

微处理机的 程序设计和软件研制

[英] F·G·邓肯 著 白英彩 译



上海科学技术文献出版社

微处理机的程序设计 和软件研制

[英] F. G. 邓肯 著

白英彩 译

上海科学技术文献出版社

微处理机的程序设计
和软件研制
【英】F. G. 邓肯 著
白英彩 译

上海科学技术文献出版社出版
(上海市武康路2号)
新华书店上海发行所发行
浙江洛舍印刷厂印刷

开本 787×1092 1/32 印张 14.25 字数 341,000

1982年8月第1版 1983年10月第2次印刷

印数 10,801—24,800

书号: 15192·217 定价: 1.74元

《科技新书目》57-262

JS65/22

内 容 提 要

《微处理机的程序设计和软件研制》系根据 F. G. Duncan 所著 "Microprocessor Programming & Software Development" (Prentice-Hall international, Inc. 1979 年版) 一书翻译的。

该书属于论述微处理机的程序设计和软件研制方面的专著。全书共包括微处理机及其系统、指令、指令系统、算术运算的程序设计、非数值操作的程序设计、用高级语言的程序设计、软件的组织与语言和结构、小型 8080 系统用的第一个软件及其文本等八章。书中运用了类似于计算机硬件描述语言使多种型号的微处理机指令有了统一的书写格式, 对于微型计算机的教学和应用颇有意义。

该书适于从事微型机的设计、生产和应用的广大工程技术人员以及大专院校和中等专科学校有关专业师生阅读。

译 序

随着计算技术的飞速发展,计算机的应用也日趋普及,特别是微型计算机的应用尤其广泛。要想很好地应用微型计算机,就必须了解其软件。所谓微型计算机的软件是指它的各种程序。它能使微型计算机充分发挥效能,为用户使用机器提供方便。因此,微型机的软件是其重要组成部分。不管是系统软件还是应用软件,对于设计、生产和应用微型计算机来说,都是非常重要的。为了更好地适应计算技术的发展形势和促进计算机应用事业,特翻译了《微处理机的程序设计和软件研制》一书。

本书是在作者及其同事们多年来从事微型机程序设计的教学和运用各种型号微处理机的基础上写成的。该书最大特点是通过建立一种符号系统(或称硬件描述语言)来表示指令,能使不同厂家的各种微处理机的指令助记符归纳为一种统一的表示法。这对于微型机的应用和教学是很有意义的。借助本书所介绍的方法,可以把为某种型号(如 6800)的微处理机编写的程序很快地变换为其它机种(如 8080/8085, Z80 等)的程序。也就是说,使得我们运用某种型号微处理机而获得的经验具有普遍意义。

该书的另一特点是比较系统地阐述了一个软件包中的各个程序段,指出了它的编程要领和技巧。书中提供的各有关机种的指令索引表格,对广大工程技术人员是颇有裨益的。

运用本书提供的指令书写格式和所介绍的编程技巧,能进一步地扩大现有微型机的功能和用途,便于深入地掌握微型机

软件的基本概念和理论,提高编程能力,对推广普及微型机应用是有益处的。

本书内容深入浅出、通俗易懂、理论联系实际,是一本较实用的教学和专业参考书,值得一读。

本书稿付印前承蒙邱百光副教授悉心审阅;全部译稿经刘寿和、顾良士和吕宗祺等工程师仔细校对。他们提出了不少宝贵意见。在此一并致谢。由于译者水平有限,错误和不妥之处在所难免,恳请批评指正。

译 者

1982年2月

原 序

《微处理机的程序设计和软件研制》一书旨在向初用微处理机的人员介绍一些程序设计知识和基本的软件设计技术。说得准确一点,本书将作为在微处理机方面希望深入理解编写和使用程序的一些新手的教科书和参考书。

笔者所以写本书是试图把几种不同型号的微处理机归纳起来。随后又与承担同样任务的师生进行了广泛的讨论,并根据有许多背景和兴趣各不相同的听讲者参加的许多课程和研究班的教学经验进行了充实。

本书主要供大专院校计算机科学系学生作为教材。在阅读本书时,假设读者尚不具备程序设计方面的知识,所需的逻辑设计方面的知识并未超出学生在一、二年级中所学得的知识。在布里斯托尔(Bristol)大学,本课程是对三年级学生讲授的。将来,当大学教学大纲广泛采用微处理机作为教学手段时,此书将作为一年级的教材。

有些阅历深的学生已熟悉了本书内容。譬如,有些学生是通过夜校和假期课程学习班,有些学生是通过自学而学完的。他们提出的非常有价值的批评和建议在本书中已予以考虑。因此,希望本书对于未来的微处理机用户有所裨益,这些用户已能胜任本领域的工作,但需要学习程序设计为其将来的工作以及确切地理解其他人的工作而打下扎实的基础。这类读者将会看到本书的主要内容是完善的,如在阅读第一章遇到困难时,则可以查阅有关逻辑设计方面的基础课本。这样一来就能较好地理

解微处理机系统的结构,并根据其需要来运用微处理机。

本书的叙述方法是实践性较强,虽然许多材料都是比较通用的,但所有详细的讨论都是以 Motorola6800、Intel 8080 和 8085、Zilog Z80 等四种用得最广的处理机为基础进行的^①。作为本书的一个组成部分,给出了一个完整的 8080 软件包,并作了全面的说明。

第一章从程序员的角度出发,概括地介绍了微处理机及其系统的硬件。

第二章详尽地描述和讨论了机器的指令系统及其作用。本章还引入了一种指令系统的通用表示法,在一定程度上可帮助读者从已掌握的机种顺利地过渡到另一种机种。这种表示法与构成由制造厂设计的“汇编语言”的所谓“助记符”系统迥然不同。它有很多优点,特别是对于不希望局限于一种机器的用户来说更有好处。对某些类型的指令(例如加法和减法)的用法作了详尽的讨论。如读者觉得这部分内容已超出了自己的需求,则在阅读第一遍时就可略过这一段,留待以后用到时作为参考。

第三章给出四种处理机(6800、8080、8085 和 Z80)的通用形式的指令系统表,以及将各指令系统中的指令(第二章已作介绍)按字母的顺序排列的索引表。这些表面上看来令人望而生畏的大块参考资料对于学习是很有帮助的。读者在后一阶段将会发现在编写程序时有这些材料和表格是很方便的。

第四章通过一系列由浅入深的例子介绍了程序设计技术。开头的一些例子为学习简单的输入、输出以及字符和数字处理方面的内容打下基础,其后的例子相当于软件包中的子程序,以

^① 当然,本书内容对其它处理机的适用程度将与这种处理机与上述四种处理机之间的差别有关。虽然其中有些材料对非 8 位字长的处理机也是适用的,但许多材料未必能有效地应用于位片式器件。

后各章将予以详细介绍。

第五章由一系列多少有点脱节的小节组成。唯一的共同之处是各节所讨论的课题都不是处理数量的问题。有些课题(例如排序问题)相当详细地讨论了不止一个实例;但是有些课题(例如文件和记录问题)由于篇幅所限只作了简要论述。

第六章介绍了有助于程序设计的一系列基本软件手段,详细地叙述了使用这些软件手段(包括在软件包里)的用户须知。本章末讨论了如何通过进一步的使用软件提高程序员的表达能力的问题。

第七章对微处理机软件的发展历史作了必要的说明,提及了可供使用的软件包,并讨论了将来各种可能的发展趋势。如果读者在做本书的程序设计练习以外,还使用某种高级语言的编译程序则,应该可以参加这种讨论,也许能达到这样的程度,即根据自己的思路来设计和补充。提供新的语言或其它更好的表达方式。

第八章包含有说明程序设计技术时所用的软件包的完整文本(附有详细的注释),而这些软件的使用须知已在第六章进行了介绍。这个软件包需占用 6K 字节的可编程只读存储器(PROM),由一个监督程序、一个 PROM 编程程序、一个汇编程序和一个反汇编程序以及一组算术子程序(包括 32 位的浮点运算程序)组成。汇编程序至少需要用 1K(最好是 2K)字节的随机存取存储器(RAM)作为工作空间。然而,正如第八章所说明的那样,部分软件包所需的 PROM 和 RAM 可以少得多。这个软件已使用了一段时间,因此对于读者来说将是有用的。当然,它也不是完美无缺的,未必能长久地满足一些要求高的用户的需要。读者既然能发现这个软件包对自己的用处、限制因素及其不足之处,以及仔细地研究了 this 软件包并了解其结构和缺陷,

也就能自行设计和实现好的软件及其它程序。

程序设计是一种实践技能，只有在具体的机器上通过实践才能获得。显然，如果读者拥有能运行本书中的软件包的 8080、8085 或 Z80 系统，则从本书所学到的东西，将比那些只使用其它某种机器的读者所学到的来得多，而后者又比无法接触任何机器的读者好得多。这三种读者聚集在一起，学习为他们的机器编制程序，则都将会有所收益。众所周知，懂两种语言的程序员远比只能用一种语言的程序员能更好地运用新的机器和语言。能为自己购买设备的初学者使用不同的两个小系统所学到的东西远比从一个较大的系统中学到的东西多得多。

本书对有关交叉汇编程序、交叉编译程序、具有很强的诊断能力的调试工具或开发系统方面的内容叙述得很少。这些东西很繁杂，会妨碍初学者的学习，即使是有经验的程序员使用起来也会感到容易混淆。这些东西当然非常方便，很有吸引力。但是正确的方法是要在其上运行某个程序的系统是研制该程序的最合适的系统；虽然错误是不可避免的，但是，混乱和不必要的复杂化则不是不可避免的；在程序设计和任何其它设计工作中，简单、设计良好并容易理解的工具比精巧、复杂而不可预言的机构更为可取。

书中肯定会有错误，会有不应该有的混乱和太复杂化的地方，这都是由于笔者的学识浅薄，书中表达的全部见解也都只是作者的一孔之见。如果书中有任何值得一提的地方，那功劳也主要是属于我的许多同事、教师、学生和朋友们，他们有意和无意地给与了许多帮助。由于为数众多，只能在此一并致谢了。

附言(1979年6月)

笔者撰写本书时，第八章的软件已经过改编供 8085 系统使用，其中

的 PROM 在 0000-1FFF(4×2716), RAM 在 2000 以上。§ 8.7 提出的一些改进建议已经实现了。使用起来稍微方便一些的汇编程序是在一只 2K 的 PROM 中 (1800-1FFF); 具有新的行式打印机命令的监督程序、PROM 编程序和反汇编程序位于另一个区域 (0000-07FF); 算术和输入-输出子程序位于第三个区域 (0800-0FFF); 而第四个区域 (1000-17FF) 供存储盒式磁带机程序和正在研制的高级软件用。这个系统的用户应该注意下述变换:

8080	8085	8080	8085
0360, 03C0	0371, 03C9	0D6F-0D7F	086F-087F
07E6-07FF	0886-089F	0DCA-0FFF	08CA-0AFF
0BE0-0BFF	08A0-08BF	2400-27FF	0C00-0FFF
0CF1-0CF7	08C1-08C7	2B80-2BFF	0B80-0BFF

F. G. D.

目 录

第一章 微处理机及其系统	(1)
§ 1.1 基本概念	(1)
§ 1.2 微处理机的内部结构	(9)
1.2.1 电源	(9)
1.2.2 时钟	(11)
1.2.3 定时和控制器	(11)
1.2.4 指令寄存器和译码器	(12)
1.2.5 运算器(ALU)	(13)
1.2.6 条件标志	(13)
1.2.7 累加器	(13)
1.2.8 程序计数器	(14)
1.2.9 堆栈指示器	(14)
1.2.10 其它寻址寄存器	(14)
§ 1.3 存储器	(15)
1.3.1 主存储器和辅助存储器	(15)
1.3.2 半导体存储器	(15)
1.3.3 地址的缓冲和译码	(19)
§ 1.4 外围设备	(22)
1.4.1 接口	(22)
第二章 指令	(24)
§ 2.1 基本概念	(24)
§ 2.2 指令的表示方式	(25)
§ 2.3 通用的表示法	(29)
2.3.1 指令各字节的表示方法	(29)
2.3.2 寄存器	(30)
2.3.3 赋值(信息的复制或传送)	(30)
2.3.4 信息交换	(30)

2.3.5	数值	(31)
2.3.6	地址和存贮单元	(32)
2.3.7	条件标志	(37)
§ 2.4	算术和逻辑运算	(37)
2.4.1	一个字节的意义	(37)
2.4.2	带符号和无符号的数值	(33)
2.4.3	加法	(40)
2.4.4	取反和取负	(46)
2.4.5	减法	(47)
2.4.6	逻辑(布尔)操作	(50)
2.4.7	“测试”和“比较”操作	(52)
2.4.8	“移位”和“循环移位”操作	(55)
2.4.9	其它算术和逻辑操作	(60)
§ 2.5	堆栈操作	(63)
§ 2.6	转移	(65)
§ 2.7	子程序	(66)
§ 2.8	输入-输出	(69)
§ 2.9	中断	(73)
§ 2.10	Z80 的其它指令	(79)
第三章	指令系统	(84)
§ 3.1	说明	(84)
§ 3.2	Motorola 6800	(87)
§ 3.3	Intel 8080 与 8085	(95)
§ 3.4	Zilog Z80	(102)
§ 3.5	四个机种的指令汇总表	(115)
第四章	算术运算的程序设计	(133)
§ 4.1	可供实用的最小系统和软件	(133)
§ 4.2	ASCII 码	(135)
§ 4.3	读取十进制数字	(135)
§ 4.4	十进制整数的输入和输出	(143)

§ 4.5 二进制整数的算术运算	(151)
4.5.1 改变数的长度.	(152)
4.5.2 比较.	(153)
4.5.3 单字长乘法.	(154)
4.5.4 多字长乘法.	(157)
4.5.5 单字长除法	(160)
4.5.6 多字长整数的除法.	(164)
§ 4.6 非整数量的二进制算术运算	(165)
4.6.1 非整数量的“定点”表示法	(165)
4.6.2 乘法	(166)
4.6.3 加法与减法	(166)
4.6.4 除法	(169)
§ 4.7 “浮点”二进制数	(170)
4.7.1 表示法与习惯	(171)
变异的表示法	(172)
4.7.2 规格化	(173)
4.7.3 辅助程序	(175)
4.7.4 浮点子程序概述	(177)
4.7.5 浮点输入与输出	(179)
4.7.6 用浮点子程序编制简单的程序	(182)
4.7.7 条件标志与浮点数	(187)
§ 4.8 二十进制(BCD)算术运算	(188)
4.8.1 无符号整数的加法	(189)
4.8.2 带符号的 BCD 整数	(191)
4.8.3 BCD 乘法	(194)
4.8.4 BCD 小数	(197)
第五章 非数值操作的程序设计	(198)
§ 5.1 非数值量	(198)
§ 5.2 复制成组的信息	(193)
§ 5.3 检索表格	(202)

§ 5.4	排序	(207)
§ 5.5	“组装式”信息——长度小于一个字节的量	(217)
§ 5.6	记录与文件	(220)
§ 5.7	特殊的外围设备	(221)
第六章	用高级语言的程序设计	(225)
§ 6.1	用机器码进行程序设计的基本工具	(225)
6.1.1	监督程序	(227)
6.1.2	汇编程序	(231)
6.1.3	反汇编程序	(234)
6.1.4	PROM 的编程与读出	(234)
§ 6.2	子程序	(235)
6.2.1	输入子程序	(235)
6.2.2	输出子程序	(238)
6.2.3	定点算术运算子程序	(241)
6.2.4	浮点算术运算子程序	(249)
§ 6.3	从机器码向高级语言发展	(250)
§ 6.4	将高级语言翻译成机器码	(260)
第七章	软件的组织、语言与结构	(263)
§ 7.1	计算机软件的发展	(263)
§ 7.2	软件的规划与设计中的—些考虑	(266)
§ 7.3	指令表示法与语言	(268)
§ 7.4	程序设计语言的结构与风格	(277)
第八章	小型 8080 系统用的第一个软件	(282)
§ 8.1	历史	(282)
§ 8.2	监督程序 PROM(0000~03FF)	(284)
8.2.1	监督程序——主体部分	(285)
8.2.2	专用子程序	(286)
8.2.3	监督程序——分支部分	(288)
8.2.4	其它子程序	(291)

8.2.5 PROM 编程器用的程序 (0360~03EB)	(298)
§ 8.3 汇编程序 (0400~07E2、0800~0BDF、0C00~0D6E、 0D85~0DC9).....	(294)
汇编程序用的子程序.....	(297)
§ 8.4 反汇编程序 (2800~2B7F)	(300)
§ 8.5 整数与定点算术运算的子程序	(302)
§ 8.6 浮点子程序	(303)
§ 8.7 错误与缺点	(307)
§ 8.8 软件的带注释的文本	(305)
§ 8.9 软件的十六进制文本	(433)

第一章 微处理机及其系统

§ 1.1 基 本 概 念

就内部结构而论,微处理机是一种超大规模集成电路,它相当于在4平方毫米左右的硅片上集成了数千个分立元件,封装成40条(典型值)引脚的双列直插式组件。就逻辑原理而论,微处理机是一种由时钟信号控制的时序电路,也即能用时钟脉冲同步地改变其内部状态,从而改变其输出信号的一种电路。时钟脉冲通常(但不是全部)是由外部产生的。微处理机各个内部状态的变化都是由其当时的内部状态以及一组输入信号共同决定的。上述说明同样也适用于JK触发器之类的器件,其差别只是复杂程度不同而已,因为微处理机可能出现大量的输入信号组态、内部状态以及输出信号组态。

然而,微处理机的工作原理却相当简单,实质上只是内部状态的重复循环。这种循环称为指令周期,指令周期总是由下述阶段组成:

(i) 将一组输入信号(一条指令)读入(锁存入)处理机的一个内部寄存器(指令寄存器)。

(ii) 处理机要经过一系列由该指令的各位所决定的状态。可能还要读入其它输入信号(数据)或产生输出信号(结果)。这就是指令的执行阶段。

(iii) 最后,产生一组输出信号(下一条指令的地址),供外部电路(一般是存贮器)决定要送入处理机的下一条指令。

使处理机完成某个特定任务所需的一串指令称为程序。要

1111405

511