

# 单板机监控程序系统 分析与设计

叶勇 袁可风 编著

高等教育出版社

# 单板机监控程序系统 分析与设计

叶 秀 袁可风 编著

高等教育出版社

## 内 容 提 要

本书详细介绍了监控程序的结构、监控程序的功能及其实现,还介绍了设计、优化、调试监控程序的一般方法,提出了汇编语言级的广义指令的概念并介绍了其实现方法。

本书附录中给出了 HJBUG 监控程序的全部程序清单,还对配 TP 801 单扳机的几个监控程序做了简明介绍。对于单扳机、微机的开发应用具有很大的实用价值。

本书可作为大专院校计算机专业、自动控制专业及有关专业的教材和教学参考书,也可供从事微机开发应用的科技工作者阅读参考。

### 单扳机监控程序系统分析与设计

叶 勇 袁可风 编著

高等教育出版社出版

新华书店北京发行所发行

河北省香河县印刷厂印装

开本787×1092 1/16 印张11.75 字数270 000

1988年12月第1版 1989年2月第1次印刷

印数 0 001—8,350

ISBN 7-04-001105-0/TP·32

定价2.50元

## 前 言

当前,微型计算机的应用正处在一个蓬勃发展的阶段,越来越多的人对它产生了浓厚的兴趣,渴望尽快地掌握这个有力的工具,以便在学完“微型计算机原理及应用”之后,能够具有开发应用单板机的能力;有的同志在实际动手开发了一两个项目之后,希望能够在理论上加以总结提高,从而更高水平地开发应用单板机。为了达到这个目的,笔者以为可以在两个方面加以努力:第一,从理论上学习开发应用微机有些什么步骤,应遵循什么原则,并实际剖析一两个完整的应用实例,从中得到启发;第二,认真剖析一个较好的监控程序,从而对单板机的了解更深入。监控程序是一个和硬件紧密结合、经过多人精心设计,反复优化的软件,在程序结构和编程技巧上都很有讲究,在一定程度上可以作为汇编语言程序设计的范例。认真地剖析一个具体的监控程序,可以学到许多编程方法和编程技巧。

为了这样两个目的,笔者于1984年和万宁康、曹俊同志合作编著了《微型计算机及应用系统设计》(华中工学院出版社1986年出版),又于1986年着手编写讲义“监控程序的分析与设计”,并在华东交通大学开设该课程,受到学生的欢迎。

承蒙西安交通大学胡正家教授和刘甘娜副教授推荐,本讲义在全国非计算机专业计算机课程教材评审组会议上被推荐为选修课教材,使笔者倍受鼓舞,立即将讲义修改定稿,并将书名改为“单板机监控程序系统分析与设计”。

编著这本书的另一个目的是,使没有学过系统软件的读者,对系统软件能够有某一侧面的了解。如果读者通过学习本书,能够了解虚拟机的概念,多少知道一些监控程序作为裸机和用户之间的界面是如何工作的,那对笔者将是一个很大的安慰。如果有部分读者读完本书后能够着手研制新的专用监控程序,那正是笔者的希望。

在编著本书时,笔者假定读者已经学完“微型计算机原理及应用”或类似的教材。

我们在教学实践中深深体会到,只有把理论与实践紧密结合,以具体机种、具体系统为对象进行分析,才能使读者较顺利地掌握新知识。因此,本书先介绍监控程序的概念,以及它应该具备一些什么职能,它的结构有些什么类型等,然后具体剖析HJBUG监控程序,最后介绍设计、优化和调试监控程序的一般方法,并对该领域的研究前沿作了一些探讨,旨在为读者指出研究方向。

本书可作为大专院校的教材或教学参考书,也适用于工程技术人员自学。由于本书实践性强,HJBUG监控程序适合于开设有关监控程序的全部实验,因此,也可作为“汇编语言程序设计”课程的课程设计用书。

全书共八章,其中第一、二、七、八章由华东交通大学叶雾副教授编著,第三、四、五、六章由华东交通大学叶雾、袁可风同志编著,附录及插图由张帮明、陈敏两同志编绘。在本书定稿过程中,所有内容均经作者互相讨论。

笔者在研制HJBUG监控程序时参考了TPBUG-B和JXBUG,尤其是在1982年底JXBUG公布之前从曹俊同志处得到了JXBUG版本清单。本书初稿完成后,承蒙李大友副教授和姜秀芳同志进行了仔细的审阅并提出了许多宝贵的意见。在本书编写过程中,还得到了华东交通大学微机应用学科组和铁道电气化学科组及江西人民出版社计算机室的大力支持。在此一并表示诚挚的谢意。

由于作者水平所限及脱稿仓促,加上没有同类专著以资参考,书中的错误及不妥之处在所难免,敬请专家及读者不吝赐教。

读者若需要固化好的监控程序及使用说明书、实验指导书,可以和高教社软件编辑室或华东交通大学计算机教研室联系。

作 者

1987年8月于南昌

# 目 录

绪论 .....	1	第四章 用户程序的运行与调试 .....	52
第一章 用户接口 .....	5	4.1 引言 .....	52
1.1 键盘命令 .....	5	4.2 用户程序的单步执行 .....	52
1.2 广义指令 .....	7	4.3 用户程序的运行 .....	54
习题 .....	8	4.3.1 用户程序的进入和退出 .....	54
第二章 监控程序的结构设计 .....	9	4.3.2 断点 .....	54
2.1 引言 .....	9	4.3.3 允许设断点的运行程序功能 .....	58
2.2 模块单门接口法 .....	10	4.4 单步调用 .....	61
2.3 模块并列门控接口法 .....	13	4.5 在 ROM 区设置断点的功能 .....	62
2.4 模块层次接口法 .....	15	4.6 调试功能总框图及其实现 .....	63
2.5 HJBUG 监控程序的结构		习题 .....	73
实例 .....	17	第五章 外部设备的操作 .....	75
2.5.1 HJBUG 的键盘配置与结构框图 .....	17	5.1 磁带机的操作 .....	75
2.5.2 HJBUG 监控程序的工作状态 .....	18	5.1.1 内存信息的转储 .....	75
2.5.3 HJBUG 监控程序的主模块 .....	20	5.1.2 磁带信息的读入和转储检查 .....	85
2.5.4 扩展功能键的接口 .....	36	5.1.3 其它转储标准 .....	95
习题 .....	37	5.2 EPROM 的写入 .....	95
第三章 计算机状态的检查与修改 .....	38	习题 .....	100
3.1 引言 .....	38	第六章 编辑及其它功能 .....	101
3.2 存储器的检查与修改、连续送数、		6.1 插入和删除功能 .....	101
层次键控门 .....	38	6.2 数据块传送功能 .....	103
3.2.1 存储器的检查与修改、连续送数 .....	39	6.3 数据串查找功能 .....	103
3.2.2 检查与修改上一个或下一个存储		6.4 其它功能 .....	105
单元 .....	41	习题 .....	106
3.2.3 相对偏移量的计算 .....	42	第七章 监控程序的设计与调试 .....	107
3.3 寄存器的读写及其状态下的		7.1 监控程序的设计 .....	107
其它命令 .....	45	7.1.1 总体设计 .....	107
3.3.1 寄存器读写 .....	45	7.1.2 实现设计 .....	108
3.3.2 寄存器读写状态下的命令键		7.2 监控程序的优化 .....	110
操作 .....	48	7.2.1 巧妙地使用指令 .....	110
3.4 I/O 端口读写及其状态下的		7.2.2 利用已有模块 .....	116
其它命令 .....	49	7.2.3 模块合并 .....	117
习题 .....	51	7.2.4 其它优化技巧 .....	117

7.3 监控程序的调试.....	119	8.2.1 专用广义指令 .....	125
7.3.1 利用现有的监控程序来调试新的 监控程序 .....	119	8.2.2 调用广义指令 .....	129
7.3.2 专用于调试新监控程序的母 监控程序 .....	122	习题 .....	131
习题 .....	123	附录 1 HJBUG 监控程序的灵活 应用 .....	132
<b>第八章 专用监控程序</b> .....	124	附录 2 HJBUG 监控程序清单 .....	138
8.1 使用 CTC 定时扫描显示.....	124	附录 3 TP 801 单板机监控程序 的简明介绍 .....	177
8.2 广义指令的实现.....	125		

# 绪 论

## 一、什么是监控程序

单板计算机是结构简单、操作简便、性能可靠、成本低廉的微型计算机,被广泛地用于工业控制、智能仪器及数据的采集与处理等各个领域。然而,单板机本身是如何进行工作的呢?

我们在“微型计算机原理”课程中,已经了解了微处理机的各种芯片的工作原理,也知道CPU在复位之后按照冯·诺依曼体制,从某个单元开始取一条指令,执行一条指令,再取一条指令,执行一条指令,……机械地重复。由此,必然要提出这样一个问题:单板机复位后执行的第一条指令是什么?它为什么能够接受人们的命令,并将运行结果告诉人们?

用各种微处理机芯片构成一台裸计算机(即不含任何软件的硬件机器),是无法交给人们使用的。必须给单板机配备一个固化的系统软件,CPU在复位之后首先进入这一软件,并在它的管理下,向人们提示必要的信息,接受、执行人们的命令。这个系统软件,叫做**监控程序**。

由此可知,监控程序是单板计算机不可缺少的关键部分。正如人之不能没有大脑,单机计算机也绝不能没有监控程序。特别是单板计算机的软硬件结合非常紧密,存储空间不够大,监控程序就显得极为重要。可以这样说,对于同一台裸机,单板机的功能的强弱,在很大程度上取决于监控程序的优劣。

若从使用者的角度来看待监控程序,它是介于使用者与裸单板机之间的一个界面。使用者无法将命令输入一台裸机,只能通过监控程序的管理来使用单板机。

## 二、监控程序的发展过程

由于单板计算机的广泛使用,也由于监控程序的优劣在很大程度上影响着单板机功能的强弱,一些计算机专业人员致力于改进、重写监控程序,从而使监控程序有一个发展过程。

以Z80系列的单板计算机为例,美国Z80 STARTER KIT单板机的监控程序ZBUG是早期的监控程序,北京工业大学TP801单板机的监控程序TPBUG、长江计算机厂的CJ801单板机的监控程序CJBUG等基本上以ZBUG为蓝本。TP801、DBJ、CJ801等单板机在我国获得广泛的生产和使用,说明ZBUG监控程序的设计是成功的。设计中贯彻了尽可能以软件代替部分硬件的思想,使Z80单板机的键盘、显示器等接口电路得到简化。

在ZBUG监控程序的管理下,该单板机不仅具有输入、修改、执行和调试机器语言程序的功能,而且有信息转储磁带和从磁带读入内存的功能(即可以使用磁带作为外存),以及固化用户程序的“EPROM写入”功能。有了这些功能,用户只要有一台单板机,即使不再有任何专用的微机开发工具,也能独立进行单板机的开发应用。可以说,ZBUG所设置的12项功能都是最基本的。

然而,这一阶段的监控程序还存在一些不足之处:其一是程序结构欠佳。ZBUG监控程序



没有考虑程序的模块化、子程序化,连最基本最常用的显示程序、读键程序都没有写成子程序,使用户在研制应用系统时无法调用现成的子程序。同时,非子程序化的监控程序,不利于研制调试,也不利于根据具体需要对它进行删改、扩充和重新组合。其二是,ZBUG 监控程序不够严谨,存在一些缺欠。例如,在设置断点运行用户程序时,若未遇到断点就用MON键或RESET键中止用户程序的运行,则断点不能被清除。又如,读键程序不够严谨。在操作键盘时,若不小心同时按动了同一行的两个键,则会出现键盘译码错误或程序“死锁”、“乱跑”的现象。此外,还有用户无法使用非屏蔽中断,I/O口修改时有时会显示与实际值不相符等问题。其三,某些操作不够简便。最后,编程技巧不足,程序不够精炼,因而在2K字节的监控中实现的功能还不够多,只设置了12项最基本的功能。

1981年初,北京工业大学首先对ZBUG作局部修改,研制了配TP801A机(裸机与TP 801机相同,仅监控程序不同)的TPBUG-A监控程序,并被国内和香港的生产厂家广泛使用。继北工大之后,株州电子研究所、华东师大、长沙铁道学院等单位也对Z80单板机的监控程序作了深入的研究和改造。这充分说明了单板机的监控程序这时已经受到了计算机专业人员的广泛重视。

这一阶段对监控程序的改进主要是对ZBUG作局部修改,精炼程序以增加功能。由于仅仅只作了局部修改,增加的功能有限。前面提到的监控程序的几项缺欠并未得到解决。

然后,人们开始把注意力放在纠正原监控程序的缺陷、改进操作、加强人机对话和改进程序结构等内在的、实质性的提高,使监控程序更趋成熟。这时程序已经相当精炼,而且基本上子程序化了,这就有可能增加较多的功能,从而在监控程序中开始有越来越多的编辑功能。其中效果最为显著的有以下几种:

1982年,江西工学院推出配TP801单板机的JXBUG监控程序。JXBUG弥补了上述的一些缺陷,解决了MON键引起的“断点滞留”问题,向用户开放非屏蔽中断,键盘操作不再出现译码错误或“死锁”、“乱跑”现象,修改了“I/O口修改”的功能。JXBUG简化了操作,加强了“提示”、“判错”和“校验”功能,并扩大了新的功能。例如,增加了插入,删除等编辑功能,可以用模拟执行的方式在ROM区程序中安装和处理断点,派生了255档可调数字式低频方波发生器等等,使单板机的功能由TPBUG-A的16项增加到JXBUG的30项。JXBUG还尽可能地将程序子程序化,为用户提供了较多的子程序。

1983年,北京工业大学对TP801-A机的硬件进行了改进,推出了TP801-B单板机,并重新设计了新的监控程序TPBUG-B。由于江西工学院未发表JXBUG的程序清单,因而在研制TPBUG-B时未能参考JXBUG。与JXBUG相比,TPBUG-B监控程序同样基本上解决了上述四项缺陷,同样向用户开放了较多的子程序,增加了更多功能,达到38项。TPBUG-B的结构更为合理,增加了光标,操作也更加方便,遗憾的是,由于TP801-B单板机的硬件不同,TPBUG-B监控程序不能适用于在我国使用最广泛、影响最大的TP801单板机。

1985年,华东交通大学参考了JXBUG和TPBUG-B监控程序,推出HJBUG监控程序。HJBUG吸取了JXBUG和TPBUG-B各自的优点,并进一步优化程序,在2K字节的监控程序中压缩进更多的功能,达到43项。HJBUG在设置断点运行用户程序时,不论是按MON键还是按

RESET 键,都不会有断点残留,从而最终根除了断点残留问题。HJBUG 设置了快速转储功能,使用2400波特率进行录带和读带操作,使原来需3分半钟转储2K字节提高到10秒钟左右,HJBUG 还保留了300波特率的录带和读带方式,使它可以和其它监控程序向上兼容。

1986年,江西工学院发表了JXBUG的新版本新版本参照TPBUG-B采用了“模块层次接口法”(见第二章)的结构,并增加了TPBUG-B中的一些功能,达到42项功能。

与此同时,华东交通大学在开发应用与教学实践中,进一步改进HJBUG监控程序,推出了HJBUG监控程序的1.1版和1.2版,HJBUG1.1版优化了程序,增加了功能,达到了47项功能,并使各项功能的使用更加方便;HJBUG1.2版进一步提高了与TPBUG、TPBUG-A的兼容性,成为后期监控程序中兼容性最强的一个监控程序。几乎在所有其它监控程序上开发的应用程序,都可以在HJBUG的1.2版本的支持下运行。

### 三、监控程序进一步发展的趋势

后期的监控程序在程序结构及程序优化方面作了较大的努力,剩下的潜力已不多了。估计今后监控程序将会向以下几个方面发展:

#### 1. 某个领域专用的监控程序。

有的领域为单板机配了一些专门的接口,并且经常要对这些接口进行某些操作,因此希望能将这些操作写入监控程序并使之命令化。例如,用于工业控制的单板机配置了A/D、D/A转换器,则往往将A/D输入等功能写入监控程序,并使之命令化。

#### 2. 适用于用户要求长时间占用CPU的监控程序。

有时用户需要长时间占用CPU。由于现有的通用监控程序占用CPU来完成扫描显示的任务,如果用户长时间占用CPU,将会使单板机的LED显示一片黑暗。这时,监控程序可以考虑用中断方式扫描显示。

#### 3. 适用于在线操作和多任务并行的监控程序。

如何对多任务、多中断进行管理和调度?由于单板机内存容量小,要处理的业务较为简单且相对固定,操作系统中常用的状态排队法在这里并不完全适用。这是一个很值得研究的问题。

### 四、为什么要学习监控程序

1. 监控程序是一种强有力的系统管理软件,在监控程序的管理下,单板机的各部分硬件才能发挥其功能。通过对监控程序的学习,可以更深入地了解微型计算机的工作原理、工作过程,进一步了解软件和硬件的关系,知道单板机的工作如何从监控程序进入用户程序,又如何退出用户程序返回监控程序;知道单板机为什么能在监控程序的管理下一步一步地单步执行用户程序,并随时检查用户程序中各寄存器的状态...

2. 通过对监控程序的学习,将使读者对单板机有透彻的了解,从而在较高的水平上开发单板机的应用。不少初学者在学完微型计算机原理后问下一步该怎么办?我们的回答是两条:一是读懂监控程序;二是读懂一两个现成的用户程序(当然,要“读懂”软件就必须同时掌握硬件),模仿性地开发单板机的应用。而读懂监控程序正是为了较高水平地开发应用。

3. 有人说,系统软件是软件的“皇冠”。监控程序正是一个很典型的系统软件,麻雀虽小,

五脏俱全。通过对监控程序的学习,将会对系统软件窥视一个概貌。操作系统是一个更高级、更全面因而更复杂的系统软件,然而对于并非专攻操作系统的人员来说,详细地剖析一个操作系统是很困难的。正因为如此,即使是计算机专业在开设操作系统课程时,也只能原则地讲述操作系统应有什么职能,它们是如何实现的,以及操作系统的结构等等,而无法把一个具体的操作系统拿出来详细剖析。监控程序则不然。正是由于它的“小”(相对操作系统而言),才有可能在几十个学时内完全剖析清楚。因而,对于非计算机专业人员,仅需学习监控程序;对于计算机专业人员,在学习操作系统之前,学习监控程序将会是很有益的。但是,必须指出的是:监控程序毕竟过于简单,丝毫没有涉及到计算机系统的存储器管理、设备管理、文件系统、作业调度等一系列问题。因此,对此感兴趣的读者可以去学习操作系统。

4. 如从软件的角度静态地观察监控程序,则它与一般的大型汇编语言程序没有本质的区别,是程序和表格的集合。也就是说,监控程序是一种软件,它是由指挥和管理单板机运转的程序和表格两部分内容构成。由于监控程序一般限制在2K字节的容量以内(有的为4K),所以必须精心设计,使之具有较强的功能。尤其是后期推出的监控程序,在程序结构和编程技巧上都很有讲究,经过反复优化,在一定程度上可以作为汇编语言程序设计的范例。认真地剖析一个具体的监控程序,可以使读者学到许多编程方法和技巧。

# 第一章 用户接口

单板机配有监控程序,用户通过监控程序来控制和使用裸机。因此,用户和监控程序之间必须建立某些联系手段,以便请求监控程序为之服务和指挥单板机工作。这些联系手段包含如下三方面内容:

- (1) 键盘命令;
- (2) 子程序调用;
- (3) 广义指令。

通常把这些内容称之为用户与监控程序的接口,简称用户接口。

## 1.1 键盘命令

单板机配有一个 24~36 键的小型键盘,在监控程序的管理下,用户可以通过键盘使用单板机。为了简化操作,一般将人们可能需要的操作实现键命令化,并实现一键一命令,而不是象计算机系统那样用回车符隔开的一串字符表示一个操作命令(也有个别单板机用一串字符表示一项命令),用户只要按下相应的键,单板机就立即执行用户所需的操作。

### 一、键盘命令功能

目前,不同监控程序的键盘命令的种类与功能虽不完全相同,但归纳其内容基本上包含如下几个方面:

#### 1. 查询和修改机内状态

常见的查询和修改机内状态的键命令有:

- (1) 寄存器读写命令。查询和修改用户程序中各寄存器的内容。
- (2) 顺序或反序查询寄存器命令。把机内用户程序的各寄存器编上号,可以按顺序或反序检查和修改各寄存器内容。
- (3) 寄存器间接寻址的存储器读写命令。键入寄存器代号,然后按该命令键,可以查询和修改该寄存器(对)间接寻址的存储器的内容。
- (4) 存储器读写命令。查询和修改内存各单元的内容,但 ROM 单元的内容只能检查而不能修改。

(5) 顺序或反序查询存储器命令。按地址顺序或反序查询和修改存储器的内容。

#### 2. 控制外设

常见的控制外设的键命令有:

- (1) I/O 端口读写命令。查询和修改 I/O 端口的内容(有的 I/O 端口内容不可修改)。
- (2) 顺序或反序查询端口命令。按端口地址顺序或反序查询和修改端口内容。

(3) 磁带机控制命令。将内存中的内容转储到磁带上,或将磁带中的内容装入内存。有普通速度和快速的两种方式。

(4) 通讯接口命令。有的单板机可以和上级计算机或其它智能设备通讯联机,并将其操作命令化。

(5) EPROM 查询及写入命令。有的单板机可以控制 EPROM 写入器,也有的单板机本身即附设 EPROM 写入器,并将其操作命令化。

(6) 打印操作命令。有的单板机配有打印机,并配有几种打印格式的键命令。

### 3. 输入和编辑用户程序

常见的输入和编辑用户程序的命令有:

(1) 键入程序命令。按内存地址顺序连续键入程序的机器码或表格的数据。

(2) 磁带机输入程序。上面第 2 条已经提到。

(3) 相对偏移量处理命令。计算和存放相对偏移量。

(4) 插入命令。在内存的程序区和数据区插入指令机器码和数据。

(5) 删除命令。在内存的程序区和数据区删除指令机器码和数据。

(6) 数据块传送命令。在内存中将整段的程序或整块的数据从一处移到另一处。

(7) 存储区置数命令。在内存的某一块空间置入数据。

(8) 数据串查找命令。在内存中查找已知的数据串。

### 4. 调试用户程序

常见的调试命令有:

(1) 运行用户程序命令。转入用户程序运行,直到按下 MON 键或 RESET 键为止,或遇到程序中设置的断点为止。

(2) 单步执行命令。在监控程序的管理下,执行一条用户程序的指令。

(3) 单步调用命令。与单步执行命令类似,但将子程序的调用过程当作一条指令处理。

(4) 断点的设置与清除命令。

### 5. 其它

有的监控程序还配有一些其它的键命令,例如,信号发生器工作命令、数制转换命令、考机命令等等。有的 CMOS 单板机的监控程序中还附设有时钟,可以通过键命令显示时间。

## 二、扩展命令键功能的措施

单板机的键盘很小,除数字键外,只有 8~20 个命令键(一般为 12 个)。如果功能命令比较多,可以采取一些措施来解决。

### 1. 分上、下档,作双功能键使用

例如 TPBUG-A 监控程序。这种方法需要一个键作上档键使用,加上执行程序命令、单步执行命令采用双功能键不妥,所以,只能将 TP 801 机的 12 个键扩展为 20 个键命令。而且,这种方法操作不够简便,后期的监控程序已不再沿用。

### 2. 状态区分多功能键

同一个命令键在不同的状态下,作为不同的命令键使用。由于不同的状态下,显示器有着

明显的差别,所以不会引起混淆。这种措施不需换档键,如果键命令安排得合理,将会使操作非常简便。巧妙的安排有时可使一个命令键具有 3~5 个功能而不产生任何困惑,因此,后期功能较多的监控程序多数采用这种方式。

### 3. 数字键兼作扩展命令键

状态区分多功能键虽然可以大大地扩展键命令的数量,然而要在同一键上标明众多的功能有时却很困难。这时可以使用数字键兼作扩展命令键使用。

### 4. 字符串命令

类似于计算机系统,采用字符串加回车键的方式可以使键命令的数量几乎不受限制。然而这种方式的<sub>操作不够简便,加上单板机的功能命令不会太多,所以这种方式的使用并不广泛。</sub>

后期的监控程序,往往将第 2,3 两项措施结合起来使用。

## 1.2 广义指令

### 一、广义指令及其引入

为了方便用户,一些监控程序(尤其是专用监控程序)往往将用户常用的操作编成子程序,并使之命令化,以便用户使用键盘操作。例如,用于数据采集和处理的单板机的监控程序,常常把 A/D 通道的输入数据的操作和某些常规数据处理方法编成子程序并使之键命令化,用户只要按相应的命令键就可以随时命令单板机执行所需的操作。有时,用户希望单板机能自动地按规定进行一系列操作。当然,他们可以使用一系列的子程序调用指令编成一段用户程序交由单板机执行,顺序地调用各键命令对应的子程序。但是这就要求用户在编制程序时不断地查阅各个子程序的入口地址,很不方便。

为了扩展监控程序的功能,提高监控程序的灵活性,方便用户,监控程序可以为用户提供一种方便灵活而又实用的手段,随时为用户服务。这种手段,就是我们所要介绍的广义指令。所谓广义指令就是指用户程序在程序这一级上请求监控程序为之服务的一种手段。它不是一条简单的机器指令,它的功能是由监控程序中的一段程序完成的。

由于广义指令不是一条简单的机器指令,所以如果用户程序中使用了广义指令,将不能被一般的反汇编程序反汇编出来,这也有利于软件的保密。

监控程序的广义指令主要有两种形式,分述如下。

### 二、专用广义指令

这种广义指令有一个字节的操作码,没有操作数或有若干字节的操作数。我们将监控程序的某些键命令编上代号,则这种广义指令就可以直接利用这些代号作为它的操作码。

由于专用广义指令的操作码很可能和某条实际机器指令的操作码相同,所以这种广义指令的使用受到很大限制,它们不能和实际的机器指令混合编程。

### 三、调用广义指令

这种广义指令有两个字节的操作码,没有操作数或有若干个字节的操作数。其操作码的第一字节,就是一条重启动指令,其操作码的第二字节,就是该命令的代号。

调用广义指令可以和普通机器指令混合使用,编制的程序可以作为普通机器语言的程序执行,因而不必定义出广义指令执行键。

关于广义指令的进一步讨论,将在第八章进行。

## 习 题

- 1.1 什么叫用户接口?
- 1.2 用户通过哪些手段与监控程序发生关系?
- 1.3 键盘命令基本上应包含哪几方面内容? 试举例说明之。
- 1.4 扩展功能键的措施有哪几种? 各有什么优缺点?
- 1.5 什么叫广义指令? 为什么要设置广义指令?
- 1.6 试比较专用广义指令和调用广义指令的优缺点。

## 第二章 监控程序的结构设计

### 2.1 引言

一个监控程序的研制过程大体上经历总体设计、实现设计、调试和维护四个阶段。其中总体设计包括功能设计和结构设计两个方面；实现设计包括确定表格、画框图和程序设计三个方面。所谓功能设计是指确定监控程序将有些什么功能，在键盘上如何安排，如何操作等等。其有关的一些问题已在第一章中提到过，下面将结合实现设计分别在第三章至第六章中详细论述。本章介绍监控程序的结构及其实现。

在进行监控程序结构设计时，必须首先确定结构设计目标，根据设计目标选择结构方案。通常提出的目标是：

#### (1) 系统的可靠性高

对于任何一个系统，可靠性都是必须优先考虑的问题。监控程序并不算太复杂，可靠性问题不象操作系统那么突出，但是如果考虑欠妥也可能产生严重后果。例如 ZBUG 监控程序在同时按下同一行的两键时会发生命令错译、“死锁”或“程序乱跑”，在实验时问题不太大，至多把待调试的程序冲掉而已。但如果在这样一个监控程序的管理下对现场进行控制，就有可能发生事故。所以，一般应把可靠性作为结构设计的主要目标。

#### (2) 易读性好

进行结构设计时应注意作到清晰易懂，这样既便于教学和交流，又便于修改和维护。

#### (3) 适应性强

所谓适应性具有两种含义，其一是它的功能和操作方式是可变的和可增删的，其二是它易于从一台单板机移到另一台上使用。例如，在 TP 801 单板机上增设 A/D、D/A 通道，要求单板机的监控程序增加 A/D、D/A 操作的键命令，若能将原有的监控程序稍加修改后就投入使用，可以减少重复性劳动。显然，在结构设计阶段就注意到这样一个问题是十分重要的，也只有在这一阶段就开始考虑这个问题才能达到预期的目的。

#### (4) 性能高

监控程序的性能主要体现在三方面：

- a. 功能多，每个功能应尽可能强；
- b. 操作简便，有提示信息以求减少操作错误；
- c. 尽可能多地允许用户使用监控程序中的子程序和表格。

监控程序是一个执行系统。如果设计成一个交织得比较紧密的复杂系统，那么对它作轻微的变动都可能影响到全局，使系统变得难于修改和扩充。人们为了使监控程序具有较大的适



应性,引进了模块接口法。顾名思义,该方法具有两个特色,一个是程序的“模块化”,一个是模块间的“接口”。所谓“模块化”就是在确定系统总的功能之后,将其划分为若干个具有独立功能的部分,并将每一独立部分编制成一个称之为模块的程序。为使每一模块不致于太复杂,往

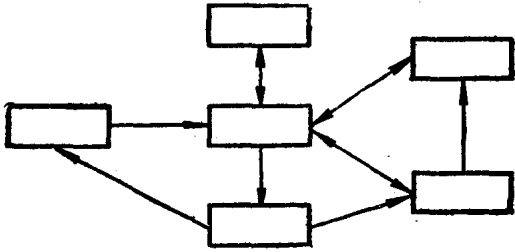


图 2.1 模块接口法示意图

往又把大模块分成更小的模块,使系统成为一种积木式的结构。所谓“接口”,则是在上述“模块化”的基础上,通过某种“接口”形式(如跳转、调用、中断等)把这些大大小小的模块连接起来,形成一个完整的体系。显然,这里的“接口”是指软件之间的接口,与第一章中所述的用户接口及硬件模块之间的“硬接口”器件是截然不同的。用这种接口法形成的系统,显然要比交

织成一体的紧密形式易于调试、增删和重新组合,并使结构关系变得简单而清晰。其特点是结构紧凑、接口简单直接、“时空”效率高、利于修改和扩充。如图 2.1 所示。

若尽可能把上述模块编写成子程序,则除了上述特点之外,还可以便于用户的调用,并且可以使监控程序本身更加精炼,减少很多重复的程序段,腾出空间来增添新的功能。

模块接口法的各模块之间的一切联系都是靠“接口”来实现的,因此,设计一个良好的接口是该方法的关键所在。

根据模块接口方法的不同,监控程序的结构可以分为两大类。第一类是模块门控接口法。这种方法是监控程序在初始化之后进入显示与扫描键盘的主程序段,等待键盘命令,一旦接到键盘命令,即进行键分析,并经过一个“门”转入相应的键命令处理模块。由于扩展键功能的方法不同,在程序结构上又可进一步分为单门结构、并列结构和多层结构。模块门控接口法在技术上较成熟,可靠性较高,目前大多数正在运行的监控程序都是采用这种结构。然而,采用这种结构要占用 CPU 来显示和扫描键盘。因此,当 CPU 去干某一件事时,将会不显示,键盘也将不接受命令。第二类结构是模块中断接口法,采用中断技术作为接口方法。这种方法是用计时器中断控制扫描显示,并在键盘操作时申请中断(因而要求硬件设计保证键盘可以申请中断),从而进行键分析进入相应的键命令处理模块。这种接口方法允许 CPU 在干别的工作的同时,显示和接受键盘命令,而且为多任务、多中断源提供了可能性。但是,如绪论中所说,这种方法用于单板机监控程序在技术上还很不成熟,仍是一片待开垦的处女地。

本章只介绍模块门控接口法的几种结构,至于模块中断接口法留待第八章再稍加讨论。

## 2.2 模块单门接口法

若监控程序的键命令数量较少,可以做到一键一功能,则其程序结构可采用模块单门接口法。例如 ZBUG 监控程序就是采用这种结构。

### 一、结构框图

模块单门接口法监控程序的结构框图如图 2.2 所示。