

微型计算机程序设计基础

(上册)

刘雅琴 赵旭文 编

国防工业出版社

微型计算机程序设计基础

(上 册)

刘雅琴 赵旭文 编

国防工业出版社

内 容 简 介

本书分上、下两册。上册有八章，第一、二章讲述程序设计的基础知识。第三、四、五章以 M6800 为主叙述程序设计模型、汇编语言及汇编语言的程序设计。第六章讲述 M6800 微型计算机的输入/输出，介绍输入/输出过程、接口电路、中断系统及应用实例。第七章讲述 M6800 开发系统及软件开发。第八章解剖 JBUG 监控程序和 24 位浮点运算程序，附有框图及程序清单。书末有附录。

本书可作为微型计算机软件训练班的教材及大专院校计算机专业的教学参考书，也可作为科研、设计和工矿企业等部门有关人员的自学参考书。

微型计算机程序设计基础

(上 册)

刘雅琴 赵旭文 编

*

国防工业出版社出版

新华书店北京发行所发行 各地新华书店经售

国防工业出版社印刷厂印装

*

787×1092¹/₁₆ 印张29 679千字

1986年10月第一版 1986年10月第一次印刷 印数：0,001—5,680册

统一书号：15034·2853 定价：5.90元

前 言

近几年来,我国已研制并生产了 DJS-050 和 DJS-060 两个系列的微型计算机,与国际上流行的 8 位微型计算机 Intel 8080 和 M6800 的功能基本相同,指令兼容,为推广应用微型计算机创造了条件。

在应用中遇到的一个重要问题是如何为这些机器编写应用程序。但在 8 位机上又很少使用高级语言,所以,迫切需要掌握的是用汇编语言进行程序设计。本书就是为满足这种需要而写的。微型计算机的程序设计入门并不难,但搞出较完美的设计也并不是轻而易举的事。

本书的上册除讲述程序设计的基础知识外,主要以 M6800 指令系统为主,重点讲述汇编语言的程序设计,并简单地介绍了程序设计中所必须了解的硬件知识。还介绍了 M6800 的开发系统及如何进行软件开发。最后讲解了两个较长的程序设计实例——JBUG 监控程序和 24 位浮点运算程序。下册是以 Intel 8080A/8085A 指令系统为主,讲述汇编语言的程序设计。本书并不讲述高级语言,有关这方面的内容国内已有专著。

目前,程序设计中的很多基本概念,各家提法不一,专业术语较为混乱。对此,本书在广泛吸取各种观点的基础上,加上编者的观点,力求这些基本概念的定義明确、严密。

本书的许多示例具有一定的实用性,并尽量将一例用几种方法来编写程序,以启示读者的程序设计思路。此外,对这些程序设计,都经编者上机调试验证。

在编写此书的过程中,得到编者所在单位的领导及许多同志的大力支持和具体的帮助;上册的全部内容 by 李木同志审阅,其中的主要章节又请何文兴,钱钧翹,王毓山、栗学忠、邵宝山和王英华等同志分别审阅,并提出了宝贵意见,在此一并表示感谢!

由于编者水平有限,错误和不足之处在所难免,请读者指正。

编 者

序

新的产业革命已经敲响了世界文明的钟声，科学技术的巨大进步，给人类带来了新的美好前景。人类正为自己智慧的创造性而震惊，并向着智力解放的目标迅跑。在这个高科学技术迅速发展的时代中，计算机、集成电路、特别是它们的共同结晶——微型计算机，给人类的进步提供了新的武器和希望。目前，微型计算机技术，即微型计算机硬件和软件是最活跃的一门学科，尤其在应用领域中，更显示出这门科学的强大生命力，并且成了巨大经济和社会效益的技术手段。

当前经济结构中，已经出现了小型化、微型化、多样化、多层化的经济趋势，这正反映了由微型计算机带来的新变化，同时又给微型计算机的应用提供了无限广阔的市场。

现在世界上已有几千万台微型计算机在忙碌着，为人类造福。它的发展趋势，如同洪水一泻千里，冲击着社会生活的各个角落，所到之处，为之焕然一新。计算机这项人类有史以来最激动人心的发明，正在影响着经济、社会、科学的发展，而且干预了人们的生产方式、工作方式、生活方式，虽然这一切还是刚刚开始，但人类无限美好的未来，将在人类智慧的推动下，乘着科学技术的风帆勇往直前。

世界新的工业革命正向我们发出一场严峻的挑战，同时也给我们提供一次跳跃机会。勇敢地迎接这个科学技术进步带来的挑战，抓住机会，实现可利用的挑战，努力实现向信息化社会的转变，尽快地建立起软件产业，将会大大加速我们实现四个现代化的建设步伐。

广泛地推广应用微型计算机就是一个跳跃的机会，不失时机地建立和发展我们的硅工业（大规模集成电路和计算机），大力推广计算机，特别是微型计算机的应用，就是一个带战略性的步骤。

发展信息技术，必须注重软件开发。《微型计算机程序设计基础》一书就是为普及推广微型计算机应用而编写的。尽管在编写中，可能有缺点和不足之处（这有待于读者批评指正），但这是个可喜的开端，也是广大微型计算机用户的一个佳音。

李 铁 映

目 录

第一章 微型计算机软件概述	1	6-1 输入/输出 (I/O)	239
1-1 微处理器、微型计算机和微型 计算机系统	1	6-2 接口电路	248
1-2 微型计算机软件	1	6-3 中断系统	265
第二章 微型计算机基础知识	11	6-4 微型计算机外部设备简介	277
2-1 微型计算机数字系统和编码	11	6-5 PIA 的应用实例	294
2-2 微型计算机的基本结构	24	第七章 M6800 的开发系统及其 软件的开发	323
2-3 微型计算机的运算	40	7-1 M6800 的开发系统	323
第三章 M6800 微型计算机的程 序设计模型	54	7-2 软件开发	332
3-1 M6800 微型计算机结构	54	第八章 JBUG 监控程序及 24 位浮 点运算程序	353
3-2 M6800 系列微型计算机的概况	57	8-1 JBUG 监控程序	353
3-3 M6800 程序设计模型	59	8-2 系统子程序	375
3-4 M6800 指令系统	61	8-3 24 位浮点运算程序	393
第四章 M6800 汇编语言	93	附录	442
4-1 汇编语言的约定	93	附录 1 M6800 指令系统表	442
4-2 伪指令	97	附录 2 M6800 指令寻址方式及执行 时间	450
4-3 汇编源程序	101	附录 3 M6800 指令机器语言矩阵表	451
4-4 汇编程序	103	附录 4 M6800 基本指令英汉 对照表 (按字母顺序)	452
第五章 M6800 汇编语言程序 设计	106	附录 5 本书常用的缩写字符英汉 对照表	454
5-1 程序框图	106	附录 6 十六进制——>十进制变换表	455
5-2 M6800 的简单程序设计技术	112	附录 7 十进制——>十六进制变换表	456
5-3 循环程序	143	附录 8 ASCⅡ 字符代码表	456
5-4 M6800 算术运算程序	163	主要参考文献	457
5-5 子程序	195		
第六章 微型计算机的输入/输出	239		

第一章 微型计算机软件概述

1-1 微处理器、微型计算机和微型计算机系统

电子计算机是当代最重大的科学技术成就之一。目前，正向巨型、微型、网络 and 智能模拟四个方面发展。七十年代出现的微型计算机，是电子工业的重大突破。它对计算机今后发展的影响是不可估量的。巨型、网络 and 智能模拟也得随着微型计算机的发展而发展。将来的社会是个信息化的社会，而微型计算机，就是进入信息化社会的重要手段。

微处理器 (Microprocessor)，简称为 μP ，也有人称微处理机，它是构成微型计算机的主要部件。在数字电路系统内，能独立地执行程序，对指令和数据进行处理，完成算术或逻辑运算。微处理器是用大规模集成电路 (LSI) 制成，是半导体领域里的一个新产品，属于一种高级的电子元件。所以说，微处理器本身并不是可用的计算机，只有配上用大规模集成电路制作的存储器和输入/输出接口电路等才构成微型计算机，简称 μC 。该机属于存储程序式计算机。微处理器在构成微型计算机时，常称为微处理单元 (MPU)。以微型计算机为中心，按各种不同的需要加上外部设备、电源，用户存储器以及使微型计算机工作的软件构成了微型计算机系统，简称 μCS 。微处理器、微型计算机和微型计算机系统的概念示意如图 1-1 所示。

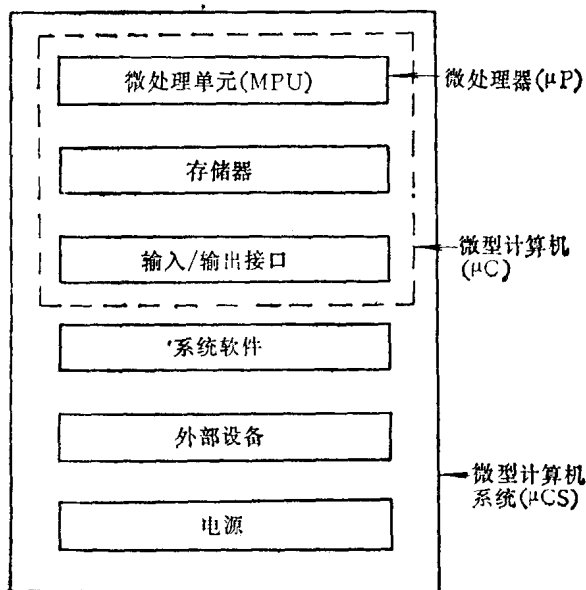


图1-1 μP 、 μC 和 μCS 的概念示意图

1-2 微型计算机软件

1-2.1 指令、程序和程序设计

一般来说，电子计算机在响应固定组合的二进制输入（即 0 和 1 逻辑变量）后，机

器本身就会执行简单的特定操作。这种由 0 和 1 组合的二进制外部输入称为计算机指令，也称指令代码。不同的计算机有不同的指令，但同类计算机的指令数是固定的。例如：M6800 有 72 条基本指令，Intel 8080 有 78 条基本指令。计算机执行不同的指令就能完成各种各样的动作。使计算机完成各种各样动作的指令序列就是程序，而编制指令序列的过程称为程序设计。这个指令序列就是人们事先安排好的计算步骤或控制过程。

1-2.2 什么是软件

电子计算机是通过人们编制程序进行工作的。当初，人们把构成计算机本身的物理部件，如 CPU、存储器、接口电路和外部设备等统称硬件；把由指令和数据组成的程序称为软件。硬件和软件是构成电子计算机系统中互不可分的两大部分。现在，软件所包含的内容已相当广泛，程序只是软件的一个组成部分。软件已成为研究如何发挥电子计算机功能的一门科学——软件科学，其中包括软件系统的构造和设计方法的研究。

微型计算机的硬件具有体积小、价格便宜、灵活性大和可靠性高等特点，因而，应用越来越广。但是，由于它的输入/输出设备不足及有限的内存容量，所以，对微型计算机的软件也产生了影响，形成它自己的软件特点，如开发系统软件、交叉软件、固化软件和分布式软件等。

1-2.3 微型计算机软件分类

1. 系统软件

系统软件是用来提高计算机的使用效率、发挥和扩大计算机的功能和用途，简化程序设计，简化使用方法的一套程序。系统软件可分为下述三种：

(1) 语言程序

在人和计算机之间，人们把符号、字母和数字等送入计算机，就相当于给计算机下达命令，让它去执行某一操作。这些符号、字母和数字就是人-机之间交换信息的语言，通称计算机语言，也叫算法语言或程序语言。这种语言可以是机器语言，面向机器语言，面向过程语言或面向问题语言。

① 机器语言

机器语言是计算机硬件能直接识别的程序语言。它是由 0 和 1 二进制代码组成。机器语言和具体的计算机或计算机系列有关。从这个角度看，机器语言就是计算机的指令系统，同一个指令码在不同的计算机上，具有不同的内容。例如，二进制数

1 0 0 0 0 0 0 0

作为 Intel 8080 的指令码，是把通用寄存器 B 的内容加上累加器的内容，结果送累加器。而作为 M6800 的指令码，是把累加器 A 的内容减去累加器 B 的内容，结果送累加器 A。

对同一个问题，两种计算机的机器语言程序也是不同的。例如：

〔例 1-1〕 8 位加法

将两个存储单元 \$ 40 和 \$ 41 的内容相加，结果存 \$ 42 单元。这里假定没有进位。\$ 表示后面的数字是十六进制数。

表 1-1a 和表 1-1b 是用机器语言编写的机器语言程序，也称二进制代码程序。这种

表1-1 a 用M6800的机器语言编写

地址 (十六进制)	二进制程序	十六进制程序
0000	1 0 0 1 0 1 1 0	96
0001	0 1 0 0 0 0 0 0	40
0002	1 0 0 1 1 0 1 1	9B
0003	0 1 0 0 0 0 0 1	41
0004	1 0 0 1 0 1 1 1	97
0005	0 1 0 0 0 0 1 0	42
0006	0 0 1 1 1 1 1 1	3F

表1-1 b 用 Intel 8080 的机器语言编写

地址 (十六进制)	二进制程序	十六进制程序
0000	0 0 1 1 1 0 1 0	3A
0001	0 1 0 0 0 0 0 0	40
0002	0 0 0 0 0 0 0 0	00
0003	0 1 0 0 0 1 1 1	47
0004	0 0 1 1 1 0 1 0	3A
0005	0 1 0 0 0 0 0 1	41
0006	0 0 0 0 0 0 0 0	00
0007	1 0 0 0 0 0 0 0	80
0008	0 0 1 1 0 0 1 0	32
0009	0 1 0 0 0 0 1 0	42
000A	0 0 0 0 0 0 0 0	00
000B	0 1 1 1 0 1 1 0	76

计算机可以直接识别, 并加以执行的程序称为目标程序(或结果程序或目的程序)。由于 M6800 和 Intel 8080 两种机器的指令系统不同, 所以, 对同一个题目的机器语言程序也完全不同, 所用的指令条数也有多有少。

此外, 从表 1-1a 和表 1-1b 还可看出二进制代码程序, 是由一二进制数字编写而成, 这有如下几个缺点:

- 1) 不便于阅读, 不直观, 难以交流;
- 2) 编写程序工作量大, 效率低;
- 3) 书写麻烦, 容易出错, 出错后难以检查。

基于上述缺点, 机器语言在实际应用上很不方便, 一般很少使用。

在计算机能够自动地把八进制或十六进制数字转换成对应的二进制数时, 可以用八进制或十六进制代码编写程序。这样, 比二进制程序多少有些改进。特别是十六进制数比较短, 检查也不那么麻烦, 出错也比较容易查找。所以, 微处理器大都采用十六进制数。

② 面向机器语言

用十六进制数编写的程序文本仍然难读, 难懂, 不易区别程序中的指令、数据或地址。

一种简单的代替机器语言的方法, 是把指令、寄存器、数据都加以命名, 并在程序

设计中, 使用这些名字代替二进制数。命名是用一种符号形式作为二进制形式的过渡, 也就是把机器指令符号化, 基本上保留了机器语言的指令格式, 只不过是使用一些特定的文字符号和数字, 按规定表示各种指令, 这些符号称为助记符。

实际上, 各个微处理器制造厂, 都为它的微处理器规定了指令的助记符。这些助记符表示出指令或寄存器所执行的实际操作功能。典型举例如下:

ADD——加法 (英文 ADD)
 SUB——减法 (英文 SUBtract)
 LDA——取数 (英文 LOAD)
 JMP——跳转 (英文 JUMP)

不同厂家给出的指令助记符略有差异, 程序员必须熟记这些命名和意义。每个厂家都规定了这些助记符所对应的十六进制代码, 即指令码, 一般都列出直观的指令表。M6800 的指令表见附录 2。用这样的指令表及有关寄存器的符号命名, 例〔1-1〕的程序可改写如表 1-2 所示。

表1-2 用助记符编写例〔1-1〕的程序

地址(十六进制)	内容(助记符)	地址(十六进制)	内容(助记符)
0000	LDAA	0004	STAA
0001	\$40	0005	\$42
0002	ADDA	0006	SWI
0003	\$41		

由表 1-2 可见, 用助记符 LDAA 代替 \$96, STAA 代替 \$97, 这是一种明显的改进。因为, 我们一看到 LDAA 这个助记符号, 不论那种微处理器, 一般都表示取数的意思, 就是把该符号下边的存储单元内容, 取数到累加器。M6800 有两个累加器, 故用 LDAA 和 LDAB 加以区分。表中用助记符编写的程序称为符号程序或汇编语言程序。所用的语言称为符号语言, 也称汇编语言。这是机器语言语句的符号化, 是一种面向机器的程序设计语言。因为, 它通常是特定的计算机或计算机系列专门设计的, 所以, 与机器语言有一一对应关系。

显而易见, 汇编语言程序比机器语言程序简单、直观和便于记忆、易懂、易写, 也便于检查和修改。然而, 计算机不能直接识别汇编语言程序, 还需要翻译成二进制代码程序, 这个翻译工作是由汇编程序来完成的。汇编程序简称汇编, 也有人称汇编器, 能把汇编语言程序翻译成计算机能够直接执行的目标程序。汇编语言的语句通常具有固定的格式。有时, 由人工按汇编格式逐条翻译成目标程序, 这叫手工汇编。如用计算机把汇编语言程序汇编成目标程序, 则称机器汇编。

目前, 由于微型计算机的内存比较小, 所以处理信息的能力也相对差一些。为弥补这一缺点, 可利用其他计算机 (大型或中、小型机), 来汇编微型计算机的汇编语言程序, 产生微型计算机的目标程序, 这一过程称为交叉汇编。与此相对而言, 用本机来汇编它的汇编语言程序称为自汇编。自汇编一般比较简单, 功能弱, 但它所占内存空间较少, 可存于 ROM, EPROM 或软磁盘里, 则称驻留汇编, 反之称非驻留汇编。

综上所述, 可把程序设计语言归纳成两大类:

1) 程序员使用的语言: 源语言, 包括汇编语言、高级语言(如 BASIC、PASCAL)等, 用这些源语言编写的程序称为源程序。

2) 计算机使用的语言: 机器语言。

有关汇编程序的详细内容, 见第五章, 这里仅把例〔1-1〕手工汇编如下:

表1-3 a 用M6800的汇编格式编制例〔1-1〕的源程序

LDAA	\$ 40	取第一个操作数
ADDA	\$ 41	加第二个操作数
STAA	\$ 42	存结果
SWI		

表1-3 b 汇编后的目标程序

存储器地址 (十六进制)	存储器内容 (十六进制)	指令 (助记符)
0000	96	LDAA \$ 40
0001	40	
0002	9B	ADDA \$ 41
0003	41	
0004	97	STAA \$ 42
0005	42	
0006	3F	SWI

表1-3 c 用Intel 8080 汇编格式编写例〔1-1〕的源程序

LDA	40H	, 取第一个操作数
MOV	B, A	
LDA	41H	, 取第二个操作数
ADD	B	, 操作数相加
STA	42H	, 存结果
HLT		

表1-3 d 汇编后的目标程序

存储器地址 (十六进制)	存储器内容 (十六进制)	指令 (助记符)
0000	3A	LDA 40H
0001	40	
0002	00	
0003	47	MOV B, A
0004	3A	LDA 41H
0005	41	
0006	00	
0007	80	ADD B
0008	32	STA 42H
0009	42	
000A	00	
000B	76	HLT

表中的H在Intel 8080 汇编格式中表示它前面的数字是十六进制数。

在微型计算机程序设计中，常常采用汇编语言。它与高级设计语言比较，专用性强、有赖于机器操作等缺点，但它却可以充分发挥微型计算机特有的灵活性，特别是对于各种过程控制，可编制出效率较高的程序。所以，本书重点介绍汇编语言以及如何用汇编语言来编制程序。即使是讲述微型计算机的基础知识，也都是为这一目的服务的。

③ 面向过程语言

为克服汇编语言编制程序中，有依赖于机器，使程序设计受到限制的缺点，人们设计了面向过程或面向问题的语言，也统称高级语言。这种语言允许你面向问题，就是把一般问题，按规定的少数通用命令（源码）写出程序（源程序），然后由翻译程序把它自动的翻译成计算机能够识别的机器语言或汇编语言程序（目标程序）。对面向过程语言有两种翻译方法，一种是解释程序，能把用高级语言写的源程序，在机器里一个一个语句地解释成对应的机器语言，边解释边执行，一条语言解释执行完毕后，再取下一条语句。解释程序的缺点是执行的速度较慢，优点是容易编写源程序，可以进行人机对话。在执行当中，可以随时修改自身的程序。这对初学者来说是非常方便的。微型计算机配备的BASIC语言是用解释程序进行翻译的，另一种翻译方法是编译程序，编译程序象接收输入数据一样来接收用高级语言写的源程序，经过它的处理后，象产生输出数据一样的产生机器语言或汇编语言程序。在执行中，这个翻译过程是不可逆的，所以，在运行时不能修改。微型计算机配备的FORTRAN，COBOL语言属于这一类型。

编译程序和解释程序之间的差别在于：解释程序不产生新的结果程序，而是直接执行源程序本身。编译程序输入的是源程序，而输出的是等价的结果程序。

高级语言的一个重要特点是与任何一种计算机无关。任何计算机，只要配备某一高级语言的编译程序，有的是解释程序，那么用这种高级语言编制的源程序，都可以在这些不同种类的计算机上执行。但是，编译程序是依赖于计算机的，也就是说，对同一高级语言，如COBOL，在不同的计算机上运行时，要配备不同的编译程序。编制一个好的编译程序，要求程序设计人员全面地了解所用机器的指令系统，并具有足够的编程经验，这些经验往往都是在编制汇编程序当中，逐渐积累起来的。

编制编译程序是费时较大的工作，工作量是以人年计算的。但是，一旦产生编译程序，对用户就十分方便，不必再去熟悉计算机本身的功能，而专心面向待解决的问题，这项工作一般由计算机厂家来完成，现在已开始用某种语言来生成编译程序，可节省时间。

前面已提过，用汇编语言或高级语言编制的程序统称源程序。这里要指出，汇编程序，解释程序和编译程序的总称为计算机的编译系统，主要任务是把源程序转换成目标程序。微型计算机系统的编译系统示意图，见图1-2。

由于高级语言已有许多专著，与微型计算机系统上所配的高级语言大同小异，所以，本书不再赘述。

（2）管理程序或操作系统

管理程序是管理计算机硬件和软件的各个部分协调工作的程序。对于程序规模大些，功能强些、或者管理复杂的管理程序又称操作系统。

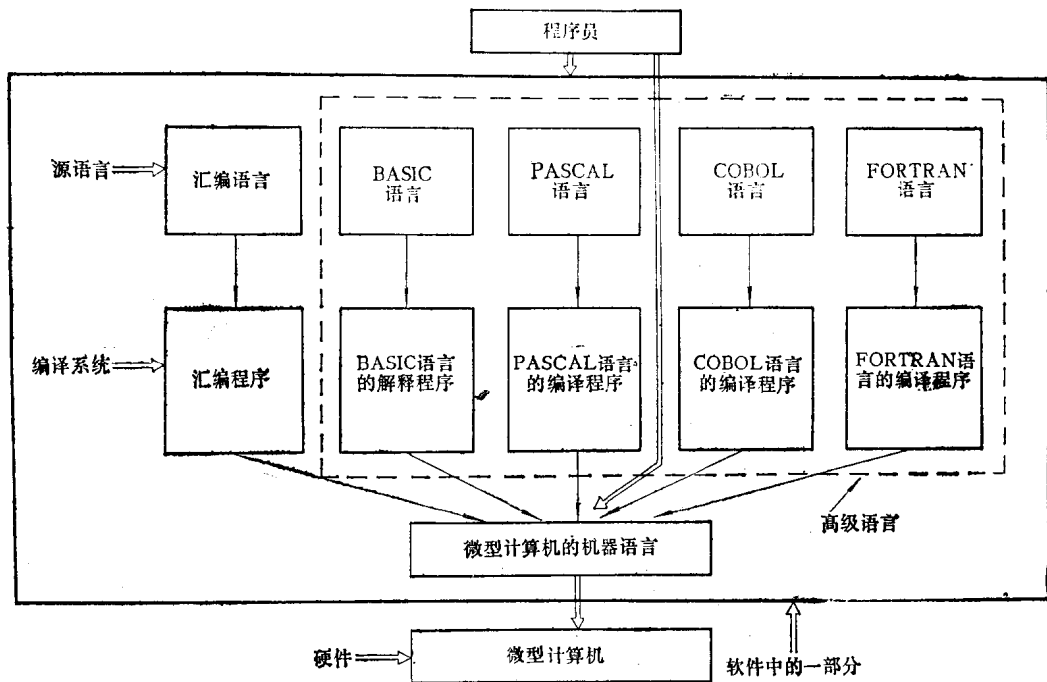


图1-2 微型计算机的编译系统示意图

单板微型计算机一般只有监控程序或监督程序，它包括人-机对话，命令处理分析，中断处理和外部设备使用等。其中，二进制引导程序，能把程序及其有关的数据等信息输入存储单元。微型计算机的监控程序是通过键盘命令来管理计算机工作的。一般都预先存在 ROM 中，使用时方便可靠。

为了充分地调度和管理计算机的各种资源，如 CPU、存储器、外部设备以及各种软件等，微型计算机也有了自己的操作系统。有了操作系统既能提高计算机使用的效率，同时又方便用户的使用。

操作系统的功能有 CPU 管理、存储管理、文件系统管理、外部设备管理以及作业（或任务）控制等。实时操作系统能及时收集和处理外来的信息，常用于实时生产过程的控制，分时操作系统是用来管理一台计算机为多个用户服务。它规定了一个时间片，与计算机对话的用户，可轮流使用机器的一个时间片。由于计算机的速度快，每个用户在思考问题、按键盘等过程中，就能为其他用户服务，能够管理多道独立的程序，同时在机器上运行，这就是多道操作系统。操作系统往往存储在磁盘中，根据需要调度各部份进入存储器，这又称为磁盘操作系统（DOS）。

目前，由于微型计算机硬件结构简单，处理能力有限，因此，它的操作系统要比中、小型计算机简单得多，一般都把操作系统存在价格低廉又有一定存储容量的软磁盘上，称为软磁盘操作系统（FDOS）。

微型计算机配备了功能较强的软磁盘操作系统及有关的高级语言，这就改变了手工汇编的方法，加快了软件的研制进程。

（3）服务程序

服务程序就是对计算机服务的程序，微型计算机有多种服务，其中一部分如下：

① 诊断程序

它用来诊断微型计算机硬件故障，有 MPU 电路的诊断、指令的检查、存储器的诊断以及各种外部设备的诊断。诊断程序可缩短维修时间，提高维修效率。

② 连接装配程序

它能按照一定的要求，把各个程序块输入内存单元，经过它的处理后，把地址浮动的各个程序块（程序段）相互有机地连在一起。

③ 文本编辑程序

它对程序块（或程序段）具有编辑功能，也就是按需要可以增加、删除或替换程序中的某些部分，使程序块完整而有机地连接起来。

④ 调试程序

它用于调试已经编好的程序，在调试过程中，可以使用各种命令，用来设置断点，显示或修改寄存器及存储单元的内容，帮助排除程序中的错误。

⑤ 系统程序包

目前，微型计算机的基本指令，除了整数加减法以外，还有乘除、小数四则运算以及各种初等函数运算。为满足用户在各方面的要求，计算机生产厂往往都配备上述运算所需的软件包。M6800 有内容丰富的软件包，用户使用十分方便。

2. 交叉软件

用其他计算机（中、小型）来编制或调试微型计算机的软件时，所使用的软件称为交叉软件。交叉软件有交叉汇编程序（见上节）、交叉编译程序、交叉模拟程序等。

（1）交叉汇编程序

交叉汇编程序和自汇编程序都有汇编功能，所以，它们的结构相同，不过交叉汇编程序通常可以用 FORTRAN 语言等来编写。

（2）交叉编译程序

它能在一台计算机上编译另一台计算机的程序。如果微型计算机使用的交叉编译语言是主计算机算法语言的一个小集，那末这种交叉编译研制工作就简单多了。

（3）交叉模拟程序

它能在一台计算机（宿主计算机）上模拟执行另一台计算机的程序。它能对目标程序的运行状态进行监控与测试，既容易发现和纠正程序中的错误，又易于发现和纠正系统中的动态性错误，并能提供多种形式的输出，给制做 ROM 或 PROM 带来很大的方便。

DJS-060 系列微型计算机模拟器，是以 DJS-130 小型计算机为宿主计算机，以 060 微型计算机的目标纸带为输入的交叉模拟程序。它不依赖 DJS-130 机的其他软件而独立的模拟 060 目标程序及其运行状态。例如，可对 060 目标程序中的任意一段进行模拟执行，显示和设置其寄存器、存储器的内容，建立在各种要求下的断点等，这些都可由用户交叉地指定。

3. 开发系统软件

微型计算机开发系统，也称发展系统，是微型计算机一种新系统。它除了具有一般微型计算机所有软件外，还有仿真器、实时跟踪部件、数据采集器和带有多个通道的测

试探头、PROM 编程序写入器等。开发系统的探头插入被测系统有关插座后,被测系统就可以共享开发系统的资源,这对被测系统中硬件和软件的调试都很有用。开发系统软件能够仿真用户程序的实时运行,通过各种命令显示或修改寄存器、存储单元的内容,在存储单元之间搜索用户指定的内容,能按指令逐条仿真执行程序,显示所需要的信息,对程序设置断点,可以使程序执行到某一条指令就停止等。

生产过程控制的应用软件,在开发系统上调试并仿真实时运行,就可以减少应用软件在现场的实验时间。

M6800 微型计算机的开发系统软件十分丰富,例如 EXORterm 220,有磁盘操作系统 (MDOS),高级算法语言 BASIC 语言、FORTRAN 语言、COBOL 语言、宏汇编、文本编辑等。

4. 应用软件

微型计算机应用日益广泛,它在生产过程中,可用作常规调节器、程序控制器、直接数字控制器等,应用软件种类繁多,不再赘述。但是,应用软件的编制过程大致如下:

- 1) 分析应用课题;
- 2) 建立数学模型 (用数学方程,逻辑式等);
- 3) 建立数据库;
- 4) 编制程序;
- 5) 编制各种文件;
- 6) 调试软件;
- 7) 可以把应用软件写入 PROM。

1-2.4 微型计算机软件发展动向

(1) 与小型计算机软件兼容

小型计算机均有丰富的软件,微型计算机在设计时与小型计算机指令系统相同,因此,可以有效地利用小型计算机的软件资源。例如微型计算机 LSI-11/23 和 PDP-11/23 软件兼容;微型计算机 μ NOVA 与小型计算机 NOVA 软件兼容。

(2) 分布式的软件

微型计算机采用分布式的应用,可以提高系统的可靠性,这时各个微型计算机的功能有主从之分,任务明确,互有联系。于是出现了分布式的操作系统和分布式的数据库。例如,UniFLEX 就是一种多用户、多任务的操作系统。它是仿照贝尔系统公司的 UNIX 型的文件结构研制而成的,它的特点是文件系统采用分层结构, I/O 与设备无关。

(3) 软、硬件结合的软件编制

在编制微型计算机软件时,软硬件紧密结合,相互配合,从而产生较好的效果。例如 M6800 微型计算机,多倍字长加法采用了十进制调整指令,使十进制加法程序简短明了。

(4) 出现交叉软件

就目前水平来看,微型计算机与小型计算相比,MPU 速度较低,内存容量有限,

外部设备也不全。为了充分利用小型计算机的资源, 弥补微型计算机的不足, 所以, 在小型计算机上编制和调试微型计算机的软件是很有利的, 这就出现了交叉软件。例如在 DJS-130 机上, 模拟 M6800 的指令系统。

(5) 专用性软件的固化

由于大规模集成电路技术的迅速发展, 成本逐渐降低, 特别是 EPROM 的出现, 就可以把已经调试好的专用性软件固化在只读存储器 (ROM) 或写入 EPROM 中。例如, 单板机 MEK6800D₂ 的监控程序就是固化在 ROM 中, 这样既提高了软件的可靠性, 又方便用户。对于用户临时专用软件可写入 EPROM 里, 如有变动还可以改写。

(6) 新颖的开发系统软件

开发系统软件是微型计算机特有的一种新颖软件。它是一个综合丰富的软件系统, 既能用来调试软件和硬件, 又能对实时系统进行模拟实验。有的开发系统甚至可以用来开发几种不同类型的微型计算机软件, 开发系统还能进行软件的固化, 它通过 PROM 编程写入器, 把已调试好的软件, 写入可编程序只读存储器中。

(7) 不断配备新的高级语言

微型计算机系统一般都配备高级语言。只是这些高级语言的容量比大、中型计算机小一些而已, 并且, 都是从大、中型计算机系统经过择优选取后, 移植到微型计算机中来。现正在把流行的新的高级语言也移植过来, 取代那些已不能满足需要的一些语言, 如 BASIC 语言等。一些新的语言有 PASCAL 语言和 C 语言。

PASCAL 语言是 1968 年由 Nicholas With 研究成的, 而 C 语言是七十年代初, 由贝尔实验室的 Dennis Ritchie 作为 Unix 操作系统的一部分提出来的。这两种语言在程序构成、数据结构与程序语言方面有很多类似之处。PASCAL 语言是采用较多的英文字, 容易读出, 便于检查, 所以, 应用较广。而 C 语言是用标点法, 它比 PASCAL 语言更活, 效率高。Unix 操作系统就是用 C 语言写成的。它提供很多软件和 C 语言程序库, 这些程序非常实用于编写系统控制用的实时操作。因此将越来越多的为人们所接受。

第二章 微型计算机基础知识

本章是此书的基础知识，将向读者介绍微型计算机的数字系统和编码。二进制数字和编码是微处理器的基本语言，十进制数字系统是人们的习惯计数法，而十六进制是微型计算机程序设计中通常采用的数字系统，了解这些知识会给程序设计打下良好的基础。

微型计算机的软件与硬件之间的联系十分密切，不了解微型计算机的硬件基础知识，就很难进行程序设计。所以，在第二节简要地介绍了微型计算机的基本结构。这些知识是用汇编语言进行程序设计所必不可少的，更详细的硬件知识请见有关专著。

本章的第三节重点讲述了微型计算机的二进制算术及逻辑运算原理，并引进“位组合格式”的概念，为上述运算写出规律性的公式。

2-1 微型计算机数字系统和编码

2-1.1 引言

微型计算机同其他类型计算机一样，是以数字为对象进行大量的数据计算和处理。程序设计更是离不开数字。而采用什么形式的数字是直接影响计算机的性能和结构的。当然，对于使用者，要想使用好微型计算机，熟悉数字系统是十分必要的。数字作为信息，有数量信息和符号信息之分。数量信息是参与运算的数据，在机器内表示成二进制，通常称为代码；最常见的符号信息是文字符号，用编码的办法表示成二进制代码。本节从十进制分析开始，重点讲述二进制和十六进制。因二进制是微型计算机的基础，十六进制是微处理器的表示形式。最后给出数，即在计算机中的三种表示法，并讨论它们的一些性质，为以后微型计算机数的运算规律打下基础。

2-1.2 数制

我们日常习惯使用十进制记数。它是由十个不同的数字符号 0、1、2、3、4、5、6、7、8、9 组成的，通常把这些数字符号叫做数码。数码处于不同位置（或数位）代表的意义不同。这十个阿拉伯数码可以表示任意大小的数。

在数的表示法中，十进制并不是唯一的记数法，与十进制原理相同的数制还有二进制、八进制和十六进制等。而代表数制的基本特征是它的底数或基数。什么是基数，在某种记数系统中所具有的基本数码的个数就是基数。例如：十进制中有十个基本数码，它的基数为 10；而二进制中只有 0 和 1 两个基本数码，它的基数为 2。八进制、十六进制以此类推。在程序设计中，往往使用几种数制。为防止数制间混乱，当写一个数时，必须在下标注上基数。例如： 4096_{10} 表示以十为基数的数字，或者在一个数的前面加上一个字符，如 \$ 40，\$ 表示 40 是十六进制数。当然，在基数已经明确的情况下，下标常略去不写。