

• 高等学校规划教材



32位微机

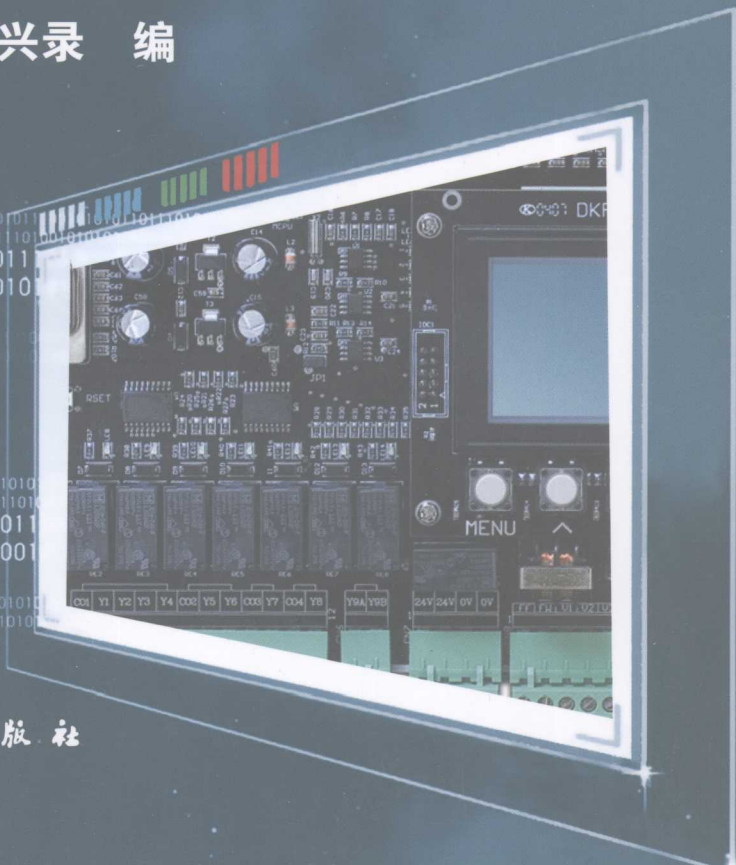
原理与接口技术实验指导



32WEI WEIJI

YUANLI YU JIEKOU JISHU SHIYAN ZHIDAO

宋廷强 陈为 马兴录 编



化学工业出版社

高等学校规划教材

32 位微机原理与接口技术 实验指导

宋廷强 陈 为 马兴录 编



化学工业出版社

· 北 京 ·

本书是《32 位微机原理与接口技术》配套的实验与课程设计指导书。全书从实验基础知识、实验设备、汇编实验、硬件接口实验等内容出发,给出了软硬件实验的基本原理、程序流程图和参考程序。基础实验中包括了 8 个汇编语言程序设计实验和 19 个硬件接口实验,在综合设计实验项目中,设置了 2 个汇编语言程序设计和 8 个硬件接口设计实验,给出了软、硬件实验的原理与设计思路,可供读者在上机实验和课程设计时参考。

本书实验项目基于先进的嵌入式微机原理实验箱进行设计,实验项目切合实际,涉及接口种类丰富,融合了多种微机应用技术,是学习微机原理与接口技术课程的实用教材。

本书可以作为高等院校理工科电子信息、自动化、电气工程、机电等相关专业任课教师的教学用书,以及本、专科学生实验参考用书,也可供工程技术人员参考。

图书在版编目(CIP)数据

32 位微机原理与接口技术实验指导 / 宋廷强, 陈为,
马兴录编. —北京: 化学工业出版社, 2009.8
高等学校规划教材
ISBN 978-7-122-06045-7

I. 32… II. ①宋…②陈…③马… III. ①微型计
算机-理论-高等学校-教学参考资料②微型计算机-接
口-高等学校-教学参考资料 IV. TP36

中国版本图书馆 CIP 数据核字(2009)第 102273 号

责任编辑: 王清颖

责任校对: 周梦华

装帧设计: 韩 飞

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 8½ 字数 225 千字 2009 年 9 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: 18.00 元

版权所有 违者必究

前 言

学习微机原理与接口技术重在实践。微机原理与接口技术课程中学习的基础知识、基本原理和基本方法技能，需要通过大量的上机实践和动手练习，才能做到学以致用，融会贯通。微机原理与接口技术实验是 32 位微机原理与接口技术课程的重要组成部分，需要与课堂教学紧密配合、相辅相成，才能达到好的教学效果，实现课程教学的目的和要求。为此，需要根据教学内容，精心设计实验项目，由浅入深，既要包含基础实验，也要有锻炼学生动手能力的设计性实验。基于这样的考虑，我们编写了与《32 位微机原理与接口技术》（马兴录，宋廷强，陈为等编）配套的实验指导教材。

本书按照实验的内容，将实验分为基础性实验和综合设计实验两部分。基础性实验包括 8 个汇编语言程序设计实验和 19 个硬件接口实验，该部分实验内容配合教材讲授进度安排，可以作为授课内容的实践与验证。综合设计实验共包含了 2 个汇编语言实验项目和 8 个硬件接口实验项目，需要综合多个知识点，是对所学内容的综合运用，可以作为期末大型实验或课程设计实验项目安排，考察学生综合运用知识，进行电路规划、电路设计、程序组织以及调试运行的能力，也可以作为学生课外开放实验的选题项目，满足指导学生开展课程设计和课外创新实践活动的需要，或者满足部分学校在更宽的范围内选择课内实验题目的需要。因此，实验项目的设置完全适应设计性、创新性、开放性实验改革的需要。

本书适用于不同教学要求的微机原理和接口技术课程。教师可根据教学要求组织实验内容，安排教学计划。

书中，基于嵌入式实验装置的实验项目由马兴录编写，基于 XL-100 的实验项目由陈为编写，宋廷强编写了本书其他内容，并负责统稿。

由于编者水平有限，书中不当之处在所难免，恳请读者批评指正。

编 者

目 录

第1章 汇编语言程序设计基础..... 1	实验 4-4 可编程中断控制器 8259A 的操作..... 44
1.1 汇编语言程序设计开发过程..... 1	实验 4-5 A/D 转换实验..... 47
1.2 汇编程序 MASM..... 2	实验 4-6 D/A 转换实验..... 50
1.3 调试工具 Debug..... 3	实验 4-7 查询方式的串行通信 接口实验..... 54
1.4 QTH-2008PC 集成开发环境..... 7	实验 4-8 中断方式下的串行通信 接口实验..... 55
1.5 UltraEdit 文本编辑工具..... 8	实验 4-9 PC 机显示及键盘接口实验..... 57
第2章 微机原理实验装置..... 11	实验 4-10 软接口实验..... 62
2.1 嵌入式微机原理实验装置..... 11	实验 4-11 128×64 点阵式 LCD 实验..... 62
2.1.1 系统构成及特点..... 11	实验 4-12 步进电机驱动实验..... 65
2.1.2 32 位微机原理教学实验环境..... 13	实验 4-13 ISD1420 语音实验..... 68
2.1.3 32 位微机实验系统硬件电路简介..... 13	4.2 基于 SXL-100 的硬件实验..... 76
2.2 SXL-100 型微机接口实验装置..... 14	实验 4-14 即插即用配置资源的获取..... 76
2.2.1 电路结构..... 14	实验 4-15 简单 I/O 端口实验..... 81
2.2.2 使用步骤..... 18	实验 4-16 8255 可编程并行接口 实现扫描式键盘..... 85
2.2.3 注意事项..... 18	实验 4-17 利用 8255 实现 LED 显示..... 88
第3章 汇编语言实验..... 19	实验 4-18 8253 定时器/计数器实验..... 92
实验 3-1 调试工具 Debug 的使用及 基本指令用法..... 19	实验 4-19 可编程中断控制器 8259 实验..... 95
实验 3-2 宏汇编程序的基本语法及 运算类程序编制..... 21	第5章 综合设计实验..... 101
实验 3-3 系统功能调用..... 23	5.1 要求..... 101
实验 3-4 中断处理程序的编制..... 25	5.2 远程监控系统的设计实例..... 102
实验 3-5 汇编语言码制转换程序设计..... 27	5.2.1 设计实验的目的、内容、要求及 实验基础..... 102
实验 3-6 表处理程序设计..... 29	5.2.2 系统设计..... 103
实验 3-7 子程序及多模块程序设计..... 31	5.2.3 实验步骤..... 105
实验 3-8 汇编语言综合实验..... 33	5.3 其他设计项目..... 107
第4章 硬件接口实验..... 34	5.3.1 程序设计类题目..... 107
4.1 基于嵌入式微机原理实验装置的 硬件实验..... 34	实验 5-1 学生成绩录入及统计程序..... 107
实验 4-1 简单 I/O 接口的操作..... 34	实验 5-2 简单的图书管理系统..... 107
实验 4-2 可编程并行接口芯片 8255A 的使用..... 36	
实验 4-3 可编程计数/定时器 8254 的使用..... 42	

5.3.2 硬件接口类题目	107
实验 5-3 电梯调度模拟系统	107
实验 5-4 7 段数码管电子闹钟	108
实验 5-5 数据采集系统设计	108
实验 5-6 步进电机控制及转速设置与 显示系统	109
实验 5-7 出租车计价器的设计	109
实验 5-8 家庭防盗报警系统的设计	109
实验 5-9 交通信号灯管理器的设计	110
附录 A 汇编出错信息 (附表 1)	111

附录 B 中断向量地址一览表 (附表 2)	113
附录 C DOS 功能调用 (附表 3)	114
附录 D BIOS 中断 (附表 4)	118
附录 E PCI BIOS 函数	121
附录 F QTHPC104 部分模块 电路原理图	123
附录 G SLB-100 实验板示意图 (附图 4)	126
参考文献	127

第 1 章 汇编语言程序设计基础

汇编语言是直接面向计算机硬件的语言，具有直接操作计算机硬件的能力。汇编语言的学习、实践环节十分重要，上机实验是快速掌握汇编语言程序设计的重要方法。本章主要介绍汇编语言程序设计的基本流程、相关工具软件的使用及程序调试方法，掌握用汇编语言设计、编写、调试和运行程序的方法，为汇编语言实验及后续的接口实验奠定坚实的基础。

1.1 汇编语言程序设计开发过程

汇编语言程序设计开发要经过编辑源程序、汇编、连接、运行、调试等步骤。由于程序开发的不确定性，每一个步骤都经过反复调试才能得到最终正确的结果。汇编语言程序设计的主要过程如图 1-1 所示。

(1) 设计编辑源程序

该步骤用于建立源程序文件，可以用普通文本编辑器编辑输入汇编语言源程序。常用的文本编辑器有 MS-DOS 下的 Edit 文本编辑程序，Windows 的记事本 (Notepad.exe)，Windows 下的 UltraEdit 文本编辑器等。用户通过屏幕编辑程序键入源程序，检查无误可将源程序存到磁盘上，该程序的扩展名为.asm。

(2) 汇编 (MASM 或 ASM) 源程序

用汇编语言编写的源程序必须是一个完整的源程序。宏汇编程序对汇编语言源程序的汇编过程包括语法检查和数据代码汇编两部分，生成目标程序和辅助信息文件。为了完成汇编任务，汇编程序一般采用两遍扫描的方法，第一遍扫描源程序产生符号表、处理伪指令等。第二遍扫描产生机器指令代码、确定数据等。

汇编程序有两种版本：一种是全型版本 (MASM)；另一种是小型版本 (ASM)。全型版本比小型版本增加了宏汇编、条件汇编及错误信息全部打印输出功能。

源程序用宏汇编程序翻译 (汇编) 后，可以得到三个中间文件。

① 目标代码文件，其扩展名为.obj，该文件将源程序的操作码部分变为机器码，但地址操作数是可浮动的相对地址，而不是实际地址，因此需经后续的连接步骤才能形成可执行文件。

② 列表文件，其扩展名为.lst，该文件把源程序和目标程序列表，以供检查程序用。列表程序由源程序和目标程序清单、段信息汇总表和符号汇总表等三部分组成，该文件可以在 DOS 状态下用 Type 命令显示或打印，以便分析调试源程序。

③ 交叉索引文件，其扩展名为.crf，该文件是一个对源程序所用的各种符号进行前后对照的

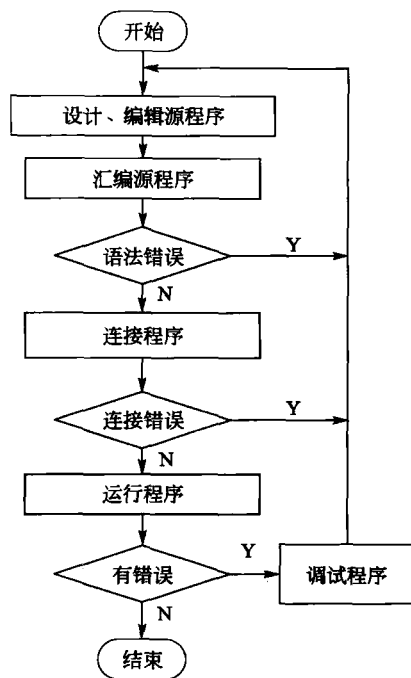


图 1-1 汇编语言的上机流程

文件。它列出了源程序中定义的符号（包括：标号、变量等）和程序中引用这些符号的情况。如果要查看这个符号表，必须使用 `Cref.exe` 工具软件，该工具软件根据.crf 文件建立一个扩展名为.ref 的文件，而后再用 DOS 的 `Type` 命令显示，就可以看到这个符号使用情况表。具体操作方法如下：

```
D:\MASM> Cref ✓  
    cref filename [.crf] : _myfile ✓  
    list filename [myfile.ref] : ✓  
D:\MASM> Type myfile.ref ✓
```

在这三个输出文件中，目标文件是必须产生的，而其他两个文件在需要时给予命令就可产生，对连接和执行汇编程序无直接的关系。

（3）目标代码连接

用汇编语言编写的源程序经过汇编程序（MASM）汇编后产生了目标代码文件（.obj），该文件将源程序操作码部分变成了机器码，但地址是可浮动的相对地址（逻辑地址），因此，必须经过连接程序 `Link.exe` 连接后才能运行。连接程序 `Link` 是把一个或多个独立的目标程序模块装配成一个可重定位的可执行文件，扩展名为.exe 文件。此外还可以产生一个内存映象文件，扩展名为.map。内存映象文件实际上是连接程序的列表文件，它给出了每个段的地址分配情况及长度。

（4）应用程序执行

当用连接程序 `Link` 将目标程序（.obj）连接定位后，产生可执行的应用程序文件（.exe）。在 DOS 提示符下，键入连接程序所产生的可执行文件的名字后，即可执行该程序。例如：

```
C:> Myfile ✓ 或  
C:> Myfile.exe ✓
```

就会把文件 `Abcd.exe` 装入内存，并从程序指定地址开始执行。在源程序 `Myfile` 中如果有显示结果的指令，则在执行程序后可以看到执行结果；如执行程序结果不正确，需要动态调试应用程序 `Myfile.exe`，则可以借助动态调试程序 `Debug.com` 来进行调试。

（5）调试程序

在编写和运行汇编程序的过程中，会遇到一些错误和问题，需要对程序进行分析和调试，调试程序 `Debug` 就是专为汇编语言设计的一种调试工具。它在调试汇编语言程序时有很强的功能，能使程序设计者接触到机器内部，能观察和修改寄存器和存储单元内容，并能监视目标程序的执行情况，使用户真正接触到 CPU 内部，与计算机产生最紧密的工作联系。

1.2 汇编程序 MASM

目前常用的汇编程序有 Microsoft 公司推出的宏汇编程序 MASM (Macro Assembler) 和 Borland 公司推出的 TASM (Turbo Assembler) 两种，二者大多数是兼容的，本书以 MASM 为例介绍宏汇编程序的使用方法。

MASM 宏汇编程序有很多版本，如 MASM5.00、MASM6.00 和 MASM6.11 等。MASM5.00 出现于 20 世纪 80 年代，支持 80386 指令集，适用于 DOS 环境汇编程序。20 世纪 90 年代推出了 MASM 6.00，该版本支持 .686、.686p 和 .mmx 等指令集；从 MASM6.11 版本开始支持 Windows 环境下的 32 位汇编语言程序。

（1）MASM 5.X 宏汇编程序的使用

MASM 5.X 及以前的版本主要包含两个文件：宏汇编程序 `Masm.exe` 和连接程序 `Link.exe`。宏汇编程序的主要使用方式有两种。

① 提问方式 在 MS-DOS 下输入命令行：`Masm` 或 `Asm`，这时 MS-DOS 将装入并启动

汇编程序，然后汇编程序就逐次向用户提问，用户必须根据要求予以回答。在回答信息的最后，可输入一个或多个开关，汇编程序将按照回答信息及开关的定义做相应处理。在回答提示信息时，可以字符“;”用于省略对后续提示的回答，也可以按下 Ctrl+C，退出汇编过程。

【例 1.1】用 MASM 命令汇编源程序

D:\MASM>Masm ✓

... ..

Microsoft (R) Macro Assembler Version 5.00

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

Source filename [.asm]: Myfile ✓

Object filename [Myfile.obj]: Myfile ✓

Source listing [Nul.lst]: Myfile ✓

Cross-reference [Nul.crf]: Myfile ✓

50678 + 410090 Bytes symbol space free

0 Warning Errors

0 Severe Errors

其中划线部分为用户键入部分，Myfile 为源程序名 (Myfile.asm)，方括号中是机器规定的默认文件名，如果用户认为方括号内的文件名就是要键入的文件名，则可只在划线部分键入回车符。如果不想要列表文件和交叉索引文件，则可在[Nul.lst]和[Nul.crf]后不键入文件名只键入回车符。

② 命令方式 以命令方式启动汇编程序时，必须在 MS-DOS 下打入下列格式的命令：

MASM <源文件>[, <目标文件>][, <列表文件>][, <交叉参考文件>][/ 开关]

命令中 MASM 后面的项目，分别顺次对应于提问方式中对各提示信息的回答，各项目之间用逗号隔开。/开关可放在任一项目之后。如果对某一提示信息采用默认值，则只需在对应的项目处连续输入两个逗号。例如：

MASM FUN,, FUN/D/X, FUN

(2) MASM 6.X 宏汇编程序的使用

从 6.0 版本以后，Microsoft 将 Assembler 和 Linker 组合成一个 ML 程序，以前需要两个步骤（编译、连接）才能生成一个 .exe 文件，现在只要一个步骤就可以完成。此时，程序扩展名一定要是 .asm，而且执行时一定要输入扩展名。例如：

ML Display.com

执行后将产生 Display.obj 和 Display.exe。因为缺省只会产生 .obj 和 .exe 文件，若要产生 .lst、.map 等文件，可在 ML 之后加 /Fl、/Fm 等参数。各参数之间一定要用空格分开。例如：

ML /Fl /Fm Display.com

表 1-1 是各参数使用的意义，全部参数均为可选参数。

表 1-1 MASM6.11 部分参数使用说明

参 数	说 明	参 数	说 明
/Fl	建立一个 .lst 文件	/Fr	建立一个 .sbr 文件
/Fm	建立一个 .map 文件	/c	只编译产生 .obj 文件

1.3 调试工具 Debug

Debug 是专门为汇编语言设计的一种调试工具，它通过步进、设置断点等方式为汇编语言程

程序员提供了非常有效的调试手段。调试程序的启动方法是在操作系统提示符下,按如下格式键入命令:

C:>Debug [d:] [path] [文件名] [参数 1] [参数 2]

其中文件名是被调试文件的名称,它须是执行文件(.exe),两个参数是运行被调试文件时所需要的命令参数,在 Debug 程序调入后,出现提示符“-”,此时,可键入所需的 Debug 命令。

在启动 Debug 时,如果输入了文件名,则 Debug 程序把指定文件装入内存。用户可以通过 Debug 的命令对指定文件进行修改、显示和执行。如果没有文件名,则是以当前内存的内容工作,或者用命名命令和装入命令把需要的文件装入内存,然后再用 Debug 的命令进行修改、显示和执行。

在 Debug 启动和执行过程中,需要注意以下事项。

① Debug 命令都是一个英文字母,后面跟着一个或多个有关参数。多个操作参数之间用",",或空格隔开。

② Debug 命令必须接着按 Enter 键,命令才有效。

③ 参数中不论是地址还是数据,均用十六进制数表示,但十六进制数据后面不要用"H"。

④ 可以用 Ctrl+Break 键来停止一个命令的执行,返回到 Debug 提示符“-”下。

⑤ 用 Ctrl+NumLock 键中止正在上卷的输出行,再通过按任意键继续输出信息。

Debug 一共提供了 18 条命令,这些命令如下。

(1) 汇编命令 A

格式: a. A [段寄存器名]:[偏移地址]

b. A [段地址]:[偏移地址]

c. A [偏移地址]

d. A

功能:用该命令可以将汇编语言程序直接汇编进入内存。

当键入 A 命令后,显示段地址和偏移地址等待用户键入汇编指令,每键入一条汇编指令回车后,自动显示下一条指令的段地址和偏移地址,再键入下一条汇编指令,直到汇编语言程序全部键入,又显示下一地址时可直接键入回车返回到提示符“-”为止。

其中 a 的段地址在段地址寄存器中,因此,在使用该命令时必须将段地址寄存器送入段地址, c 的地址在 CS 中, d 的段地址在 CS 中,偏移地址为 100H。

(2) 显示内存命令 D

格式: a. D [地址]

b. D [地址范围]

c. D

功能:显示指定内存范围的内容。

显示的内容为两种形式:一种为十六进制内容,一种为与十六进制相对应的 ASCII 码字符,对不可见字符以"·"代替。

对于 a、c 每次显示 128 个字节内容, b 显示的字节数由地址范围来决定。

若命令中有地址,则显示的内容从指定地址开始,若无地址(如 c)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。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段地址,起始偏移地址为 100H,即 CS:100。

对于 a 中的地址为偏移地址, 段地址为 CS 的内容, 对 b 中的地址范围, 可以指定段地址和起始偏移地址和终止偏移地址。

(3) 修改存储单元内容命令 E

格式: a. E [地址] [内容表]

b. E [地址]

功能:

a. 用命令所给定的内容表去代替指定地址范围的内存单元内容。

b. 一个单元一个单元地连续修改单元内容。

其中: 内容表为一个十六进制数, 也可以是用单引号括起的一串字符。

(4) 填充内存命令 F

格式: F [范围] [单元内容表]

功能: 将单元内容表中的内容重复装入内存的指定范围内。

(5) 内存数据传送命令 M

格式: M [源地址范围] [目标起始地址]

其中源地址范围和目的起始地址为偏移地址, 段地址为 DS 的内容。

功能: 把源地址范围的内容传送至以目标起始地址开始的存储单元中。

(6) 比较命令 C

格式: C [源地址范围], [目标地址]

其中源地址范围是由起始地址和终止地址指出的一片连续的存储单元, 目标地址为与源地址所指单元对比的目标地址起始地址。

功能: 从源地址范围是由起始的地址单元开始逐个与目标起始地址往后的单元顺序比较每个单元内容, 比较到源终止地址为止。比较结果如果一致则不显示任何信息, 如果不一致, 则以 [源地址] [源内容] [目的内容] [目的地址] 的形式显示失败单元地址及内容。

(7) 搜索指定内容命令 S

格式: S [地址范围] [表]

功能: 在指定地址范围内搜索表中内容, 搜索到就显示表中元素所在地址。

(8) 检查和修改寄存器内容命令 R

格式: a. R

b. R [寄存器名]

功能: a. 显示 CPU 内部所有寄存器的内容和全部标志位的状态。

b. 显示和修改一个指定寄存器的内容和标志位的状态。

其中对状态标志寄存器 FLAG 以外的形式显示, 显示时, 8 个状态标志的显示次序和符号如表 1-2 所示。

表 1-2 状态标志显示形式

标志位	状态	显示形式(置位 / 复位)	标志位	状态	显示形式(置位 / 复位)
溢出标志 OF	有/无	OV/NV	零标志 ZF	零/非	ZR/NZ
方向标志 DF	增/减	DN/UP	辅助进位 AF	有/无	AC/NA
中断标志 IF	开/关	EI/DI	奇偶标志 PF	偶/奇	PE/PO
符号标志 SF	负/正	NG/PL	进位标志 CF	有/无	CY/NC

(9) 跟踪与显示命令 T

格式: a. T [=地址] 或 T [地址]

b. T [=地址][条数] 或 T [地址][条数]

功能:

a. 执行一条指定地址处的指令, 停下来, 显示 CPU 所有寄存器内容和全部标志位的状态, 以及下一条指令的地址和内容。

b. 多条跟踪命令, 从指定地址开始; 若命令中用 [地址] 给定了起始地址, 则从起始地址开始, 若未给定, 则从当前地址 (CS:IP) 开始, 执行命令中的 [条数] 决定一共跟踪几条指令后返回 Debug 状态。

(10) 反汇编命令 U

格式: a. U [地址]

b. U [地址范围]

功能: 将指定范围内的代码以汇编语言形式显示, 同时显示该代码位于内存的地址和机器。

若在命令中没有指定地址则以上一个 U 命令的最后一指令地址的下一个单元作为起始地址; 若没有输入过 U 命令, 则以 Debug 初始化段寄存器的值作为段地址, 以 0100H 作为偏移地址。

(11) 命名命令 N

格式: N 文件名

功能: 在调用 Debug 时, 没有文件名, 则需要用 N 命令将要调用的文件名格式化到 CS:5CH 的文件控制块中, 才能用 L 命令把它调入内存进行调试 (其他形式参考 DOS 手册)。

(12) 读盘命令 L

格式: a. L [地址] [驱动器号] [起始扇区号] [所读扇区个数]

b. L [地址]

c. L

功能:

a. 把指定驱动器和指定扇区范围的内容读到内存的指定区域中。其中地址是读入内存的起始地址, 当输入时没有给定地址, 则隐含地址为 CS:100H。起始扇区号指逻辑扇区号的起始位置。所读扇区个数是指从起始扇区号开始读到内存几个扇区的内容。驱动器号为 0 或 1, 0 表示 A 盘, 1 表示 B 盘。

b. 读入已在 CS:5CH 中格式化的文件控制块所指定的文件。在使用该命令前用 N 命令命名, 即可将要读入的文件名格式化到 CS:5CH 的文件控制块中, 其中地址为内存地址。

c. 同 b, 地址隐含在 CS:100H 中。

当读入的文件有扩展名 .com 或 .exe, 则始终装入 CS:100H 中, 命令中指定了地址也没用。其中 BX 和 CX 中存放所读文件的字节数。

(13) 写盘命令 W

格式: a. W [地址] [驱动器号] [起始扇区号] [所写扇区个数]

b. W [地址]

c. W

功能:

a. 把在 Debug 状态下调试的程序或数据写入指定的驱动器中, 起始扇区号、所写扇区个数为要占盘中几个扇区。写盘指定扇区的操作应十分小心, 如有差错将会破坏盘上的原有内容。如果在命令行中的地址只包含偏移地址, W 命令认为段地址在 CS 中。

b. 当键入不带参数的写盘命令时 (或只键入地址参数的写盘命令), 写盘命令把文件写到软

盘上。该文件在用 W 命令之前用命名命令 N 将文件格式化在 CS:5CH 的文件控制块中。

c. 只有 W 命令以前而没有任何参数时, 与 N 配合使用进行写盘操作。

在用 W 命令以前在 BX 和 CX 中应写入文件的字节数。

(14) 输出命令 O

格式: O [端口地址] [字节值]

功能: 向指定端口地址输出一个字节。

(15) 运行命令 G

格式: G [=地址] [地址 [地址...]]

功能: 执行用户正在调试的程序。

其中地址为执行的起始地址, 以 CS 中内容作为段地址, 以等号后面的地址为偏移地址。再后面的地址为断点地址。在命令中只有起始地址, 没有断点地址, 则程序在执行时不中断。Debug 规定最多设置 10 个断点地址。设置多个断点用于调试较大的程序, 即程序中有多个模块、多个通路时用, 比较方便, 在执行时不论走哪条通路, 程序都可以在断点处停下来, 以便调整程序。

断点地址为程序中断处的偏移地址, 段地址在 CS 中。

当执行在 Debug 状态下汇编的小段程序时, 只用 G 命令即可。

(16) 十六进制运算命令 H

格式: H 数据 1 数据 2

其中数据 1 和数据 2 为十六进制数据。

功能: 将两个十六进制数相加、减, 结果显示在屏幕上。

(17) 结束 Debug 返回到 DOS 命令 Q

格式: Q

功能: 程序调试完退出 Debug 状态, 返回到 DOS 状态下。

Q 命令不能把内存的文件存盘, 要想存盘必须在退出 Debug 之前用 W 命令写盘。

1.4 QTH-2008PC 集成开发环境

QTH-2008PC 是一个集成开发环境, 内部集成了汇编程序、连接器和调试器, 主要界面如图 1-2 所示。界面上包括项目区、程序区和消息区, 所有的操作都可以通过菜单、工具条或快捷键操作, 可以完成从汇编语言源程序输入、汇编、连接和调试的全过程, 其内部是按照项目进行管理, 图 1-2 是采用该工具进行操作的主要过程。

① 项目的建立 项目 (Project) 是用于管理由一个或多个源程序文件组成的应用程序的。选择“新建项目”, 在自己的文件夹下创建新的项目。项目名称后缀默认为“.pci”。

在建立项目时会弹出对话框, 有个“加入模块文件”的选项, 如果汇编语言源程序已经输入并保存, 这时, 可以选中此选项, 将汇编语言源程序加入项目。

注意: Qthpci 软件中对文件名遵循 DOS 的命名约定, 即文件名或文件夹名称不能超过 8 个西文字符, 不能使用汉字, 不能有空格等。

② 源程序的建立 新建一个文件, 输入源程序, 并保存。保存时注意添加后缀.asm。

③ 源程序加入项目 在项目的源文件上点击右键或选择主菜单中的“项目”, 选择“加入模块文件”, 将刚才保存的汇编语言源程序加入项目。

④ 编译 对源程序进行编译。如源程序有错误, 按照信息框中的错误提示, 对源程序进行修改。

⑤ 编译连接 对编译通过的程序进行连接。



图 1-2 QTH-2008PC 集成开发环境

⑥ 调试 当信息框中出现 0 个错误 0 个警告并且调试工具变为可用时,说明编译连接通过可以开始调试了。该环境集成了两种调试器:动态调试程序 Debug 和 Turbo Debugger。可以采用其中的任何一个进行程序调试。

1.5 UltraEdit 文本编辑工具

UltraEdit 是 Windows 下非常优秀的文本编辑软件,有很好的可配置性,支持语法高亮等。配置 UltraEdit,实现在 UltraEdit 环境下编辑汇编源程序、汇编、连接、调用 Debug 调试,非常方便。配置完成后,可实现快捷键调用各编译、连接、调试功能。

(1) 打开 UltraEdit, 编辑并保存汇编源文件

先新建一个文件,并保存为.asm 为后缀的文件,此后,可以有语法高亮指示。输入汇编程序,并保存。图 1-3 是利用 UltraEdit 输入源程序的界面。

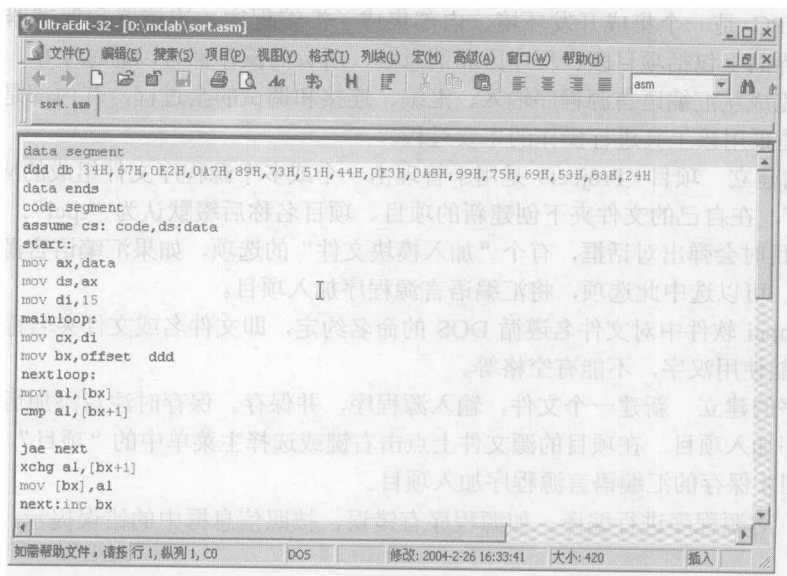


图 1-3 UltraEdit 编辑环境

(2) 编译

从 UltraEdit 菜单栏中选择: 高级->masm 编译 (或 Tasm 编译) 菜单, 此菜单会调用 masm.exe (ml.exe) 或 tasm.exe, 输出结果放在当前编辑屏幕的下方, 如图 1-4 所示。

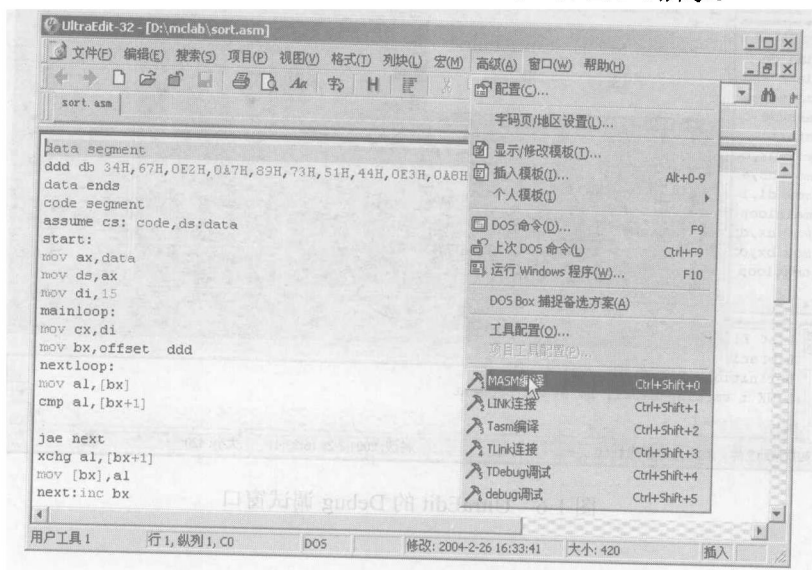


图 1-4 利用 UltraEdit 菜单选择编译功能

(3) 连接

从 UltraEdit 菜单栏中选择: 高级->Link 连接, 此菜单会调用 Link.exe 输出结果也放在当前编辑屏幕的下方, 如图 1-5 所示。

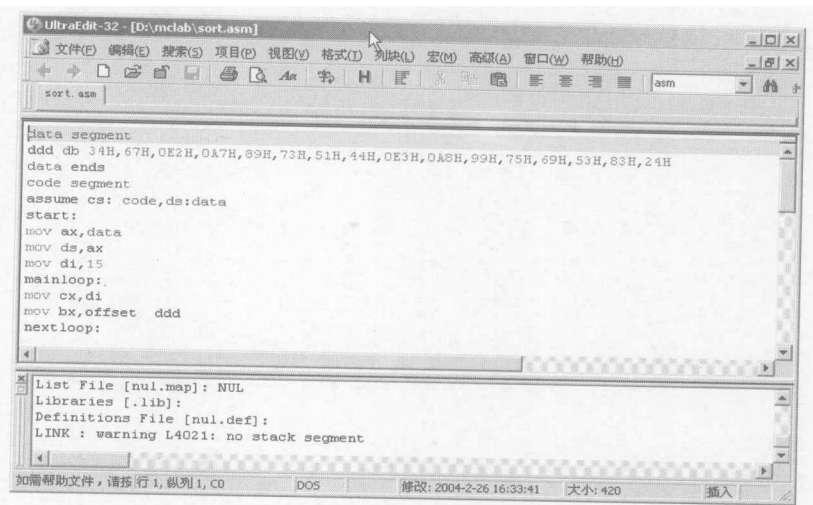


图 1-5 Link 结果显示

(4) 调试

从 UltraEdit 菜单栏中选择: 高级->Debug 调试, 此菜单会新开一个 DOS 窗口, 在其中运行 Debug.com, 调试完毕, 可以输入-q 返回 DOS。该方式下, 可以通过 Alt+Enter 实现 DOS 窗口模式与全屏模式之间的快速切换。图 1-6 给出了利用 UltraEdit 进行调试的界面。

第 2 章 微机原理实验装置

2.1 嵌入式微机原理实验装置

QTH-2008PC 实验系统为 32 位微机原理和 32 位微机接口技术提供了实验平台。

2.1.1 系统构成及特点

嵌入式微机原理实验装置采用了先进的嵌入式系统的设计思想，实验箱采用模块化结构设计，将微机原理、接口技术及部分嵌入式实验功能集于一体，有利于学生综合能力及创新能力的培养。

该装置以工业级嵌入式 PC/104 586 主板为核心，外扩 10 多种不同的实验模块，主要包括：简单 I/O 模块、8255 模块、8254 模块、8259 模块、小键盘模块、开关及 LED 模块、A/DD/A 模块、LCD 模块、串行通信模块、电机模块、语音模块、开关量 I/O 模块、12 位 AD 模块等，如图 2-1 所示。

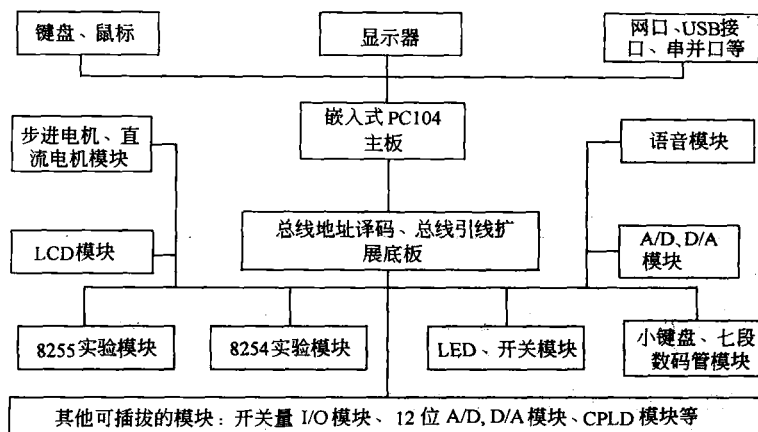


图 2-1 嵌入式微机原理实验装置总体结构示意图

32 位微机原理部分的实验平台是以 PC/104 586 主板为核心构成的。用户可以通过该平台进行 32 位计算机下的实验程序编制、运行及调试。32 位微机接口应用部分的平台，用于支持基本接口应用学习。QTH-2008PC 系统提供了一个仿真 ISA 接口的平台，并提供了大量开发实验例子，可以参考这些实例快速掌握接口应用的实现方法。QTH-2008PC 提供实验内容如表 2-1 所示。

该实验平台具有以下功能及特点。

(1) 先进的 32 位微机原理实验教学平台

系统提供了 586 微机上的调试操作软件，允许用户在 32 位微机实模式和保护模式下调试并运行实验程序，为用户提供了一个窥探 80X86 微处理器运行机制的窗口，使用户可以迅速了解 32 位微机的工作机制和过程，并掌握其编程方法，为学习 32 位微机接口技术和应用打下基础。