

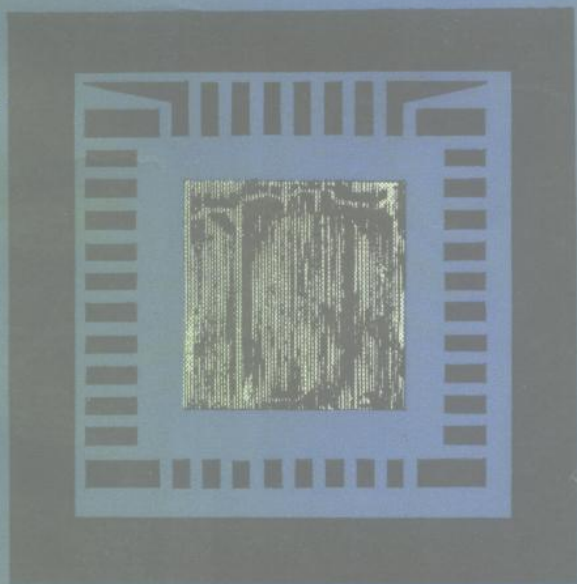
微型计算机 软件基础技术

6801 Z-80

6809

6802

6803



6800

8080

Z8

微型计算机 软件基础技术

(M6800系, 8080系, Z-80系)

[日]横井与次郎 著

唐长钧 徐子亮 译

煤炭工业出版社

内 容 简 介

本书是微型计算机程序设计的入门书。内容包括微型计算机程序设计基础知识, M6800系列、8080系列、Z-80系列的指令系统和汇编程序基础, 宏指令以及文本编辑。

本书可供微型计算机用户和从事微型计算机系统开发、应用及教学方面的科研、工程技术人员、操作员使用, 也可作为大专院校有关专业师生的参考书。

责任编辑: 李 秀 荣

横井与次郎著

マイクロコンピュータ

ソフトウェア 基礎技術

昭和55年12月15日再版 2刷発行

ラジオ技術社

東京都千代田区神田淡路町1-9

微 型 计 算 机

软 件 基 础 技 术

(M6800系, 8080系, Z-80系)

〔日〕横井与次郎 著

唐长钧 徐子亮 译

*

煤炭工业出版社 出版

(北京安定门外和平门路16号)

煤炭工业出版社印刷厂 印刷

新华书店北京发行所 发行

*

开本850×1168¹/₃₂

印张16

字数 419 千字

印数1—12,800

1986年3月第1版

1986年3月第1次印刷

书号15085·2707 定价3.15元

JS332/21

译 者 的 话

本书译自日本无线电出版公司1980年出版的《微型计算机软件基础技术》一书。该书曾在日本三次出版发行,深受读者欢迎。作者是横井与次郎。全书共分十章。第一章介绍了进行微型计算机程序设计所必要的基础知识;第二章,第三章,第四章分别介绍了M6800系列,8080系列,Z-80系列的硬件基础知识,并详细地分析了这三种系列微型处理器的指令系统,寻址方式,指令的功能和动作,为编写汇编语言程序打下基础;第五章是微型计算机程序设计的开发方法,讨论了程序设计的步骤,汇编语言的功能和使用方法,并介绍了程序设计的工具—微型计算机的开发支援系统;第六,七,八章分别叙述了M6800系列,8080系列,Z-80系列汇编语言程序设计的基础知识,并通过实例说明汇编语言的使用方法;第九章讨论了宏指令及其使用方法,举例说明了宏汇编,伪指令和浮动汇编的使用方法;第十章讨论了文本编辑程序的功能及其使用方法。

故本书介绍了目前广泛使用的三种8位微处理器及其程序设计技术。熟读此书,基本上可以掌握M6800系列,8080系列,Z-80系列的汇编语言程序设计。

本书的特点是内容丰富,层次分明,通俗易懂。对初学者来说是一本较理想的入门书。

由于我们水平有限,译文中难免有这样或那样的缺点和错误,敬请读者批评指正。

参加本书翻译的有唐长钧(第二章,第五章,第六章,第九章,第十章),徐子亮(第一章,第三章,第四章,第七章,第八章)。

译者

1984年1月20日

作 者 的 话

本书是开发控制用微型计算机程序所必需的 软件技术入门书，内容包括微型计算机软件基础知识，各种微型计算机指令系统以及汇编程序语法基础和基本程序三部分。

众所周知，微型计算机可用于数据处理和控制。数据处理软件今后与大型计算机相同，主要是使用高级语言，几乎不使用汇编语言。

但是在设计控制用的程序时，由于微型计算机的输入输出基本上使用“二进制位”的位组合格式，所以编制程序时，即使使用高级语言，在编译时往往也列出汇编格式的列表清单，如果不会用汇编语言编写程序，至少也要能看懂。

此外，在程序调试时，有的地方与 I/O 有微妙的定时关系，有的地方要求进行位操作，所以往往需要分析汇编清单中的机器代码部分。

因此，虽不用机器代码编制程序，也必须充分了解所使用的微型计算机的指令系统和机器代码。

基于这样的理由，不管高级语言如何发展，既然控制微型计算机的用途是数字处理的计算机化，那末汇编语言和机器代码是绝对必要的。如果没有这些知识而要进行程序的研制和调试，不仅程序设计效率低，程序员的精神负担也重，编制出的程序问题也多。

因此对于至今尚无设计控制用计算机程序的经验，而又想掌握它的人们，最好采用下述的入门方法。

首先要了解微型计算机硬件和软件的一般基础知识。

本书的第 1 章和第 5 章就一般的微型计算机软件技术作了说明。

但本书没涉及数字技术中最基本的内容，例如二进制运算基础及与（AND）、或（OR）、非（NOT）三个基本逻辑的内容和使用，想要学习这些内容的读者，可阅读有关这方面的大量文献。

其次要了解实际使用的微型计算机的硬件基本知识。

为了理解指令系统，本书在第2章说明了MC6800, MC6802, MC6801, MC6809；在第3章说明了8080A和8085；在第4章说明了Z-80和Z-8微型计算机的硬件基础，同时还分别对指令系统的助忆码，动作及机器代码作了说明。

机器代码不须一个一个死记硬背，但必须了解根据助忆码查得的机器代码，或根据机器代码查得助忆码的方法。

最后要学习汇编程序的语法，实际编制程序，掌握汇编的方法，进行源程序的汇编，再将汇编成的目的程序装入目的机器，使其运行，并进行调试。

本书涉及到的微型计算机机种有M6800系列的MC6800/02, MC6801/03, MC6809；8080系列的8080和8085；Z-80系列的Z-80和Z-8。

这本书由于涉及了这么多的机种，故不可能对各种机种的指令系统都作详尽的分析，用汇编语言编制程序的实例也不多，几乎没涉及各机种程序设计技巧。但是，对主要机种的指令系统作了较充分的说明。

从事微型计算机业务的个人或企业，若能专门致力于某种机种加以研究是再好不过的了，但这往往是不可能的。

对专门研究一种微型计算机的人来说，若要掌握其他的机种，或要了解其他机种的软件概况，或要对几种机种相互比较，本书将是非常有用的。

经常听到想要学习微型计算机的人问起，在这么多的微型计算机中，选用哪一种较好呢？

要对各种各样微型计算机硬件和软件的概况和特征加以说明是不成问题的，但对个人或企业来说，哪种机种较为合适，这在

不了解使用微型计算机的条件和立场之前，无论如何是不能作出判断的。

因此，本书只能从第三者的立场出发，说明各种微型计算机的概况，至于选用哪种机种，则由使用者分析自己的条件和立场，自己作出决定。

* * *

与本书内容有关的工业所有权问题，因对其调查需要用很长时间，也很费手续，本人不十分清楚。如果有什么疑问，敬请读者自己调查。

本书的出版受到无线电技术出版社铃木勇治总编辑和菅井彰吾的帮助，在此表示感谢。

在引用本书内容时请注明作者、书名、出版社及页数。

横井与次郎

1979年11月

目 录

第1章 微型计算机程序设计所需要的基础知识

1.1 微型计算机硬件和软件的关系	1
1.2 理解微型计算机软件所需的硬件基础知识	3
1.2.1 微型计算机的基本结构	3
1.2.2 微型计算机CPU的结构	5
1.2.3 微型计算机中的存贮器	6
1.2.4 微型计算机的字长和IC工艺结构	11
1.2.5 总线系统	12
1.3 计算机中数据的表示方法	13
1.3.1 记数法	13
1.3.2 微型计算机中负数的表示法	15
1.3.3 ASCII码	20
1.4 程序组成的要素和流程图	22
1.4.1 程序设计的三个要素	22
1.4.2 流程图的符号和形式	25
1.5 理解指令系统的基础知识	30
1.5.1 微型计算机指令字的组成	30
1.5.2 指令字的寻址方式	31
1.5.3 操作数的寻址方式	31
1.5.4 跳越转移, 分支转移, 跳步, 返回的操作和子程序	37
1.5.5 微型计算机中指令的分类和概述	41
1.6 微型计算机硬件操作的基础知识	46
1.6.1 微型计算机的基本时序	46
1.6.2 CPU的外部控制种类	48
1.6.3 复位, 再启动	48
1.6.4 暂停, 保持, 停机, 准备好	48

1.6.5 中断.....	48
---------------	----

第2章 M6800系列微处理器的指令系统

2.1 理解指令系统所需的MC6800/MC6802/MC6801的 硬件结构知识.....	50
2.1.1 MC6800的硬件结构.....	50
(1) CPU内部结构和内部寄存器.....	50
(2) 中断功能.....	53
2.1.2 MC6802的硬件结构.....	55
2.1.3 MC6801/MC68701/MC6803的硬件结构.....	55
2.2 MC6800/MC6802的指令系统.....	60
2.3 MC6800/6802的指令寻址方式.....	62
2.3.1 MC6800/MC6802指令寻址方式的内容.....	62
2.3.2 MC6800/6802中有效地址的计算方法.....	64
2.4 MC6800/6802的指令概述.....	68
2.5 MC6800/6802指令的分类.....	75
2.6 MC6800/6802的指令分析和动作说明.....	76
2.6.1 数据的传送和移动.....	76
2.6.2 程序运行的控制.....	81
2.6.3 数值运算和逻辑运算.....	92
2.7 MC6801/MC6803的指令系统.....	98
2.8 MC6801/6803指令概况.....	100
2.9 MC6801/6803的指令分析和动作说明.....	108
2.9.1 数据传送和移动.....	108
2.9.2 程序运行的控制.....	111
2.9.3 数值运算和逻辑运算.....	111
2.10 理解指令系统所需的MC6809/6809E的硬件结构知识.....	115
2.10.1 CPU内部结构和CPU内部寄存器.....	115
2.10.2 中断功能.....	129
2.11 MC6809/09E的指令系统.....	130
2.12 MC6809/09E指令概况.....	131
2.13 MC6809/6809E的寻址方式.....	137
2.13.1 MC6809/09E寻址方式的内容.....	137

2.13.2 MC6809/09E中有效地址的计算方法	153
2.14 MC6809/09E的指令分类	159
2.15 MC6809/6809E的指令分析和动作说明	161
2.15.1 数据的传送和移动	161
2.15.2 程序运行的控制	166
2.15.3 数值运算和逻辑运算	169

第3章 8080系列微处理器的指令系统

3.1 理解指令系统所需的8080A和8085的硬件知识	175
3.1.1 8080A的硬件组成	175
(1) CPU的内部组成和CPU内部寄存器	175
(2) 中断功能	179
(3) 与I/O通道口的关系	181
3.1.2 8085的硬件结构	183
(1) CPU内部结构和CPU内部寄存器	183
(2) 中断功能	183
3.2 8080A的指令系统	186
3.3 8080A的寻址方式	190
3.4 8080A指令概述	193
3.5 8080A的指令分析和动作说明	203
3.5.1 8080A的指令分类	203
3.5.2 数据传送和移动指令	203
3.5.3 程序执行的控制指令	215
3.5.4 数值运算和逻辑运算指令	225
3.6 8085的指令系统	230
3.7 8085指令操作说明	235

第4章 Z-80系列微处理器的指令系统

4.1 理解指令系统所需的Z-80 硬件结构知识	239
4.1.1 CPU的内部结构和CPU内部寄存器	239
4.1.2 中断功能	248
4.2 Z-80的指令系统	251

4.3	Z-80指令的寻址方式	262
4.3.1	Z-80指令寻址方式的内容	262
4.3.2	Z-80有效地址的计算方法	266
4.4	Z-80的指令概况	267
4.5	Z-80指令的分类	281
4.6	Z-80的指令分析和动作说明	281
4.6.1	数据传送和移动指令	281
4.6.2	控制程序执行指令	293
4.6.3	数值运算和逻辑运算指令	296
4.7	理解Z8指令系统所需的硬件结构知识	303
4.7.1	CPU内部结构和CPU内部寄存器	303
4.7.2	内部ROM和外部存贮器的地址	311
4.7.3	Z8的中断功能	312
4.8	Z8的指令系统	315
4.9	Z8指令的寻址方式	315
4.9.1	Z8指令寻址方式的内容	315
4.9.2	Z8的有效地址计算方法	324
4.10	Z8指令概述	330
4.11	Z8指令的分类	330
4.12	Z8指令的分析和动作说明	330
4.12.1	数据的传送和移动	330
4.12.2	程序执行的控制	334
4.12.3	数值运算和逻辑运算	336

第5章 微型计算机程序的开发方法

5.1	微型计算机程序编制的顺序	339
5.2	微型计算机使用程序设计语言的种类及其内容	341
5.3	汇编程序的功能及其使用方法	345
5.3.1	汇编程序的功能	345
5.3.2	汇编程序的使用方法	348
5.4	微型计算机开发支援系统概况	353

第6章 M6800系列的汇编程序基础

6.1	M6800系列的汇编程序语法	358
-----	----------------------	-----

6.1.1	汇编程序语法的不同	355
6.1.2	源程序的形式	358
6.1.3	字段的格式	364
6.2	伪指令的功能及其使用方法	371
6.3	M6800程序举例	383
6.3.1	成组传送程序	383
6.3.2	1字节乘法程序	393
6.3.3	2字节乘法程序	395
6.4	MC6801的汇编程序	397
6.5	MC6801程序举例	398
6.5.1	成组传送程序	398
6.5.2	2字节乘法程序	400
6.6	MC6809的汇编程序	405
6.6.1	MC6809与MC6800汇编程序在语法上的主要不同	405
6.7	MC6809程序举例	406
6.7.1	成组传送程序	406
6.7.2	2字节乘法程序	413
6.8	汇编清单的错误信息	416

第7章 8080系列的汇编程序基础

7.1	8080系列的汇编语法	418
7.2	伪指令	420
7.3	8080程序设计实例	428
7.3.1	成组传送程序	428
7.3.2	1字节乘法程序	433
7.3.3	2字节乘法程序	436
7.4	8085汇编程序的语法	440
7.5	8085程序设计实例	440
7.6	编译程序和汇编程序	445

第8章 Z-80系列汇编程序基础

8.1	Z-80汇编程序的语法	448
-----	-------------	-----

8.1.1	Z-80汇编程序语法和助忆码	448
8.1.2	TDL Z-80浮动汇编程序的助忆码	448
8.1.3	TDL Z-80浮动汇编程序的伪指令	455
8.2	Z-80程序设计实例	456
8.2.1	成组传送程序	456
8.2.2	1字节乘法程序	458
8.2.3	2字节乘法程序	460
8.3	Z8的汇编程序的语法	464
8.4	Z8的程序实例	465

第9章 宏指令的语法及其使用方法

9.1	宏功能及其语法	469
9.2	使用PDA-80宏汇编程序的程序实例	471
9.2.1	使用8080汇编程序中伪指令的程序实例	471
9.2.2	成组传送程序	474
9.3	使用TDL Z-80浮动汇编程序的程序实例	481
9.3.1	成组传送程序	481
9.3.2	MC6809 TFR, EXG指令的交叉汇编	484

第10章 文本编辑功能和使用方法

10.1	文本编辑的功能	489
10.2	行号编辑程序举例	490
10.3	指示器编辑程序举例	494

第 1 章

微型计算机程序设计所需要的基础知识

1.1 微型计算机硬件和软件的关系

所有的计算机，包括微型计算机在内，都由硬件（Hardware）和软件（Software）两大部分组成，因此使用微型计算机所需要的知识也可分为硬件技术和软件技术两部分，如图1.1所示。

硬件，顾名思义指的是设备，即指构成计算机的集成电路 IC、存储器、将它们相互连接起来的插件板（印刷电路板），接插件之间的连线，以及将上述部分装成一体的机架等。

软件是以程序为中心的信息（Information）组合。由该信息组合来实现硬件所给定的功能。

但是，软件不光是指作为处理手段的程序，还有怎样实现必要功能的思想体系，即算法（Algorithm）。

微型计算机的软件技术，毫无疑问要包括所有计算机共同的程序设计技术，此外还必需包括一般的数字技术。

所谓一般的数字技术就是与数字电路中所必要的门和触发器、计数器、移位寄存器、译码器、多路转换器、奇偶校验/发生器等功能电路有关的知识。

由于计算机的控制器本质上是分时控制的，所以时序技术也是必需的。

另外，和程序设计同样重要的是修改已编写好程序中的错误，即程序调试（纠错：错误的检出，订正）技术。

程序在编制阶段错误是难免的，问题是如何迅速地找出错误的位置和原因，并加以订正，这直接影响到程序设计的效率。一般说来，程序的调试时间接近或者超过程序的编写时间。

设计数字装置时，不管是布线逻辑，还是计算机方式，如图1.2所示，都要从系统设计开始，这一点是相同的，但对于布线

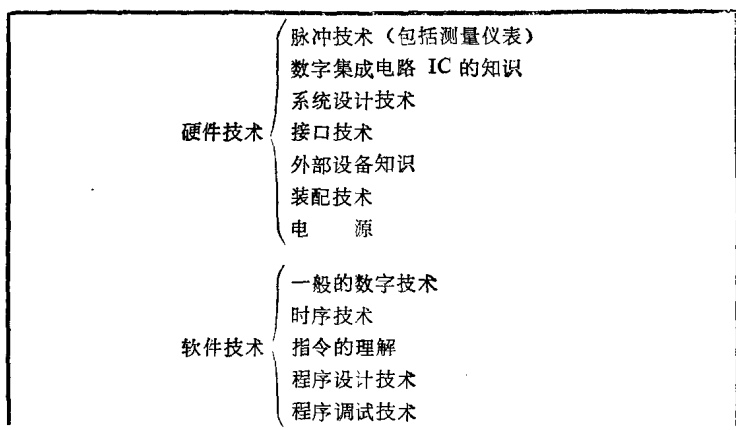


图 1.1 使用微型计算机所必要的知识

逻辑来说, 电路设计一旦完成, 该系统的功能就被确定了, 而对计算机来说, 则在体系结构设计完, 并确定电路结构之后, 再设计程序, 然后把它存入存储器, 这时功能才被确定。

因此, 不管硬件如何好, 如果程序设计不好, 计算机的性能仍不能满足要求。所以程序设计对计算机来说是至关重要的。然而程序也不是决定一切的, 体系结构的不合理, 若用软件来弥补, 往往会产生许多问题, 故软件和硬件都是十分重要的。

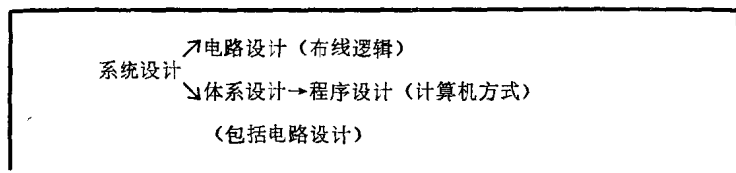


图 1.2 布线逻辑和计算机设计方法的不同

在计算机系统中, 各种功能主要是由硬件来实现还是由软件来实现, 这是一个重要的问题。此外, 处理速度, 经济性, 可靠性, 校验和维修容易, 变动方便等都应充分加以研究再予决定。

1.2 理解微型计算机软件所需的硬件基础知识

1.2.1 微型计算机的基本结构

如1.1节所述,微型计算机的硬件和软件是紧密相关的,为了理解软件,必须先理解硬件。

为此,有必要了解微型计算机的基本结构,CPU的内部结构,存贮器的种类和组成等。

当然,在实际编写程序时,还有必要了解各种微型计算机的具体特点,这些将在第2章,第3章,第4章指令系统说明之前进行介绍。

本章主要说明理解以后各章所必需的硬件基础知识。

图1.3是计算机的基本体系结构(Architecture),由存贮器(Memory),CPU(Central Processing Unit)和输入输出器件(Input/output Device)组成。

在这些方框中,CPU可以说是代表计算机的特征的,决定计算机和布线逻辑区别的就是存贮器。

利用计算机进行逻辑处理时,其基本组成如图1.4所示。图中左面虚线部分与图1.3完全相同,是计算机的基本部件,此外就是与布线逻辑相同的逻辑部件和输入输出(I/O)接口。

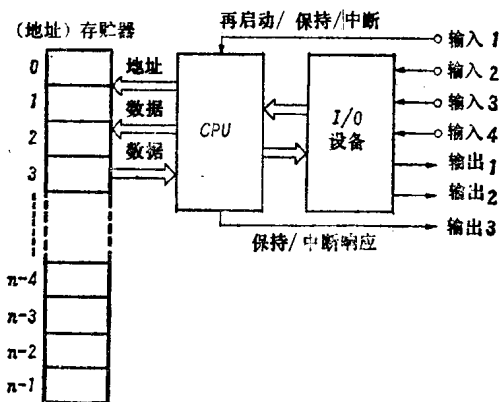


图 1.3 计算机的基本结构

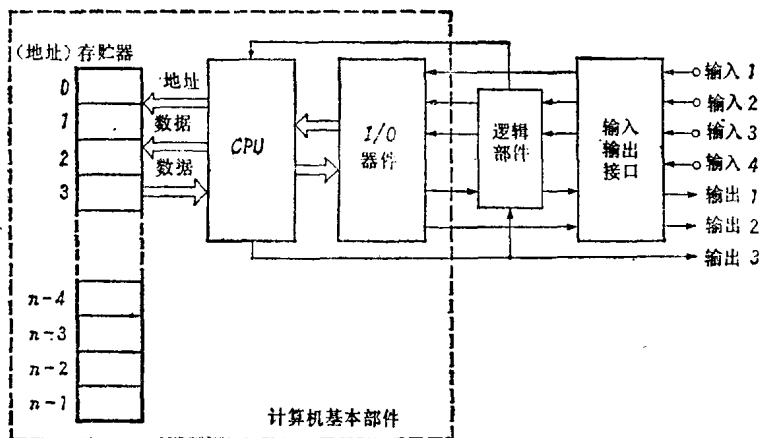


图 1.4 由计算机进行逻辑处理的基本结构

这里，为了说明原理，把逻辑部件单独画出，而通常，其功能包含在I/O器件或I/O接口中。当然，该逻辑部件是辅助性的，整体的大部分功能都由计算机部件承担。

冯·诺依曼 (J.von Neumann) 提出的存贮程序方式的计算机，是把各种输入的处理方法以程序指令的集合形式存放在存贮器中，再按顺序将其读出译码，并执行。

这样，用计算机进行逻辑处理是由程序来实现的，相对于布线逻辑，则称之为程序处理方式。

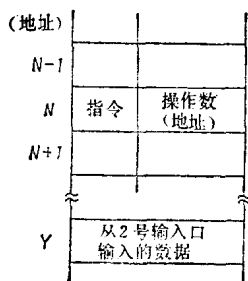


图 1.5 存贮器中的指令和数据

计算机如何处理加在各输入端的数据或数字信息，取决于处理数据所存放的存贮器地址或输入通道口号码的指令(Instruction)内容。

例如，如图1.5所示，从2号输入口来的数据（不仅是数值也可以是逻辑信息）存放在地址为Y的单元里，如果把地址Y作为操作数（指令的处理对象）的指令存放在地址为