

8080 8085

# 软件设计

C·A 泰特斯等著 张梅岗译 人民邮电出版社 [上册]

# 8080/8085 软 件 设 计

## 上 册

C. A. 泰特斯 等著

张 梅 岗 译

符 明 何 诚 审校

人 民 邮 电 出 版 社

8080/8085 Software Design  
by christopher A. Titus, Peter R. Rony, David  
G. Larsen, and Jonathan A. Titus  
Howard W. Sams & CO., Inc., 1979

## 内 容 提 要

本书通俗地讲述软件设计，分上、下两册。上册主要讨论 8080/8085 的基本指令和高级指令、应用程序、系统程序。书中给出了很多程序设计实例。这些程序具有很强的通用性和实用性。通过这些实例，总结了 8080/8085 软件设计的经验和技巧，告诉读者如何根据具体情况，设计出可靠的、有效的软件。

本书在讲述上采用循序渐进的方式，适于不具备很多计算机知识的人员阅读。同时，可供从事计算机应用的工程技术人员、大专院校有关专业师生参考。

## 8080/8085软件设计

### 上 册

C.A.泰特斯 等著

张 梅 岗 译

符 明 何 诚 审校

人民邮电出版社出版

北京东长安街27号

河北省邮电印刷厂印刷

新华书店北京发行所发行

各地新华书店经售

开本：787×1092 1/32      1985年7月 第一版  
印张：15 页数：240      1985年7月河北第一次印刷  
字数：335 千字      印数：1—20,000 册  
统一书号：15045·总3057—无6338  
定价：2.20元

## 译者的话

微型电子计算机体积小、重量轻、成本低、可靠性高，作为一种轻便而通用的电子控制装置，得到越来越广泛的应用。微型计算机开创了计算技术真正得以普及应用的新阶段，它不仅给电子工业和其它部门带来了深刻的影响，而且，它将对人类的物质文明起着重要的作用，将出人意料地改变人们的生活方式。

近几年来我国微型计算机的研制和生产，发展很快，主要产品有 DJS-030、DJS-040 DJS-050 和 DJS-060 几个系列。DJS-050 相当于美国 Intel 公司的 8080 A 微处理机系列。这种处理机及其系列部件，我国已经能够批量生产，今后的问题主要在于推广应用。随着微型计算机的广泛应用，软件显得越来越重要。

《8080/8085 软件设计》一书，通俗地讲述软件设计，内容丰富。全书分上、下两册出版。书中不仅讨论了 8080/8085 的基本指令和高级指令、应用程序、系统程序和文件库，而且还介绍了组装 8080/8085 微型计算机系统所需要的硬件及其组装方法，所以它可以作为微型计算机应用的参考书。我们还要特别指出，书中给出了三百多个程序设计实例，具有广泛的通用性和实用性，拿到机上就可以运行，解决实际问题。因此，本书

可作为程序设计手册使用。《8080/8085 软件设计》总结了这种机器的软件设计的经验和技巧，告诉读者如何根据应用环境和现有资料，在充分理解和分析问题的基础上设计出可靠的、有效的、完善的软件。本书提供了许多改进软件设计的技巧，可供程序设计者借鉴。本书可供大专院校计算机专业的师生、广大工程技术人员和计算机爱好者作参考。

最后，需要对本书程序中的指令格式作一点说明。本书仍采用了原著者所使用的指令格式。严格地讲，这种书写格式是不很恰当的。如原书中 `MOVAE` 的正确格式应为 `MOVA, E`。读者在书写程序或使用本书所给的程序时，应采用图 3-6 给出的指令格式。

湖南大学谢卓杰教授认真地审阅了全书，在此表示衷心感谢。

我还要特别感谢顾徐珍同志，她协助我工作，付出了辛勤的劳动。

译者衷心希望广大读者对本书的错误和不妥之处给予批评指正。

一九八三年十月

## 序 言

英特尔 (Intel) 公司制出了四位字长的微处理器“芯片”4004, 从而向人类世界引入了微处理器和微型计算机的概念。这在当时看来还不象是一场革命。随着对早期集成电路的改进, 微型计算机革命得到了有力的推动, 越来越多的厂家开始生产微处理器集成电路和微型计算机。目前, 这些由一种化学元素——“纯化的砂子”作成的器件, 实际上已经开始改变着人们的生活方式。

在超级市场, 电子现金出纳机不仅能完成机械现金出纳机的各种任务, 而且还能进行编目控制。有些缝纫机再也不需要用复杂的齿轮和凸轮去完成特殊的针法了。当需要使用特殊的缝针法时, 可以用微型计算机来控制针的位置。在不久的将来, 你甚至可以“教”缝纫机做最新式的服装。

计算器可以取代计算尺以完成复杂的计算任务, 目前计算器的功能要比一、二十年前生产的先进的计算机更强。钟表不再需要上发条了, 术语“顺时针”和“逆时针”不久也许将成为历史名词。今天, 你甚至可以使用一种电子记录现有余款的支票簿。再也不必每月都进行结账了。在不久的将来, 微波炉“知道”的烹调技术也许比你知道的还要多。

当然, 使用微型计算机解决某个问题, 需要硬件和软件。

硬件包括微处理器集成电路、某些存储器集成电路、发光二极管的接口电子器件，以便微型计算机能对外围设备（指示灯、电动机、阀门、线圈和仪表）进行控制。软件由微型计算机执行的指令序列组成，有了它，微型计算机才能处理数据或控制外部设备。遗憾的是，软件的设计和实现并不象硬件的设计和实现那样确定。

现在，有些普通微型计算机的用户用 BASIC 程序设计语言给计算机编程序。要这样做，就必须把 BASIC 的解释程序（一种汇编语言程序）存入到 5000 到 8000 个存储单元中。有了这种汇编语言程序，你就能把 BASIC 程序装入微型计算机，并且实际地加以执行。象世界上的其他任何事物一样，用 BASIC 语言给微型计算机编程序，有优点，也有缺点。由于用 BASIC 语言编程序存在着缺点，因此，在微型计算机上进行程序设计时，常常使用汇编程序设计语言。这种语言正是我们要在本书讨论的程序设计语言。

毫无疑问，汇编程序设计语言在教和学两方面都比较困难。你不可能用汇编程序设计语言在 10 分或 15 分钟内编写一个程序，让计算机算出 1 到 1000 之间的各个数的立方根（有效数字为 6 位）。如果使用 BASIC 程序设计语言，也许能做到这一点。但是，使用汇编程序设计语言是有优点的。事实上，用汇编语言可以完成的许多任务，用 BASIC 语言则不能完成。特别是在过程控制、外围设备控制、高速计算和实时数据采集等方面更是如此。实际上，装有微型计算机的大多数消费电子产品都用汇编语言编制程序；例如：电视游戏机，微波炉，缝纫机，电子现金出纳机（出售点终端），煤气泵，血压监控器等等。所以，汇编程序设计语言是一种功能很强而有用的语言，值得学习。

在本书的第一章，我们并不讨论汇编程序设计语言的程序设计，而只是介绍 8080 和 8085 微处理器集成电路的特性；包括这两种微处理器芯片内的寄存器。这些寄存器是很重要的，因为你在用汇编程序设计语言编程序时要反复用到它们。紧接着的三章，讨论 8080 和 8085 能实际执行的汇编语言程序指令。由于你对理解长而复杂的程序用的各条重要指令还不够熟悉，所以在头几章，读者见不到长而复杂的程序。掌握了这些重要指令之后，读者才能开始用汇编程序设计语言给微型计算机编写程序，完成有用作业。本书其余各章讨论 8080 汇编程序设计语言指令的应用，诸如算术运算、数制转换、输入/输出设备（外围设备）控制等。

本书具有一些与众不同的特点，我们相信读者对这些特点会深为赞赏。我们并没有用一章来专门讨论二进制、八进制和十六进制各种数制。我们只在某些章节中告诉读者如何用纸和笔来做这些数制的转换运算。但是，我们自己更感兴趣的是微型计算机为完成上述运算所必须执行的指令序列。如果读者需要了解更多的有关基本算术运算的知识，可以参考有关计算机程序设计的其他著作。读者还会发现，凡是以 8080 微处理器为基础而组装的微型计算机，大都可以使用本书给出的程序。我们在本书给出的硬件或软件例子并不只是代表某个厂家的硬件特征。本书所给出的程序例子，下列公司的 8080 微计算机都可以运用，MITS 公司，处理机技术公司 (Processor Technology)，英特尔公司 (Intel Corporation) 国家半导体公司 (National Semiconductor corporation)，数字团体公司 (Digital Group)，控制逻辑公司 (Control Logic)，IMS 公司 (IMS Associates) 和 E & L 仪器公司 (E & L Instruments)。当然，各个微型计算机系统的外围设备可以各有不同，正如各个系统

的外围设备的地址可以各有不同一样。

在输入/输出这一章，即第七章，我们给出了用软件控制的外围设备的电路方框图。如果读者不知道外围设备怎样具体与微型计算机连接，那么控制该外部设备的程序软件也就没有什么用处了。此外，如果读者希望在自己的微型计算机上应用某个具体程序，则也许需要了解该程序控制的外围设备的电路方框图，以便照着进行硬件的连接。

本书的前三章讨论八进制和十六进制数形式的指令操作码。我们比较喜欢用八进制数表示操作码。因为我们深深地感到，用八进制数表示 8080 的汇编语言程序指令的操作码比较容易记住，而且，当读者检查八进制数操作码时，很容易确定这条指令与哪些寄存器或存储器的地址单元有关联。如果用十六进制数表示操作码，作出这种判断是非常困难的。不少程序员却比较喜欢用十六进制数表示操作码，因为他们只需要记住两位数字的操作码，而不需要记住三位数字的操作码。因此，我们只能采取在程序例子中两种数制都用的办法。我们建议：程序员初次给 8080 或 8085 微型计算机编写程序，最好使用八进制数，因为八进制数比较容易记住。日后，使用十六进制数也许更方便。如果读者确定不了使用哪种数制，则最好两种都试一下，然后选用自己感到最方便的一种。在本书经常可以看到十六进制数，它们是用括号括起来的，例如：(5F)。

我们的另一个目的，就是要对本书的程序例子如何操作给出详细的说明解释。我们并不说：“这儿的程序，你捉摸摸摸它是如何操作的。”如果采用这种方法，读者可能所得甚少。相反地，我们从能完成某个具体任务的最简单的指令序列开始，逐步开发出一个个程序。这些简单程序常常有某些局限性；碰到这种情况，我们就给它增添几条指令，使微型计算机能更好

地完成更通用的任务或某一特定的任务。因此，通过学习以前的例子，我们就能对某个问题设计出最佳的解题程序。

读者通览本书后可以发现，书中程序例子的计算机打印输出格式与其他书的作者所用的格式有所不同。我们选择的这种格式，有助于初学者掌握汇编语言程序设计的概念。其中最重要的概念之一是，必须把多字节的指令存储在连续的存储单元内。如果读者看一下其他书的程序例子的计算机打印输出，则很难掌握这一概念。正是这个原因，我们编写了常驻编辑程序/汇编程序(TEA)，用来产生程序表；我们认为，对于一本关于汇编语言的程序设计的书，这种做法是最合适的。值得强调的是，读者只要熟悉汇编语言程序设计的适当的方法，无论用什么编辑程序/汇编程序软件包来开发汇编语言程序都是可以的。请记住，编辑程序和汇编程序只不过是达到目的的手段。它们是用来帮助我们开发适当操作的汇编语言程序的工具。

软件的优秀特征之一是容易复制和分布使用。正是由于这一特点，不少专业杂志和业余杂志刊登了许多软件中的程序表，提供源程序和目标程序纸带或音频录音磁带。另外，还有许多软件库，其中包括：英特尔用户库 (Intel User's Library, 3065 Bowers Ave, Santa Clara, CA 95051)，微型计算机软件库(The Microcomputer Software Depository, 2361 E • Foothill Blvd, Pasadena, CA 91107)。

《8080/8085 软件设计》所阐述的有关程序设计的许多概念已经编入了大学研究生班的教材，由布莱克斯堡的 Tychon 公司发行。该教材分三个教程：微型计算机设计(626)，微处理器接口技术(628)，和 8080/8085 微处理器的软件设计(690)。如果读者希望得到这些教程，可以写信向 Tychon 公司的教程部经理索取 (P. O. Box 242, Blacksburg, VA 24060)。也可

以通过连续教育中心和弗吉尼亚工学院州立大学 (Blacksburg, VA 24061), 希望他们提供。为了了解更多的情况, 请打电话给 Linda Leffel 博士(703)(961—5241)。

C. A. 泰特斯

P. R. 罗尼

D. G. 拉森

J. A. 泰特斯

# 目 录

## 程序例目录

### 第 一 章

|                        |   |
|------------------------|---|
| 8080/8085 微处理器介绍 ..... | 1 |
|------------------------|---|

8080微处理器—机器语言和汇编语言—8085微处理器—本书用的数制—程序格式—8080和8085的相同之处

### 第 二 章

|                       |    |
|-----------------------|----|
| 8080/8085 的基本指令 ..... | 15 |
|-----------------------|----|

数据传送指令—用读/写存储器存储数据—立即数传送指令—简单的寄存器对指令—输入指令和输出指令—逻辑指令和算术运算指令(8位)—逻辑指令—算术运算指令—转移、传送控制和判定指令—小结

### 第 三 章

|                   |    |
|-------------------|----|
| 子程序与基本指令的应用 ..... | 38 |
|-------------------|----|

调用子程序—延时子程序—条件调用指令和返回指令—基本指令的运用—输入/输出设备的同步—电传打字机输入/输出和字符处理—电传打字机程序和终端程序—电锁—小结

## 第 四 章

### 8080/8085 的高级指令..... 163

寄存器对各种操作—堆栈指示器—DAD 类指令—直接装入指令和存储指令—利用堆栈来存储数据、地址和状态信息—再启动指令(单字节调用指令)—使用寄存器对 H 操作—A 寄存器的附加指令—进位指令—最后的结论

## 第 五 章

### 算术子程序..... 221

整数加—整数减—整数乘—整数除—BCD 算术运算—四位 BCD 数的操作—浮点算术操作—特殊的功能

## 第 六 章

### 数制的转换..... 300

三位 ASCII 八进制数—二进制数的转换—八位二进制数—ASCII 八进制数转换—两位 ASCII 的十六进制数—二进制数转换—八位二进制数—ASCII 的十六进制数的转换—三位 ASCII 的十进制数—二进制数转换—八位二进制数—ASCII 十进制数转换—十六位二进制数—ASCII 十进制数转换—问题在于转换还是不转换—在计数器程序和子程序中应用 DAA 指令—删除无效零—小结

## 第 七 章

|                       |     |
|-----------------------|-----|
| 微型计算机的输入/输出(I/O)..... | 372 |
|-----------------------|-----|

I/O 数据传送—总线控制—8080 与简单的 I/O 设备—8080 与键盘—用硬件编码器的键盘的软件和硬件—软件驱动的多路转换(扫描的)键盘—ASCII 键盘与 8080 微型计算机的连接—8080 和发光二极管显示器—存储器映象输入/输出—带有硬件编码器的存储器映象输入/输出键盘—存储器映象输入/输出的、多路转换(扫描的)键盘—存储器映象输入/输出发光二极管显示器—十位数字的多路转换显示器—小结

# 程序例目录

## 第一章

|     |                          |    |
|-----|--------------------------|----|
| 1-1 | 8080 微处理器的典型程序 .....     | 7  |
| 1-2 | 8080 的几条指令助记符及其操作码 ..... | 9  |
| 1-3 | 程序表的格式 .....             | 13 |
| 1-4 | 另一种可能的程序表格式 .....        | 14 |

## 第二章

|      |                                         |    |
|------|-----------------------------------------|----|
| 2-1  | 把同一个数值装入 B、C、D 和 E 寄存器 .....            | 22 |
| 2-2  | 在四个寄存器复制一个数据字节的两种方法 .....               | 23 |
| 2-3  | 把存储器单元 030 123 (1853) 的内容传到 D 寄存器 ..... | 25 |
| 2-4  | 把一个立即数据字节存入存储器 .....                    | 26 |
| 2-5  | LXIH 指令及其等效的 MVI 指令 .....               | 27 |
| 2-6  | 使用 LXIB、LXID 和 LXIH 指令 .....            | 29 |
| 2-7  | 把 8 位数据值传送给第 015 (0D) 号外部设备 .....       | 32 |
| 2-8  | 从第 103 (43) 号外部设备接收一个 8 位数 .....        | 32 |
| 2-9  | 屏蔽一个 ASCII 字符的高四位有效位 .....              | 44 |
| 2-10 | 用一条 ANDC 指令屏蔽四位高有效位 .....               | 44 |
| 2-11 | 输入、屏蔽循环移位以及合并两个 ASCII 字符 .....          | 46 |
| 2-12 | 用 ADDB 指令把 B 的内容加到 A 的内容上 .....         | 51 |
| 2-13 | 把 A 寄存器的内容和 B 寄存器的内容相加的程序 .....         | 51 |

|      |                                    |    |
|------|------------------------------------|----|
| 2-14 | A、B 两个寄存器的内容相加, 产生一位进位             | 51 |
| 2-15 | 两个数相加, 进位置逻辑 1 的程序                 | 53 |
| 2-16 | 两个 16 位数相加                         | 54 |
| 2-17 | 把 ADC 类指令用于 16 位加法运算               | 55 |
| 2-18 | 从 A 寄存器的内容减去 E 寄存器的内容              | 56 |
| 2-19 | B 寄存器的内容减去 E 寄存器的内容                | 56 |
| 2-20 | 做减法操作时, 产生一位借位                     | 57 |
| 2-21 | 从较小的数减去较大的数                        | 57 |
| 2-22 | 两个 16 位数相减                         | 58 |
| 2-23 | 寄存器对 D 的内容减去寄存器对 B 的内容             | 58 |
| 2-24 | 寄存器对 D 的内容减去寄存器对 B 的内容, 产生了一位借位    | 59 |
| 2-25 | 寄存器对 D 的内容减去寄存器对 B 的内容             | 59 |
| 2-26 | 用 INRB 指令使 B 寄存器的内容加 1             | 62 |
| 2-27 | 用 DCRE 指令使 E 寄存器的内容减 1             | 62 |
| 2-28 | INXH 指令的应用                         | 63 |
| 2-29 | 把存储器的内容装入 D 寄存器和 E 寄存器             | 64 |
| 2-30 | 把存储器的内容装入 D 寄存器和 E 寄存器的改进程序        | 64 |
| 2-31 | DCXH 指令的具体说明                       | 65 |
| 2-32 | HLT 指令的应用                          | 67 |
| 2-33 | 使用 NOP 指令在程序中留出空单元                 | 68 |
| 2-34 | 转移指令的格式                            | 69 |
| 2-35 | 返回到程序的起点                           | 70 |
| 2-36 | 用问号中止输入程序                          | 72 |
| 2-37 | 屏蔽其他 ASCII 字符(数字 0~9 和问号(?) 除外)的程序 | 74 |

|      |                                          |    |
|------|------------------------------------------|----|
| 2-38 | 先测试 ASCII 数字字符 .....                     | 76 |
| 2-39 | 使用循环移位指令和进位标识位来测试所选择的<br>一位 .....        | 78 |
| 2-40 | 用 ANI 指令和零标识位测试 A 寄存器的某一位 .....          | 79 |
| 2-41 | 用循环移位指令测试一个 8 位字的若干位 .....               | 81 |
| 2-42 | 按数据位 $D_6$ 、 $D_3$ 和 $D_4$ 的顺序进行测试 ..... | 82 |
| 2-43 | 使用 ANI 指令测试一个字的三位数据位 .....               | 83 |
| 2-44 | 等待一位数据变成为逻辑 0 .....                      | 84 |
| 2-45 | $D_2$ 位的置位操作 .....                       | 85 |
| 2-46 | 位清零或者位复位指令 .....                         | 86 |

### 第 三 章

|      |                                          |     |
|------|------------------------------------------|-----|
| 3-1  | 用 LXISP 指令对堆栈指示器进行装入操作 .....             | 93  |
| 3-2  | 200 毫秒延时子程序 .....                        | 103 |
| 3-3  | 30 秒延时子程序 .....                          | 104 |
| 3-4  | 30 秒延时简化的子程序 .....                       | 105 |
| 3-5  | 0.200 秒延时子程序, 程序中使用一条寄存器对减 1<br>指令 ..... | 106 |
| 3-6  | 调用 HAFMIN 子程序, 产生一小时延时的程序 .....          | 108 |
| 3-7  | 在存储器中没有存储换行符的 ASCII 字符打印程序<br>.....      | 111 |
| 3-8  | 二进制与 ASCII 码的十六进制的转换子程序 .....            | 114 |
| 3-9  | 简单的电传打字机输出子程序 .....                      | 118 |
| 3-10 | 在电传打印机上打印 B .....                        | 119 |
| 3-11 | 灵活地打印字符的方法 .....                         | 122 |
| 3-12 | 把键盘字符回送到打字机的程序 .....                     | 123 |
| 3-13 | 在输入/输出 (I/O) 程序中应用 ANI 指令带来的灵            |     |