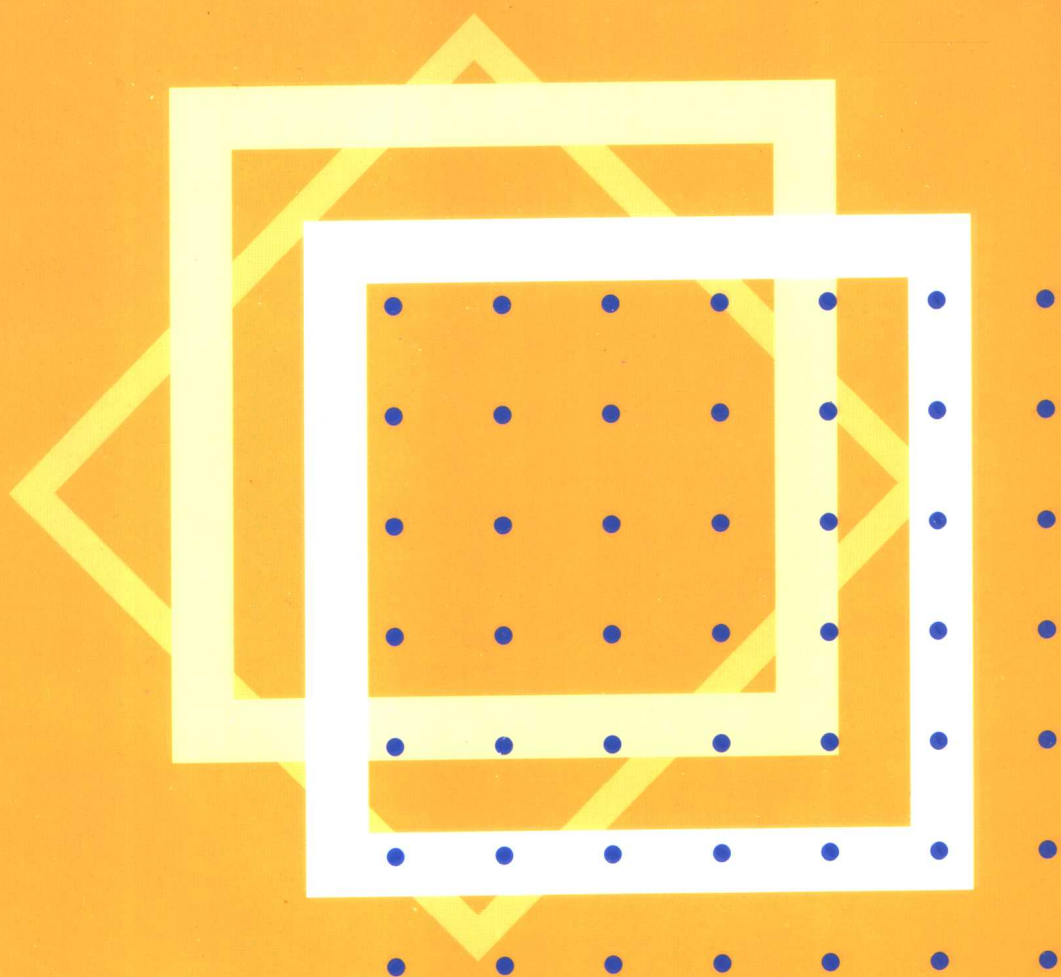


现代数字系统 实践教程

吴新开 曾照福 刘昆山 编著

XIANDAI SHUZIXITONG SHIJIAN JIAOCHENG



现代数字系统实践教程

吴新开 曾照福 刘昆山 编著

人民邮电出版社

图书在版编目 (CIP) 数据

现代数字系统实践教程 / 吴新开, 曾照福, 刘昆山编著.

—北京: 人民邮电出版社, 2004.6

ISBN 7-115-12119-2

I. 现... II. ①吴...②曾...③刘... III. 数字系统—高等学校—教材 IV. TP271

中国版本图书馆 CIP 数据核字 (2004) 第 022848 号

内 容 提 要

本书作为《电工电子实践教程》的姊妹篇, 重点介绍了现代数字系统——微机原理与接口技术、可编程序控制器及可编程逻辑器件应用技术的实践与实验指导内容。书中实验以设计性、综合性和研究性实验内容为主, 辅以部分验证性实验内容。本书可作为工科院校本科及研究生学习电工电子基础课程配套的实验指导教材, 也可供工程技术人员学习现代数字系统设计技术时参考。

现代数字系统实践教程

◆ 编 著 吴新开 曾照福 刘昆山

责任编辑 张 鹏

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67129264

北京汉魂图文设计有限公司制作

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 18

字数: 437 千字

2004 年 6 月第 1 版

印数: 1-6 000 册

2004 年 6 月北京第 1 次印刷

ISBN 7-115-12119-2/TN · 2255

定价: 24.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

《电工电子实践教程》一书出版后,得到了社会各界特别是高等学校同仁们的支持与鼓励,使《电工电子实践教程》在一年时间内发行量突破 12000 册。这表明全国高等学校迫切需要一本应用教学改革研究成果、使电工电子实验内容真正达到国家教育部示范性实验室建设标准的实践教材。因此,我们决定将微机原理与接口技术、可编程逻辑器件与应用技术和可编程序控制器应用技术等课程的实践教学内容编写出版《现代数字系统实践教程》一书,作为对《电工电子实践教程》的补充和扩展。

随着科学技术的发展与进步,电工电子技术正以空前的速度向各行业、各领域渗透,特别是计算机、微电子技术的应用与发展,使传统的电工电子技术发生了重大变革。

1. 微电子技术与计算机技术的发展,使电子系统的可靠性得到了空前的提高。现在,人造卫星、宇宙飞船、巡航导弹、自动化生产线、自动化工厂无一不在采用现代数字系统进行控制。

2. 微电子技术与计算机技术的发展,使电子系统的体积和重量大幅度减小。现代电子技术的发展与应用,使复杂的计算机系统在性能要求不变的情况下,能集成在一块超大规模集成电路芯片上,从而使系统的体积变得越来越小,重量变得越来越轻。

3. 微电子技术与计算机技术的发展,使电子系统变得越来越复杂,其功能变得越来越强。许多过去连想都不敢想的事情,现在变得越来越容易,许多功能的实现,不再依靠传统的硬件电路来实现,而是采用软件来实现,因此,软、硬件功能的相互替换与交融,使系统的功能得以迅速扩大。

4. 微电子技术与计算机技术的发展,促使模拟电子系统向数字电子系统转变,使系统的性能得到了提高。特别是数字系统不存在模拟电子系统的温度漂移等优点,使数字电子系统得到了飞速的发展与应用,如传统的电视接收机系统,正在向数字式彩色电视机系统进行转变。

正是数字电子电路的上述优点,使数字系统的设计及综合变得越来越重要,新的数字电路和系统也不断涌现,为了帮助学生掌握现代数字系统的综合设计技术,我们在此教程中,将主要介绍现代数字系统的基本知识、基础实验内容和综合设计所必需的基础知识,内容包括微机原理的基础实验和综合性实验内容,可编程序控制器 FXON 的基本知识和应用技术、基础实验和设计性、综合性、研究性实验内容,可编程逻辑器件(PLD)的基础知识、应用技巧、基础实验和综合性、研究性实验内容。

本教材作为基础课程的实践教程,在编写过程中,得到了湖南科技大学有关领导和老师们的关心和支 持,如果没有他们的支持与关心,要在短期内出版这样一本教材是不可能的,在此谨向支持和关心该教材的编写和出版的所有同仁致以衷心的感谢。在整个编写过程中,还得到了许多研究生和本科生的帮助,他们负责了全书图表的绘制,在此也要感谢他们为本书的出版所付出的辛勤劳动。

本教材由曾照福讲师编写第一篇,刘昆山高级工程师编写第二篇,吴新开教授编写第三

篇，整书由吴新开教授负责统稿、文字润色。

尽管我们做了许多工作，力争使错误做到最小，但错误仍不可避免，恳请使用此书的教师和学生继续提出宝贵意见，以便该教材修改重印。

编 者

于湖南科技大学南校区

目 录

第一篇 微机原理与接口技术

第 1 章	汇编语言程序的上机及调试	3
1.1	汇编语言程序的上机过程	3
1.1.1	用编辑程序建立和修改源程序	3
1.1.2	用 ASM 或 MASM 命令产生目标文件	3
1.1.3	用 LINK 命令产生可执行文件	5
1.1.4	用 DEBUG 调试程序	5
1.2	动态调试程序 DEBUG 的使用	6
1.2.1	DEBUG 程序的调用	6
1.2.2	DEBUG 的主要命令	6
第 2 章	汇编语言与微机应用单元实验	11
2.1	汇编语言实验	11
2.1.1	实验 1 DEBUG 软件的使用	11
2.1.2	实验 2 汇编语言程序的上机和调试	14
2.1.3	实验 3 顺序程序设计实验	15
2.1.4	实验 4 循环程序设计	16
2.1.5	实验 5 分支程序设计	17
2.1.6	实验 6 子程序设计	17
2.1.7	实验 7 DOS 系统功能调用	18
2.1.8	实验 8 显示目录	18
2.1.9	实验 9 BIOS 功能调用	20
2.2	微机应用单元实验	21
2.2.1	实验 10 简单输入输出程序设计	21
2.2.2	实验 11 8255 并行接口实验	23
2.2.3	实验 12 数码管显示器接口实验	24
2.2.4	实验 13 8253 (8254) 计数器/定时器实验	25
2.2.5	实验 14 声音接口实验	26
2.2.6	实验 15 8259A 中断控制器实验	27
2.2.7	实验 16 RAM 扩充实验	27
2.2.8	实验 17 DMA 传送实验	29
2.2.9	实验 18 8250 异步通信控制器实验	30
2.2.10	实验 19 A/D 转换实验	31

2.2.11	实验 20 D/A 转换实验	32
第 3 章	微机原理应用综合设计	34
3.1	微机应用系统的一般设计方法	34
3.2	综合设计要求	38
3.2.1	硬件实验装置自检	38
3.2.2	电子时钟设计	38
3.2.3	采用中断控制的数据采集系统设计	39
3.2.4	模拟交通灯控制	39
3.2.5	电子琴	40
3.2.6	A/D-D/A 联合实验	40
3.2.7	语音记录与重放	40
3.2.8	微机化数字式密码锁实验	43
第二篇 PLC 原理及应用技术		
第 4 章	FXON-60MR 实验装置及编程技巧简介	47
4.1	FXON-60MR 主机特性介绍	47
4.1.1	技术指标	47
4.1.2	特殊元器件	49
4.1.3	基本逻辑指令	51
4.1.4	步进顺控图指令	58
4.1.5	功能指令	60
4.2	FX-10P-E 手持式编程器介绍	81
4.2.1	概述	81
4.2.2	HPP 的组成与操作面板	82
4.2.3	编程操作	84
4.2.4	监视/测试操作	87
4.3	计算机编程软件 SWOPC-FXGP/WIN-C 使用说明	88
4.3.1	概述	88
4.3.2	操作环境	89
4.3.3	Fxgpwin 软件的界面与基本环境	89
4.4	实验装置简介	97
第 5 章	PLC 基础实验	100
5.1	指令训练实验	100
5.1.1	实验 1 与、或、非基本逻辑指令	100
5.1.2	实验 2 置位、复位及脉冲指令	102
5.1.3	实验 3 栈及主控指令	104
5.1.4	实验 4 定时器、计数器指令	106

5.1.5	实验 5 步进、顺序控制指令	107
5.1.6	实验 6 跳转、比较、传送指令	110
5.1.7	实验 7 四则运算与逻辑运算指令	111
5.1.8	实验 8 移位指令	112
5.2	基础实验	113
5.2.1	实验 9 电机正反转及能耗制动	113
5.2.2	实验 10 回转刀盘选刀控制	114
5.2.3	实验 11 自动送料系统	115

第 6 章	PLC 综合应用实验	118
6.1	实验 12 交通灯信号控制	118
6.2	实验 13 自动轧钢控制	119
6.3	实验 14 抢答器及八段码显示	120
6.4	实验 15 产品自动检测分拣系统	120
6.5	实验 16 电梯控制	121

第三篇 可编程逻辑器件应用技术

第 7 章	在系统可编程逻辑器件	125
7.1	可编程逻辑器件及其分类	125
7.1.1	PLD 的优点	125
7.1.2	PLD 的分类	125
7.1.3	PLD 电路表示法	127
7.2	可编程逻辑器件的基本结构	128
7.2.1	PROM 的结构	128
7.2.2	PAL 和 GAL 的结构	128
7.2.3	PLA 的结构	129
7.2.4	PAL 的输出和反馈结构	129
7.3	在系统可编程逻辑器件	130
7.4	ispLSI 器件编程概述	137

第 8 章	ABEL_HDL 语言和 ISP Synario 设计软件	139
8.1	ABEL_HDL 语言	139
8.1.1	ABEL_HDL 语言的基本语法	139
8.1.2	DIRECTIVES 指示字	146
8.1.3	ABEL_HDL 源文件的基本结构	148
8.2	ISP Synario 设计软件	151
8.2.1	ISP Synario 设计软件的常用命令	151
8.2.2	“Processes for Current Source” 源过程窗口的处理	152
8.2.3	ISP Synario 软件的几类窗口	153

8.2.4	电路原理图编辑器各类主要菜单的功能	153
8.2.5	ISP Synario 软件的文件后缀及其含义	155
8.3	ISP Synario 软件设计实践	156
8.3.1	ISP Synario 软件的安装	156
8.3.2	ISP Synario 原理图输入设计	156
8.3.3	设计编译与仿真	159
8.3.4	ABEL 语言和原理图混合输入	160
8.3.5	在系统编程操作方法	164
第 9 章	VHDL 语言及 EDA 设计工具简介	165
9.1	VHDL 语言基础	165
9.1.1	VHDL 语言基本语法	165
9.1.2	VHDL 语言的结构	170
9.1.3	VHDL 语言的基本描述方法	179
9.1.4	面向仿真与综合的 VHDL 语言设计	190
9.2	'ALTERA 公司的“MAX+PLUS II”EDA 设计工具的使用方法	196
9.3	XILINX 公司的“Foundation Series”EDA 设计工具的使用方法	200
第 10 章	实验设计实例及应用设计项目	204
10.1	数字电路基础实验	204
10.1.1	交通控制灯监视电路	204
10.1.2	码变换电路	206
10.1.3	双向移位寄存器	209
10.1.4	计数器	212
10.1.5	排序器	215
10.1.6	特征位串检测器	218
10.2	ISP 技术应用设计方法	220
10.2.1	ISP 技术设计方法简介	220
10.2.2	ISP 技术设计方法举例	221
10.3	ISP 技术应用设计项目	224
10.3.1	篮球 30 秒定时控制电路设计实例 (ABEL 语言)	224
10.3.2	图像序列累加器的 FPGA 设计 (VHDL 语言)	227
10.3.3	ISP 技术应用设计项目	230
附 录		235
附录 1	8086/8088 指令系统	235
附录 2	MS-DOS 的功能调用	251
附录 3	BIOS 调用	255
附录 4	宏汇编错误信息表	260

附录 5 I/O 通道插槽 (ISA 插槽) 引脚信号	263
附录 6 I/O 地址分配	265
附录 7 常用集成电路引脚图	266
附录 8 ISP 技术综合实验系统简介	269
附录 9 部分可编程逻辑器件生产厂的下载电路	272
附录 10 常用可编程逻辑器件引脚图	275
主要参考文献	278

第 一 篇

微机原理与接口技术

第 1 章 汇编语言程序的上机及调试

1.1 汇编语言程序的上机过程

汇编语言程序的上机过程如图 1-1 所示。

由图 1-1 可以看出，汇编语言的上机过程主要由 4 个部分组成。

1.1.1 用编辑程序建立和修改源程序

汇编源程序是文本文件，可以用任何文本文件的编辑软件来建立和修改源程序。常用的编辑软件有 EDLIN、EDIT、NE、QE、WORDSTAR、WPS（N 命令）、PE、TC 等，推荐采用 EDIT 或 TC 来编辑源文件（注意：文件名必须以 ASM 作为扩展名）。

1.1.2 用 ASM 或 MASM 命令产生目标文件

源程序建立后，就可用汇编程序 ASM 或 MASM 将源文件转换为用二进制代码表示的目标文件。在汇编过程中，汇编程序对源程序进行两遍扫描，第一遍扫描源程序产生符号表、处理伪指令等，第二遍产生机器指令代码、确定数据等。

1. 汇编过程

假设源程序文件名为 ZZF.ASM，执行命令：

C:\>masm

Microsoft (R) Macro Assembler Version 5.00

Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Source filename [.ASM]: ZZF

Object filename [ZZF.OBJ]:

Source listing [NUL.LST]: ZZF

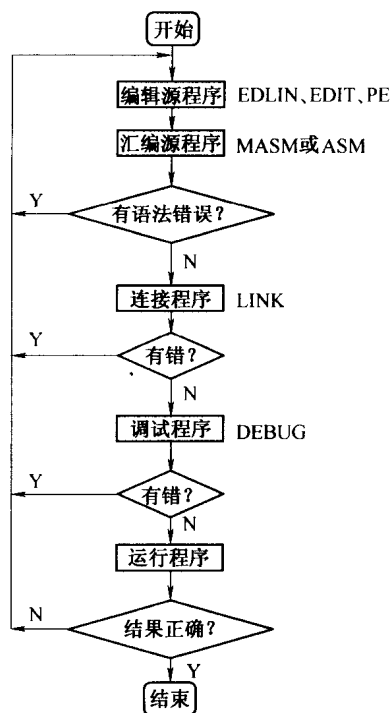


图 1-1 汇编语言上机调试流程图

Cross-reference [NUL.CRF]: ZZF ✓

50424 + 419832 Bytes symbol space free

0 Warning Errors

0 Severe Errors

其中划线部分为用户键入部分，ZZF 为源程序名 (ZZF.ASM)，方括号内的为机器规定的默认文件名，通常直接按回车键，表示采用默认文件名。如果不想要列表文件 (.LST) 和交叉索引文件 (.CRF)，则可在[NUL.LST]和[NUL.CRF]后直接按回车键。

在回答完上述 4 个询问后，汇编程序就对源程序进行汇编。若汇编过程中发现源程序中有语法错误，则提示出错信息，指出是什么性质的错误，即错误的类型，最后列出错误的总数。此时，可根据错误类型及列表文件给出的错误提示，找出问题和原因，再用编辑程序修改源程序，重新汇编，直到汇编通过为止。当然汇编过程只能指出源程序中的语法错误，并不能指出算法错误和其他错误。

如果在汇编时不需要产生列表文件和交叉索引文件，则调用汇编程序时用分号结束。如输入 MASM ZZF; 即可。

如果要产生目标文件 (.OBJ) 和列表文件，不需要交叉索引文件，则在分号前面加两个逗号。如输入 MASM ZZF,; 即可。

如果 4 个文件都要，则只要在分号前面加 3 个逗号。如输入 MASM ZZF,,; 即可。

2. 列表文件 (.LST)

汇编后建立的列表文件可用任何文本文件的编辑软件打开或用 TYPE 命令显示、打印，以便分析调试源程序。

列表文件由 3 部分组成：

(1) 源程序和目标程序清单

列表文件同时列出源程序和对应的机器码，第一列给出每条指令所在行号；第二列给出从段的首地址开始的每条指令存放的偏移地址；接下来的是数字列，给出对应每条语句的机器码和对应于存放在堆栈段和数据段的值，在机器码后加上“R”的指令表示这条指令在连接时可能产生与列出来的偏移地址不同的地址，因为这些偏移地址可能与其他模块有关；最右边为源程序。

(2) 段信息汇总表

在段信息汇总表中列出该程序用了哪几个段，每个段的定位类型以及段的组合类型。

(3) 符号汇总表

在列表文件的最后部分列出了用户在源程序中定义的符号名、类型、值和所在段。如果在源程序中存在某些语法错误时，列表文件将出错提示显示在出错指令行的下面，这样可从列表文件中很快查找到错误行，以便调试。另外，由于列表文件给出了各条指令的偏移地址，对调试程序时设置断点很方便。

3. 交叉索引文件 (.CRF)

交叉索引文件给出了用户定义的所有符号，对每个符号都列出了将其定义的所在行号和

引用的行号，并在定义行号上加上“#”号，同列表文件一样，交叉索引文件也是为了便于调试而设置的，对于一些规模较大的程序，交叉索引文件为调试工作带来很大的方便。

如果要查看这个符号表，必须用 **CREF.EXE** 文件将交叉索引文件转换为扩展名为 **.REF** 的文件，然后再用任何文本文件的编辑软件打开或用 **TYPE** 命令显示。具体操作方法如下：

```
C:\>cref✓  
Microsoft (R) Cross-Reference Utility  Version 5.00  
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.  
Cross-reference [.CRF]: ZZF✓  
Listing [ZZF.REF]: ✓
```

3 Symbols

1.1.3 用 **LINK** 命令产生可执行文件

由汇编程序建立的目标文件必须经过连接后，才能成为可执行文件。连接过程如下：

```
C:\>link✓  
Microsoft (R) Overlay Linker  Version 3.60  
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.  
  
Object Modules [.OBJ]: ZZF✓  
Run File [ZZF.EXE]: ✓  
List File [NUL.MAP]: ZZF✓  
Libraries [.LIB]: ✓
```

其中划线部分为用户键入部分，**ZZF** 为目标文件名，方括号内的为机器规定的默认文件名，通常直接按回车键，对扩展名为 **MAP** 的列表分配文件是否要建立，由用户决定，它给出每个段在内存中的地址分配情况及长度，可用任何文本文件的编辑软件打开或用 **TYPE** 命令显示。

最后一个询问是问是否在连接时用到库文件，对于连接汇编语言源程序的目标文件，通常是不需要的，可直接键入回车键。

与汇编程序一样，可以在连接时用分号结束后续询问。若连接后只要产生可执行文件（**.EXE**），则键入 **LINK 目标文件**;✓即可。若要产生 **.MAP** 文件，则在分号前加两个逗号，即键入 **LINK 目标文件**;;✓。

1.1.4 用 **DEBUG** 调试程序

由 **LINK** 命令产生的可执行文件在 **DOS** 操作系统下就可以运行了。但是，绝大部分程序

初次运行结果都是不正确的，必须通过 DEBUG 程序来进行调试、运行，以发现和改正错误。

1.2 动态调试程序 DEBUG 的使用

1.2.1 DEBUG 程序的调用

在 DOS 的提示符下，可键入命令：

```
C:\>DEBUG [drive:][path] filename [test file-parameters]
```

其中 [] 的内容为可选项。

[drive:] 为驱动器号，指要调入 DEBUG 状态的可执行文件在哪个驱动器中，如 A:、B:、C: 等。

[path] 为路径，指要调入 DEBUG 状态的可执行文件在哪个目录下或子目录下。

filename 指要调入 DEBUG 状态的可执行文件的文件名，该文件可以通过编辑、汇编、连接后产生的可执行文件，也可以是在 DEBUG 状态下通过写盘命令写入磁盘的文件。

[test file-parameters] 为可选参数，是给定文件的说明参数。

在启动 DEBUG 时，如果输入文件名，则 DEBUG 程序把指定文件装入内存，用户可以通过 DEBUG 有关命令对指定文件进行修改、显示或执行。如果没有文件名，则是以当前内存的内容工作，或者用命名命令或装入命令把需要的文件装入内存，然后再通过 DEBUG 命令进行修改、显示或执行。

当启动 DEBUG 程序后，屏幕上出现 “-”，如无任何提示说明正确，可进行调试。

1.2.2 DEBUG 的主要命令

DEBUG 命令是在 DEBUG 提示符 ‘-’ 下，由键盘键入的。每条命令以单个字母的命令符开头，然后是命令的操作参数，操作参数与操作参数之间用空格或逗号隔开，操作参数与命令符之间用空格隔开，命令的结束符是回车键 ‘↵’ (<ENTER>)。命令及参数的输入可以是大小写的结合，Ctrl+Break 键可中止命令的执行。Ctrl+Num Lock 键可暂停屏幕卷动，按其他键继续。命令所用数均为十六进制数，且不必写 H。若 DEBUG 检查出一个语法错误，则显示具有错误的行和指示错误所在。

1. 汇编命令 A (Assemble)

格式：(1) A<段寄存器名>: <偏移地址>

(2) A<段地址>: <偏移地址>

(3) A<偏移地址>

(4) A

功能：键入该命令后显示段地址和偏移地址并等待用户从键盘逐条键入汇编命令，逐条汇编成代码指令，直到显示下一地址时用户直接键入回车键返回到提示符 ‘-’。

其中(1)用指定段寄存器内容作段地址,(3)用 CS 内容作段地址,(4)以 CS: 100 作地址。

以后命令中提及的各种“地址”形式,均指(1)、(2)、(3)中 A 后的地址形式。

2. 比较命令 C (Compare)

格式: C<源地址范围>, <目标地址>

其中<范围>是由<起始地址><终止地址>指出的一片连续单元,或<起始地址>L<长度>。

功能:从<源地址范围>的起始地址单元起逐个与目标起始地址以后的单元顺序比较单元的内容,直到源终止地址为止。遇有不一致时以<源地址><源内容><目标内容><目标地址>形式显示失配单元及内容。

3. 显示内存命令 D (Dump)

格式:(1) D<地址>

(2) D<范围>

(3) D

功能:以两种形式显示指定内存范围内容。一种形式为十六进制内容,一种形式为相应字节的 ASCII 码字符,对不可见字符以“.”代替。

格式(1)从<地址>指定的内存单元开始显示 128 个字节的内容;格式(2)显示的字节数由地址范围来决定,在范围中包含起始地址和结尾地址格式;格式(3)则从上一个 D 命令所显示的最后一个单元的下一个单元开始显示 128 个字节。若以前没有使用过 D 命令,则以 DEBUG 初始化的段寄存器的内容为起始地址,起始偏移地址为 100H,即 DS: 100。

若在命令中所打入的地址中,只包含起始地址的偏移量,则 D 命令认为段地址包含在 DS 中。

4. 修改内存命令 E (Enter)

格式:(1) E<地址><单元内容>

(2) E<地址><单元内容表>

其中<单元内容>是一个十六进制数,或用引号‘或“括起来的字符串;<单元内容表>是以逗号分隔的十六进制数,用‘或“括起来的字符串,或者是二者的组合。

功能:(1)将指定内容写入指定单元后显示下一地址,可连续键入修改内容,直到新地址出现后键入回车键为止。(2)将<单元内容表>逐一写入由<地址>起始的一片单元。

5. 填充内存命令 F (Fill)

格式: F<范围><单元内容表>

功能:将单元内容表中的值逐个填入指定范围,单元内容表中内容用完后重复使用。

6. 运行命令 G (Go)

格式: G<=地址><断点地址 1[断点地址 2 ...断点地址 10]>

功能:从指定<地址>开始执行程序(如地址缺省则从当前 CS:IP 开始),运行至<断点地址 1>停止,显示所有寄存器及标志位内容与下一条指令,如后面还有断点,可键入 G 继续执行。

示例: -G 001A 则执行从当前 CS: IP 至 001A 的指令。

注意:地址设置必须从指令的第一字节设起。