

微型计算机程序设计

〔日〕石田晴久 编

史嘉权 戴梅萼 译 吕景瑜 校

国防工业出版社

微型计算机程序设计

〔日〕 石田晴久 编

史嘉权 戴梅萼 译

吕景瑜 校

国防工业出版社

内 容 简 介

本书对微型计算机的程序设计作了系统而全面的阐述。全书共分十五章。作为程序设计的基础知识,头两章讲解了微型计算机的结构和外部设备。全书的核心内容由三部分组成:第一部分是微型计算机系统程序的设计,其中包括监控程序、编辑程序、汇编程序、Tiny BASIC 解释程序、微型 Plan 语言的处理程序、交叉汇编程序、模拟程序和交叉编译程序的设计;第二部分是微型计算机应用程序的设计,其中包括浮点运算程序、随机数生成程序、计算机音乐程序、宇宙战争游戏程序、莫尔斯码通信程序的设计;第三部分用专门的章节介绍了微型计算机程序设计的技巧。

本书内容充实、丰富,不但对基本原理进行了通俗的讲解,而且,对各主要程序作了深入的分析解剖。特别应该提到的是书中还给出了上述程序的完整而详细的注释清单,这无疑是微型计算机程序设计人员极感兴趣的内容。

本书主要供从事微型计算机程序设计的科技人员参考,也可作为大专院校计算机专业师生的补充读物。

マイクロコンピュータのプログラミング

石田晴久 编

昭和53年初版1刷発行

共立出版株式会社

昭和56年初版11刷発行

*

微型计算机程序设计

〔日〕 石田晴久 编

史嘉权 戴梅尊 译

吕景瑜 校

*

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24 554千字

第一版 1986年9月第一次印刷 印数: 0,001—6,800册

统一书号: 15034·3076 定价: 4.90元

译者的话

近年来,计算机获得了迅速的发展,微型计算机由于其体积小、价格便宜、使用方便,而引起了人们的广泛兴趣。在面向未来、面向世界、面向现代化的今天,已经掀起了一个“微型计算机热”。许多科技人员、大专院校师生迫切希望得到一本系统而具体地讲述微型计算机程序设计技巧和方法的书,所以,译者真诚地向广大读者推荐石田晴久等著的这一著作。

石田晴久是日本微型计算机方面的知名教授,近年来发表了不少著作。他所主笔的著作有这样一个特点:既有相当的深度和广度,又带有浓厚的趣味性。他在书中总是能够抓住读者最感兴趣、最想解决的问题作深入浅出而又具体的说明。

本书的宗旨在于讲解微型计算机程序设计的方法、原理和技巧,为此,书中安排了十分充实的内容。书中介绍了程序设计人员普遍感兴趣的监控程序、编辑程序、汇编程序和交叉软件的设计思想及其精巧的程序设计技术。所介绍的Tiny BASIC解释程序虽然只有2K字节,却包含了高级语言处理程序的基本内容。可谓麻雀虽小,五脏俱全。加上书中给出了详尽的注释,有心的读者仔细看后便能举一反三。后面章节所述的Plan语言处理程序采用了通过中间代码进行二次变换的办法,而中间代码作为解决程序可移植性的途径之一已引起计算机科技人员很大的重视。对于从事或有志于从事程序可移植性课题的研究人员,这部分内容无疑具有重要的参考价值。微型计算机上的浮点运算程序和随机数生成程序也是许多程序设计人员普遍感兴趣的,但以往详细介绍这两个程序的总体结构并给出注释清单的文献却极少,本书在这方面显然提供了有价值的参考资料。此外,莫尔斯码通信程序、音乐程序、根据星球大战的故事而编制的宇宙战争游戏程序和女皇程序等则富于趣味而给人以有益的启迪。

本书的最大特点是实用性强。对于基本的系统程序和应用程序,书中不但具体地阐述了程序的设计思想,并且给出了详细的注释清单,这正是一般书籍和刊物很少提供的内容,而这对于每一位想深入理解和真正掌握微型计算机程序设计方法和技巧,或者分析、移植、设计微型计算机系统程序和应用程序的读者来说都是极为有益的。本书在日本连续印刷十一次,仍然畅销,恐怕就在于上面所述的原因吧!

由于译者水平有限、时间仓促,缺点和错误在所难免,敬请读者批评指正。

序

近年来,随着微型计算机越来越广泛地使用,其软件的重要性正逐渐被人们所认识。但是,目前在微型计算机方面广为流传的资料仍偏重于硬件。统观书籍杂志,也以介绍元件功能和硬件设计的居多,而有关软件方面的资料甚少。因此,编写本书的目的是希望有益于微型计算机软件的开发、学习和研究。

本书的特点

本书的主要特点是直截了当地给出了一些用汇编语言编写的微型计算机基本程序的注释清单。为了使这些程序的叙述具有一致性,书中涉及的微型计算机系统,都属于如下规格的最标准的系统:

指令系统 8080、8085 和 Z80 (Z80 的部分指令)

汇编语言 Intel 的标准宏汇编格式

目的代码 Intel 的标准装配程序格式

但是,本书所阐述的程序设计思想,对于 6800 等其它微型计算机基本上也是适用的。

本书的构成

本书共分五篇。

第一篇介绍了研究微型计算机软件所需的最低限度的指令系统和硬件。在指令系统部分,还介绍了一般手册中很少提到的一些重要内容。在硬件部分,着重从用户的角度介绍了比较重要的外部设备和接口。

第二篇讲解了作为基本软件的监控程序、编辑程序、汇编程序、BASIC 解释程序以及 Plan 语言的处理程序。在阐述这些程序的设计思想的同时,还给出了完整的源程序列表文件并作了注释。值得注意的是,其中的监控程序、编辑程序、汇编程序都具有独特的技巧,从而使程序非常简练。另外, Tiny BASIC 解释程序虽然已出现了东京大学的版本,但本书仍按最初的版本作了详细阐述。后面的 Plan 语言是纯粹日本设计的计算机语言。倘若能以本书为开端,不断开发 Plan 语言以外的新的程序设计语言及其处理程序,则是值得庆幸的。

第三篇通过具体的源程序阐述了程序设计的技巧和方法。近年来,在使用大型计算机的场合,更多地强调程序的可读性,即使浪费存储器也在所不惜。但是,对于存储容量受到较大限制的微型计算机来说,存储器的浪费是不允许的。而要编写简单明了的程序,就要掌握符合结构程序设计理论的技巧和方法。

第四篇讲述有关交叉汇编程序、模拟程序、交叉编译程序的设计思想和使用方法。在小型计算机或中、大型计算机上,特别是在分时系统(TSS)上,交叉软件会成为开

发软件的有力武器。本篇还涉及到PL/M语言的应用，而PL/M语言今后将成为重要的正规计算机语言。

第五篇介绍了几个读者感兴趣的应用软件。其中，音乐程序和莫尔斯码通信程序是用汇编语言编写的，而宇宙战争游戏程序则是用BASIC语言编写的。

附录为最小的BASIC标准草案。除了解释程序只有2K字节的Tiny BASIC以外，在设计更正规的BASIC系统时，至少应该包含这个草案的全部功能。

程序的版权

最后，再谈几句关于程序的版权问题。本书中公开发表的程序，是个人或者大学、研究所等事业单位使用后确定的。编者希望，在其它地方使用或引用时，应该充分尊重设计者（和出版社）的版权（及编辑版权）。本书的发表实际上是把软件无偿地公开了，对这种做法，有人认为是以软件开发为生的软件公司或系统公司的一个打击。其实，与这种议论恰恰相反，这是我们提倡学习风气的一种尝试，并不会助长轻视软件价格的风潮。而通过这种尝试，可使广大读者充分了解到软件开发是何等重要；对于用一句话就可说明的问题（例如浮点运算），要用程序实现起来，又要花费多少功夫！所以，希望广大读者对软件的价格重新作出正确的评价。

最后，书中除美国软件专业杂志Dr. Dobb's Journal发表过的一些程序外，其他程序均由东京大学的小野芳彦先生汇总，小野芳彦先生还对源程序列表文件进行了统一和整理工作。在此，各位执笔者向他再一次表示感谢。

石田晴久

1978年1月

目 录

第一篇 微型计算机基础

第一章 微型计算机的结构	1
1.1 前言	1
1.2 8080/8085微型计算机的结构	2
1.3 8080/8085指令系统	4
1.3.1 数据传送类指令	4
1.3.2 8位和16位加法指令	8
1.3.3 减法指令	9
1.3.4 逻辑运算指令	11
1.3.5 移位指令	11
1.3.6 其他指令	12
1.3.7 寄存器的增量和减量指令	12
1.3.8 转移指令	12
1.3.9 子程序调用指令	12
1.3.10 返回指令	13
1.3.11 有关堆栈的指令	13
1.3.12 输入输出指令	13
1.3.13 中断控制指令	13
1.4 指令的长度和执行时间	13
1.5 中断处理	14
1.6 8080的软件	15
第二章 微型计算机的外部设备	17
2.1 外部设备和接口的基础知识	17
2.2 8080系统的输入输出方式	18
2.3 程序控制的输入/输出的设计思想	19
2.4 DMA的结构	20
2.5 ASR33电传打字机的连接	21
2.6 电视电传打字机的应用	23
2.7 廉价的纸带输入机	24
2.8 其他外部设备	25

第二篇 基本的系统程序

第三章 监控程序	27
3.1 调试的步骤	27
3.2 外部规格和内部规格	29
3.3 用于SDK-80的监控程序(第3版)注释清单	34
第四章 微型编辑程序	49
4.1 概述	49

4.2 文件的内部结构	563
4.3 程序概述	57
4.4 微型编辑程序注释清单	58
第五章 微型计算机的汇编程序	61
5.1 软件的研制	61
5.2 汇编语言的语法规则	62
5.2.1 字符	62
5.2.2 语句	62
5.2.3 语句的结构	62
5.2.4 常数	65
5.2.5 操作数的表示方法	65
5.2.6 指令	66
5.2.7 伪指令	66
5.2.8 注意事项	67
5.3 汇编程序的使用方法	67
5.4 汇编程序的分析	68
5.4.1 标号表	70
5.4.2 指令类型和指令表	70
5.4.3 操作数的分析	72
5.5 程序说明	73
5.6 微型自汇编程序注释清单	75
第六章 Tiny BASIC 解释程序	102
6.1 Tiny BASIC 概述	102
6.1.1 Tiny BASIC	102
6.1.2 程序举例	102
6.2 Tiny BASIC 系统概述	104
6.2.1 Tiny BASIC系统的结构	104
6.2.2 Tiny BASIC系统的硬件结构	104
6.2.3 系统程序的输入	104
6.3 Tiny BASIC 解释程序注释清单	108
第七章 微型Plan语言的处理程序	140
7.1 系统的特点	140
7.2 程序执行的次序 (用于8080)	140
7.3 微型 Plan 语言的语法	142
7.4 Q 码	145
7.4.1 数据区	145
7.4.2 代码区	146
7.4.3 寄存器组	146
7.5 R 码	152
7.6 编译程序、装配程序、解释程序	154
7.7 微型 Plan 语言编译程序注释清单	157
7.8 微型Plan/8080 R码装配程序和 Q 码解释程序注释清单	168
7.9 编译程序名称表初始化用的纸带	185

第三篇 程序设计技巧

第八章 程序设计技巧	186
------------------	-----

8.1 i8080 指令的特点	186
8.2 设计小型的程序	187
8.2.1 巧妙地使用寄存器	187
8.2.2 充分利用堆栈	188
8.2.3 尽可能减少无条件转移指令	190
8.2.4 有效地使用CALL指令	192
8.2.5 递归调用	194
8.2.6 在指令中编指令	195
8.3 程序的高速化	197
8.3.1 n 个女皇的问题	198
8.3.2 提高执行速度的技巧	199
8.4 其他方法	206
8.5 结束语	207
第九章 浮点运算程序	208
9.1 只有整数位的情况	208
9.2 浮点数的表示形式	208
9.3 二进制形式的浮点数	209
9.4 浮点运算的流程	210
9.4.1 乘法运算	210
9.4.2 除法运算	210
9.4.3 加减法运算	211
9.5 二进制浮点运算	211
9.5.1 数据的表示形式	211
9.5.2 运算程序的功能	212
9.5.3 输入/输出设备和内存环境	214
9.5.4 寄存器的保护	214
9.6 浮点运算程序包的说明	214
9.6.1 运算程序	214
9.6.2 输入/输出程序	217
9.7 浮点运算程序	219
9.8 结束语	221
9.9 浮点运算程序注释清单	221
第十章 随机数生成程序	249
10.1 什么叫随机数	249
10.2 计算机能产生“随机数”吗	249
10.3 什么是随机性	250
10.4 随机数生成程序实例	250
10.5 产生随机数的方法	251
10.6 随机性的验证	252
10.6.1 统计性的假设检验	252
10.6.2 χ^2 检验	253
10.6.3 无规则性的检验——检验统计性好吗	255
10.7 产生符合其他分布的随机数	256
10.8 产生各种类型的均匀随机数	257
10.8.1 程序运行的环境	257
10.8.2 8 位随机数子程序 RND8 (图10.5)	257

10.8.3 产生8位以下任意范围的随机数子程序RND8X	257
10.8.4 产生16位随机数的子程序RND16(图10.11)	258
10.8.5 RND16的应用之一——RNDX子程序(图10.17)	261
10.8.6 RND16的应用之二——RND子程序	261
10.9 后记	261
10.10 随机数生成程序注释清单	262

第四篇 交叉软件

第十一章 交叉汇编程序 模拟程序 交叉编译程序	268
11.1 交叉汇编程序	269
11.1.1 汇编语言	269
11.1.2 汇编语言的语法	269
11.1.3 汇编方式和数据结构	272
11.1.4 交叉汇编程序的使用	273
11.2 模拟程序	282
11.2.1 模拟程序的原理	282
11.2.2 模拟程序的实际内容	285
11.2.3 调试程序	290
11.2.4 模拟程序的使用	291
11.3 交叉编译程序	295
11.3.1 交叉编译程序的现状	295
11.3.2 交叉编译程序的特点	297
11.3.3 PL/M的使用	298
11.4 宏指令	305

第五篇 应用程序

第十二章 计算机音乐程序	307
12.1 计算机和音乐	307
12.2 自动演奏的技巧在于对时间的程序控制	307
12.3 电子音乐的基础	307
12.4 TK-80中的电子发音盒	308
12.5 CPU产生音阶的精度限制	312
12.6 乐谱输入方式举例	313
12.7 合成器概述	314
12.8 TK-80的改造和软件	316
12.9 音乐自动演奏程序注释清单	318
第十三章 宇宙战争游戏程序	326
13.1 深受欢迎的星球大战游戏	326
13.2 星球大战游戏程序的设计思想	326
13.3 微型星球大战游戏的例子	328
13.4 下一个是日本的宇宙战舰游戏吗	329
13.5 星球大战游戏程序清单	330
第十四章 莫尔斯码通信程序	333
14.1 莫尔斯码通信和微型计算机	333

14.2	莫尔斯码与数据结构	333
14.3	莫尔斯码发送程序的设计	336
14.4	适用于SDK-80 的莫尔斯码发送程序	340
14.5	莫尔斯码发送程序注释清单	341
第十五章 最小 BASIC 标准草案		351
15.1	适用范围	352
15.2	参考标准	352
15.3	术语	352
15.4	字符和字符串	354
15.5	程序	355
15.6	常数	356
15.7	变量	358
15.8	表达式	359
15.9	内部函数	360
15.10	自定义函数	361
15.11	LET 语句	362
15.12	控制语句	362
15.13	FOR 语句和 NEXT 语句	364
15.14	PRINT 语句	365
15.15	INPUT 语句	367
15.16	READ 语句和 RESTORE 语句	369
15.17	DATA 语句	369
15.18	数组说明	370
15.19	REM 语句	371
15.20	RANDOMIZE 语句	371
附录		372
附录 1	标准的组成	372
附录 2	语法的描述方法	372
附录 3	所谓符合标准	373
附录 4	由设计者规定的功能	374

第一篇 微型计算机基础

第一章 微型计算机的结构

1.1 前言

微型计算机有 4 位并行处理、8 位并行处理、12 位并行处理以及 16 位并行处理等多种类型。其中，最典型的要算 8 位并行处理的微型计算机，这种类型的机器每次刚好处理一个字节的数据（数或文字）。而 8 位类型中，又有各种各样的机型，如 Intel、日本电气、三菱、东芝的 8080、8085，Zilog、夏普、日电的 Z-80，Motorola、日立、富士通的 6800，Mostek 的 6502 等等。

本书着重介绍 8 位机中有代表性的机种 8080 和可与 8080 使用共同程序（软件上向上兼容）的 8085。由于 8080 指令系统后来成了 Zilog 公司研制的 Z-80 指令系统的子集，所以，本书中阐述的程序及其设计思想也适用于 Z-80 系统。

Intel 公司的 8080 是在原有的 8008 的基础上，将其功能加以扩充而设计的，因此，其指令系统并不很完善。但是，8080 多少也能做些 16 位的算术运算等操作，因此，功能也算相当强了。由于 8080 是最早出现的正规的单片 CPU（微处理器），所以，获得了广泛的应用。

8085 与 8080 相比，除新增加的两条指令外，其他指令和 8080 完全相同。但是，8085 是高集成度的芯片，它包含了时钟发生器 8224 和系统控制器 8228 这两个芯片的功能，而 8224 和 8228 这两个芯片是 8080 系列微型计算机必不可少的。此外，8080 需要 +5V、-5V 和 +12V 三种电源，而 8085 只需要 +5V 一种电源就可以了，8085 的外围芯片（如存储器等）也只需要 +5V 一种电源。不过，因为 8085 地址的低 8 位和 8 位数据使用同一数据总线，所以，当要求在外面把地址的低位和数据分开时，会稍微麻烦些。但是，总的来说，现在想要使用 8080 系列微型计算机的用户选用 8085 还是明智的。

后来，在使用 8080 的过程中感到功能不够，于是，又在 8080 的 78 条指令的基础上增加了 80 条指令，从而研制出具有 158 条指令的 Z-80。但是，反过来，由于 Z-80 指令过多，要自如地使用它们也很不容易。不过，在充分利用成组检索和成组传送等指令的场合，用 Z-80 是很有利的。本书中的程序可以原封不动地用到 Z-80 上，不言而喻，在这种情况下，并不能充分发挥 Z-80 的功能。当然，如果把部分程序改为用 Z-80 的指令来编写，用起来会更好些。

对于不同的 CPU 来说，程序设计的原理并没有显著的差别，所以，即使用 8080 以外的 CPU，例如使用 6800 CPU，那么，本书中的程序也是十分有价值的。特别是，编排具体程序以前的程序设计步骤以及程序设计思想，对于任何计算机都大致相同。

1.2 8080/8085微型计算机的结构

Intel公司的SDK-85单板机可作为用8085构成微型计算机的一个例子，其结构如图1.1所示。因为在8085中，地址的低8位和数据使用共同的总线（总线用于传递信息），所以，一般需要把它们分离开。但是，如果使用象8155和8755这样的专用存储器芯片，由于分离工作可在这些芯片内进行，因而就不必增加额外的电路了。另外，在图中，键盘和显示器的控制采用了型号为8279的大规模集成电路（LSI）芯片。就象SDK-85那样，

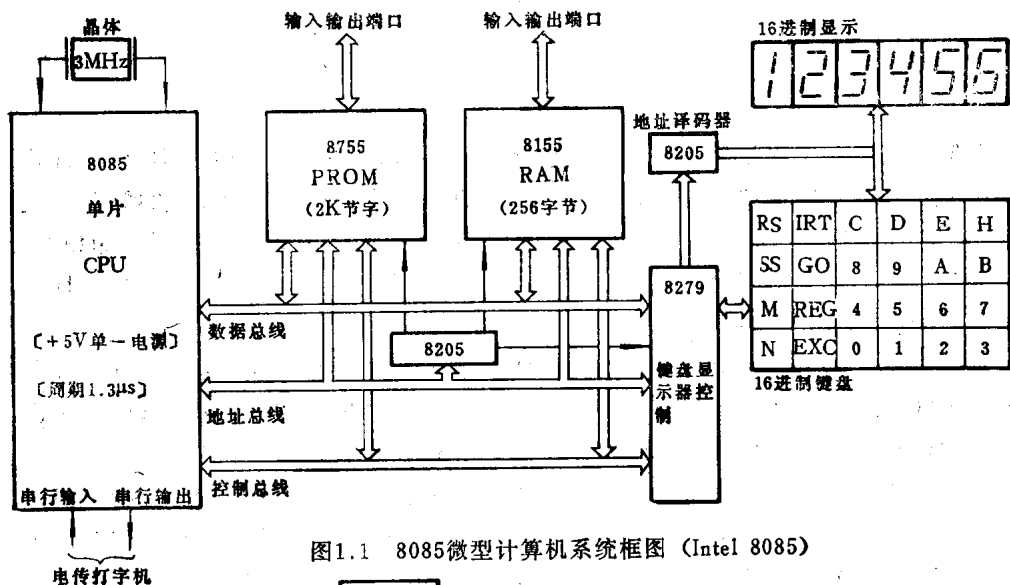


图1.1 8085微型计算机系统框图 (Intel 8085)

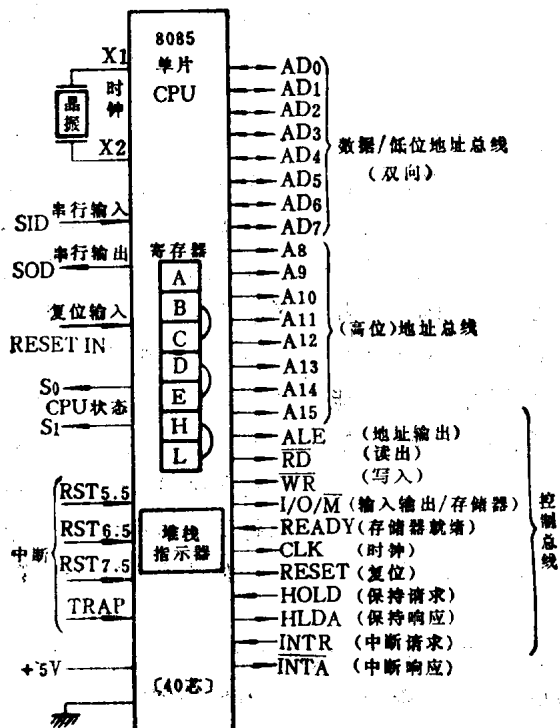


图1.2 8085CPU引线信号

当CPU用高集成度的8085时，外围电路往往也采用大规模集成电路。从硬件生产的角度看，外围电路并不比CPU更复杂，而麻烦的还是软件。

图1.2示出8085 CPU的40个信号。特别值得注意的是，它具有串行输入/输出端。假如用程序实现串行往并行和并行往串行的转换以及对速率的控制，就可以把电传打字机、CRT显示器、盒式磁带记录器等接到串行输入/输出端上。控制总线信号中有一个ALE(Address Latch Enable——启动地址锁存器)信号，其含义如下：

- { 为1时，AD0~AD7上的信息为地址。
- { 为0时，AD0~AD7上的信息为数据。

HOLD和HLDA信号在存储器直接存取(DMA)方式下使用。所谓DMA是指不通过CPU，而在存储器和外部设备之间直接进行数据传送。用DMA方式传送数据时，由外部设备发出保持(HOLD)请求信号，以便使存储器和CPU之间的连线断开(电阻看作无穷大)，当收到由CPU发出的保持响应(HLDA-Hold Acknowledge)信号之后，便进入DMA方式。关于8085的中断问题，以后再介绍。

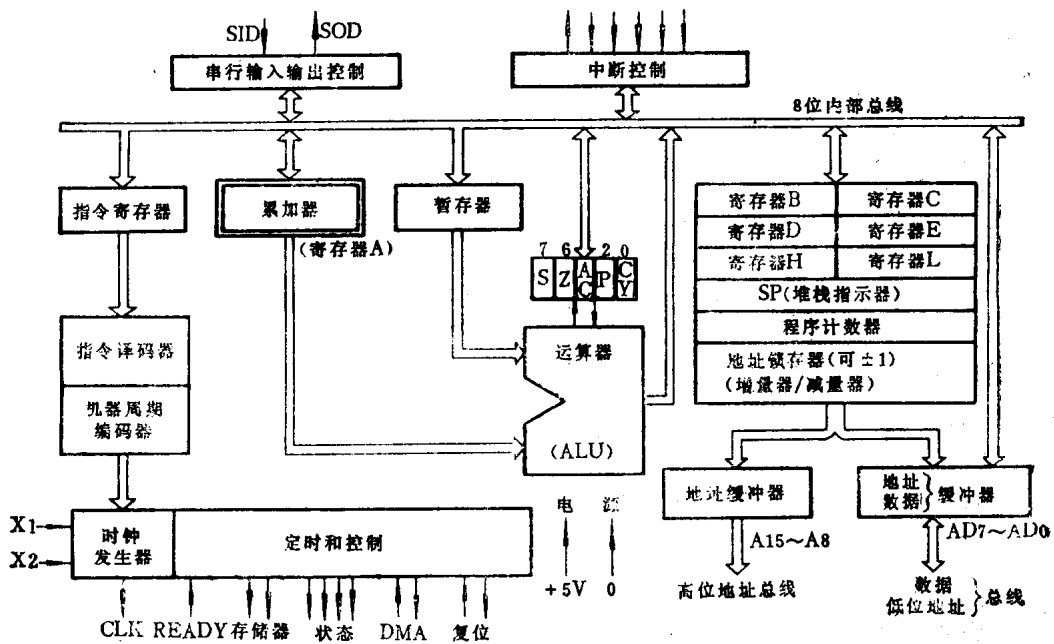


图1.3 8085 CPU的内部结构

图1.3是编写程序时需要了解的8085 CPU(和8080在本质上是同样的)的内部结构。通过这个图，可把8080/8085结构上的特点归纳如下：

(1) 8位的累加器(即寄存器A)是运算的中心，8位运算的结果总是保存在累加器中。

(2) 8位通用寄存器有B、C、D、E、H、L。

(3) 寄存器运算时，先把寄存器的内容送到暂存器，然后在运算器(ALU)中和累加器的内容进行运算，再根据运算的结果设置如下的标志(每个标志占寄存器的一位)：

S = 1	负 (D7 = 1)	S = 0	正或零 (D7 = 0)
Z = 1	零 (D7 = D6 = ... = D0 = 0)	Z = 0	非零
AC = 1	低四位 ≥ 1010	AC = 0	低四位 ≤ 1001
P = 1	奇偶性为偶 (even)	P = 0	奇偶性为奇 (odd)
CY = 1	有进位	CY = 0	无进位

其中, 辅助进位标志AC用于二—十进制转换。例如, 十进制数12用二进制表示为00001100, 如果运算结果为12 (因为比9大), 则AC = 1[●]。

(4) 寄存器B和C、D和E、H和L可以连起来作为寄存器对(16位)使用。其中, H和L连起来使用时具有地址寄存器的功能, 并且在16位运算中具有16位累加器的功能。

(5) 运算器(ALU)中可进行的运算有: 8位整数加、减, 左移(乘以2), 右移(除以2), 逻辑运算(取反、与、或、异或), 比较(只是按减法设置标志, 而A的内容并不改变)等。另外, H和L可看作累加器, 通过两次使用运算器而实现16位的加法运算。

(6) 寄存器、寄存器对及存储单元除了可以把其内容送到运算器作运算以外, 还可以利用图1.3右下方的增量器、减量器使其内容加1或减1。

(7) 可以在存储器(以字节为单位设置地址)中按地址从大到小设置堆栈, 堆栈的栈顶由16位的堆栈指示器(SP)指出。

(8) CPU和外面的数据传送是按8位并行进行的。但是, 8085也可以传送串行数据。

(9) 有中断功能。8080的中断请求信号线只有一条, 而8085却有五条。

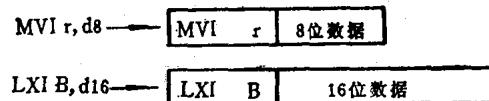
1.3 8080/8085指令系统

要在微型计算机上熟练地使用汇编语言, 就必须熟悉微型计算机的指令系统。这里, 以8080/8085为例, 按照表1.1的顺序简要地说明一下各指令的含义。

1.3.1 数据传送类指令

如图1.4所示, 传送类指令主要是把8位或16位的数据在寄存器与寄存器之间或者寄存器与存储器之间进行传送。其中, MOV M, r指令中的M表示存储器的一个单元(存储器中的一个字节), 其地址为H-L(寄存器H和L连起来组成的寄存器对)中的内容, r为A、B、C、D、E、H、L中的任意一个。

此外, 所谓立即数(Immediate)就是放在指令中(低位字节)的数据。立即数型指令的格式如下:



在上面的例子中, MVI指令表示把该指令低位的一个字节的数据放到寄存器r中, 而LXI指令则表示把该指令低位的两个字节的数据放到寄存器对B-C中。

● 当AC = 1时, 如果执行累加器的十进制调整(DAA——Decimal Adjust A)指令, 则在A的低4位上加6, 于是A的内容成为00001100 + 00000110 = 00010010。作为二—十进制数, 每4位一组, 就成为12。

表1.1 8080/8085指令一览表〔标志中的“○”表示设置（即受影响），●表示置0〕

(1)	数据传送	8085 的 状态数	标 志				
			Z	S	P	CY	AC
MOV	r1, r2	寄存器 r2 的内容送到寄存器 r1	4				
MOV	M, r	寄存器 r 的内容送到 H-L 所指的内存单元	7				
MOV	r, M	H-L 所指的内存单元的内容送到寄存器 r	7				
MVI	r, d8	8 位立即数送到寄存器 r	7				
MVI	M, d8	8 位立即数送到 H-L 所指的内存单元	10				
STAX	B	把 A 的内容存到 B-C 所指的内存单元	7				
STAX	D	把 A 的内容存到 D-E 所指的内存单元	7				
LDAX	B	把 B-C 所指的内存单元的内容取到 A	7				
LDAX	D	把 D-E 所指的内存单元的内容取到 A	7				
STA	ad	按直接寻址方式存 A 的内容	13				
LDA	ad	按直接寻址方式取数到 A	13				
LXI	B, d16	16 位立即数送到寄存器对 B-C	10				
LXI	D, d16	16 位立即数送到寄存器对 D-E	10				
LXI	H, d16	16 位立即数送到寄存器对 H-L	10				
SHLD	ad	按直接寻址方式存 H-L 的内容	16				
LHLD	ad	按直接寻址方式取数到 H-L	16				
XCHG		D-E 和 H-L 的内容互相交换	4				
(2)	加法						
ADD	r	寄存器 r 的内容加到 A	4	○	○	○	○
ADC	r	寄存器 r 的内容连同进位加到 A	4	○	○	○	○
ADD	M	H-L 所指的内存单元的内容加到 A	7	○	○	○	○
ADC	M	H-L 所指的内存单元的内容连同进位加到 A	7	○	○	○	○
ADI	d8	8 位立即数加到 A	7	○	○	○	○
ACI	d8	8 位立即数连同进位加到 A	7	○	○	○	○
DAD	B	B-C 的内容加到 H-L	10			○	
DAD	D	D-E 的内容加到 H-L	10			○	
DAD	H	H-L 的内容加到 H-L	10			○	
DAD	SP	堆栈指示器的内容加到 H-L	10			○	
(3)	减法						
SUB	r	A 减去寄存器 r 的内容	4	○	○	○	○
SBB	r	A 减去借位及寄存器 r 的内容	4	○	○	○	○
SUB	M	A 减去 H-L 所指的内存单元的内容	7	○	○	○	○
SBB	M	A 减去借位及 H-L 所指的内存单元的内容	7	○	○	○	○
SUI	d8	A 的内容减去 8 位立即数	7	○	○	○	○
SBI	d8	A 的内容减去借位及 8 位立即数	7	○	○	○	○
(4)	逻辑运算						
ANA	r	寄存器 r 和 A 的内容相“与”	4	○	○	○	●
XRA	r	寄存器 r 和 A 的内容“异或”	4	○	○	○	●
ORA	r	寄存器 r 和 A 的内容相“或”	4	○	○	○	●
CMP	r	寄存器 r 和 A 的内容进行比较	4	○	○	○	○
ANA	M	H-L 所指的存储单元的内容和 A 的内容相“与”	7	○	○	○	●
XRA	M	H-L 所指的存储单元的内容和 A 的内容“异或”	7	○	○	○	●
ORA	M	H-L 所指的存储单元的内容和 A 的内容相“或”	7	○	○	○	●
CMP	M	H-L 所指的存储单元的内容和 A 的内容进行比较	7	○	○	○	○
ANI	d8	8 位立即数和 A 的内容相“与”	7	○	○	○	●

(续)

			8085 的 状态数	标 志				
				Z	S	P	CY	AC
XRI	d 8	8 位立即数和 A 的内容“异或”	7	○	○	○	●	●
ORI	d 8	8 位立即数和 A 的内容相“或”	7	○	○	○	●	●
CPI	d 8	8 位立即数和 A 的内容进行比较	7	○	○	○	○	○
(5) 移位								
RLC		A 的内容循环左移	4				○	
RRC		A 的内容循环右移	4				○	
RAL		A 的内容连同进位循环左移	4				○	
RAR		A 的内容连同进位循环右移	4				○	
(6) 其他指令								
CMA		A 的内容取反	4					
STC		进位标志置 1	4				○	
CMC		进位标志取反	4				○	
DAA		累加器的内容按十进制进行调整	4	○	○	○	○	○
(7) 增量/减量								
INR	r	寄存器 r 的内容加 1	4	○	○	○	×	○
DCR	r	寄存器 r 的内容减 1	4	○	○	○	×	○
INR	M	H-L 所指的内存单元的内容加 1	10	○	○	○	×	○
DCR	M	H-L 所指的内存单元的内容减 1	10	○	○	○	×	○
INX	B	B-C 寄存器对的内容加 1	6					
INX	D	D-E 寄存器对的内容加 1	6					
INX	H	H-L 寄存器对的内容加 1	6					
DCR	B	B-C 寄存器对的内容减 1	6					
DCR	D	D-E 寄存器对的内容减 1	6					
DCR	H	H-L 寄存器对的内容减 1	6					
(8) 转移								
JMP	ad	无条件转移	10					
JC	ad	有进位则转移	7/10					
JNC	ad	无进位则转移	7/10					
JZ	ad	为零则转移	7/10					
JNZ	ad	非零则转移	7/10					
JP	ad	为正则转移	7/10					
JM	ad	为负则转移	7/10					
JPE	ad	奇偶性为偶则转移	7/10					
JPO	ad	奇偶性为奇则转移	7/10					
PCHL		H-L 的内容送到程序计数器	6					
(9) 子程序调用								
CALL	ad	无条件调用	18					
CC	ad	有进位则调用	9/18					
CNC	ad	无进位则调用	9/18					
CZ	ad	为零则调用	9/18					
CNZ	ad	非零则调用	9/18					
CP	ad	为正则调用	9/18					
CM	ad	为负则调用	9/18					
CPE	ad	奇偶性为偶则调用	9/18					