

21世纪高等学校规划教材 | 计算机科学与技术



微机原理、汇编 与接口技术教程

王晓虹 苏维龙 邓红卫 编著

清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术



微机原理、汇编 与接口技术教程

王晓虹 苏维龙 邓红卫 编著

清华大学出版社
北京

内 容 简 介

本书以 80x86 微处理器为背景,详细地介绍微机原理、汇编语言及接口技术。全书共 12 章,第 1~4 章介绍微机原理部分,包括运算器、存储部件及控制器、微处理器的工作原理及应用。第 5~8 章介绍汇编语言部分,包括汇编语言程序设计的基础知识、指令系统、程序设计方法及典型案例分析。第 9~12 章介绍接口技术部分,包括输入输出接口、中断及定时/计数接口、并行与串行接口、A/D 及 D/A 转换接口的工作原理及应用。

本书深入浅出,注重应用示例分析,具有较强的系统性、实用性。

本书可作为高校计算机本科非计算机专业的教材和自学考试的教材,也可作为教师、计算机应用技术人员参考书。

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

微机原理、汇编与接口技术教程/王晓虹等编著. —北京:清华大学出版社,2016

21 世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-45005-4

I. ①微… II. ①王… III. ①微型计算机—理论—高等学校—教材 ②汇编语言—程序设计—高等学校—教材 ③微型计算机—接口技术—高等学校—教材 IV. ①TP36 ②TP313

中国版本图书馆 CIP 数据核字(2016)第 208257 号

责任编辑:贾 斌 薛 阳

封面设计:傅瑞学

责任校对:梁 毅

责任印制:王静怡

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

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:19.75

字 数:478 千字

版 次:2016 年 9 月第 1 版

印 次:2016 年 9 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

产品编号:066098-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

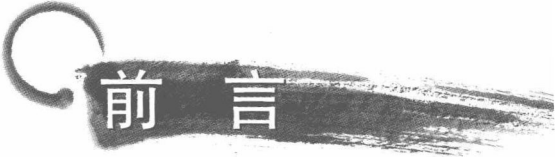
- (1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 21 世纪高等学校规划教材·信息管理与信息系统。
- (6) 21 世纪高等学校规划教材·财经管理与应用。
- (7) 21 世纪高等学校规划教材·电子商务。
- (8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn



前言

随着计算机应用的普及和深入,计算机在各个领域的作用日益提高,并且前景广阔。各行各业的从业人员对计算机知识和技能掌握得越好,就越能充分发挥计算机的作用,也可以更好地与计算机专业人员合作。所以,在高等院校中,微机原理、汇编与接口技术是很多专业开设的一门有关计算机原理及应用方面的课程。本课程主要讲述计算机的工作原理、各种接口芯片的结构及工作过程、利用汇编语言来实现对硬件的驱动和操作。

本教材以精讲多练,鼓励学生自主学习为原则,在保证知识的连续性、完整性的同时,对传统的教材内容进行了较大的改动,力求内容精炼、突出重点、注重应用。

本书由辽宁石油化工大学的王晓虹、苏维龙、邓红卫、王福威执笔。其中,全书内容由王晓虹、苏维龙组织和审核。

本书中的错误和不妥之处敬请批评指正。

编者

2015年12月

目 录

微机原理部分

第 1 章 运算器	3
1.1 计算机中的数据	3
1.1.1 数据的表示	3
1.1.2 二进制数	5
1.1.3 二进制加法	8
1.2 逻辑门	9
1.3 加法器	10
1.3.1 加法器的构成	10
1.3.2 加法的溢出	13
1.4 有符号数的加法运算	15
1.4.1 原码	15
1.4.2 补码	16
1.4.3 补码加法的溢出	17
1.5 补码减法	19
习题 1	21
第 2 章 存储部件	23
2.1 寄存器	23
2.1.1 基本 RS 触发器	23
2.1.2 时序逻辑电路	24
2.1.3 时标 RS 触发器	25
2.1.4 D 触发器	25
2.1.5 寄存器的入口	26
2.1.6 寄存器的出口	26
2.1.7 总线	27
2.2 主存储器	28
2.2.1 随机存取存储器 RAM	28
2.2.2 只读存储器 ROM	32
2.2.3 存储器容量扩展	33
2.3 辅助存储器	37

2.3.1 非易失性存储器	37
2.3.2 文件与文件夹	37
2.3.3 脱机存储	37
习题 2	38
第 3 章 控制器	39
3.1 机器语言	39
3.2 微程序控制器	41
3.3 控制信号的次序	45
3.4 组合逻辑电路控制器	47
3.5 逻辑表达式的化简	51
3.5.1 与运算	51
3.5.2 或运算	51
3.5.3 非运算	52
3.5.4 常用运算规律	52
习题 3	53
第 4 章 微处理器	54
4.1 8086/8088 微处理器	54
4.1.1 8086/8088CPU 的结构	54
4.1.2 8088/8086 的寄存器	56
4.1.3 访问存储器	60
4.1.4 存储器的组成	60
4.1.5 存储器的段结构	61
4.1.6 逻辑地址与物理地址	62
4.2 堆栈	62
4.3 8086/8088 的 CPU 总线	63
4.4 8088 的主要时序	67
4.4.1 最小组态下的时序	68
4.4.2 最大组态下的时序	69
习题 4	72

汇编语言部分

第 5 章 86 系列微机的指令系统	77
5.1 寻址方式	77
5.1.1 隐含寻址	77
5.1.2 立即寻址	77
5.1.3 寄存器寻址	77

5.1.4	存储器操作数的寻址方式	78
5.1.5	段基值的隐含约定	80
5.1.6	隐含段的改变	80
5.2	指令系统	81
5.2.1	指令系统概述	81
5.2.2	传送类指令	82
5.2.3	算术运算类指令	86
5.2.4	位操作指令	94
5.2.5	转移类指令	97
5.2.6	串操作指令	99
5.2.7	处理器控制类指令	102
5.3	指令应用举例	103
习题 5		105
第 6 章	汇编语言	107
6.1	汇编语言语句种类及格式	107
6.1.1	语句种类	107
6.1.2	语句格式	108
6.2	汇编语言的数据	109
6.2.1	常数	109
6.2.2	变量	110
6.2.3	标号	113
6.2.4	段名和过程名	114
6.3	汇编语言运算符	114
6.3.1	算术运算符	114
6.3.2	逻辑运算符	116
6.3.3	关系运算符	116
6.3.4	属性值返回运算符	117
6.3.5	属性修改运算符	120
6.3.6	运算符的优先级	121
6.4	常用伪指令	122
6.4.1	段定义伪指令	122
6.4.2	段指定伪指令	125
6.4.3	汇编地址计数器和定位伪指令	125
6.4.4	等值语句	126
6.4.5	等号语句	127
6.4.6	源程序结束伪指令	127
6.5	伪指令应用举例	128
6.6	汇编语言程序的调试	130
6.6.1	汇编程序调试步骤	130

6.6.2	汇编	131
6.6.3	连接	133
6.6.4	调试与运行	134
习题 6		141
第 7 章	汇编语言程序设计	144
7.1	程序设计方法概述	144
7.1.1	程序设计的步骤	144
7.1.2	程序的基本控制结构	146
7.1.3	程序设计方法	147
7.2	汇编语言源程序的基本格式和编程步骤	147
7.3	顺序结构程序设计	150
7.4	系统功能调用	152
7.4.1	系统功能调用方法	152
7.4.2	常用系统功能调用	153
7.5	分支结构程序设计	156
7.5.1	转移指令	157
7.5.2	分支结构程序设计	158
7.5.3	多分支结构程序设计	162
7.6	循环结构程序设计	164
7.6.1	循环程序的控制方法	164
7.6.2	单重循环程序设计	166
7.6.3	多重循环程序设计	169
7.7	子程序设计	175
7.7.1	子程序概念	175
7.7.2	子程序的定义	176
7.7.3	子程序的调用和返回	176
7.7.4	子程序设计方法	177
7.8	宏功能程序设计	186
7.8.1	宏的概念	186
7.8.2	宏定义和宏调用	186
7.8.3	参数的使用	189
习题 7		195
第 8 章	汇编语言应用案例	199
8.1	字符处理	199
8.1.1	灵活运用串操作指令	199
8.1.2	字符处理举例	200
8.2	数码转换	202
8.2.1	BCD 码十进制调整原理	202

8.2.2 BCD 码十进制调整指令的应用	203
8.3 表的应用	205
8.3.1 应用 XLAT 指令	205
8.3.2 应用举例	206
8.4 逻辑运算	207
8.4.1 关于逻辑运算与移位指令	207
8.4.2 应用举例	208
8.5 声音与乐曲	209
8.5.1 声音的产生	209
8.5.2 演奏乐曲	209
习题 8	211

微型计算机接口部分

第 9 章 输入输出接口	215
9.1 接口技术	215
9.1.1 基本概念	215
9.1.2 功能与组成	218
9.2 I/O 端口地址译码技术	220
9.2.1 I/O 端口及共用技术	220
9.2.2 I/O 端口地址编址方式	221
9.2.3 I/O 端口地址译码的方法	223
9.2.4 I/O 端口地址译码电路设计	224
9.3 接口中的数据	226
9.3.1 接口中的寄存器	226
9.3.2 接口中的三态缓冲器	226
9.3.3 接口中的数据应用举例	227
习题 9	229
第 10 章 定时/计数技术和中断技术	230
10.1 定时/计数技术	230
10.1.1 8254 的内部逻辑结构	230
10.1.2 8254 的外部特性	231
10.1.3 8254 的工作方式	232
10.1.4 8254 的编程命令	235
10.1.5 8254 的应用	236
10.2 中断技术	238
10.2.1 中断控制器	238
10.2.2 中断服务程序的设计	245
习题 10	248

第 11 章 并行接口与串行接口	250
11.1 并行通信与并行接口	250
11.1.1 并行通信	250
11.1.2 并行接口	250
11.2 可编程并行接口芯片 8255A	251
11.2.1 8255A 的基本特性	251
11.2.2 8255A 外部特性	251
11.2.3 8255A 内部结构	252
11.2.4 8255A 的控制字	254
11.2.5 8255A 工作方式及应用	255
11.3 串行通信与串行接口	263
11.3.1 串行通信	263
11.3.2 串行接口	264
11.3.3 串行通信的数据传送方向	265
11.3.4 串行通信的基本方式	265
11.3.5 串行通信中的术语	266
11.3.6 串行通信中的调制与解调	267
11.4 可编程串行接口芯片 8251A	268
11.4.1 8251A 的外部特性	268
11.4.2 8251A 的内部结构与功能	270
11.4.3 8251A 的控制字	272
11.4.4 8251A 的应用	275
习题 11	278
第 12 章 A/D 与 D/A 转换器接口	279
12.1 A/D 转换器	279
12.1.1 A/D 转换器工作原理	279
12.1.2 A/D 转换器性能指标及连接特性	281
12.1.3 A/D 转换器 ADC0809	282
12.2 D/A 转换器	284
12.2.1 D/A 转换器工作原理	285
12.2.2 D/A 转换器的性能指标及连接特性	286
12.2.3 D/A 转换器 DAC0832	287
习题 12	289
附录 A IBM PC 的键盘输入码和 CRT 显示码	290
附录 B 出错信息	292
参考文献	300

微机原理部分

- 第1章 运算器
- 第2章 存储部件
- 第3章 控制器
- 第4章 微处理器

电子计算机是作为计算工具被发明出来的,计算是其最基本的功能。计算机中实现计算功能的部件被称作运算器,电子元件是构成运算器的基础。根据电子元件的特性,按照运算规则,把电子元件组织在一起,就可以实现具有计算能力的运算器。

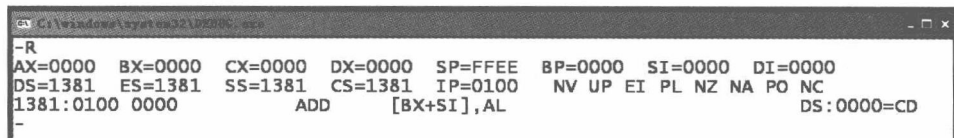
1.1 计算机中的数据

人们日常习惯使用的十进制数,在计算机中实现起来并不方便。所以,计算机中没有采用十进制数。进而,运算规则也不是十进制的运算规则。

1.1.1 数据的表示

首先,了解一下计算机中数据的样子。

打开一台装有 Windows XP 的微机(32 位的 Win7 也可以,如果是 64 位的,就要安装一个 32 位模拟器)。用鼠标单击屏幕左下角的“开始”按钮,然后单击弹出菜单中的“运行”,用键盘输入“DEBUG”,按“回车”键。将会看到一个弹出的窗口,左上角有一个减号,后面跟着一个闪动的下划线。那个减号叫做提示符,是在提示“可以输入 DEBUG 命令了”。那个闪动的下划线,就是输入 DEBUG 命令的位置。输入“R”、“回车”,屏幕上会出现如图 1.1 所示的画面。



```
C:\windows\system32\DEBUG.exe
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0100 NV UP EI PL NZ NA PO NC
1381:0100 0000      ADD     [BX+SI],AL      DS:0000=CD
-
```

图 1.1 DEBUG 中的 R 命令

等号右面的数,就是计算机中的数据。等号左面是保存数据的部件的名称。这个部件叫做寄存器,AX、BX...都是寄存器。刚才在提示符后面输入的那个字母 R,就是“显示寄存器内容”命令(在 DEBUG 中大小写字母的含义是一样的)。

既然寄存器是保存数据的,其内容就可以修改。输入“R AX”、“回车”,屏幕上出现如图 1.2 所示的画面。

```
-R AX
AX 0000
:-
```

图 1.2 DEBUG 中的 R AX 命令

此时,计算机在等待操作者输入新的数据。输入“8”、“回车”,这时,寄存器 AX 中的数据就变成了 8。用 R 命令看一下寄存器的新值:输入“R”、“回车”,屏幕上出现如图 1.3 所示的画面。

```
:8
-R
AX=0008 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0100 NV UP EI PL NZ NA PO NC
1381:0100 0000 ADD [BX+SI],AL DS:0000=CD
:-
```

图 1.3 寄存器 AX 的新值

计算机中的数据,当然是可以进行计算的。下面实现一个“寄存器 AX 加 1”的功能。输入“A 100”、“回车”,屏幕上出现如图 1.4 所示的画面。

```
-A 100
1381:0100 _
```

图 1.4 DEBUG 中的 A 100 命令

输入“INC AX”、“回车”。屏幕上出现如图 1.5 所示的画面。

```
1381:0100 INC AX
1381:0101 _
```

图 1.5 输入 INC AX 指令之后

再输入一个“回车”,结束输入。用 R 命令看一下寄存器的值:输入“R”、“回车”,屏幕上出现如图 1.6 所示的画面。

```
-R
AX=0008 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0100 NV UP EI PL NZ NA PO NC
1381:0100 40 INC AX
:-
```

图 1.6 DEBUG 的 T 命令执行之前

寄存器 AX 的内容还是 8,并没有看到“寄存器 AX 加 1”的效果。继续输入“T=100”、“回车”。屏幕上出现如图 1.7 所示的画面。

```
-T=100
AX=0009 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0101 NV UP EI PL NZ NA PE NC
1381:0101 0000 ADD [BX+SI],AL DS:0000=CD
:-
```

图 1.7 DEBUG 的 T 命令

可以看到,寄存器 AX 中的数据已经变成了 9,实现了加 1 功能。这是因为刚才输入的 INC AX,其功能是“寄存器 AX 加 1”。INC AX 是一句汇编指令,是汇编语言中的一条指令。汇编语言中有很多种指令,可以对计算机中的数据进行各种处理。INC 是用来描述指令功能的,被称作助记符。INC 是 increase 的缩写,助记符都是英语单词或缩写。汇编语言

是这本书向大家介绍的一个重要内容。

前面输入的 DEBUG 命令 A 100, 其中 A 的含义是向计算机中输入汇编指令。100 的含义是输入的指令保存在内存中 100 这个位置。内存是计算机中另一个保存数据的部件。寄存器的数量比较少, 一般只有十几个到几十个。而内存的容量则大得多, 可以保存很多数据。现在的微机, 常见的是配备几个 G 的内存。1G=2³⁰=1 073 741 824, 比十亿稍多一点。数据放入内存之后, 为了以后使用的时候能够找到它, 需要记住数据所存放的位置。内存是通过编号来区分不同的位置的。上面提到的那个 100, 就是内存中一个位置的编号。这个编号, 有一个很形象的名称——地址。

输入完 INC AX 指令之后, 输入“回车”, 光标移到下一行。这时, 计算机是在等待输入下一条指令。需要再一次输入“回车”, 退出“输入汇编指令”状态, 退回到“输入 DEBUG 命令”状态。

随后输入的 R 命令, 没有看到寄存器 AX 有变化。这是因为 A 命令只是把指令 INC AX 存入了内存。要想让指令发挥作用, 需要“执行指令”。

随后输入的 T=100, T 的含义是执行一条汇编指令, 100 的含义是所执行的是内存中编号是 100 的那个位置的指令。输入“回车”之后, 内存中 100 这个位置存放的那句指令 INC AX, 就被执行了。于是, 寄存器 AX 中的 8 就变成了 9。

内存中的指令, 可以再次执行。输入“T=100”、“回车”, 屏幕上出现如图 1.8 所示的画面。

```
-T=100
AX=000A BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0101 NV UP EI PL NZ NA PE NC
1381:0101 0000      ADD     [BX+SI],AL      DS:0000=CD
```

图 1.8 第二次执行 T 命令

寄存器 AX 再次加 1, 从 9 变成了 A。A 是什么? 9 加 1 应该是 10 呀?

1.1.2 二进制数

原来, 这个 A 是个十六进制数, 就等于十进制的 10。那么, 为什么要用十六进制数呢?

原因是这样的。计算机要保存一个十进制数, 就需要一种具有 10 个稳定状态的物理器件, 10 种状态分别对应 0~9。这种器件很难实现。而具有两个稳定状态的器件就很容易实现。这样, 计算机采用二进制数, 就很容易实现数据的保存。另外, 二进制的运算规则也比十进制简单得多, 运算部件的实现也就容易得多。所以, 计算机内部采用的是二进制数。

二进制数有 0 和 1 两个符号, 含义与十进制一样。下一个数应该是 2。但是, 二进制数里没有 2, 只有 0 和 1, 1 是最大的 1 位数。现在的情况类似于十进制中的 9。9 是最大的 1 位十进制数。如果想要表示比 9 大 1 个数, 只能用增加位数的方法, 表示为一个两位十进制数 10。同样的道理, 1 是最大的 1 位二进制数, 如果想要表示比 1 大 1 个数, 也只能用增加位数的方法, 表示为一个两位二进制数 10。这个 10 表示的是 2。随后是: 11 表示 3, 100 表示 4, 101 表示 5 等等。

其实, 二进制数与十进制数还是很像的。一位数时, 用一个符号表示。一位十进制数最