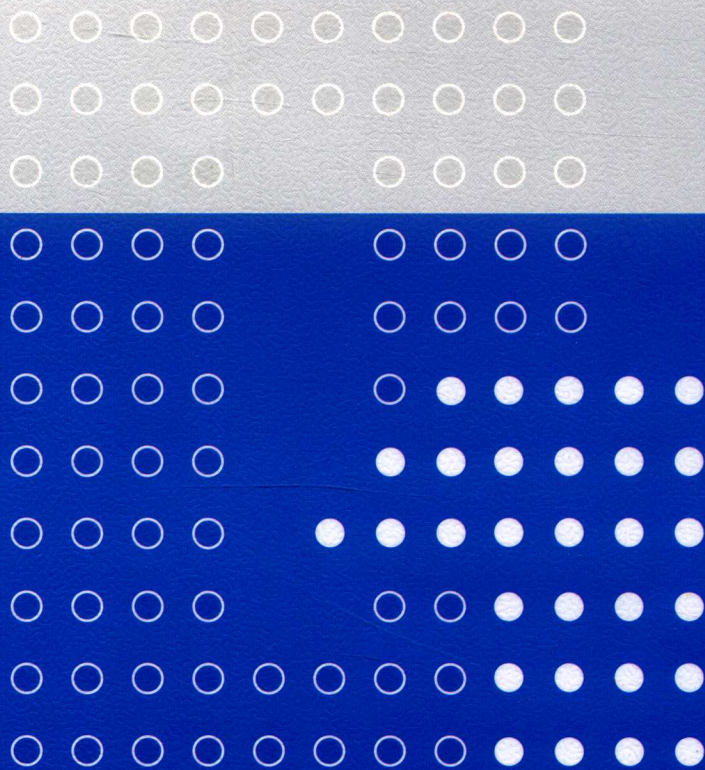




“十二五”普通高等教育本科国家级规划教材 计算机系列教材

# 汇编语言与接口技术 知识精要与实践



上定 杜呈透 编著



清华大学出版社

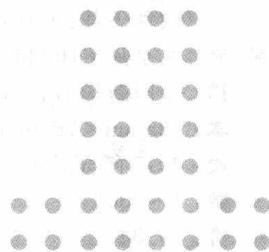


“十二五”普通高等教育本科国家级规划教材

计算机系列教材

王让定 杜呈透 编著

# 汇编语言与接口技术 知识精要与实践



清华大学出版社  
北京



## 内 容 简 介

本书是《汇编语言与接口技术(第4版)》知识点精讲、拓展及实践教材,全书分为两个部分。第一部分是知识精要,共10章,主要结合主教材的知识结构,对知识点进行精讲、拓展,每一章都系统总结了主教材的主要知识点,并给出主教材的大部分课后习题的详细分析讲解和参考答案。第二部分是应用实践,共4章,主要是对理论知识的实践应用,以增进知识内化,内容包括汇编语言程序设计实验步骤、主要调试工具、汇编语言程序设计实验以及微机原理接口技术实验,每个实践项目严格按照教学进度安排,由易到难,并设计若干提高实验、综合实验以及创新性实验。本书的实验不限于某个实验箱系统,尽量做到通用。

本书可作为高等学校本科信息类相关专业的教材,也可供广大工程技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

汇编语言与接口技术知识精要与实践/朱莹,王让定,杜呈透编著. —北京:清华大学出版社,2017

(计算机系列教材)

ISBN 978-7-302-46472-3

I. ①汇… II. ①朱… ②王… ③杜… III. ①汇编语言—程序设计—教材 ②微型计算机—接口技术—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2017)第 024641 号

责任编辑:张 民 战晓雷

封面设计:常雪影

责任校对:胡伟民

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京泽宇印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:9.5

字 数:240千字

版 次:2017年5月第1版

印 次:2017年5月第1次印刷

印 数:1~2000

定 价:29.00元

产品编号:073823-01

为了更好地学习“汇编语言与接口技术”这门课程,培养学生的应用开发能力,我们编写了《汇编语言与接口技术知识精要与实践》,以王让定教授和朱莹老师主编的《汇编语言与接口技术(第4版)》教材(以下简称“主教材”)为蓝本,对关键知识点进行精讲和拓展,并设计了丰富的实践项目以深化学习。

本教材的作者多年承担“汇编语言”和“微机接口技术”等课程的理论与实践教学,在教学实践过程中,我们深深体会到,要提高综合应用信息技术的能力,关键在于更好地运用已有信息系统的接口技术。本书在知识点总结和实验安排上向综合应用和创新延伸、拓展,以起到引导读者继续深入学习的作用。另外,本书所有实验不限于某个实验系统,尽量做到通用,这也是本书的一个特色。

本书共分两个部分。第一部分是知识精要,共10章,每一章首先对主教材的本章知识点进行总结,然后针对主教材中的典型课后习题做详细的分析讲解,给出参考答案,在习题精讲的过程中对知识点进行精练和扩充,便于复习与自学。第二部分是实践应用,共4章,主要内容如下:对80x86 CPU的寄存器组、寻址方式和指令系统进行归纳总结;详尽地介绍汇编语言程序设计上机实验步骤、常用调试工具;给出汇编语言程序设计实验,共有8个专题,每个专题的实验内容都由易到难,循序渐进;最后介绍了80x86微机接口技术及其应用实验。其中,最后一章首先介绍最简单的微机系统组成及I/O地址译码原理电路,然后根据主教材的教学内容安排了6个接口技术基础实验,这些实验是常用的微机接口方法,包括定时器、串并行接口技术、模数/数模转换技术和中断技术,每个实验都包括基本应用和提高部分。另外,为了提高学生的综合应用能力和创新能力,最后一章还设计了综合创新类实验。

未编入本书的主教材部分课后习题也备有参考答案,欢迎选用本教材的老师与作者联系索取。

本书由朱莹、王让定、杜呈透编写,其中第一部分(第1~10章)由朱莹执笔,第二部分的第12章和第14章的4.3节由杜呈透执笔,第二部分的其余章节由朱莹执笔。王让定负责全书的统稿。

本书的编程题目都在计算机上调试通过。编程环境为MASM 6.0和Visual C++。

愿本书对广大读者有所帮助。限于作者水平和经验,书中难免有疏漏和不妥之处,恳请广大专家和读者不吝批评指正。

## 前言 《汇编语言与接口技术知识精要与实践》

欢迎选用本书的教师和学生与作者联系,共同探讨与“汇编语言与接口技术”课程和本教材相关的问题。作者的 E-mail 地址是 zhuying@nbu.edu.cn, duchengtou@nbu.edu.cn, wangrangding@nbu.edu.cn。

作者

## 第一部分 知识精要

- 第 1 章 微型计算机概述 /3
  - 知识点总结 /3
  - 习题精解 /3
- 第 2 章 80x86 微处理器 /6
  - 知识点总结 /6
  - 习题精解 /6
- 第 3 章 80x86 指令系统和寻址方式 /12
  - 知识点总结 /12
  - 习题精解 /12
- 第 4 章 80x86 汇编语言程序设计 /25
  - 知识点总结 /25
  - 习题精解 /25
- 第 5 章 存储器技术 /50
  - 知识点总结 /50
  - 习题精解 /50
- 第 6 章 输入输出接口及数据传输控制方式 /54
  - 知识点总结 /54
  - 习题精解 /54
- 第 7 章 串并行接口技术 /56
  - 知识点总结 /56
  - 习题精解 /56
- 第 8 章 中断和 DMA 技术 /67
  - 知识点总结 /67
  - 习题精解 /67
- 第 9 章 模拟接口技术 /71
  - 知识点总结 /71
  - 习题精解 /71
- 第 10 章 高速串行总线 /77
  - 知识点总结 /77

习题精解 /77

## 第二部分 应用实践

### 第 11 章 80x86 机器组织 /81

11.1 80x86 寄存器 /81

11.2 80x86 数据寻址方式 /82

11.3 80x86 指令集 /84

### 第 12 章 汇编语言程序设计步骤 /85

12.1 软件上机过程简述 /85

12.2 DOS 基本命令摘要 /86

12.3 汇编与宏汇编程序 /87

12.4 连接程序及程序执行 /88

12.5 调试工具介绍 /89

12.5.1 DEBUG /89

12.5.2 TD 的使用 /93

### 第 13 章 汇编语言程序设计实验 /96

13.1 算术运算类编程实验 /96

13.1.1 实验 1: 多字节加减法运算编程实验 /96

13.1.2 实验 2: 十进制数的 BCD 加减法编程实验 /97

13.1.3 实验 3: 二进制乘除法编程实验 /97

13.2 数值转换实验 /98

13.3 分支程序设计实验 /101

13.4 循环程序设计实验 /103

13.5 字符串操作及输入输出实验 /105

13.6 子程序设计实验 /106

13.6.1 实验 1: 求无符号字节序列中的最大值和最小值 /108

13.6.2 实验 2: 求  $N!$  /108

13.7 排序程序设计实验 /109

13.8	查表程序设计实验	/110
<b>第14章</b>	<b>80x86 微机接口技术及其应用实验</b>	<b>/112</b>
14.1	8086 微机系统结构配置及 I/O 地址译码电路	/112
14.1.1	8086 微机系统结构配置	/112
14.1.2	I/O 地址译码电路	/115
14.2	接口技术基础实验	/116
14.2.1	8253 定时/计数器应用实验	/116
14.2.2	8255 并行接口实验	/118
14.2.3	8251 串行接口应用实验	/119
14.2.4	D/A 转换实验	/121
14.2.5	A/D 转换实验	/123
14.2.6	中断控制器 8259A 实验	/124
14.3	接口技术综合实验	/126
14.3.1	LED 显示实验	/126
14.3.2	电子发声设计实验	/128
14.3.3	竞赛抢答器	/130
<b>附录 A</b>	<b>80x86 指令集简介</b>	<b>/133</b>



# 第一部分

## 知识精要



# 第 1 章 微型计算机概述

## 知识点总结

通过本章的学习,应对微处理器和微型计算机有较完整的认识。本章内容包括微处理器和微型计算机的基本概念、组成以及系统总线等,重点是以下几方面内容:

- 掌握微处理器、微型计算机及计算机系统的概念。
- 掌握微型计算机系统的软硬件组成及微机系统的基本工作原理。
- 掌握微型计算机系统的主要技术参数,包括字长、存储容量、速度、外部设备、配套软件、扩展性及性价比等。
- 了解总线基础和常用的总线。
- 了解微处理器的产生、发展、应用及未来的发展趋势。

## 习题精解

1. 解释和区别下列名词术语:

- (1) 微处理器和微型计算机系统。
- (2) 硬件和软件。
- (3) 系统软件和应用软件。
- (4) 机器语言、汇编语言和高级语言。
- (5) 汇编语言程序和汇编程序。
- (6) 总线、内部总线和系统总线。

参考答案:

(1) 微处理器是由一片或几片大规模集成电路组成的具有运算器和控制器功能的中央处理器(即 CPU)。

微型计算机系统是以微处理器为中心,加上只读存储器(ROM)、随机存取存储器(RAM)、输入输出接口电路和系统总线缓冲器。

(2) 硬件系统就是计算机系统中的各种物理装置,包括控制器、运算器、内存储器、I/O 设备以及外存储器等,它是计算机系统的物质基础。

软件系统是相对于硬件系统而言的,即所有程序、数据和相关文件的集合。从狭义的角度上讲,软件是指计算机运行所需的各种程序。

**注意:** 没有硬件,谈不上应用计算机;但是,光有硬件而没有软件,计算机也不能工作。

(3) 系统软件是应用软件的运行环境,是人和硬件系统之间的桥梁,人们就是通过它们来操作计算机的。系统软件是由计算机的设计者或销售商提供给用户的,是硬件系统

首先应安装的软件。

应用软件是用户利用计算机及其所提供的系统软件、程序设计语言为解决各种实际问题而编写的程序。

(4) 机器语言是能够直接被计算机识别和执行的语言。

汇编语言是用一种能够帮助记忆的符号,即用英文或缩写符来表示机器的指令,并用这种助记符表示的机器语言。

高级语言又称为算法语言,它的产生是为了能够使程序设计语言适于描述各种算法,使程序设计中使用的语句与实际更接近,并使程序设计可以脱离具体计算机结构。

(5) 汇编程序是把用汇编语言编写的源程序翻译成机器语言表示的目标程序的翻译程序,如 MASM. EXE。

汇编语言程序是由组成汇编语言的助记符指令汇集成的程序,如 XX. ASM。

(6) 为了简化硬件电路设计和系统结构,常用一组线路,配置以适当的接口电路,与各部件和外围设备连接,这组共用的连接线路称为总线。

内部总线是微机内部各外围芯片与处理器之间的总线,用于芯片一级的互连,如 I<sup>2</sup>C 总线、SPI 总线、SCI 总线等。

系统总线是微机中各插件板与系统板之间的总线,用于插件板一级的互连,如 ISA 总线、PCI 总线等。

2. 画出典型的 8 位微处理器的结构框图,说明各组成部分的作用。

参考答案:

一个典型的 8 位微处理器的结构如图 1.1 所示,主要包括以下几个重要部分:累加器、算术逻辑运算单元(ALU)、状态标志寄存器、寄存器阵列、指令寄存器、指令译码器和定时及各种控制信号的产生电路。

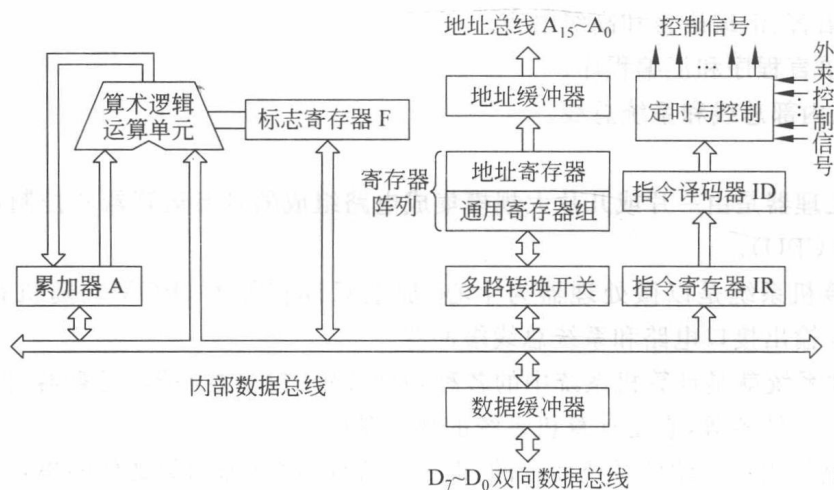


图 1.1 典型的 8 位微处理器的结构

(1) 累加器和算术逻辑运算部件。

累加器和算术逻辑运算部件主要用来完成数据的算术和逻辑运算。ALU 有两个输入端和两个输出端。其中一端接至累加器,接收由累加器送来的第一个操作数;另一端

通过数据总线接到寄存器阵列,以接收第二个操作数。参加运算的操作数在 ALU 中进行规定的运算,运算结束后,将结果送至累加器,同时将操作结果的特征状态送标志寄存器。

累加器是一个特殊的寄存器,它的字长和微处理器的字长相同,累加器具有输入输出和移位功能,微处理器采用累加器结构可以简化某些逻辑运算。由于所有运算的数据都要通过累加器,故累加器在微处理器中占有很重要的位置。

#### (2) 寄存器阵列。

- 通用寄存器组:可由用户灵活支配,用来寄存参与运算的数据或地址信息。
- 地址寄存器:专门用来存放地址信息的寄存器。
- 指令指针(IP):它的作用是指明下一条指令在存储器中的地址。每取一个指令字节,IP 自动加 1,如果程序需要转移或分支,要把转移的偏移地址放入 IP。
- 变址寄存器(SI,DI):变址寄存器的作用是用来存放要修改的地址,也可以用来暂存数据。
- 堆栈指示器(SP):用来指示 RAM 中堆栈栈顶的地址。SP 寄存器的内容随着堆栈操作的进行自动发生变化。

#### (3) 指令寄存器、指令译码器和定时及各种控制信号的产生电路。

- 指令寄存器(Instruction Register,IR):用来存放当前正在执行的指令代码。
- 指令译码器(Instruction Decoder,ID):用来对指令代码进行分析、译码,根据指令译码的结果输出相应的控制信号。
- 时序逻辑电路产生各种操作电位、不同节拍的信号、时序脉冲等执行相应命令所需的全部控制信号。

#### (4) 内部总线和总线缓冲器。

内部总线把 CPU 内各寄存器和 ALU 连接起来,以实现各单元之间的信息传送。内部总线分为内部数据总线和地址总线,它们分别通过数据缓冲器和地址缓冲器与芯片外的系统总线相连。缓冲器用来暂时存放信息(数据或地址),它具有驱动放大能力。

## 第 2 章 80x86 微处理器

### 知识点总结

通过本章的学习,应该对 80x86 CPU 内部结构、寄存器、引脚信号和总线操作时序有很深刻的认识。学习时注意以下几方面的内容:

- 掌握 80x86 CPU 的内部结构及各部分的功能。掌握 80x86 CPU 的内部寄存器组。
- 掌握 80x86 CPU 的外部引脚结构,了解 80386 以上微机的外部引脚结构。
- 掌握 80x86 CPU 的两种工作模式下系统总线的构成,理解 80386 以上微处理器的工作方式。
- 理解 80386 以上微处理器的 3 种工作模式,即实地址模式、保护模式和虚拟 8086 模式。
- 掌握 80x86 CPU 总线工作时序。理解总线周期、时钟周期及指令周期的概念。
- 掌握 80x86 CPU 20 位的物理地址的形成过程,掌握 8086/8088 CPU 的存储器组织和堆栈的概念。
- 掌握 80x86 存储器组织。
- 了解最新微处理器的特点。

### 习题精解

1. 8086/8088 CPU 的地址总线有多少位? 其寻址范围是多少?

**参考答案:**

20 位,其寻址范围是 00000H~FFFFFFH。

2. 8086/8088 CPU 分为哪两个部分? 各部分主要由什么组成?

**参考答案:**

从功能上来看,8086/8088 CPU 可分为两部分,即总线接口部件(Bus Interface Unit, BIU)和执行部件(Execution Unit, EU)。

(1) 执行部件(EU)。

功能: 负责指令的执行。

组成:

- 8 个 16 位通用寄存器组(AX、BX、CX、DX、SP、BP、SI、DI)。
- 1 个 16 位的状态标志寄存器(FLAGS)。
- 算术逻辑运算器(ALU)。
- 暂存寄存器。

- EU 控制单元。

## (2) 总线接口部件(BIU)。

功能：负责与存储器及 I/O 接口之间的数据传送操作。具体来看，完成取指令并送入指令队列缓冲器，配合执行部件的动作，从内存单元或 I/O 端口取操作数，或者将操作结果送内存单元或者 I/O 端口。

组成：

- 4 个 16 位段寄存器(DS、CS、ES、SS)。
- 1 个 16 位指令指针寄存器 IP(指向下一条要取出的指令代码)。
- 20 位地址加法器(用来产生 20 位地址)。
- 6 字节(8088 为 4 字节)指令队列缓冲器。

### 3. 什么叫队列？8086/8088 CPU 中指令队列有什么作用？其长度分别为多少字节？

参考答案：

队列(queue)是只允许在一端进行插入，而在另一端进行删除的运算受限的线性表，队列的修改是依先进先出(FIFO)的原则进行的。

8086/8088 的指令队列分别为 6 个字节和 4 个字节，在执行指令的同时，可从内存中取出后续的指令代码，放在指令队列中，可以提高 CPU 的工作效率。

### 4. 8086/8088CPU 中有几个通用寄存器？有几个变址寄存器？有几个指针寄存器？通常哪几个寄存器也可作为地址寄存器使用？

参考答案：

#### (1) 通用寄存器。

8086/8088 有 4 个 16 位的通用寄存器(AX、BX、CX、DX)，可以存放 16 位的操作数，也可分为 8 个 8 位的寄存器(AL、AH、BL、BH、CL、CH、DL、DH)来使用。其中 AX 称为累加器，BX 称为基址寄存器，CX 称为计数器，DX 称为数据寄存器，这些寄存器在具体使用上有一定的差别，如表 2.1 所示。

表 2.1 内部寄存器主要用途

寄存器	用途
AX	字乘法，字除法，字 I/O
AL	字节乘，字节除，字节 I/O，十进制算术运算
AH	字节乘，字节除
BX	转移
CX	串操作，循环次数
CL	变量移位，循环控制
DX	字节乘，字节除，间接 I/O

#### (2) 指针寄存器。

系统中有两个 16 位的指针寄存器 SP 和 BP，其中 SP 是堆栈指针寄存器，由它和堆栈段寄存器 SS 一起确定堆栈在内存中的位置；BP 是基数指针寄存器，通常用于存放基

地址。

### (3) 变址寄存器。

8086/8088 CPU 系统中有两个 16 位的变址寄存器 SI 和 DI, 其中 SI 是源变址寄存器, DI 是目的变址寄存器, 都用于指令的变址寻址方式。

### 6. 试求出下列运算后的各个状态标志, 并说明进位标志和溢出标志的区别。

1278H + 3469H

54E3H - 27A0H

3881H + 3597H

01E3H - 01E3H

### 参考答案:

ZF=0, AF=1, PF=1, CF=0, OF=0, SF=0

ZF=0, AF=0, PF=0, CF=0, OF=0, SF=0

ZF=0, AF=0, PF=1, CF=0, OF=0, SF=0

ZF=1, AF=0, PF=0, CF=0, OF=0, SF=0

**分析:** 处理器内部以补码表示有符号数, 各种数据编码都有其数据表示范围, 如 8 个二进制位能够表示的整数范围是  $-128 \sim 127$ , 16 位表示的范围是  $-32768 \sim 32767$ 。如果在运算过程中产生的数据超出这个表示范围, 称为溢出。

**注意:** 有符号数溢出, 说明运算结果不正确。

进位标志 CF 和溢出标志 OF 是两个意义不同的标志:

- 进位标志表示无符号数运算结果是否超出范围, 进位标志置位时运算结果仍然正确。
- 溢出标志表示有符号数运算结果是否超出范围, 溢出标志置位时运算结果一定不正确。

例 1:  $3AH + 7CH = B6H$

无符号数运算:  $58 + 124 = 182$ , 范围内, 无进位。

有符号数运算:  $58 + 124 = 182$ , 范围外, 有溢出。

例 2:  $AAH + 7CH = (1)26H$

无符号数运算:  $170 + 124 = 294$ , 范围外, 有进位。

有符号数运算:  $-86 + 124 = 28$ , 范围内, 无溢出。

处理器对两个操作数进行运算时, 按照无符号数求得结果, 并相应设置进位标志 CF; 同时, 根据是否超出有符号数的范围设置溢出标志 OF。应该利用哪个标志, 由程序员来决定。也就是说, 如果认为将参加运算的操作数是无符号数, 就应该关心进位; 认为是有符号数, 则要注意是否溢出。

判断运算结果是否溢出有一个简单的规则: 只有当两个相同符号数相加, 而运算结果的符号与原数据符号相反时, 才会产生溢出, 因为此时的运算结果显然不正确。其他情况下则不会产生溢出。

**注意:** 一般来讲, 不是每次运算后所有的标志都改变, 只是在某些操作之后, 才对其某个标志进行检查。



8. 什么是逻辑地址？什么是物理地址？它们之间有什么联系？各用在何处？

参考答案：

逻辑地址是程序设计时根据需要产生的一种地址，它不是真正的单元地址。逻辑地址由段基址和偏移地址两部分组成，它们都是 16 位无符号数。例如，某存储单元的逻辑地址为 1000H:2000H，表示该存储单元的段基址是 1000H，偏移地址是 2000H。

物理地址是与存储单元一一对应的地址，如 8088/8086 可直接寻址 1MB 的存储空间，其地址区域为 00000H~FFFFFFH。物理地址是由段地址与偏移地址共同决定的，段地址来自段寄存器(CS、DS、ES、SS)，是 16 位地址，由指令提供该单元相对于相应段起始地址的 16 位偏移量，它可以来自 CPU 中不同的 16 位寄存器(IP、SP、BP、SI、DI、BX 等)。

计算物理地址的表达式如下：

$$\text{物理地址} = \text{段地址} \times 16 + \text{偏移地址}$$

10. 若(CS)=5200H，物理转移地址为 5B230H 时，则当 CS 的内容被设为 7800H 时，物理转移地址应为多少？

参考答案：

根据计算物理地址的表达式，有

$$5B230H = 5200H \times 16 + \text{偏移地址}$$

解得偏移地址为 9230H。

若(CS)=7800H，则物理地址 = 7800H × 16 + 9230H = 81230H。

12. 8086 CPU 工作在最小模式(单 CPU)和最大模式(多 CPU)时的主要特点是什么？有何区别？

参考答案：

所谓最小模式，就是系统中只有一个 8088/8086 微处理器，在这种情况下，所有的总线控制信号都是直接由 8088/8086 CPU 产生的，系统中的总线控制逻辑电路被减到最少，该模式适用于规模较小的微机应用系统。

8088/8086 系统在最小模式下的典型配置如图 2.1 所示，它具有以下几个方面的特点。

(1) MN/ $\overline{\text{MX}}$ 端接 +5V，决定 CPU 的工作在最小工作模式下。

(2) 有一片 8284A，作为时钟信号发生器。

(3) 有一片 8282 或 74LS273，用来作为地址信号的锁存器。

(4) 当系统中所连接的存储器和外设较多时，需要增加数据总线的驱动能力，这时，需用两片 8286/8287 作为总线收发器。

最大模式是相对于最小模式而言的，最大模式用在中、大规模的微机应用系统中，在最大模式下，系统中至少包含两个微处理器，其中一个为主处理器，即 8086/8086 CPU，其他的微处理器称为协处理器，它们是协助主处理器工作的。在这种情况下，所有的总线控制信号由总线控制器产生。8088/8086 在最大模式下的典型配置如图 2.2 所示。

可以看出，最大模式和最小模式在配置上的主要差别在于：在最大模式下，要用 8288 总线控制器来对 CPU 发出的控制信号进行变换和组合，以得到对存储器或 I/O 端口的读/写信号和对锁存器 8282 及总线收发器 8286 的控制信号。