

微 处 理 器 和微型计算机开发系统

——以微处理器为基础的系统设计——

〔美〕穆罕默德·拉菲居扎芒 著

陈伟鑫 等译 林定基 等校

微处理器和微型计算机开发系统

——以微处理器为基础的系统设计

〔美〕 穆罕默德·拉菲居扎芒 著

陈伟鑫 等译 林定基 等校

国防工业出版社

(京)新登字106号

内 容 简 介

本书从使用者的角度出发,全面系统地讨论了各种微处理器(8、16、32位,并包括单片微处理器)及微处理器开发系统(包括Intel、Tektronix、GenRad和Hewlett-Packard等公司的产品)。详细地讨论了各种开发系统的结构、特点和应用,特别是以HP64000开发系统为典型,具体地阐述了开发系统的使用方法和完整步骤,并提供三个设计案例。

本书适合从事微型计算机技术开发应用的科技人员阅读,亦可供大专院校有关专业作为教学参考书。

MICROPROCESSORS AND MICROCOMPUTER DEVELOPMENT
SYSTEMS—Designing Microprocessor-Based System
MOHAMED RAFIQUZZAMAN
HARPER & ROW 1984

微处理器和微型计算机开发系统

——以微处理器为基础的系统设计

〔美〕穆罕默德·拉菲库扎曼 著

陈伟鑫 等译

林定基 等校

国防工业出版社发行

(北京市海淀区紫竹院南路23号)

(邮政编码 100044)

新华书店经售

国防工业出版社印刷厂印装

787×1092 1/16 印张35³/₄ 插页1 825千字

1992年2月第一版 1992年2月第一次印刷 印数: 0,001—2,000册

ISBN 7-118-00683-1/TP·87 定价: 29.30元

译者前言

随着微处理器和微型计算机在各个领域的广泛应用,对于微型计算机开发系统的需求亦日益增长。本书的目的就是帮助读者熟悉各种微处理器(包括单片微机 8048)、接口芯片,熟悉为微处理器应用系统设计软件、硬件所需要的开发系统。对于前者,本书介绍了典型的8、16、32位微处理器;对于后者,则介绍了微处理器生产厂家针对本厂产品所研制的开发系统,又介绍了仪器公司(如Hewlett-Packard等)所研制的、适用面广的开发系统。特别是具体地介绍了典型开发系统的使用方法、步骤,以及微型计算机应用系统的开发实例。

诚然,本书部分内容引自有关厂家所发表的技术资料,而其中有的已散见于国内有关书刊。但从总体上看,本书内容全面、具体实用,对于我们利用开发系统来组织微处理器和微型计算机应用项目的设计与调试有所助益。本书可供从事微处理器技术应用开发的广大科技人员参阅。亦可作为大专院校有关专业的教学参考书。为此,特将本书译出,以饯读者。

本书第10章及部分附录为兰家隆副教授所译,余为本人所译。

本书承林定基教授、黄志辉老师悉心校阅,本人对此深表谢意。但限于本人水平,不妥或差错之处仍所难免,恳请读者不吝指正。

陈伟鑫

1988年

序

当前,微处理器正广泛应用于各个领域,有代表性的是用于过程控制、通信系统、数字仪器、电子游戏和家用设备等。因此,对计算机工程师和科学家说来,完全有必要掌握以微处理器为基础的各种应用系统的芯片级设计。微型计算机开发系统是为这类微处理器应用系统设计硬件和软件的工具。

本书的目的是使读者熟悉典型的8、16和32位微处理器、接口芯片,熟悉为微处理器应用系统设计和开发硬件和软件所必需的开发系统。本书也给出了运用一个典型的开发系统去设计微处理器应用系统的具体步骤。书中以简明形式提供的大量例子是从实际应用中提炼出来的。

本书自成系统,并包含了诸如微型计算机结构和编程之类的课题。本书是以基本数字逻辑为基础的。微处理器基本课程有助于学习本书,但非必需,因为本书已有大量基本课题。由于包括了上述基本内容,因此本书也可作为教学参考书用。

作者着重选择了所有微型计算机开发系统所共有的特性和原理作为阐述重点,本书通过一个特定的开发系统对一个指定的微处理器的实验开发来阐述这些内容。HP64000和Intel 8085即为此而用。本书选用HP64000来说明一个典型开发系统的特点是基于如下原因:

1. 64000是市场上唯一提供软键特点的系统,因此容易操作。
2. 64000是多功能的和基于多终端的。
3. 64000提供了典型开发系统的特点,诸如编辑程序、编译程序、汇编程序、连接程序、仿真程序、调试程序和逻辑分析仪等。

本书用Intel 8085及其支持芯片来说明微处理器应用系统的芯片级设计,并详尽地给出了设计一个专用系统的逐条设计步骤。作者认为一旦读者理解了使用64000开发系统的基本概念,则用其它开发系统和任何微处理器进行以微处理器为基础的设计就十分容易了。

本书分为十二章。第一章以导论的观点给出了开发系统的基本特点,以及这类系统在为以微处理器作基础的产品设计硬件、软件时所起的作用。第二章到第四章包括了与Intel 8085相关的典型微计算机的结构、编程和硬件概念。第五章讨论了8085的I/O技术。在第十章和第十一章讨论开发以8085为基础的系统时,以及在第六章和第七章讨论典型的8、16和32位微处理器时,均需大量引用本章的内容。第八章对典型的微处理器接口芯片作了详尽讨论。此类芯片包括ROM、EPROM、RAM、I/O、键盘和DMA控制芯片。第九章为微处理机开发系统基础。其内容有编辑程序、编译程序、汇编程序、连接程序、仿真程序和逻辑分析仪。第十章是开发系统概论,讨论Intel、Tektronix、GenRad(前身为Futuredata)和Hewlett-Packard等公司的产品、第十一章详尽地探讨了有如HP64000这样一类开发系统的原理和操作。第十二章给出了用HP64000去开发8085基础系统的软件和硬件的系统步骤。

本书适用于很多方面。首先可供二年制或四年制大学开设微处理器基础系统设计课程时选作教材，根据工程学、计算机科学或工艺设计各科的教学安排，该课程可在半年或四分之一学年内结束。这里要求大学生已具备数字逻辑基础知识，并熟悉布尔代数、卡诺图、晶体管和 MOS 器件等。本书也适用于希望扩充有关微处理器、微计算机开发系统的知识，提高设计技术水平的工程技术人员。

原 著 者

目 录

第一章 微型计算机开发系统和以微处理器为基础的系统设计导论	1
1.1 基本特点	1
1.2 系统开发流程	5
1.2.1 软件开发	5
1.2.2 硬件开发	6
1.2.3 诊断设计	7
1.3 一个专用系统的硬件和软件设计	7
1.3.1 软件开发	8
1.3.2 硬件	8
1.3.3 诊断	8
第二章 微型计算机基础	9
2.1 微处理器系统概述	9
2.1.1 基本微处理器系统	9
2.1.2 程序	10
2.1.3 输入/输出	10
2.1.4 存储器	12
2.1.5 小型机和微型机	15
2.2 数制和布尔逻辑	15
2.2.1 十进制和二进制	15
2.2.2 八进制	16
2.2.3 十六进制	16
2.2.4 关于数位的术语	16
2.2.5 基本布尔运算	16
2.2.6 算术和逻辑运算	17
2.2.7 数的表示	19
2.3 软件基础	22
2.3.1 作为逻辑装置的微计算机	23
2.3.2 流程图	23
2.3.3 程序语言	24
思考题和习题	28
第三章 微计算机系统的硬件和 I/O 技术	29
3.1 基本硬件概念	29
3.1.1 总线概念	29
3.1.2 三态总线	30
3.1.3 系统总线	31
3.1.4 输入和输出	33
3.1.5 多个器件和存储器的地址译码	34
3.2 地址译码	38
3.2.1 地址译码器	38
3.2.2 线性选择译码	39
3.2.3 逻辑比较器译码	40
3.2.4 组合逻辑译码	40

3.2.5 I/O映射译码	41
3.3 存储器和外部设备	41
3.3.1 存储器的类型	41
3.3.2 ROM和RAM	41
3.4 典型的微处理器 I/O 技术	42
思考题和习题	43
第四章 典型的 8 位微处理器和微计算机	44
4.1 Intel 8085	44
4.1.1 引言	41
4.1.2 8085的引脚和信号	45
4.1.3 指令周期和执行	47
4.1.4 机器周期	49
4.1.5 程序的执行	49
4.1.6 8085指令系统回顾	49
4.1.7 子程序和堆栈	57
4.1.8 数学运算	59
4.2 Intel 8048微计算机	71
4.2.1 引言	71
4.2.2 指令系统	73
4.2.3 寻址方式	75
4.2.4 I/O功能	76
4.3 Zilog Z80	79
4.3.1 引言	79
4.3.2 寻址方式	81
4.3.3 指令系统	81
4.3.4 输入/输出(I/O)	94
4.4 Motorola 6800	95
4.4.1 引言	95
4.4.2 寻址方式	97
4.4.3 指令系统	97
4.4.4 I/O功能	103
4.5 Motorola 6809	104
4.5.1 引言	104
4.5.2 寻址方式	106
4.5.3 指令系统	106
4.5.4 I/O功能	109
思考题和习题	111
第五章 8085输入/输出	113
5.1 8085的可编程 I/O	113
5.1.1 8355/8755 I/O端口	113
5.1.2 8155 I/O端口	113
5.2 8085中断系统	116
5.2.1 TRAP	116
5.2.2 RST7.5	116
5.2.3 RST6.5	117
5.2.4 RST5.5	117
5.2.5 INTR	117
5.3 8085 DMA	123

5.4 8085的SID线和SOD线	124
思考题和习题	125
第六章 Intel 8086和Zilog Z8000	127
6.1 Intel 8086	127
6.1.1 8086结构	129
6.1.2 寻址方式	132
6.1.3 指令系统	133
6.1.4 输入/输出(I/O)	138
6.1.5 8086的引脚和信号	140
6.2 Zilog Z 8000	144
6.2.1 Z 8000 CPU组织	144
6.2.2 寄存器结构	146
6.2.3 寻址方式	147
6.2.4 指令系统	148
6.2.5 中断	160
6.2.6 Z 8000的引脚和信号	160
6.2.7 一个典型的Z 8000系统	162
思考题和习题	165
第七章 MOTOROLA 68000和INTEL 432	166
7.1 MOTOROLA 68000	166
7.1.1 处理器结构	167
7.1.2 寻址方式	169
7.1.3 指令系统	171
7.1.4 68000的引脚和信号	178
7.1.5 68000系统图	180
7.1.6 68000字节寻址	181
7.1.7 系统特点	181
7.1.8 循环	189
7.1.9 68000的外围电路	190
7.1.10 68000与6846ROM I/O定时器(RIOT)接口	190
7.2 Intel 432	194
7.2.1 通用数据处理处理器(GDP)——iAPX43201和iAPX43202	195
7.2.2 接口处理器(IP)——iAPX43203	208
7.2.3 432操作系统——iMAX	215
7.2.4 432的应用	215
7.2.5 结论	215
思考题和习题	215
第八章 典型的微处理器接口芯片	217
8.1 典型的EPROM、RAM和I/O芯片	217
8.1.1 Intel 2716 EPROM	217
8.1.2 带I/O的Intel 8355/8755 ROM/EPROM	219
8.1.3 带I/O和定时器的Intel 8155/8156 RAM	222
8.2 典型的串行I/O接口芯片	232
8.2.1 Motorola MC6850异步通讯接口适配器(ACIA)	232
8.3 键盘/显示控制器芯片	239
8.3.1 Intel 8279键盘/显示控制器芯片	241
8.4 直接存储器存取(DMA)控制器芯片	257
8.4.1 Intel 8257 DMA控制器	257

思考题和习题	264
第九章 微型计算机开发系统基础	266
9.1 基本特点	266
9.1.1 硬件	266
9.1.2 操作系统和调试技术	269
9.2 软件开发工具	272
9.2.1 编辑程序	272
9.2.2 汇编程序	274
9.2.3 反汇编程序	279
9.2.4 连接程序	281
9.2.5 装入程序	281
9.2.6 命令文件	282
9.2.7 高级语言编译程序	282
9.2.8 解释程序	282
9.2.9 监控程序	282
9.2.10 操作系统	283
9.3 微型计算机开发系统的操作控制台	283
9.4 微型计算机开发系统的海量存储	284
9.4.1 无存储	284
9.4.2 纸带	284
9.4.3 盒式磁带	285
9.4.4 软盘	285
9.4.5 硬盘	285
9.5 开发系统的结构	285
9.5.1 主/从系统	285
9.5.2 单处理器系统	286
9.6 调试和综合	287
9.6.1 联机仿真器	287
9.6.2 调试程序	287
9.6.3 仿真调试	288
9.6.4 实时调试	289
9.6.5 综述	290
9.7 微处理器用高级语言	290
9.7.1 BASIC	291
9.7.2 PL/M	291
9.7.3 COBOL	292
9.7.4 PASCAL	293
思考题和习题	296
第十章 通用型微型计算机开发系统	298
10.1 H-P 公司的HP64000	298
10.2 Intel 开发系统	299
10.2.1 Intel 120型和225型	299
10.2.2 Intel 286型微型计算机开发系统(MDS)	301
10.2.3 Intel 290型网络开发系统 1 (NDS1)	303
10.2.4 Inteltec主机连接(IML)	306
10.3 Tektronix开发系统	322
10.3.1 Tektronix 8001型微处理器开发实验室(MDL)	322
10.3.2 Tektronix 8002A	324

10.3.3 Tektronix 8500系列 (MDL).....	326
10.4 GenRad 系统.....	340
10.4.1 2300独立软件.....	340
思考题和习题	356
第十一章 HP 64000	357
11.1 系统介绍	357
11.2 开发站概述	357
11.3 启动	359
11.3.1 上电.....	359
11.3.2 装入系统软件.....	359
11.3.3 软键.....	360
11.3.4 专用功能键.....	360
11.4 编辑程序	361
11.4.1 编辑程序的使用.....	361
11.5 HP 64000 汇编程序.....	363
11.5.1 汇编命令.....	363
11.5.2 64000的宏	365
11.5.3 汇编程序伪指令的使用.....	369
11.5.4 几种常见错误及其结果举例.....	373
11.6 HP 64000 连接程序.....	381
11.6.1 连接程序的初始化.....	381
11.7 HP 64000 仿真器.....	382
11.7.1 仿真设备.....	382
11.7.2 分析.....	382
11.7.3 带符号调试.....	382
11.7.4 64000仿真器结构	383
11.7.5 对不同处理器的仿真.....	383
11.7.6 仿真会话的开始.....	383
11.8 命令文件	388
11.8.1 简单命令文件举例.....	388
11.8.2 另一个命令文件举例.....	388
11.8.3 命令文件举例——传送参数.....	389
11.9 模拟 I/O	389
11.10 举例说明在典型 64000 系统上进行软件和硬件的开发.....	389
11.10.1 64000引导.....	390
思考题和习题	425
第十二章 设计课题	428
12.1 设计课题之一	428
12.1.1 课题说明.....	428
12.1.2 设计课题之一的解.....	428
12.2 设计课题之二	441
12.2.1 课题说明.....	441
12.2.2 设计课题之二的解.....	441
12.2.3 方法学.....	448
12.2.4 结果.....	449
12.2.5 结论.....	449
12.3 设计课题之三	453

12.3.1 课题说明	453
12.3.2 设计课题之三的解	453
12.3.3 系统硬件讨论	453
12.3.4 系统软件讨论	458
思考题和习题	469
附录 A 8085指令系统	472
附录 B Intel 8279	507
附录 C 以8085为基础的微型计算机——HP5036 A的系统图	524
附录 D HP64000的其他资料	525
附录 E 其他程序列表	533

第一章 微型计算机开发系统和以微处理器为基础的系统设计导论

为了有效地开发微处理器应用系统，有必要使用微型计算机开发系统。微型计算机●开发系统用于设计和调试微处理器应用系统，有时也用于编制文件。本章将概述典型的微型计算机开发系统的特性以及设计和调试某些专用微处理器系统的方法。

1.1 基本特点

当首批4位和8位微处理器投入市场时，与微处理器系统有关的硬件价格相当高，因此劳务费用（逻辑设计、软件、调试等）对总的产品价格影响不大。随着新型微处理器元件价格的持续降低，开发新产品的劳务费用就起更大作用了。微计算机开发系统的主要功能在于简化产品的开发阶段，从而使设计者（不论是单独的设计师，还是庞大的设计组）获得最大效率。

开发系统允许硬件和软件平行设计，因此在任何实际模拟硬件出现之前，就可对设计进行测试和调整。某些开发系统允许软件工程师无需使用最后硬件来进行软件模拟。

开发系统分为两大类：由器件制造商提供的系统和由设备制造商生产的系统。两类系统之间的主要区别在于系统所适应的微处理器品种范围。由器件制造厂商（如Intel、Motorola、RCA等）提供的系统局限于开发供应者自己生产的特定芯片。按照这种方式，Intel的开发系统不能用来开发以Motorola芯片为基础的系统。另一类系统则具有较大的通用性。这类通用系统（如Tektronix、Hewlett-Packard、AMI等公司的产品）可以提供软件和仿真硬件，以开发大多数常用的微处理器。

两大类开发系统基本上有三种可能的类型：即单用户系统、分时系统和网络系统。单用户系统由一个开发站组成，因此一次只能供一个用户使用。单用户系统价格低，适用于小系统开发。分时系统通常由转储信息的终端组成。这些终端通过数据线连接到一个控制全部操作的集中的微计算机系统上。一个网络系统通常由许多灵巧的CRT显示器组成。这些显示器能完成大部分开发工作，并可通过数据线接到一个大型中央计算机上。网络中的这个中央计算机通常负责分配磁盘存储空间，并能将某些程序下放到用户工作站计算机中。网络开发系统的应用日益广泛，因为这种系统允许更多的数据吞吐，这样就允许用户又快又容易地开发程序。随着站数的增加，网络系统每站的价格将比单用户系统低。

微型机开发系统由微处理器设计所需的硬件和控制这些硬件所必需的软件组合而成。硬件中的基本部件是中央处理器、CRT终端，海量存储设备（软盘或硬盘），通常还有

● 通常称为微计算机、微型机及微机。

一个在线仿真器 (ICE)。

中央处理器是开发系统的核心。不论是单用户系统中的微处理器、或是多用户分时系统中的大型处理器,中央处理器总是负责对开发系统进行全面控制。在单用户系统中,中央处理器执行操作系统软件,管理输入/输出 (I/O) 设备,执行开发程序 (编辑程序、汇编程序、连接程序等),并给所执行的程序分配存储空间。在大型多用户网络系统中,中央处理器可以负责海量存储器的分配,而本地处理器可以负责实现 I/O 功能和执行开发程序。

CRT 终端在用户和操作系统之间,或用户和所执行的程序之间提供接口。用户经 CRT 键盘输入命令或数据。执行中的程序通过 CRT 屏幕为用户显示数据。早期的开发系统不用 CRT,而用电传打字机和开发系统通信。现在用 CRT 和操作系统通讯,其速度比用老式的电传打字机快十倍以上。单用户系统的 CRT 可以和中央处理器直接相连,而分时系统中的 CRT 可以通过电话线 (用调制解调器) 和远程计算机相连。

对于用户数据的海量存储,有必要使用某种磁盘驱动器。某些流行的复杂操作系统和高级开发程序需要用大容量的随机存取存储器 (RAM) 系统进行操作,用小容量的存储器存储用户数据,例如正在开发的程序的原码。为此,操作系统必须分配可用的存储器和磁盘上程序未用的部分。存储器管理的一种普通的方法是采用覆盖文件,使用覆盖是因为程序的一部分不是经常需要,因此在有限的 RAM 空间中不予保留,当需要覆盖时,就从磁盘读出文件,并覆盖到存储器中另一个不再需要的程序上。覆盖技术和其它存储管理技术的应用,允许当今通用的小型计算机能运行原来不可能运行的复杂程序。用户文件 (如源码) 也存入磁盘。

大多数单用户系统使用软盘做为海量存储器,而分时系统和网络系统则倾向于使用硬盘作为海量存储器。磁盘系统的价格和性能直接相关。软盘系统的价格低,但存储容量也小。软盘的存取时间 (从请求数据开始到取得数据为止的这段延迟时间) 比硬盘或标准存储器都慢。硬盘的存取时间比软盘短得多,而存储容量却大得多,但硬盘系统价格较高,不适宜用于小用户。如果购买硬盘系统,所付出的代价通常可以缩短开发时间得到补偿。

ICE 是一种极有价值的工具,它可以使通用计算机演变成开发系统。ICE 是一种硬件电路,通过它可将开发系统接对待开发的微处理器 (目标) 系统上。ICE 插入目标系统以代替目标微处理器,而其电特性和功能与目标微处理器相同。通过控制 ICE,开发系统可以对被开发的微处理器执行测试程序。当 ICE 操作目标系统时,开发系统可以显示参数 (如寄存器内容)。借助 ICE,可以在开发系统的 RAM 中调试程序,而不必将程序编入只读存储器 (ROM) 再插进目标系统。可以用 ICE 来设置断点 (即设计者希望暂停程序并检查程序状态的程序位置),用户还可以检查程序的进程,如果需要可以从同一点再启动程序。程序中的错误可以在几秒钟内在 RAM 中加以修改,而一个新的可编程只读存储器 (PROM) 的编程过程可能要花费几分钟宝贵的开发时间。当目标系统的硬件结构和软件日趋复杂时,ICE 就成为微机开发的一种必备工具。

以上所列举的设备可以构成一个基本开发系统,但大多数系统还具有其它设备,如附加的打印机和 PROM 编程器等。需要打印机为用户提供被开发程序的硬拷贝记录。许多简单的程序完全可以在 CRT 上开发,但对于高级程序的开发则需要打印机来提供所必

需的文件编制。程序完全调试好以后，则由 PROM 编程器将机器码装入 PROM，以供最终测试样机用。图1.1所示为一个典型的开发系统。

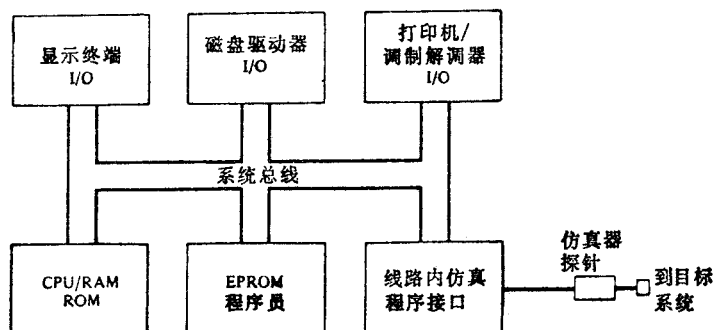


图1.1 一个典型开发系统的框图

对于一个开发系统来说，硬件是必需的，但若没有适当的开发软件，系统仍不能应用。随着开发工作的日益复杂，操作系统为用户承担了许多例行任务，使用户的时间具有更高的效率。许多开发系统是自助的，用户不必再从操作手册中查找命令和操作步骤了。

对于微处理器开发所必需的主要程序是操作系统、编辑程序、汇编程序、连接程序、编译程序和调试程序。

操作系统的职责是执行用户的命令。操作系统（如 CP/M、MP/M、UNIX 等）处理输入/输出（I/O）功能、存储器管理，以及将程序从海量存储器装入 RAM，以便执行该程序。开发系统越庞大，操作系统也就庞大，能力也越强。

编辑程序用来开发用户程序。不论用户程序是用汇编语言还是用 PASCAL 之类的高级语言编写的，都必须首先通过编辑程序进入开发系统。编辑程序允许用户输入并编辑源码，可以在源码中插进或删除单个字符或整行。

早期的编辑程序只允许一次编辑一行，因此称为行编辑。首先指定需编辑的行，然后用规定的命令去更改该行的内容。

大多数流行的编辑程序都是上下文编辑程序，即将一段源码实时显示在 CRT 上。因源码完整地显示在 CRT 上，可以看到上下文，所以取名上下文编辑。对原码进行编辑时，更改的情况直接显示在 CRT 屏幕上。不必用复杂的命令去编辑源码，用户可以用简单的光标移动命令指向需修改的源码部分。当光标定位后，即可输入新的文本。许多编辑程序允许块操作，即可将很多行，甚至整个文件移动或复制到其它位置。

源码经编辑生成后，必须转换成目标处理器可以执行的十六进制码（目的码）。汇编语言源码由汇编程序负责转换，而高级语言源码则由编译程序负责转换。

汇编程序将通过编辑程序输入的 ASCII 符号数据翻译成目标微处理器所使用的目的码，即 16 进制码。除标准的汇编语言指令外，汇编程序还将识别称作伪指令的特殊命令。伪指令是随汇编语言一同输入的命令，但在汇编时被汇编程序识别后，则产生特定的格式或数据。例如，ORG 语句被用来建立程序的起始地址或源地址。伪指令的存在允许用户在每页清单上放置标题，给清单规定便于阅读的格式，在程序中插入数据表，建立程序符号等等。

现行的汇编程序大都具有宏功能。宏功能是在源码中专门标注出来的一组汇编语言指令。给宏功能取这个名字，以便每次用户希望执行相同的一组汇编语言命令时，可以去调用它。宏功能与标准汇编语言CALL（调用）命令的差别在于宏功能仅仅是汇编程序在使用时所具有的一种功能，宏不是一个子程序（子程序是代有RET命令的一组指令），但宏是实际上插入最终目的码中的复制码。利用宏功能可以节约打印时间，因为不必一再打印重复的码。

大多数现行的汇编程序都是二次扫描汇编程序。该程序通过对源码的二次扫描生成目的码。第一次扫描收集该程序所用的全部标号和符号，并对其规定所需的地址和数值，再把它们放进一个大表格中。第二次扫描时，每当找到一个标号就生成实际的目的码，并引入表中，这样就得到了用于该标号处的地址或数据。早期的汇编程序属于单次扫描类型，因而不便使用标号，这是因为在一个标号可能被使用之前必须在源码中定义，这就难以引用或转移到程序后面才出现的代码。

连接程序用来将各段程序（或者只有一个程序）组合成一个最后的工作程序。在许多设计师共同编写一个程序而各人仅编写其中一段的场合，通常就要用到连接程序。当各段程序已经写好，并已汇编成功后，连接程序将所有的单段程序作必要的地址变更，使各程序段组合起来，按连接时所规定的地址产生一个可执行的最后程序。用这种方法，一个代码段可以这样编写和调试，使它先工作在特定的地址上，然后再与其它程序连接在一起，在不同的地址上去执行。连接程序的输出称为绝对码，即为供给目机的实际机器码。

调试程序为用户程序的交互执行和测试提供了一种极为有效的手段。当用户程序已经汇编和连接后，由调试程序装入绝对码。

对于某些小型开发系统，其中央处理器与目标系统的相同。绝对码可以装入该中央处理器并执行，无需用户硬件。这种方法对检验程序的逻辑性是很满意的，但不能在用户的硬件上按实际操作条件检验程序。

大多数开发系统用一种仿真程序来具体地调试程序。仿真程序是为使一种处理器能执行另一种处理器的机器码而编写的程序。它可以为许多不同种类的处理器的编写仿真程序，以允许设计师在一个开发系统上能为多种处理器开发程序。

一旦仿真程序投入运行，用户就可以检查目标程序的操作，在特定位置暂停以检验程序状态，或在程序运行时动态显示程序操作（如寄存器或存储单元的内容）。

仿真程序也允许用户修改目标程序的内容，而无需重新编辑和汇编。这就允许做小的修改程序并在它们插入原码之前就得到测试。

某些程序员为了方便起见愿意用高级语言编写程序，而不管实际的汇编语言格式。编译程序是用来处理有如PASCAL或PL/M这类高级程序语言的，使用编译程序可将高级语言的语句转换成目标微处理器的机器码。编译程序和高级语言使用方便，可节约很多编程时间，但不能生成效率很高的机器码。因此，由编译程序生成的机器码与目标系统的容量相比较，可能显得过于庞大，因此无法使用。

可以下这样的结论，开发系统的持续发展将成为微处理器系统最后价格的决定性因素。一个设计师越快越方便地开发一个微处理器的程序，则最终价格越低。当硬件价格下跌而劳务价值上升时，先进的开发系统对微处理器系统生产成本的下降，越来越成为决定性的因素。

1.2 系统开发流程

开发微处理器应用系统的全过程，通常包括三个方面：软件设计、硬件设计和程序诊断设计。分配给系统程序员的任务是编写应用软件，而逻辑设计师的任务是设计硬件。通常两组设计师还都要承担开发测试系统用的诊断程序的任务。对于小系统，一个工程师就可以承担所有三方面的工作；而对于大系统，可能每一方面的工作都需要分配几个工程师去完成。图1.2所示为系统开发全过程的流程图。注意软件开发和硬件开发可平行进行，以节约时间。

1.2.1 软件开发

开发软件的第一步是消化系统设计任务书，并编制一份完成预期任务、以达到任务书要求的流程图。此流程图务必结构合理，以保证有效的程序流。

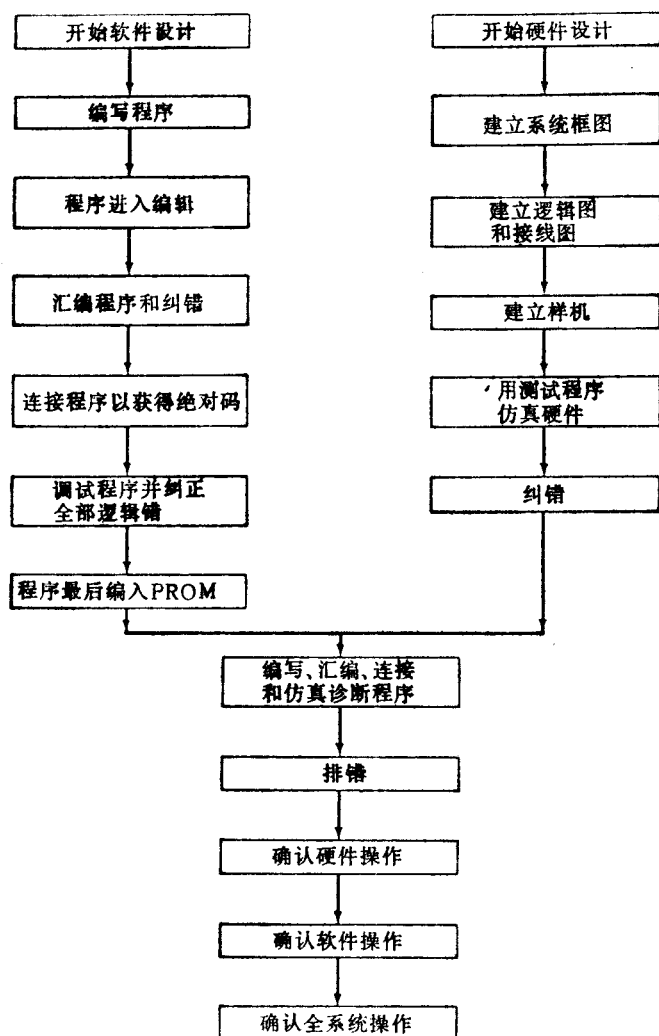


图1.2 微处理器系统开发