙印装有限公司

787mm×1092mm 1/16 印张14 字数376千字 2010年8月北京第1版第1次印刷

购书咨询：010-64518888（传真：010-64519686） 售后服务：010-64518899

网 址：<http://www.cip.com.cn>

凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定 价：28.00 元

版权所有 违者必究

前 言

2001年,根据微型计算机原理及应用课程的教学要求,我们组织编写了《微型计算机原理及应用》教材,并于2007年进行了修订,且第二版被评为普通高等教育“十一五”国家级规划教材。

两版教材先后被国内多所院校选作相关专业同类课程的本科生教材,得到了广大师生的广泛好评。

在多年的教学过程中,许多老师、学生反映,在有限的课程学时内,对教材中的各部分内容能够基本接受,但对各部分内容之间的相互关系,不能很好地理解,特别是面对具体习题时,不能很好地入手,有效地解决。

针对上述矛盾,我们觉得有责任将多年来,通过对相关问题的理解、思考形成的体会,总结成教学参考书的形式,提供给大家。

本书旨在帮助学生学习主教材基本概念和基本理论的基础上,能够把握课程的整体内容体系,充分掌握微型计算机系统各部分之间的相互关系,并能够从解决具体问题的角度,进一步巩固所学知识,发现问题、解决问题,顺利通过学期考试和研究生入学考试。

本书共分10章,其中前9章与主教材的前9章对应,每章内容包括基本要求、内容体系、重点及难点分析、典型例题解析4个部分。

基本要求部分,简明扼要地列出本章内容的基本要点。

内容体系部分,说明本章内容所涉及的问题,指出相关问题与其他问题及课程整体内容之间的相互关系。

重点及难点分析部分,对本章内容所涉及的核心问题,进行深入的阐述和分析。

典型例题解析部分,说明如何利用基本概念和基本理论,来解决具体问题的方法。

书的第10章为综合应用及考研试卷与解答,给出了综合应用例题及南京理工大学近年硕士研究生入学试卷及分析、解答。

本书是作者结合多年来在课堂教学过程中对相关问题的理解和体会写成的,其内容体系完整,结构关系清晰,叙述通俗易懂。

本书每章的基本要求、内容体系、重点及难点分析部分由赵邦信撰写,典型例题解析的大部分由林嵘撰写,最后由赵邦信统稿。侯晓霞作为主教材的主编对本书的编撰工作提出了非常重要的思路和建议,在此对侯老师表示衷心的感谢。

由于编著者水平有限,书中难免存在不妥之处,诚请专家和广大读者给予斧正。

编著者
2010年6月

目 录

第 1 章 微型计算机概述	1
1.1 基本要求	1
1.2 内容体系	1
1.3 重点及难点分析	2
1.4 典型例题解析	15
第 2 章 8086/8088 指令系统	22
2.1 基本要求	22
2.2 内容体系	22
2.3 重点及难点分析	23
2.4 典型例题解析	36
第 3 章 汇编语言程序设计	45
3.1 基本要求	45
3.2 内容体系	45
3.3 重点及难点分析	46
3.4 典型例题解析	55
第 4 章 存储器系统	66
4.1 基本要求	66
4.2 内容体系	66
4.3 重点及难点分析	68
4.4 典型例题解析	78
第 5 章 定时与计数	85
5.1 基本要求	85
5.2 内容体系	85
5.3 重点及难点分析	86
5.4 典型例题解析	90
第 6 章 输入输出控制	97
6.1 基本要求	97
6.2 内容体系	97
6.3 重点及难点分析	99
6.4 典型例题解析	117
第 7 章 串并行通信及其接口技术	126
7.1 基本要求	126
7.2 内容体系	126
7.3 重点及难点分析	128
7.4 典型例题解析	139
第 8 章 总线技术	150

8.1 基本要求	150
8.2 内容体系	150
8.3 重点及难点分析	151
8.4 典型例题解析	154
第9章 D/A、A/D 转换与接口技术	157
9.1 基本要求	157
9.2 内容体系	157
9.3 重点及难点分析	158
9.4 典型例题解析	166
第10章 综合应用及考研试卷与解答	171
10.1 综合应用实例及分析与解答	171
10.2 硕士研究生入学考试试卷一	181
10.3 硕士研究生入学考试试卷一分析与解答	185
10.4 硕士研究生入学考试试卷二	191
10.5 硕士研究生入学考试试卷二分析与解答	195
10.6 硕士研究生入学考试试卷三	202
10.7 硕士研究生入学考试试卷三分析与解答	206
附录 有关辅助芯片的引脚信号与操作	214
参考文献	216



第1章 微型计算机概述



1.1 基本要求

- ① 了解微型计算机与计算机之间的相互关系；
- ② 了解微处理器及微型计算机系统的发展过程和性能特点；
- ③ 掌握微机系统的组成、结构及层次关系；
- ④ 掌握 Intel8086/8088 微处理器的编程结构、部分引脚及所传送的信号；
- ⑤ 了解并初步掌握微机系统中内部存储器和 I/O 端口的组织；
- ⑥ 理解程序存储、程序控制的概念，掌握微机系统的运行过程；
- ⑦ 了解时序的概念，掌握最小模式下总线读/写操作的基本时序，了解一些特殊过程的时序。



1.2 内容体系

本章从系统总体的角度，概括性地介绍了微型计算机系统的组成、结构及工作过程，无论从微机系统本身，还是从学习该课程的角度来看，本章所涉及的内容，对后续章节内容的学习和理解，都会起到纲领性的作用。

也就是说，如果不建立微机系统组成、结构及工作过程的整体概念，就无法领会和理解微机系统各组成部分之间的相互关系及它们的功能和作用，就不能进一步深入理解本课程所涉及的其他概念。

本章从微型计算机与计算机之间的相互关系出发，介绍了微机系统的特点，需要注意的是，所有这些特点都是由于微机系统采用了大规模及超大规模集成电路器件的缘故。

与计算机系统一样，微型计算机系统也必须由硬件系统和软件系统两大部分组成，其中硬件系统同样包括运算器、控制器、存储器及输入和输出设备五大部分，不同地是，微机系统中的运算器、控制器被做在了一片集成电路芯片中，就是微处理单元 MPU (Micro Processor Unit) 或中央处理单元 CPU (Central Processing Unit)，而存储器部分则演变成了包括高速缓冲存储器 (Cache)、主存储器和辅助存储器的三级存储体系结构。

微型计算机硬件系统各个组成部分之间的相互连接关系，就是微机系统的结构。微机系统采用总线结构，即利用一组公共的信号线，连接组成微机系统的各个功能部件，采用总线结构，可以把各功能部件之间的连接关系，转化为各功能部件与总线之间的连接关系，从而使微机系统具有配置灵活的结构特点。

由于微机系统采用总线结构，所以有关总线的概念、分类，总线在系统中的位置，总线的功能、作用及总线的标准等问题，对于本课程的学习和理解，都显得非常重要。

微机系统是由电子器件构成的信息处理工具，其本身的组成和结构呈现出一定的层次关系，而本课程的内容，正是按照这种层次关系，采用从核心到外围，即按 CPU→内部存储器→I/O 接口→外部 I/O 设备（包括外部存储器）的顺序来安排的。

处于微机系统不同层次的各个组成部分,各具有不同的特点,系统对它们的功能和性能要求,也存在着很大的差异。基本的情况是,越处于核心的部分,其组成、结构越具有规范性,对整个系统的功能和性能越具有决定性。如 CPU 是整个微机系统的核心,是系统中必不可少的部分;反之,越是处于外围的部分,其性能的离散性就越大,越可以由用户灵活配置,如微机系统中的外部 I/O 设备,完全可以由用户根据实际需要,选择是否配置,还可以由用户选择不同的类型。

按照上述安排,本章首先以 Intel 公司生产的 16 位微处理器 8086 及准 16 位微处理器 8088 为背景,详细介绍微处理器的内部组成、结构及外部引脚。

微处理器的内部结构又称为编程结构,直接与其内部数据处理过程相关,只有掌握了微处理器的编程结构,才能理解指令的功能,才能利用指令进行程序设计;微处理器外部引脚的重点,则在于这些引脚与芯片外部的数据传送,只有掌握了微处理器外部引脚及其所传送的信号,才能进行微机应用系统的硬件设计。

微机系统的工作过程不仅与系统的核心,即微处理器直接相关,还与系统中的内部存储单元及 I/O 端口的组织方式有关,为进一步介绍微机系统的工作过程,本章简要介绍了系统中内部存储单元及 I/O 端口的组织及读/写控制问题,涉及一系列基本概念,对于理解微机系统的组成、结构特别是微机系统的工作过程,是至关重要的。

在此基础上,结合冯·诺依曼有关程序存储及程序控制概念,对微机系统取指令→分析指令→执行指令的工作过程进行深入理解,是学习和理解本课程的关键。

另外,本章还对微机系统运行过程中,各种信号及动作之间的时间顺序,即时序问题,进行了简单介绍。

系统运行的过程就是执行程序的过程,而程序是由指令系统中的指令组成的,所以系统在执行一段程序的过程中,各种动作的顺序,就是组成该程序段的各条指令执行过程时序(指令周期)的组合;另一方面,指令的执行过程,包括若干总线操作,相应地,指令执行过程的时序(指令周期),则是由所包含的总线操作时序(总线周期)组合而成的,而系统运行过程所涉及的总线周期的种类数只有有限的 4 种,即存储器读/写周期和 I/O 端口的读/写周期。

所以,如果掌握了 4 种总线周期及所对应的总线操作,又能够把不同指令周期分解为这些总线周期的组合,则相当于掌握了完整的指令周期;如果掌握了指令系统中每条指令所对应的指令周期及程序中所包含的指令及组合方式,就可以完全掌握微机系统执行任意程序的过程中,相应的时序关系。

《微型计算机原理及应用》(第二版)中介绍的总线空闲、中断响应、总线响应和恢复及系统复位等,属于系统运行过程中的一些特殊过程,或可认为是系统执行两条指令之间的过渡过程,当然也对应着一定的时序关系。

1.3 重点及难点分析

1.3.1 计算机与微型计算机

计算机是一种能够自动、快速、准确地进行信息处理的电子工具。计算机所进行的信息处理,包括信息的采集、输入、传输、存储、运算、输出、显示和控制等多个方面,上述过程中,涉及的设备、方法等,都属于计算机技术的范畴。

计算机能够处理及处理之后产生的信息,既可以是数据、符号信息,也可以是声音、图形、图像或其他类型的信息。

由于计算机是利用电子器件的不同状态来区分和表示不同信息,而利用电子器件的状态在程序控制作用下的快速转换来进行信息处理的,因而与传统的信息处理过程及信息处理工具,包括人脑相比较,具有信息处理速度快、实时性能好、能保存及处理大量不同类型信息等优点。

计算机是进行信息处理的电子工具,因而其组成、结构必须满足信息处理的功能要求,具体来说,计算机必须由硬件系统和软件系统两大部分组成。

计算机的硬件系统,是构成计算机这种信息处理工具的物理实体,是具有信息的输入、存储、运算、输出及能够对整个信息处理过程进行协调控制的功能部件的组合,包括输入设备、存储器、运算器、输出设备和控制器五大部分。

计算机软件则是自成体系的所有程序及在程序设计时所形成的文档资料的总和,实质上是人们以程序的形式为计算机安排的信息处理规则和方法的组合。

总之,计算机的硬件,建立了计算机应用的物质基础,是整个系统的躯体,而软件则给躯体加入了思维的功能,硬件与软件的有机结合,才是完整的计算机系统。

计算机是一种电子工具,它是由电子器件构成的,因而电子技术的发展、新型电子器件的不断出现,是计算机技术不断发展的强大推动力。从1946年世界上出现第一台电子计算机ENIAC(Electronic Numerical Integrator And Calculator)到目前为止,先后经历了电子管、晶体管、中、小规模集成电路和大规模及超大规模集成电路等阶段。

微型计算机是大规模及超大规模集成电路计算机,属于第四代计算机。微型计算机系统结构上的独特特点,是把构成计算机硬件系统的两个最核心的部分,即运算器和控制器,集成到一片集成电路芯片上,这个芯片就是中央处理单元CPU或微处理单元MPU,又称为微处理器。

集成电路是一种包含成千上万甚至更多基本电路单元的新型电子器件,是能够按照构成电子、电气电路、设备的要求,利用特殊的工艺,在体积很小的半导体芯片中制作出来的,微机系统中的CPU及许多部件,都采用了集成电路器件。

与采用分列元件相比,采用集成电路器件来构成计算机硬件系统的直接结果,可以从两个方面来表述,即在功能要求相同的情况下,整个计算机系统的体积、重量、功耗将会大幅度减小或降低;而在体积、重量、功耗相同的情况下,计算机系统的性能将会大幅度提高。

微型计算机是大规模集成电路的产物,由于采用了大规模或超大规模集成电路器件,使得微型计算机除了具有一般计算机的优点之外,还具有体积小、重量轻、价格低廉、组态灵活、可靠性高、应用面广等方面的特点。

1.3.2 微型计算机系统的组成、结构及层次关系

微型计算机硬件系统由中央处理器CPU(又称微处理器MPU)、内部存储器,并通过I/O接口连接必要的外部I/O设备(包括外部存储器)组成,图1-1是微型计算机硬件系统的基本组成和结构。

图1-1中,中央处理器CPU是整个微机系统的核心,它能够按照程序代码的要求,实现对整个系统运行过程的控制,并完成相应的信息处理任务;内部存储器通常由半导体器件担当,有RAM、ROM两种不同的类型,用以存放程序、数据的代码及系统运行处理的结果;CPU和内存构成了微机系统的主机,I/O接口则是主机与外部I/O设备之间的连接电路,通过输入接口,主机才能接收来自于外部输入设备提供的输入信息;通过输出接口,主机才能把经过处理之后产生的实时信息,提供给外部输出设备。

微型计算机系统采用总线结构形式,即利用一组公共信号传输线,实现各功能部件之间的相互连接,进而实现微机系统运行过程中,各种不同类型信息的实时传送。

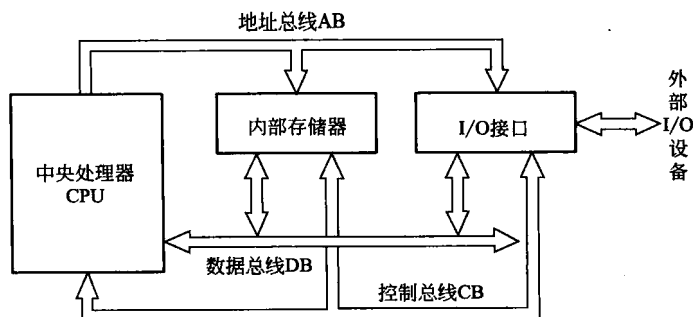


图 1-1 微型计算机系统的基本组成与结构

按所传送信息类型的不同，连接微机系统各功能部件的总线可以分成三种不同的类型：地址总线、数据总线和控制总线。

① 地址总线 AB (Address Bus) 地址总线是微机系统用以传送地址信号的连线。地址总线的位数决定了微机系统可以直接寻址的内存单元、I/O 端口的数量。例如，8086/8088CPU 的地址线共 20 位，全部用以寻址内存单元时，可寻址内存单元的数量为 $2^{20} = 1\text{MB}$ ；一般情况下，系统利用地址总线中的低 16 位来寻址 I/O 端口，则可以寻址的 I/O 端口数量为 $2^{16} = 64\text{KB}$ 。

地址总线为单向、三态总线。所谓单向，是指地址信号只能由 CPU 发出，去寻址内存单元或 I/O 端口；三态是指地址总线除了高、低电平两种状态之外，还可以处于高阻抗状态，高阻抗状态即所谓的隔离状态。

② 数据总线 DB (Data Bus) 数据总线是微机系统用以传送数据的信号线，数据总线的位数决定了微机系统一次可以并行传送二进制数码的数量，一般情况下，等于相应微处理器的字长，如 8086CPU 有 16 位数据线，表明 8086CPU 可以同时进行 16 位二进制数码的传送，即 8086CPU 的字长为 16 位。

数据总线均为双向三态总线，即数据信息，可以分时地由 CPU 传送给内存单元、I/O 端口或由内存单元、I/O 端口传送给 CPU。

③ 控制总线 CB (Control Bus) 控制总线用以传送 CPU 对内存单元、I/O 端口的控制信息，也可用以获取内存单元、I/O 端口传送给 CPU 的状态信息。

控制总线中的各种信号线，情况比较复杂，有的为输入、有的为输出，有的是单向、有的是双向，有的是三态、有的为非三态。

微机系统的各个组成部分，在系统运行过程中，具有不同的功能和作用，有些部分是必不可少的，有些部分则是可以灵活配置的，甚至是可有可无的，按照这种关系，整个系统的结构表现出明显的层次性。图 1-2 是微型计算机系统从核心到外围的层次关系。

由图 1-2 可见，中央处理器 CPU 包括运算器和控制器两部分，是微机系统的核心，是系统中必不可少的部分，CPU 的类型、性能对整个系统起决定性的作用。

CPU 与内部存储器，构成微机系统的主机，其中内部存储器可根据系统的功能要求，选择不同类型、不同规格的存储器件灵活配置。

主机通过 I/O 接口电路，连接必要的外部 I/O 设备（包括外部存储器），构成微机的硬件系统，其中，外部存储器和外部 I/O 设备的配置更加灵活。

在一定硬件配置的基础上，再加上必要的系统软件，就构成了商品微机系统，其中系统软件应该由生产厂商作为系统的一部分，随硬件系统一同提供给用户。

在商品微机系统的基础上，用户针对各自的使用目的和要求，研制、开发应用硬件和应

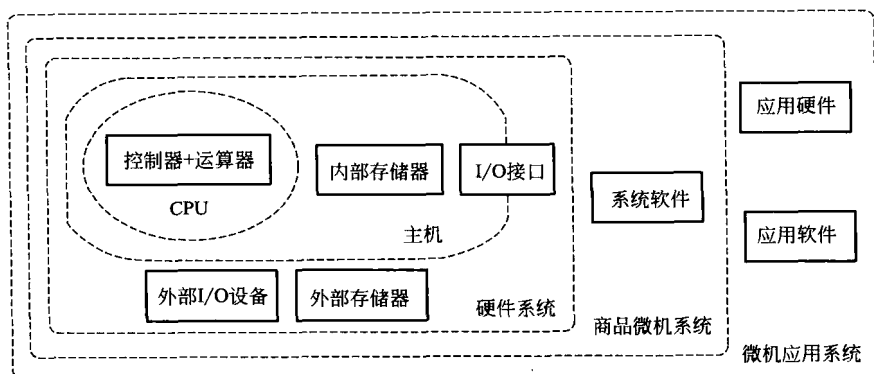


图 1-2 微型计算机系统的层次关系

用软件，就构成了微型计算机应用系统。

1.3.3 Intel8086/8088 微处理器的内部结构

(1) 简述

8086 是 Intel 公司于 1978 年推出的 16 位微处理器产品，其内部数据处理及与片外的数据传送，都可以 16 位同时进行。

为保持与前期 8 位微处理器产品的兼容性，1979 年，Intel 公司还生产了一种准 16 位的微处理器产品 Intel8088，它与 8086 的内部结构基本相同，但对外的数据总线只有 8 条，即 8088CPU 内部，可以同时处理 16 位二进制数码，但与芯片外部，每次却只能进行 8 位二进制数码的传送。

微处理器的内部结构，并非指 CPU 的物理结构和各个组成部分的布局，而是程序员进行程序设计时，所要关注的组成和结构，称之为编程结构。

图 1-3 是 8088 微处理器的内部编程结构，可以看出，CPU 由总线接口部件 BIU (Bus Interface Unit) 和执行部件 EU (Execution Unit) 两大部分组成。

(2) 总线接口部件

总线接口部件 BIU，负责 CPU 与芯片外部之间的数码传送，包括取指令代码及存、取数据代码，由以下几个部分组成。

① 20 位地址加法器 为实现对内存单元及 I/O 端口的寻址，以便进行读/写操作，BIU 必须向外提供 20 位物理地址，而 CPU 内部并没有能够直接存放 20 位物理地址的寄存器，只有一些 16 位的寄存器。解决的方法是，在进行读/写操作时，由两个 16 位的地址（即段地址和偏移地址）通过一定的运算，临时形成所需要的 20 位物理地址，能够实现上述运算的功能部件，就是 BIU 中的 20 位地址加法器。

② 段寄存器 BIU 中共有 4 个 16 位的段寄存器，即代码段寄存器 CS、数据段寄存器 DS、扩展段（附加段）寄存器 ES、堆栈段寄存器 SS。

段寄存器用以存放并向 20 位地址加法器提供所需的 16 位段地址，需要注意的是，CPU 访问内存的不同区域，或按不同的方式访问内存时，需由不同的段寄存器提供段地址。

③ 指令指针寄存器 IP 16 位的指令指针寄存器 IP，用以存放指令代码所在单元的偏移地址，与代码段寄存器 CS 的内容，共同形成指令代码所在存储单元的 20 位物理地址。

④ 指令队列 为提高系统的运行速度，8088CPU 在 BIU 中设置了 4 个字节的指令队列寄存器，简称指令队列，用以暂存取自于内存代码段的程序代码，以实现取指令及分析、执行指令的并行操作。与 8088CPU 不同的是，8086CPU 中有 6 个字节的指令队列寄存器。

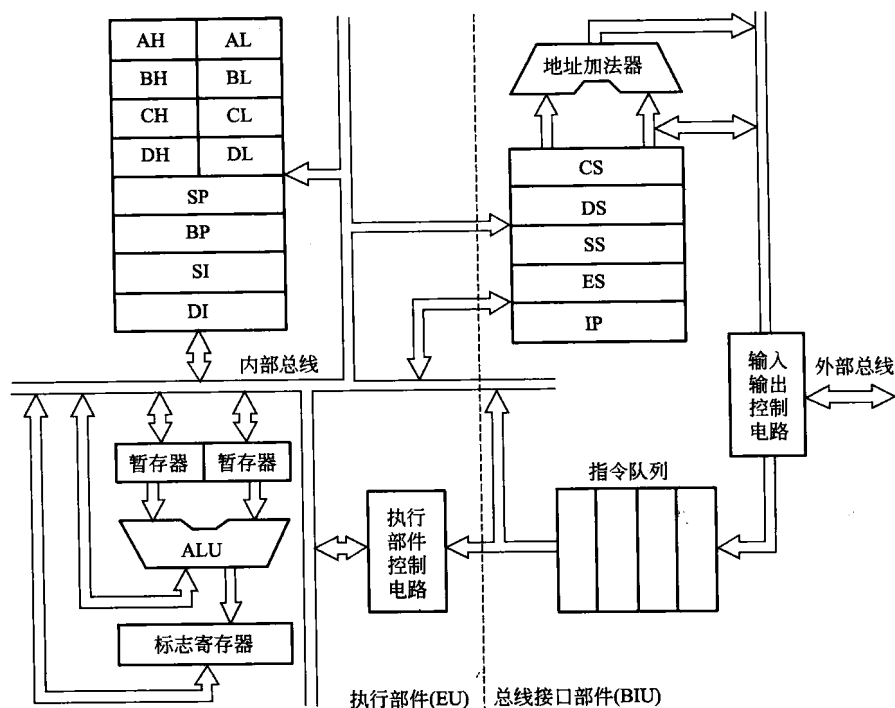


图 1-3 8088 微处理器的内部编程结构

(3) 执行部件

执行部件 EU，负责对指令的代码进行译码分析，并根据结果，协调整个系统执行相应的功能，执行部件由下列几个部分组成。

① 算术逻辑单元 ALU (Arithmetic Logic Unit) 是 CPU 中的运算部件，一方面，ALU 能够进行算术、逻辑运算及数据的传送和移位操作；另一方面，它能够提供运算结果的标志，给标志寄存器。

② 通用寄存器组 EU 中设置了 4 个 16 位的通用寄存器，即累加寄存器 AX、基址寄存器 BX、计数寄存器 CX、数据寄存器 DX，除了某些指令中，规定了特殊用途之外，通用寄存器可由程序员灵活支配使用。

4 个 16 位的通用寄存器，都可以被分为高、低两个 8 位寄存器独立使用。例如，累加器 AX 可分为 AH、AL 两个 8 位的通用寄存器，基址寄存器 BX 可分为 BH、BL 两个 8 位的通用寄存器，其余类推。

③ 专用寄存器组 EU 中有 4 个 16 位的专用寄存器，分别是基址指针寄存器 BP、堆栈指针寄存器 SP、源变址寄存器 SI、目的变址寄存器 DI。一般情况下，专用寄存器用以存放指定类型的信息，当然，在不冲突的情况下，也可由程序员支配和使用。

④ 标志寄存器 FR (Flag Register) 是 EU 中的 16 位标志寄存器，其内容即为处理器状态字 PSW (Processor Status Word)，共包含 9 个标志位，用以保存 CPU 运行结果的状态信息，或人们对 CPU 运行方式的控制信息，相应地，标志寄存器中的标志可以分为状态标志和控制标志两类。

状态标志反映了 ALU 运行到当前时刻微处理器的某些状态，是 ALU 进行逻辑分析、判断并进而实现程序转移的依据。状态标志共 6 个，分别是符号标志 SF、零标志 ZF、奇偶标志 PF、进位标志 CF、辅助进位标志 AF 和溢出标志 OF。

控制标志，可由程序员通过指令设置，用以规定 CPU 执行某种操作时的方式，体现了人们对 CPU 运行方式的可干预性。控制标志共 3 个，它们分别是方向标志 DF、中断标志 IF 和跟踪标志 TF。

处理器状态字 PSW 中的各个标志位，在标志寄存器中的安排，如图 1-4 所示，各个标志的功能和作用，如表 1-1 所示。

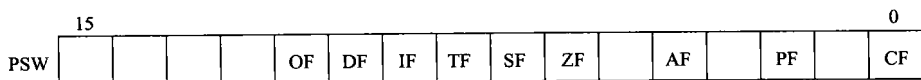


图 1-4 PSW 各位的安排

表 1-1 PSW 中各标志位的功能和作用

状态标志	置位的条件	控制标志	置位时规定微处理器采用的工作方式
SF	运算结果的最高位为 1	DF	串操作按反向方式进行
ZF	运算结果为 0		
PF	运算结果的低 8 位中,1 的个数为偶数	IF	允许 CPU 响应外部可屏蔽中断
CF	运算结果有进位或借位		
AF	字节运算的结果,低 4 位有向高 4 位的进位或借位	TF	使 CPU 按跟踪方式执行指令
OF	运算结果有溢出		

(4) BIU 与 EU 之间的动作协调

BIU 与 EU 是 8086 及 8088CPU 的两大组成部分，它们在系统的运行过程中既相互独立又相互配合，使得 CPU 可以在执行当前指令的同时，取后续指令的代码，从而使 CPU 的工作效率能够得到大幅度的提高。

BIU 和 EU 相互协调工作的原则，可以归纳为下列几个要点。

① EU 总是从指令队列的前端，顺序获取指令代码并分析、执行，指令队列中的后续代码依次前移。

② BIU 保证指令队列中有 4 个（8086）或 3 个（8088）以上字节的指令代码，即若指令队列中有 2 个（8086）或 1 个（8088）以上空字节时，BIU 就会自动将内存代码段中的后续指令代码取到指令队列中，以保证指令执行过程的连续性。

③ 若 EU 对当前指令代码的分析表明有访问内存或 I/O 端口的总线操作要求，则 BIU 在取完当前字节的指令代码后，进行相应的总线操作。

④ 指令队列中的字节数满足要求，且当前指令没有存/取数据的总线操作请求时，BIU 处于空闲状态。

⑤ 在执行转移、调用或返回类指令时，指令队列被自动清除，而装入由上述三类指令所指向的另一区域中的指令代码。

1.3.4 Intel8086/8088 微处理器的工作模式和外部引脚

(1) 最小模式与最大模式的概念

微处理器是构成微机应用系统的核心，是以集成电路芯片的整体形式，由专业的制造厂商研制的。在不同的应用场合下，针对不同的性能要求，微机系统的组成、结构存在着很大的差别，因而对 CPU 的性能要求也不尽相同。

Intel 公司在研制 8086/8088 微处理器芯片时，充分考虑了各类用户对其性能的不同要

求, 允许用户通过硬件设置使 CPU 工作在最小与最大两种不同的模式下。

当 CPU 的第 33 引脚, 即 $\text{MN}/\overline{\text{MX}}$ 引脚接高电平 (+5V) 时, CPU 处于最小模式; 反之, 当 $\text{MN}/\overline{\text{MX}}$ 引脚接低电平 (地) 时, CPU 处于最大模式。

最小模式用于构成小规模微机应用系统, 最小模式的微机应用系统中, 只配置单一的微处理器 (即 8086 或 8088), 系统工作过程中所需的总线控制信号均由 CPU 直接产生。

最大模式用于构成中、大规模的微机应用系统, 最大模式的微机应用系统中, 必须配置总线控制器 8288, 由其按照主处理器的要求, 产生系统工作过程中所需的各种总线控制信号。

最大模式下的微机应用系统中, 除由 8086 或 8088 作为主处理器之外, 还可以配置协处理器, 如用于数值运算的 8087 及用于输入/输出管理的 8089 等。

(2) Intel8086/8088 的外部引脚

8086/8088CPU 都是双列直插式集成电路芯片, 如图 1-5 所示, 是 8086/8088 微处理器的外部引脚分布图, 由图 1-5 可见, 它们都有 40 个引脚, 相互之间只在个别引脚上存在差别。

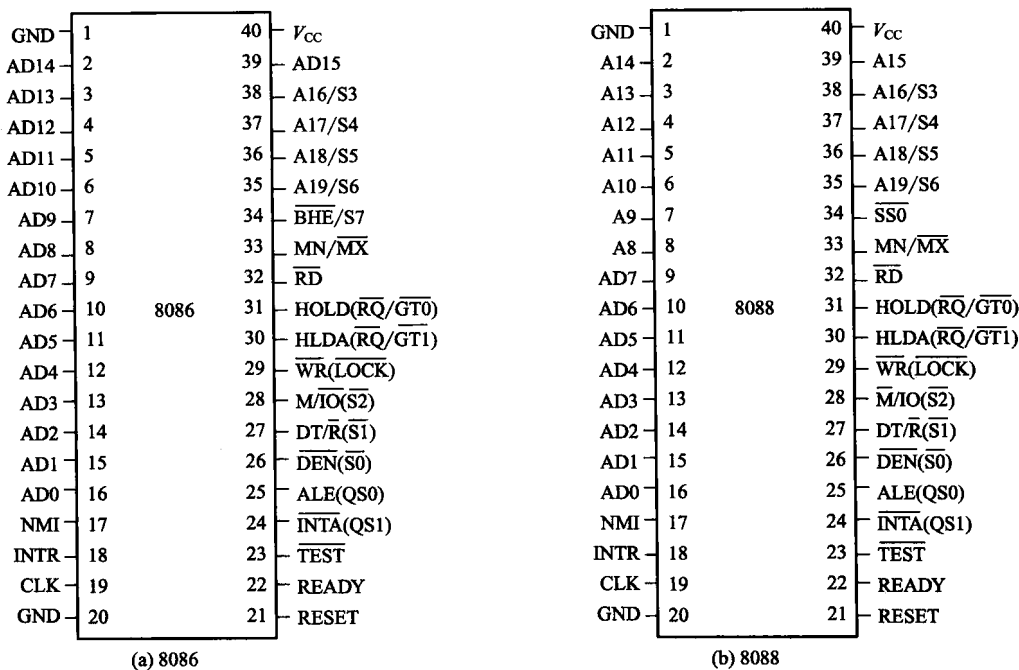


图 1-5 8086/8088CPU 的引脚分布图

需要注意的是, 两种微处理器芯片中的第 24~31 共 8 个引脚, 在不同的工作模式下, 名称及传送的信号是不相同的。下面, 我们分类对 8086/8088CPU 芯片中, 比较重要的引脚及所传送的信号进行介绍。

① 与地址、数据及状态信号有关的引脚 包括 AD15~AD0、A19/S6~A16/S3、ALE 及 $\text{BHE}/\text{S7}$ 等, 其中 AD15~AD0 为地址/数据信号输入/输出引脚, 分时输出低 16 位地址信号及输入/输出 16 位数据信号; A19/S6~A16/S3 为地址/状态信号输出引脚, 分时输出高 4 位的地址信号及 CPU 内部的状态信号。

由于 8088CPU 只有 8 位数据线, 所以只需用 AD7~AD0 引脚, 分时输出地址信号及进行数据信号的输入/输出, 而 A15~A8 只用以输出中 8 位地址信号。

受芯片集成度的限制, 8086/8088CPU 的部分引脚采用分时复用的机制, 即利用相同的引脚, 在不同的时段内传输不同的信号, 如上述地址/数据及地址/状态信号的分时传送。为支持地址/数据及地址/状态引脚的分时复用, 需把先行输出的地址信号锁存到外部锁存器中去, 而 CPU 的 ALE 引脚, 则用以输出上述锁存操作所需要的选通信号。

8086CPU 有 16 位数据线, 能够同时进行 16 位数据的传送, 当然, 也可以只利用其中的高、低 8 位实现 8 位数据的传送, $\overline{\text{BHE}}/\text{S7}$ 引脚中的 $\overline{\text{BHE}}$ 若为有效的低电平, 表明某次数据传送过程中, 高 8 位数据总线上正驱动着有效的数据, 而该引脚分时输出的状态信号 S7, 没有任何实际意义。

对于 8088CPU 来说, 由于其只有 8 位数据总线, 因而不存在 $\overline{\text{BHE}}/\text{S7}$ 引脚, 代之以 $\overline{\text{SSO}}$ 引脚, 由它和 $\text{DT}/\overline{\text{R}}$ 、 $\overline{\text{M}}/\text{IO}$ 一起决定 8088 芯片当前总线周期的读/写操作类型。

② 与读、写操作控制有关的引脚 包括 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{M}}/\text{IO}$, 其中 $\overline{\text{RD}}$ 输出低电平时, 表明要对由地址信号选中的内存单元或 I/O 端口进行读操作; $\overline{\text{WR}}$ 输出低电平时, 表明要对由地址信号选中的内存单元或 I/O 端口进行写操作; $\overline{\text{M}}/\text{IO}$ 则用以区分访问内存单元还是访问 I/O 端口, 需要注意的是, 8086CPU 的相应引脚为 $\text{M}/\overline{\text{IO}}$ 。

③ 与中断请求及响应有关的引脚 包括 NMI、INTR、 $\overline{\text{INTA}}$ 引脚, 其中 NMI、INTR 用以接收来自 CPU 外部的非屏蔽及可屏蔽中断请求信号, $\overline{\text{INTA}}$ 则用以向中断源输出中断应答信号。

④ 与总线请求及响应有关的引脚 最小模式下, 为 HOLD 和 HLDA 引脚, 其中 HOLD 用以接收来自于系统中其他总线设备的总线请求信号, CPU 通过 HLDA 引脚输出总线响应应答信号; 最大模式下, 为 $\text{RQ}/\overline{\text{GTI}}$ 、 $\text{RQ}/\overline{\text{GT0}}$ 两个引脚, 每个引脚都可以分时接收总线请求信号及输出总线响应应答信号。

⑤ 与外部总线收发器有关的引脚 包括 $\overline{\text{DEN}}$ 及 $\text{DT}/\overline{\text{R}}$, 若芯片通过总线收发器 8286 接系统数据总线, 则由 $\overline{\text{DEN}}$ 提供 8286 的启动信号, 而由 $\text{DT}/\overline{\text{R}}$ 引脚输出决定 8286 数据传送方向的控制信号。

⑥ 总线周期状态信号输出引脚 包括 $\overline{\text{S2}}$ 、 $\overline{\text{S1}}$ 、 $\overline{\text{S0}}$, 是最大模式下特有的引脚, 用以实现 CPU 与总线控制器 8288 的连接, 其不同组合, 可以使总线控制器 8288 产生不同操作所需要的总线控制信号。

除上述常用引脚外, 8086/8088CPU 还有复位信号输入引脚 RESET、时钟信号输入引脚 CLK 等, 这里不再赘述。

1.3.5 8086/8088 微机系统的内部存储器组织

(1) 简述

内部存储器是微机系统主机的重要组成部分, 用以存放程序及数据的代码。以 8086/8088CPU 为核心的微机系统中, 内存以字节 (Byte) 为单位形成存储阵列, 来保存数码信息。

CPU 对内存的读/写操作, 既可以字节 (Byte) 为单位来进行, 也可以双字节, 即字 (Word) 为单位来进行, 某些特殊情况下, 还可以双字 (DWord) 为单位来进行。

地址相邻的两个字节存储单元的内容, 可以合成一个字, 其中高位字节占高地址单元, 低位字节占低地址单元, 系统约定低地址单元的地址, 即为字存储单元的地址。为提高对字存储单元的读/写速度, 通常应使字存储单元从偶地址单元开始。

(2) 存储容量

8088/8086CPU 有 20 位地址信号输出引脚, 可以直接寻址的字节存储单元数为 $2^{20} = 1\text{MB}$, 即 512KW。

(3) 物理地址

8088/8086CPU 可直接寻址 1MB 的存储单元, 与这些存储单元一一对应的 20 位地址 00000H~FFFFFH, 称为存储单元的物理地址。

(4) 内部存储器的分段及段地址

8088/8086CPU 可直接寻址的存储单元数为 1MB。增大内存容量, 当然是提高系统性能的前提条件, 但在处理具体问题时, 直接面对这样一个大容量的存储空间, 显得有些不太方便, 为了解决这个矛盾, 系统中采用了存储器分段的方法。

所谓存储器分段, 就是根据存放信息的类型及读/写操作方式的不同, 把存储空间划分为若干子空间, 以便程序对各类不同信息进行存储、管理和使用。

以 8088/8086CPU 为核心的微机系统, 其内部存储器采用固定分段大小和浮动分段起始位置的处理方法。存储器每个分段所包含的单元数, 即段的大小是固定的, 为 16 位地址信号能够寻址的单元数 ($2^{16}=64\text{KB}$); 存储器分段的起始位置是浮动的, 只要某内存单元的 20 位物理地址中低 4 位全为 0 (或称能够被 10H 整除), 则该单元就可以作为一个存储器分段的起始单元。

(5) 内部存储单元的段地址与偏移地址

采用分段管理机制后, 某存储单元的位置可由其所在段起始单元的物理地址, 及该单元在段内与起始单元间间隔的单元数共同确定。

由于段起始单元物理地址的低 4 位恒定为 0, 所以只需关注其高 16 位地址, 称为相应存储器分段的段地址; 而把某存储单元在段内与起始单元之间间隔的单元数, 称为相应单元的段内偏移地址, 简称偏移地址, 或有效地址、地址位移量等。

把某存储单元的段地址及偏移地址, 写在一起, 中间用冒号隔开, 如 2000H:3000H, 称为相应存储单元的逻辑地址。其中, 段地址由 CPU 中的段寄存器提供, 而偏移地址可以是指令代码的一部分, 也可以来自于 CPU 中不同的 16 位寄存器, 还可以来自于内存中的字存储单元。

(6) 物理地址与逻辑地址的关系

由存储器分段及逻辑地址的概念可知, 存储单元的物理地址是由逻辑地址中的段地址与偏移地址共同决定的, 即

$$\text{物理地址(20 位)} = \text{段地址(16 位)} \times 10\text{H} + \text{偏移地址(16 位)}$$

例如, CPU 访问内存代码段时, 约定由代码段寄存器 CS 及指令指针寄存器 IP 提供段地址和偏移地址, 系统启动时 CS 及 IP 寄存器的初始内容分别为 FFFFH 和 0000H, 所以初始指令代码所在单元的物理地址为 FFFF0H。

通常, 系统在从 FFFF0H 单元开始的连续单元中, 固化一条无条件转移指令的代码, 可转移到实际待执行程序代码所在的起始单元。

(7) 段地址及偏移地址的匹配规则

CPU 要访问内存, 必须提供 20 位的物理地址, 而物理地址必须由 16 位的段地址和 16 位的偏移地址共同决定。

其中, 16 位的段地址, 由 CPU 总线接口部件中 4 个段寄存器中的某一个提供; 16 位偏移地址可以来自于指令中的立即数, 也可以来自于 CPU 中某个 16 位寄存器, 还可以来自于内部存储器的字存储单元, 取决于指令中存储器操作数的寻址方式。

系统访问内存时, 提供段地址的段寄存器及形成偏移地址的寻址方式, 有一定的匹配规则, 可以概括如下。

① 取指令代码时, 由代码段寄存器 CS 和指令指针寄存器 IP 分别提供段地址和偏移

地址。

② 栈操作时，由堆栈段寄存器 SS 和堆栈指针寄存器 SP 分别提供段地址和偏移地址。

③ 按非栈操作方式对栈区域进行读/写操作时，段地址由堆栈段寄存器 SS 提供，偏移地址必须由 BP 寄存器提供或参与提供。

④ 对数据段进行读/写操作时，段地址由数据段寄存器 DS 提供，偏移地址由指令直接提供或由 BX、SI、DI 寄存器提供或参与提供。

⑤ 串操作时，必须由 DS 及 SI 提供源串所在内存区域的段地址和偏移地址，而由 ES 及 DI 提供目的串所在内存区域的段地址和偏移地址。

⑥ 可以用指令超越前缀，明确提供段地址的段寄存器。

1.3.6 8086/8088 微机系统的 I/O 端口组织

(1) 接口及 I/O 接口

接口是微机系统各功能部件之间的连接电路，具有数据暂存、缓冲及信号类型变换和时序匹配等方面的功能，CPU 与 I/O 设备之间的接口，称为 I/O 接口。

(2) I/O 端口

I/O 端口是 I/O 接口电路中的一个（组）用以暂存输入/输出数据，且可由 CPU 访问的寄存器。微机应用系统中，可能同时配置多个 I/O 设备，即使是某特定的 I/O 接口，由于需要传送不同类型的信息，也可以设置多个 I/O 端口，所以微机应用系统中，I/O 端口的数量不止一个。

(3) I/O 端口地址

微机应用系统中，可能设置多个 I/O 端口，为区分不同的 I/O 端口，必须为每个 I/O 端口设置一个确定的地址，即为相应的 I/O 端口地址。

(4) 8086/8088 系统中 I/O 端口的数量

以 8086/8088CPU 为核心的微机应用系统中，约定以地址总线的低 16 位来寻址 I/O 端口，因此，整个系统可以寻址的 I/O 端口数为 $2^{16} = 64\text{KB}$ 个。

I/O 端口以字节为单位组织数据的存储，与内存单元的读/写操作方式相应，系统在进行端口的读/写操作时，支持把地址相邻的两个字节端口，组合成一个字端口（通常低位字节在偶地址端口，高位字节在奇地址端口），并规定低地址为字端口的地址。

(5) I/O 端口的访问

I/O 端口是 CPU 与 I/O 设备之间进行信息传送的桥梁，CPU 对端口的访问，有读和写两个基本方面。

为实现对 I/O 端口的读/写操作，必须解决两个方面的问题。其一，利用 CPU 发出的 $\overline{M}/\text{IO}(\overline{M}/\overline{\text{IO}})$ 信号，区分访问内存单元还是 I/O 端口；其二，利用读/写控制信号 ($\overline{\text{RD}}$ 、 $\overline{\text{WR}}$) 确定是对 I/O 端口进行读操作还是写操作。

1.3.7 微机系统运行过程及其分析

(1) 程序存储与程序控制

微机系统之所以能够在没有人工直接干预的情况下，自动完成各种信息处理任务，是因为人们事先为它编制了工作程序。微机系统的工作过程，就是执行程序的过程。

程序是由一条条指令组合而成的，对机器而言，指令是以二进制代码的形式出现的，程序的代码就是组成程序段所有指令的代码。把执行一项信息处理任务所对应的程序代码，以字节为单位，按顺序存放在内部存储器的一段连续的区域中，就是程序存储，而程序第一字节代码所在单元的地址，称为相应程序段的首地址，包括段地址和偏移地址。

把程序段首地址中的段地址和偏移地址分别送代码段寄存器 CS 及指令指针寄存器 IP