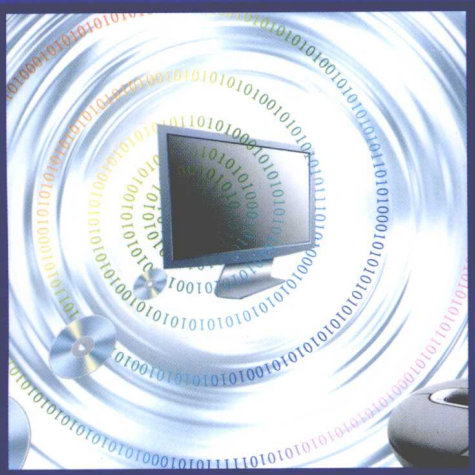
高职高专“十一五”规划示范教材

北京市高等教育精品教材立项项目



杨延双
魏坚华　编著
张晓冬

微机原理及汇编语言教程

（第2版）

北京航空航天大学出版社

高职高专“十一五”规划示范教材

北京市高等教育精品教材立项项目

微机原理及汇编语言教程

（第2版）

杨延双
魏坚华　编著
张晓冬

北京航空航天大学出版社

内 容 简 介

详细介绍 PC 系列的微型计算机原理与汇编语言。共 10 章，包括：Intel 系列处理器；汇编语言指令系统和汇编语言的程序设计；存储器；中断系统；输入/输出系统；总线技术及通信接口和常用外设接口。每章后面均有习题。书后有 7 个上机实验指导书，并在附录中给出了参考答案。第 2 版新增加内容体现了微机技术新发展和知识系统的完整性。总体上，本书内容丰富，深入浅出，注重实用，是面向高等职业学校而编写的，也可以作为非计算机专业本科教材及相关技术人员参考。

图书在版编目(CIP)数据

微机原理及汇编语言教程/杨延双，魏坚华，张晓冬编著. —2 版. —北京：北京航空航天大学出版社，2010.4

ISBN 978-7-5124-0054-2

Ⅰ.①微… Ⅱ.①杨…②魏…③张… Ⅲ.①微机—理论—高等学校—教材②汇编语言—程序设计—高等学校—教材 Ⅳ.①TP36②TP313

中国版本图书馆 CIP 数据核字(2010)第 058055 号

微机原理及汇编语言教程(第 2 版)

杨延双 魏坚华 张晓冬 编著

责任编辑 文幼章

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：bhpress@263.net 邮购电话：(010)82316936

北京市媛明印刷厂印装 各地书店经销

*

开本：787×1092 1/16 印张：17.75 字数：454 千字

2010 年 5 月第 2 版 2010 年 5 月第 1 次印刷 印数：4 000 册

ISBN 978-7-5124-0054-2 定价：29.00 元

第二版前言

本书的第一版《微机原理及汇编语言教程》自出版以来，得到了广大师生和读者的肯定，还收到了许多有益的反馈建议。该书第一版被评为“北京市高等教育精品教材”，更使得本书的修订工作得到广大师生和读者的关注和支持。他们的反馈信息为本书的修订提供了有利的帮助，在此深表感谢。

近年来我国更加重视高等职业教育的发展，本教材面向高等职业教育，特色明显，有利于学生对未知领域的掌握和运用。本书追求不断的完善，体现微型计算机技术的不断发展，以及知识体系的完整性，更好地满足教与学的要求。

本书第 2 版新增和修订的内容有：第 2 章增加 2.6 节是新一代微处理器 Itanium(安腾)；第 3 章 3.6 节增加的是 Pentium 的有关的内容：1)MMX 编程环境，2)MMX 指令操作数，3)MMX 技术指令；第 5 章增加 2 节：5.6 Pentium 程序设计举例，5.7 汇编语言和 C 语言的混合编程；第 6 章增加的是 6.4 节：闪速存储器(flash memory)；第 7 章修改了 7.4 节为：PC 机的中断处理，增加 7.4.3 节 I/O 控制中心的中断管理，7.4.4 高级可编程中断控制子系统；第 9 章增加的 1 节是 USB 总线；第 10 章全是新增加的：PC 机通信接口及常用外设接口。

为配合教学，提供电子版 PPT，为供教师备课参考及便于读者掌握知识要点。

本书第 2 版由杨延双拟定了编写内容和大纲，并统稿。新增的内容中 2.6 节由杨延双编写；3.6 节、5.6 节、5.7 节由张晓冬编写；6.4 节、7.4 节、9.6 节及第 10 章由魏坚华编写。全书由肖创柏教授审阅，并提供了宝贵意见。尹子赓、尹志忠、王万亭参加了书稿的资料整理和校对等工作。在本书的编写过程中，得到了蒋宗礼教授的大力支持和帮助。谨在此一并表示衷心的感谢。

由于作者水平所限，书中内容难免有不当或错误之处，敬请专家和广大读者批评指正。

编　者

2009 年 8 月

前　言

微机原理与汇编语言是学习和掌握计算机技术的重要内容。在传统的教学计划中将“微机原理”和“汇编语言程序设计”分开单独设课。而近年来高等职业教育蓬勃发展，由于其校情、师生、生源等与普通高等教育有所不同，所以在课程设置上也有其特殊的需求和自身的特点。我们为适应这种教育发展形势而编写了这本《微机原理及汇编语言教程》。

本课程的前修课程为“数字逻辑”。本课程是“微机接口技术”、“操作系统”、“计算机体系结构”等课程的必要先修课。由于本课程在计算机专业必修课中的位置，更体现了它的重要性。读者通过本课程的学习，可深入了解微型计算机系统的组成、工作原理，掌握汇编语言程序设计技术，为微型计算机技术的应用打下良好基础。

全书由 9 章组成。第 1 章为概述；第 2 章全面介绍了 Intel 系列微处理器；第 3～5 章为汇编语言系统和汇编语言程序设计；第 6 章介绍了存储器；第 7 章对中断系统结构、工作原理和中断控制器进行了详细的论述；第 8 章介绍了输入/输出接口的概念和控制方式；第 9 章为总线技术，介绍了常用的总线标准。实验部分编排了 7 个上机实验，并在附录中给出了参考答案。

本书作者都是有多年的教学经验和实践经验的教师。本书内容丰富，深入浅出，注重实用，是面向高等职业教育的特点而编写的，有利于学生对未知领域的掌握和运用。

本书的第 1,2,6,9 章由杨延双编写；第 3,4,5 章由张晓冬编写；第 7,8 章及实验由魏坚华编写。全书由张载鸿教授主审。

在本书的编写过程中，得到了张载鸿教授的大力支持，并提供了宝贵意见；尹子赓、尹志军、刚冬梅承担了书稿的录入、校对等工作。在此一并感谢。

由于作者水平有限，书中难免存在错误及不妥之处，敬请专家和广大读者批评指正。

编　者

2001.10

目　录

第1章 概 述

自1971年世界上第一台微型计算机诞生以来，随着微型计算机技术的飞速发展和微型计算机的空前普及，其应用领域已遍及各行各业及社会生活，深刻影响着社会、政治和经济的发展，改变了人们的学习方式、工作方式和生活方式。微型计算机的发展和影响是近代科学技术发展史上所罕见的。本章介绍微型计算机的发展概况和有关基本概念。

1.1 微型计算机发展概况

电子计算机按其体积和性能分为巨型机、大型机、中型机、小型机和微型机。微型计算机是以微处理器为核心，配上存储器、输入/输出接口电路和系统总线所构成。微型计算机的发展通常是以微处理器的升级而换代的。

微处理器的问世是在1971年。美国Intel公司生产的4004微处理器采用了PMOS技术。在4.2 mm×3.2 mm的硅片上集成了2 250个晶体管，可进行4位二进制的并行处理。后来Intel公司正式生产了通用的4040微处理器。这种4位的微处理器以体积小，价格低而引起人们的兴趣。以4004为核心组成的MCS-4是世界上第一代微型计算机。

1974年—1978年，Intel公司推出的8080/8085，Zlog公司推出的Z80，Motorola公司推出的MC6800/6802等，被称为第二代微处理器。这一代8位的微处理器的特点是采用NMOS电路，集成度达5 000个晶体管/片以上，时钟频率为2 MHz～4 MHz。这时的微处理器的设计和生产技术已相当成熟。

第三代是以16位微处理器的出现(1978年)为标志的。典型产品为Intel的8086、Zlog的Z 8000和Motorola的MC68000。它们采用了HMOS工艺，集成度为20 000～60 000管/片，时钟频率为4 MHz～8 MHz，平均指令执行时间为0.5 μs。

第四代微型计算机(1985年—1992年)是32位微型机。典型的微处理器产品有80386/486、MC 68020、Z 80000等。

1993年Intel公司推出的Pentium微处理器，宣布了第五代微处理器的诞生。这种64位的Pentium微处理器芯片采用了新的体系结构。芯片的集成度达(5×10^6～9.3×10^6)管/片，时钟频率达150 MHz～300 MHz。

1995年PentiumⅡ问世，1999年PentiumⅢ公布。目前，市场上的主流产品是PentiumⅣ系列。更高性能的微处理器将不断推出，微型计算机的发展速度是惊人的，而其性能/价格比日渐提高，微型计算机技术的应用越来越广泛。

2001年Intel第一款64位的产品安腾(Itanium)处理器隆重推出，2002年又推出了

Itanium2处理器。它们是瞄准高端企业市场的。

1.2 微型计算机的特点与分类

微型计算机与巨型机、大型机、中型机、小型机相比,最主要的特点是体积小、功耗低、价格低廉、可靠性高、硬件结构设计灵活、安装维修方便及具有丰富的软件。微型计算机的这些特点极大地赢得了用户的欢迎,使其应用日益广泛。

微型计算机的分类可以从不同角度进行:如果从制造工艺来分,可将微型机分为 MOS 型和双极型;若从组装形式划分,则可以分为单片、单板和多板微型机;按微处理器的字长来划分,通常可分为 4 位机、8 位机、16 位机、32 位机、64 位机及位片式等。位片式微处理器是以位为单位,由若干个位片组合而构成不同字长的微型计算机,其特点是结构灵活。目前,市场上的主流产品是 64 位机。

1.3 微处理器、微型计算机和微型计算机系统

1.3.1 微处理器

微处理器由一片或几片大规模集成电路组成,具有运算和控制功能的中央处理器部件(central processing unit)简称 CPU。

微处理器在内部结构上一般包括:

- 算术逻辑部件 ALU;
- 寄存器组;
- 程序计数器、指令寄存器和译码器;
- 时序和控制部件。

微处理器是微型计算机的核心。

1.3.2 微型计算机

微型计算机是以微处理器为核心,再配上存储器、输入/输出接口电路和系统总线,如图 1.1所示。

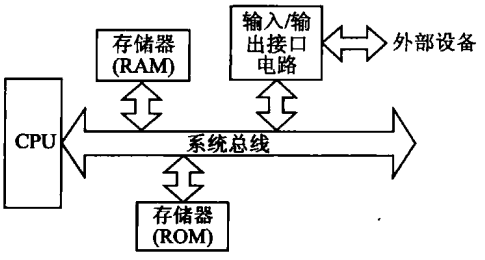


图 1.1 微型计算机的结构框图

- 存储器 包括只读存储器 ROM 和随机存取存储器 RAM。它们用来存储程序和数据。
- 输入/输出接口电路 用来控制微机与外部设备之间的信息交换。

● 系统总线 用来在微型计算机的部件和部件之间进行信息传输的一组总线，通常包括地址总线、数据总线和控制总线。

地址总线用来传送地址信息，为单向输出。地址总线的位数决定了 CPU 可以直接寻址的内存范围。如地址总线为 16 位时，可直接寻址范围为 $2^{16}=64K$ 单元。

数据总线是用来传输数据的，双向。数据总线的位数和 CPU 的位数相对应。如对于 16 位的 CPU，其数据总线的宽度为 16 位。

控制总线用来传输控制信息。

1.3.3 微型计算机系统

以微型计算机为中心，再配上外部设备和相应的软件就组成了微型计算机系统。微机系统包括两大部分：硬件和软件，如图 1.2 所示。

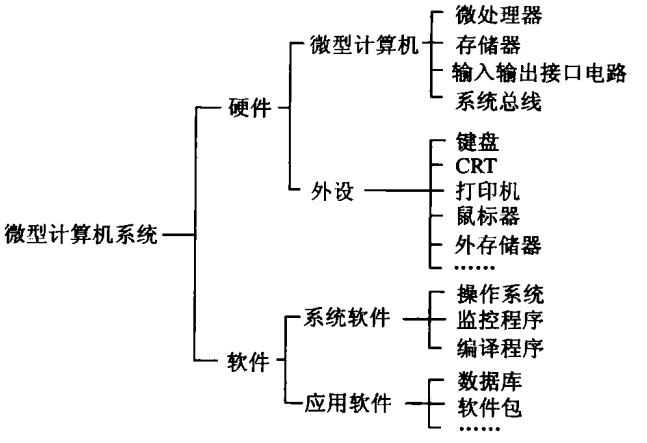


图 1.2 微型计算机系统

习 题

1. 微型计算机有哪些主要特点？
2. 说明微处理器、微型计算机和微型计算机系统有什么不同及三者之间的关系。

第2章 Intel系列微处理器

2.1 微处理器的基本结构

从内部结构上,微处理器一般都包含下列功能部件:

- 运算器:算术逻辑部件ALU;
- 寄存器:累加器和通用寄存器组;
- 程序计数器、指令寄存器和指令译码器;
- 时序和控制部件。

微处理器内部的算术逻辑部件是用来执行基本的算术运算和逻辑运算的。它可以进行加、减、乘、除的算术运算和与、或、非、异或等逻辑运算。累加器和通用寄存器组用来存放参加运算的数据、中间结果及存储运算结果的状态标志,也用来存放地址。程序计数器总是指向下一条要执行的指令;指令寄存器存放从存储器中取出的指令码。而指令译码器是对指令码进行译码和分析,以完成指定的操作。时序和控制部件具有指挥整个系统操作次序的功能。

现代的微处理器均为单片型,即由一片或几片超大规模集成电路制成。其集成度越来越高,性能也越来越高。

2.2 微处理器的工作原理

微处理器是通过执行程序来完成预定任务的,即逐条从存储器中取出程序中的指令并完成指令所指定的操作。

微处理器执行程序一般是通过反复执行以下步骤而实现的。

首先,从程序计数器所指向的存储器单元中取出一条指令(由于程序一般存放在内存的一个连续区域,所以顺序执行程序时,每取一个指令字节,程序计数器就自动加1),存放到指定的寄存器。其次,由指令译码器对指令码进行译码和分析,来确定指令的操作。若指令要求操作数,则确定操作数的地址,读出操作数。之后,执行指令内容(算术运算或逻辑运算)。最后,指令译码器译码时产生的相应控制信号送到时序和控制逻辑电路,控制CPU内部及整个系统来协调工作,从而完成指令所指定的操作任务。

2.3 16位微处理器

8086是Intel系列的16位微处理器。它采用高密度的硅栅H-MOS工艺制造,内部包

含近 29 000 只晶体管。它采用 40 根引脚双列直插式封装，单一 的 5 V 电源和单相时钟。8086 有 16 位数据线和 20 位地址线，可寻址空间为 2^{20} B，即 1 MB。

8088 是 Intel 公司继 8086 之后又推出的一种准 16 位的微处理器。8088 内部是 16 位 CPU，而外部的数据总线是 8 位的。这样就可以与当时已有的一整套 Intel 外围设备接口芯片直接兼容。

8086/8088 是 Intel 系列 CPU 中最具有代表性的 16 位微处理器。随后 Intel 公司陆续推出的 80x86 都是按其模式加以升级的，均保持与 8086/8088 兼容。

2.3.1　8086 的内部结构和引脚

2.3.1.1　8086 的内部结构

1. 框　图

8086 的内部结构框图如图 2.1 所示。

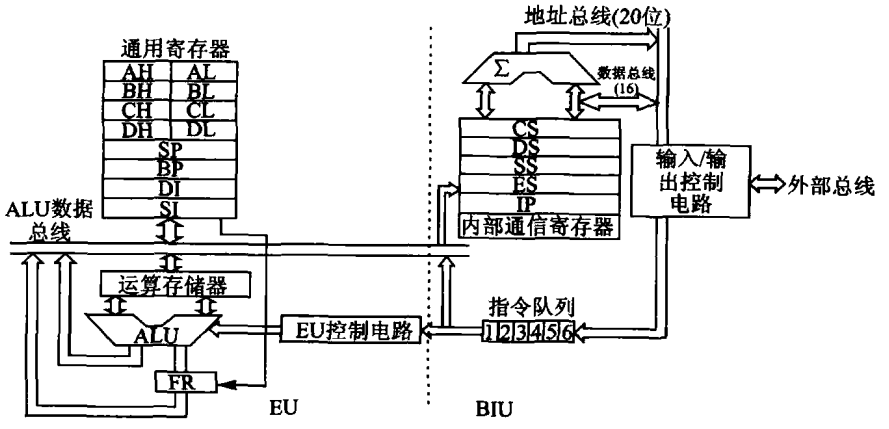


图 2.1　8086 的 CPU 内部结构框图

8086 的 CPU 由两个功能部件：EU(Execution Unit)和 BIU(Bus Interface Unit)所构成。下面分别介绍这两个功能部件。

(1) 总线接口部件 BIU

总线接口部件由下列部分组成：4 个 16 位的段寄存器 CS，SS，DS，ES；16 位的指令指针寄存器 IP；20 位的地址加法器 Σ；内部通信寄存器；6 字节的指令队列和输入/输出控制电路。

BIU 的功能负责与存储器、I/O 端口传送数据，即当指令队列空时，BIU 从内存取指令放入指令队列；当 CPU 执行指令时，BIU 配合 EU 从指定的内存单元或外设端口中取出数据供 EU 使用；当运算结束时，BIU 运算结果传送到指定内存单元或外设端口中。

BIU 中的地址加法器 Σ 是进行物理地址计算的，即将段寄存器中的 16 位数值和偏移量相加而得到 20 位物理地址。

(2) 执行部件 EU

执行部件包括 8 个 16 位通用寄存器、算术逻辑单元 ALU、标志寄存器 FR、运算寄存器和执行部件控制电路。

执行部件负责执行指令。EU 从指令队列取出指令代码，将其译码，发出相应的控制信息，并对通用寄存器和标志寄存器进行管理。

2. 寄存器

8086CPU 内部共有 14 个寄存器。下面分别加以介绍。

(1) 通用寄存器

8086CPU 有 8 个 16 位的通用寄存器。这 8 个通用寄存器可分为两组。一组称为数据寄存器,包括 AX,BX,CX 和 DX,用来存放数据或地址。这 4 个数据寄存器可以作为 16 位寄存器使用,也可以作为 8 位寄存器使用,即每一个 16 位数据寄存器都可分成两个独立的 8 位寄存器,见图 2.1 中的 AH 和 AL,BH 和 BL,CH 和 CL,DH 和 DL。其中 AH,BH,CH,DH 为高 8 位,AL,BL,CL,DL 为低 8 位。

通用寄存器的另一组包括堆栈指针寄存器 SP、基地址寄存器 BP、源变址寄存器 SI 和目的地址寄存器 DI。这 4 个寄存器只能作为 16 位寄存器使用,主要用来存放存储器或输入/输出端口的地址,也可以用来存放数据。

(2) 段寄存器

8086CPU 有 4 个 16 位的段寄存器:代码段寄存器 CS、数据段寄存器 DS、附加段寄存器 ES 和堆栈段寄存器 SS。段寄存器用来存放段基地址(或称为首地址),即 CS 中存放有当前执行程序所在段的首地址;DS 存放数据段首地址;SS 存放当前堆栈段的首地址;ES 存放当前附加段首地址。

(3) 指令指针寄存器 IP(16 位)

IP 用来指明将要执行的下一条指令的偏移地址。

(4) 标志寄存器 FR(16 位)

标志寄存器中的 7 位未用,所用 9 位的含义如图 2.2 所示。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
×	×	×	×	OF	DF	IF	TF	SF	ZF	×	AF	×	PF	×	CF

×:表示该位未用。

图 2.2 8086 标志寄存器的标志

标志寄存器的 9 个标志位可按照功能的不同分为两类:状态标志和控制标志。

状态标志用来表示算术运算和逻辑运算结果的特征,有如下 6 位。

- CF:进位标志。当进行加或减法运算时,若最高位产生进位或借位时,则 CF 为 1;否则 CF 为 0。另外,循环指令、移位指令也会影响这一标志位。
- PF:奇偶标志。当本次运算结果中的低 8 位中 1 的个数为偶数时,PF 为 1;否则 PF 为 0。
- AF:辅助进位标志。当本次运算,如果第 3 位往第 4 位有进位或借位时,AF 为 1;否则 AF 为 0。此标志在 BCD 码运算中作为是否进行十进制调整的依据。
- ZF:零标志。当运算结果为 0 时,ZF 为 1;否则 ZF 为 0。
- SF:符号标志。当运算结果最高位为 1 时,表示结果为负数,SF 为 1;否则 SF 为 0。
- OF:溢出标志。当运算结果产生溢出,即运算结果超出了相应类型数据所能表示的范围,OF 为 1;否则 OF 为 0。

控制标志用来控制 CPU 的操作,有如下 3 位。

- TF:单步标志(跟踪标志)。当 TF 为 1 时,CPU 按单步工作方式执行指令。在单步工

作方式下，CPU 每执行完一条指令就自动产生一次内部中断。此功能便于程序的调试。

- IF：中断标志。这是控制可屏蔽中断的标志。当 IF 为 1 时，允许 CPU 响应可屏蔽中断；当 IF 为 0 时，则 CPU 不能响应可屏蔽中断请求。
- DF：方向标志。用来控制串操作指令的执行。若 DF 为 1，则串操作按减地址方式进行操作；DF 为 0 时，则串操作指令按增地址方式进行操作。

2.3.1.2　8086CPU 的引脚

8086 的 CPU 的引脚分配如图 2.3 所示。

由于 8086 的 CPU 可有两种工作模式：最小模式（单 CPU 模式）和最大模式（多 CPU 模式），所以 8086CPU 的引脚 24 ～31 都有两种定义功能。图 2.3 括号中的引脚定义为最大模式下的定义。

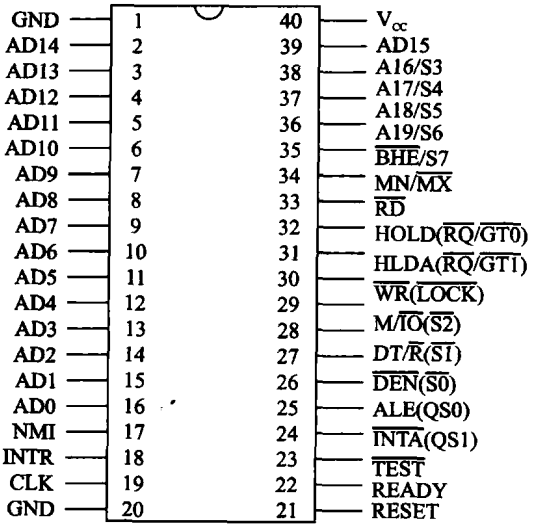


图 2.3　8086 的 CPU 的引脚信号

下面对 8086CPU 的各引脚功能作简要说明。

1）AD15～AD0（Address / Data BUS）：分时复用的地址/数据总线。传送地址时，三态输出；传送数据时，可以双向三态输入/输出。在 DMA 操作期间，这些引脚为浮空状态。

2）A19/S6～A16/S3（Address / Status）：分时复用的地址/状态线。作为地址线用时，A19～A16 与 A15 ～A0 一起构成访问存储器的 20 位物理地址。当 CPU 访问 I/O 端口时，A19～A16 均保持为 0。作状态线用时，S6 ～S3 输出状态信息。用 S4 和 S3 的不同编码表示当前使用哪一个段寄存器来对存储器寻址。S4，S3编码与所使用的段寄存器的对应关系如表 2－1 所列。S5 表示可屏蔽中断允许标志的当前状态。当 IF ＝1 时，S5 为 1。S6 保持恒为低电平。在 DMA 操作期间，这些引脚均为浮空状态。

表 2－1　S4，S3 状态编码

S4	S3	段寄存器
0	0	ES
0	1	SS
1	0	CS（或未用任何段寄存器）
1	1	DS

3）$\overline{\text{BHE}}$/S7（BUS High Enable / Status）：数据总线高 8 位有效信号。当 CPU 读/写存储器或 I/O 端口时，$\overline{\text{BHE}}$用作体选信号，与 AD0 配合来表示总线使用情况，如表 2－2 所例。在其它时间作为状态信号 S7，但 S7 未定义。

表 2-2 BHE 和 AD0 编码的含义

$\overline{BHE}$	AD0	总线使用情况
0	0	AD15～AD0 16 位数据总线上进行字节传送
0	1	AD15 ～AD8 8 位数据总线上进行字节传送
1	0	AD7～AD0 8 位数据总线上进行字节传送
1	1	无效

4) NMI(Non-Maskable Interrupt):非屏蔽中断请求信号输入,上升沿触发。非屏蔽中断不受 IF 的影响,不能用软件屏蔽。CPU 一旦检测到 NMI 请求有效,就会在当前指令结束后,执行中断类型号为 2 的中断处理程序。

5) INTR(Interrupt Request):可屏蔽中断请求信号输入,高电平有效。CPU 在每条指令的最后一个时钟周期对 INTR 进行采样。一旦发现 INTR=1 时,并且当前中断允许标志 IF=1 时,则 CPU 在当前指令结束后,响应中断请求,转入中断响应周期。

6) $\overline{INTA}$(Interrupt Acknowledge):中断响应信号输出,低电平有效。表示 CPU 响应了外部中断请求信号 INTR,发给请求中断的设备的回答信号。

7) $\overline{RD}$(Read):读信号。三态输出,低电平有效。表示当前 CPU 正在读存储器或 I/O 端口。

8) $\overline{WR}$(Write):写信号。三态输出,低电平有效。表示当前 CPU 正在写存储器或 I/O 端口。

9) M/$\overline{IO}$(Memory / IO):存储器或 I/O 端口访问控制信号,三态输出。当 M/$\overline{IO}$=1 时,表示 CPU 当前正在访问存储器;而 M/$\overline{IO}$=0 时,则表示 CPU 当前正在访问 I/O 端口。

10) READY:准备好信号输入,高电平有效。当 READY=1 时,表示 CPU 要访问的存储器或 I/O 设备已准备好传送数据。而当 READY 无效时,则要求 CPU 插入等待状态 Tw 来延长总线周期,直到 READY 信号有效为止。

11) RESET:复位信号输入,高电平有效。当 CPU 接收到 RESET 信号后,停止现行操作,并对标志寄存器,段寄存器 DS,SS,ES,指令指针寄存器 IP 和指令队列清零,而将 CS 置为 FFFFH。

12) $\overline{TEST}$:测试信号输入,低电平有效。在 CPU 执行 WAIT 指令时,每隔 5 个时钟周期对$\overline{TEST}$进行一次测试。若测试到$\overline{TEST}$为 0,则等待状态结束,CPU 继续执行下一条指令;否则 CPU 继续处于等待状态。

13) ALE(Address Latch Enable):地址锁存允许信号输出,高电平有效。当 ALE 输出有效时,表示当前在地址/数据总线上输出的是地址信息。地址锁存器将 ALE 作为锁存信号,对地址进行锁存。在 DMA 操作时,ALE 不能浮空。

14) DT/$\overline{R}$(Data Transmit/Receive):数据发送/接收控制信号输出。在最小模式系统中,用来作为数据收、发送器 8286/8287 的数据传送方向的控制。当 DT/$\overline{R}$为高电平,则进行数据发送;当 DT/$\overline{R}$为低电平时,则进行数据接收。在 DMA 方式中,它处于浮空状态。

15) $\overline{DEN}$(Data Enable):数据允许信号,低电平有效。在最小模式系统中作为数据收、发器 8286/8287 的选通信号。

16) HOLD(Hold Request):总线请求信号输入。在最小模式系统中,当 Hold 为高电平时,表示有其它共享总线的主控模块向 CPU 请求使用总线。

17) HLDA(Hold Acknowledge):总线请求响应信号输出,高电平有效。当 HLDA 有效

时，表示 CPU 对其它主模块的总线请求作出响应，并立即让出总线控制权，即所有三态线都进入浮空状态。

18) CLK(Clock)：主时钟信号输入。CLK 是 CPU 和总线控制的基准定时时钟。

19) MN/$\overline{MX}$(Minimum/Maximum)：工作模式选择信号输入。若此引脚接＋5 V 时，则 CPU 工作在最小模式系统中；若此引脚接地时，则 CPU 工作在最大模式系统。

下面对 8086CPU 工作在最大模式时，引脚 24 ～31 的定义作简要说明。

1) $\overline{S2}$，$\overline{S1}$，$\overline{S0}$(BUS Cycle Status)：总线周期状态信号输出。这三个信号的组合，表示 CPU 总线周期的操作类型，如表 2－3 所列。系统总线控制器 8288 利用这些状态信号来产生访问存储器或 I/O 接口的控制信号。

2) QS1，QS0(Instruction Queue Status)：指令队列状态信号输出。用来表示 CPU 中指令队列当前的状态，其信号组合含义如表 2－4 所列。

3) $\overline{LOCK}$：封锁信号输出。当$\overline{LOCK}$为低电平时，系统中其它总线主控者就不能占有总线。这个信号由软件设置。当在指令前加上$\overline{LOCK}$前缀时，则在执行这条指令期间$\overline{LOCK}$保持有效。

4) $\overline{RQ}/\overline{GT0}$，$\overline{RQ}/\overline{GT1}$(Request/Grant)：请求/同意控制信号。输入时表示其它主控者向 CPU 请求使用总线；输出时表示 CPU 对总线请求的响应信号。$\overline{RQ}/\overline{GT0}$和$\overline{RQ}/\overline{GT1}$都是双向的。两个引脚可以连接两个主控者，其中$\overline{RQ}/\overline{GT0}$比$\overline{RQ}/\overline{GT1}$的优先级高。

表 2－3　$\overline{S2}$，$\overline{S1}$，$\overline{S0}$ 的代码组合和对应的操作

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	操作过程
0	0	0	发中断响应信号
0	0	1	读 I/O 端口
0	1	0	写 I/O 端口
0	1	1	暂停
1	0	0	取指令
1	0	1	读内存
1	1	0	写内存
1	1	1	无源状态 *

* 无源状态：此时，一个总线操作过程就要结束，另一个新的总线周期还未开始。

表 2－4　QS1，QS0 代码组合的含义

QS1	QS0	含　义
0	0	无操作
0	1	从指令队列中取第一个字节
1	0	指令队列已空
1	1	从指令队列中取后续字节

2.3.2　8086 的存储器组织

8086 有 20 根地址线，具有直接寻址空间 1 MB。这 1 MB 的内存单元按照 00000H～FFFFFH 来编址。由于 8086 的内部寄存器都是 16 位的，所以用寄存器不能直接对 1 MB 的

内存空间进行寻址。8086 将 1 MB 空间进行分段,每个逻辑段最大为 64 KB,段内寻址可用 16 位地址码直接寻址。各个逻辑段可以在实际的存储空间中完全分开,也可以相互重叠,如图 2.4 所示。对于任何一个物理地址,可以被包含在一个逻辑段中或多个逻辑段中,只要能得到它所在逻辑段的首地址和在段中的相对地址,就可以对它进行访问。为了简化操作,要求每个逻辑段必须从 1MB 空间的任一个能被 16(16 字节的存储空间作为一节)除尽的边界开始,即每个段的首地址低 4 位总为 0。将段首地址的高 16 位存放在相应的段寄存器中,这样通过预置段寄存器的内容,就可以访问不同的存储区域。段内存储单元的地址,用相对于段首地址的偏移量来表示。

在 8086 中,每一个存储单元都有一个唯一的物理地址,即在 1MB 存储空间中唯一识别每一个存储单元的地址。范围是 00000H～FFFFFH。CPU 访问存储器时,由地址总线送出的是物理地址信息,而程序中涉及的地址是逻辑地址。逻辑地址由段首地址(也称为段基址)和偏移量(也称为偏移地址)来组成。段基址和偏移地址都是无符号数。物理地址是由逻辑地址变换而来的。物理地址的计算公式为:

$$物理地址=段基址\times 16+偏移地址$$

这是由总线接口部件 BIU 的地址加法器Σ来完成的。其操作如图 2.5 所示。

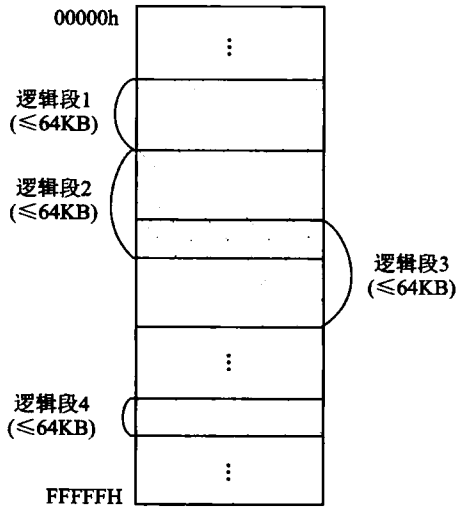


图 2.4 逻辑段分段方式示意图

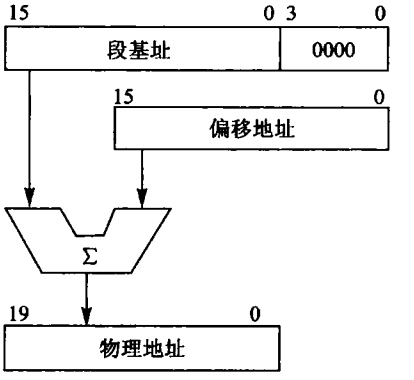


图 2.5 物理地址的形成过程

例如,段寄存器 CS=0100H,编移地址 IP=0066H,则物理地址为 01066H。

通常内存单元的逻辑地址表示为:段基地址:偏移地址。例如:0100H:0066H。

2.3.3 80286 的内部结构

Intel 公司在 1982 年推出的 80286 微处理器是较先进的 16 位微处理器。其内部操作和寄存器均为 16 位的,不再使用分时复用的引脚,具有独立的 24 条地址线和 16 条数据线。80286 与 8086 保持向上兼容,并增加了虚拟存储器管理、四级特权保护机构及多任务处理等。

80286 具有两种工作方式:实地址方式和虚地址保护方式。在实地址方式下,相当于一个高速的 8086 运行;在虚地址保护方式下,系统具有 16MB(2^{24} B)的实际地址空间,为每个任务