高等学校教材

微型计算机
汇编语言程序设计

大连工学院　王月霞　编

電子工業出版社

内 容 简 介

本教材主要讲述Z80汇编语言程序设计，其中包括：与程序设计有关的硬件简介、Z80指令系统、寻址方式、伪指令；程序的顺序、分支、循环和子程序结构及其程序设计；浮点数运算程序的设计；分类与检索程序的设计；输入/输出与中断程序的设计以及汇编语言程序的运行等等。最后简要地介绍Intel8086、Zilog8000和Motorola 68000三种十六位微型机的寻址方式和指令系统。本书中含有大量程序实例和作业题。

微型计算机汇编语言程序设计

王月霞 编

责任编辑 王惠民

*

电子工业出版社（北京市万寿路）

妙峰山印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

开本 787×1092 印张 29.875 字数 686千字

1985年11月第1版 1985年12月第1次印刷

印数：15500册 定价：6.35元

统一书号 15290·184

出 版 说 明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校工科电子类专业课教材的编审、出版的组织工作。从一九七七年底到一九八二年初，由于各有关院校，特别是参与编审工作的广大教师的努力和有关出版社的紧密配合，共编审出版了教材159种。

为了使工科电子类专业教材能更好地适应社会主义现代化建设培养人才的需要，反映国内外电子科学技术水平，达到“打好基础、精选内容、逐步更新、利于教学”的要求，在总结第一轮教材编审出版工作经验的基础上，电子工业部于一九八二年先后成立了高等学校《无线电技术与信息系统》、《电磁场与微波技术》、《电子材料与固体器件》、《电子物理与器件》、《电子机械》、《计算机与自动控制》、中等专业学校《电子类专业》、《电子机械类专业》共八个教材编审委员会，作为教材工作方面的一个经常性的业务指导机构，并制定了一九八二～一九八五年教材编审出版规划。列入规划的教材、教学参考书、实验指导书等共217种选题。在努力提高教材质量，适当增加教材品种的思想指导下，这一批教材的编审工作由编审委员会直接组织进行。

这一批教材的书稿，主要是从通过教学实践、师生反映较好的讲义中评选择优和从第一轮较好的教材中修编产生出来的。广大编审者，各编审委员会和有关出版社都为保证和提高教材质量作出了努力。

这一批教材，分别由电子工业出版社、国防工业出版社、上海科学技术出版社、西北电讯工程学院出版社、湖南科学技术出版社、江苏科学技术出版社、黑龙江科学技术出版社和天津科学技术出版社承担出版工作。

限于水平和经验，这一批教材的编审出版工作肯定还会有许多缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评建议，共同为提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

前　言

本教材系由《计算机与自动控制专业》教材编审委员会《自动控制专业》教材编审小组评选审定，并推荐出版。

该教材由大连工学院王月霞编写，杭州电子工业学院章鉴汀担任主审。编审者均依据自动控制编审小组审定的编写大纲进行编写和审阅的。

本课程参考教学时数为50～60学时。鉴于近年来我国各高校引进的八位机以Z80为CPU的居多，故本教材主要讲述Z80汇编语言程序设计。其内容为Z80指令系统、寻址方式、伪指令，程序的顺序、分支、循环和子程序等四种基本结构及其程序设计；为了从计算机原理向汇编语言程序设计过渡，教材中还简单地介绍了与程序设计有关的计算机原理部分内容。为了满足自动控制专业的需要，本教材加强了输入/输出和中断程序设计的内容；分类和检索程序的设计也是常常遇到的，但由于自控专业不开设数据结构课程，故也在本教材中讲述。最后简要地介绍Intel 8086、Zilog 8000和Motorola 68000三种十六位微型计算机的寻址方式和指令系统。汇编语言程序设计是一门实践性很强的课程，因此，本书编写了若干实例与习题，并强调上机实践，以培养软件开发能力。

使用本教材时应注意在讲授的同时至少安排20学时以上机操作实习。并且，讲述该教材应着重讲述程序设计的方法和算法，而具体程序可留给学生自己阅读。第一章内容可根据学生情况进行删减。若删去第五、十两章，并选取各章中前半部分作为基本内容，则可在30学时内讲完。

在本教材编写过程中，钱飞参加草拟了部分习题，由广华和本系83届研究生参加了部分程序的调试工作，腾玉邦参加了部分接口实验。王众讬教授认真全面地审阅了初稿。参加初稿审阅工作的还有唱江盛，高铭学、蔡小平、杨德礼和邵军哲等同志，他们为本书提出了许多宝贵意见，这里表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编者

1984・12

目　录

第一章 引 论

电子计算机自1946年问世以来得到飞速的发展，应用领域也越来越广。近十多年来，随着大规模集成电路的出现和发展，诞生了一代新型的电子计算机——微型计算机（Microcomputer）。它利用大规模集成电路技术，把计算机的中央处理单元（CPU-Central Processing Unit）集成在一个芯片上，称为微处理器（Microprocessor）。典型的8位微处理器有8080、6800、Z80、6502等。16位微处理器有8086、68000、Z8000等等。同样，利用大规模集成电路技术也制造出来了读写存储器芯片RAM，只读存储器芯片ROM和各种通用的以及专用的输入/输出接口电路芯片。人们利用这些典型的芯片，再配备上必要的输入、输出设备(键盘、显示器、打印机等)以及软件就构成了微型计算机。

由于微型计算机采用了大规模集成电路，不仅大大缩小了计算机的体积，更由于芯片能够大批生产，从而大大地降低了成本，提高了可靠性。微型计算机自七十年代初诞生以来，几经换代，现在它的功能已赶上或超过了以往的小型机，甚至初期的大、中型机。将现在微型电子计算机与世界上第一台计算机相比较，体积几乎缩小到三万分之一，价格则下降到十万分之一，而速度增大了二十多万倍，效率提高了一百万倍。微型计算机应用领域也更加广阔，不仅应用于国防、科研、教学、工农业生产，甚至普及到了办公室和家庭。微型机的广泛应用，大大地减轻了人们脑力劳动的负担。因此，它有微型电脑之称。随着微型电脑日新月异突飞猛进的发展，产业社会正酝酿着一场新的革命，它将把人类推向一个全新的信息社会。因此，为了迎接这场工业革命的到来，适应信息社会的需要，必须大力普及微型计算机。普及微型计算机的任务之一，就是普及微型计算机汇编语言程序设计。在正式讲述汇编语言程序设计之前，利用这一章复习一下微型机原理课所学过的有关微型计算机结构、寻址方式、指令系统等知识，同时也讲述一些汇编语言的预备知识。

第一节 机器语言、汇编语言和高级语言

我们都知道，为在电子计算机上执行某一任务，就要为这一任务编写一个程序。再把写好的程序输入到计算机中，然后让计算机执行程序。当程序执行完毕，输出计算结果，任务也就完成了。对于微型机来说，也是如此。

那么，什么是程序？程序用什么来编写呢？程序是使计算机执行特定任务的指令序列。为某一任务编写程序，就是编排一个执行该任务的指令序列。为了使指令序列把人的意图传达给电子计算机，这就要求程序既能表达人的意图，又要使计算机明白它的任务。于是，产生了人和电子计算机之间进行交流所用的语言——程序设计语言（Programming Language）。随着电子计算机的发展，程序设计语言也是由低级向着高级发展，即由机器语言发展为汇编语言和高级语言。

一、机器语言

机器语言（Machine Language）就是用二进制编码的机器指令。

早期的电子计算机采用机器能直接识别的语言——机器语言编写程序。因为电子计算机的硬件只能识别二进制数0和1，所以，电子计算机诞生的初期，是用二进制数编码的指令来编写程序的。

为了人们阅读和书写方便，二进制编码的指令可以写成八进制或十六进制形式。

例如，我们要用Z80指令写一程序。把存储单元2200H（十六进制数）中存储的数，加上存储单元2201H中存储的数，将它们的和送入存储单元2202H中去。用十六进制形式指令书写的程序如下：

```
210122
3A0022
86
320222
76
```

将该程序输入到Z80机后，即可直接执行。

电子计算机可直接识别和执行的程序就是机器语言程序。它又称为目的程序（Objec Program）。

机器语言程序存在着严重的不足：

① 机器语言程序不能用人们熟悉的形式来描述计算机要执行的任务，因而难读，难懂，不便于人们理解、记忆和交流。

② 用机器语言编写程序是一件十分繁琐的事情，极易出错。一旦有错，也很难发现。因此，程序难以调试（Debug）或者叫排错。

二、汇编语言

为了克服机器语言带来的缺点和困难，后来人们采用便于记忆、并能描述指令操作功能的符号表示指令的操作码，也就是用符号给各指令起个名字。这些名字叫助记符或记忆符（Mnemonics）。助记符一般是说明指令功能的英语词汇或者英语词汇的缩写。例如用ADD作加法指令助记符，OR作逻辑“或”指令助记符等等。此外，为了编写程序的方便，也可用符号表示参与操作的数据、CPU寄存器和存储单元地址等。例如，用A表示CPU的累加器，用符号ADDR表示一个存储单元地址等。

这种用助记符和符号地址表示的指令叫汇编格式指令。用汇编格式指令编写的程序叫汇编语言程序（Assembly Language Program）。

前面所介绍的加法运算的例子，用汇编语言编写的程序是这样的：

```
LD    HL, 2201H
LD    A, (2200H)
ADD   A, (HL)
LD    (2202H), A
HALT
```

第一行是传送指令，操作码助记符LD（Load）是装入的意思。这条指令的功能是：把常数2201H装入HL寄存器对中。常数2201H在这里是存储单元的地址码。

第二行也是传送指令，它把存储单元2200H的内容装入到CPU累加器A中、

第三行是加法指令，操作码助记符是ADD。该指令的功能是寄存器对HL指示的存储

单元（这里是2201H单元）的内容与累加器A中的内容相加，然后将它们的和送回A中。

第四行也是传送指令。它把累加器A中存放的结果送入到存储器2202H单元。

第五行是暂停指令。表示程序到此暂时停止。

从此例可以看出，用汇编语言编写程序方便得多，这种程序对任务执行过程有一定程度的描述，便于人们的阅读和记忆。但带来的问题是电子计算机硬件不能直接识别和执行这样的程序。

用汇编语言编写的程序叫源程序（Source Program）。为在电子计算机中执行用汇编语言写的源程序，必须把它翻译成为用机器语言写的目的程序。这种翻译工作可由人通过查表的办法来进行，这叫人工汇编。人工汇编是一种单调、重复、量大、繁重而效率极低的工作。它完全可以由电子计算机来完成。用电子计算机翻译的过程叫机器汇编。

机器汇编是由一种起翻译作用的程序——汇编程序（Assembler）来完成的，如图1-1-1所示。

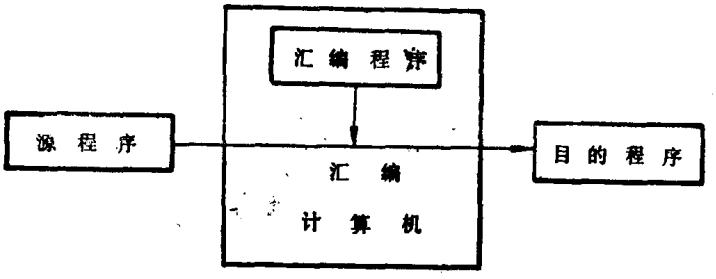

图 1-1-1 汇编过程示意图

计算机执行汇编程序时，将用汇编语言写的源程序，翻译为机器语言的目的程序。将目的程序再输入到计算机中就可以执行而得到结果了。

虽然汇编语言比起机器语言有很大的进步，但仍然有不少不可克服的缺点：

① 由于汇编指令与机器指令是一一对应的，因而所编写的程序，对任务的描述与人们的习惯差距仍然不小，不经过专门训练，仍然是难读、难懂。

② 用汇编语言进行程序设计，需要程序设计人员对所用微型机的性能有详细的、深入的了解。比如该机有什么寄存器和指令，指令如何使用寄存器，采用哪些寻址方式和其它有关知识等等。

③ 每种微型机都有它自己的汇编语言，它反映着机器自身系统结构。因而一种微型机的汇编语言程序是不能直接移植给另一种型号的机器的。这就有碍微型机程序的交流，给推广应用微型机带来很大不便。

三、高级语言

克服汇编语言带来的许多困难的办法是使用高级语言（High Level Language）。高级语言近乎人的语言，是独立于机器的。程序设计人员不再需要对机器指令和结构有深入的了解。一个用高级语言编写的程序可以在各种不同型号的机器上运行（只要配有该种语言的编译或解释程序）。高级语言的程序由语句组成，每个语句相当于若干条机器指令，因此语句的功能很强，用语句编写程序要容易得多，而且这种程序更接近于人们用习惯语言对任务的描述。

高级语言编写的程序也叫源程序，它必须经过编译程序（Compiler）翻译成目的程序才能执行。或者通过解释程序（Interpreter）边解释边执行。

目前用于微型机的高级语言种类繁多。如BASIC、FORTRAN、COBOL、PL/1 Pascal等等。使用者可根据任务的性质选择合适的高级语言。

高级语言具有容易学习、使用方便、通用性强、便于交流等等许多优点。一般适用于科学计算、事务处理等等。

尽管高级语言使用方便，但对于应用于实时控制来说，高级语言也还有某些不足之处，下面例举两点：

① 程序要访问具体硬件时，高级语言不如汇编语言方便。

② 为执行某一任务，用高级语言编写的程序，经编译后产生的目的程序，与较好的程序员用汇编语言编写，经过汇编后的目的程序相比，要多占用0.5～1倍的存储单元，执行程序花费的时间要长出0.5～2倍。

这就是说，使用汇编语言编写程序，能充分发挥机器硬件的作用，能高效地使用机器。汇编语言还是编写电子计算机通用软件的工具。因而掌握汇编语言是十分重要的。

机器语言程序使用很不方便，目前只有那些要求机器造价低、存储容量小的单板机上才使用机器语言程序。但这样的机器语言程序也可用具有汇编语言的同类微型机汇编出来。可见，即使应用单板机，编程序也应掌握汇编语言。

第二节　汇编语言的支持硬件简介

从上节可知，由于高级语言不依赖于机器硬件，所以用高级语言进行程序设计，不要求设计人员了解机器具体的内部硬件结构。可是，汇编语言却不同，它与机器硬件密切相关，不了解有关机器内部硬件结构，就不能从事汇编语言程序设计。因此，这一节就本书所用的Z80汇编语言有关的硬件结构作一简要介绍。

一、Z80硬件系统

Z80是美国Zilog公司开发的八位微处理器。以微处理器Z80作为CPU，配上ROM和RAM，各种输入/输出（I/O）接口电路和必要的外部设备（如键盘、显示器、打印机和外存储设备等）就组成了微型计算机。

Z80同其它的微处理器一样，采用总线结构（如图1-2-1所示）。共有三组总线：地址

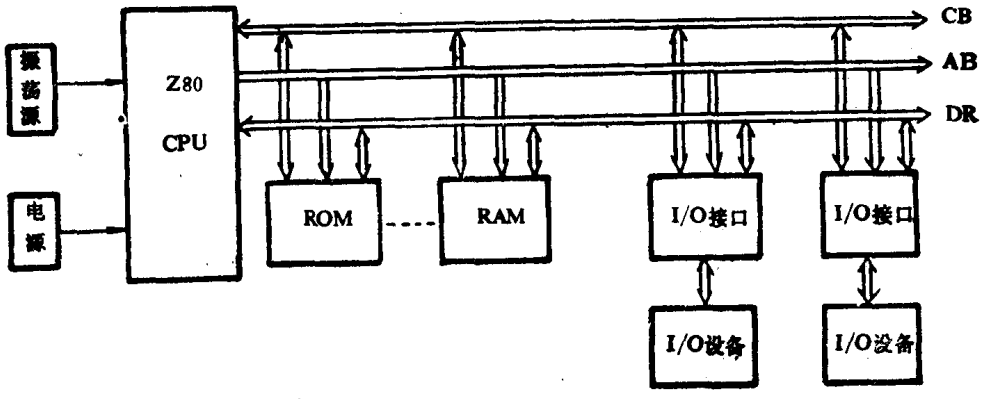

图 1-2-1　微型机结构框图

总线AB，控制总线CB和数据总线DB。

（一）地址总线AB (Address Bus)

地址总线为16位即A_{15}—A_0，因而，可寻址的存储单元为$2^{16}=64K$。可编程序I/O接口也通过地址总线来寻址，它使用地址总线的低8位，故可寻址256个I/O端口 (Port)。

地址空间的分配因机而异。TRS-80机的地址分配图如图1-2-2所示。

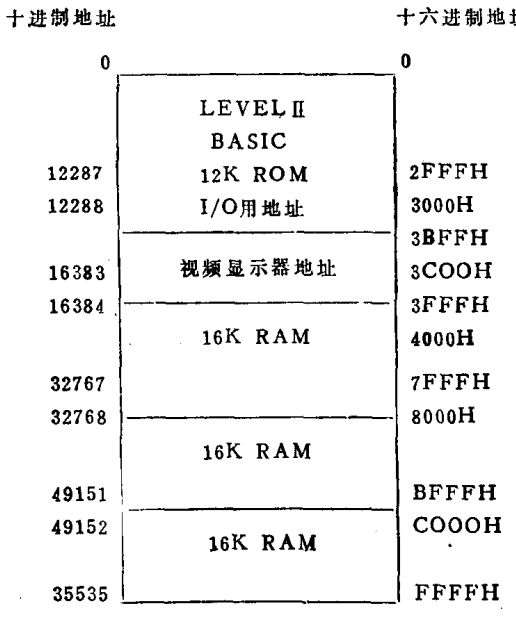

图 1-2-2，TRS-80机地址分配图

0～2FFFH寻址12K ROM。ROM存储LEVEL Ⅱ BASIC解释程序。

3000H～3FFF寻址I/O接口，其中3COOH～3FFFH寻址视频显示器的RAM单元。3000H～3BFFH供输入/输出设备使用。

4000H～FFFFH可寻址48K RAM。运行TRS-80编辑/汇编程序至少要配有16K RAM。16K的系统中，源程序缓冲区约为8.5K。

（二）数据总线DB (Data Bus)

数据总线为8位双向总线。用于CPU与存储器，CPU与I/O接口之间的数据传送。由于采用总线结构，也可以通过DMA (Direct Memory Access) 接口，在存储器和外部设备之间直接传送数据。

（三）控制总线CB (Control Bus)

控制总线传送各种控制信号，使CPU、存储器、I/O接口相互配合，协调工作。

（四）外部设备

为了运行Z80汇编程序，最小微型计算机系统应配备有ASCII（美国信息交换标准码）键盘，视频显示器和作为外存储的盒式磁带机。稍大些的系统应增配软盘驱动器和行式打印机。

二、Z80 CPU内部寄存器

微处理器Z80 CPU内部结构框图如图1-2-3所示。从图上看到，主要组成部件可分三类：算术逻辑运算单元ALU；指令、总线和CPU控制部件；CPU内部寄存器。这三

类部件中与汇编语言程序设计有密切关系的是CPU内部寄存器，图1-2-4是CPU内部寄存器的示意图。

Z80 CPU内部寄存器用208位静态RAM实现，它组成十八个8位寄存器和四个16位寄存器。按用途可分为三种类型：专用寄存器，通用寄存器和累加器与状态标志寄存器。

（一）专用寄存器

专用寄存器有程序计数器PC、堆栈指针SP，变址寄存器IX和IY，中断向量寄存器I和存储器刷新寄存器R。

1．程序计数器PC(Program Counter)

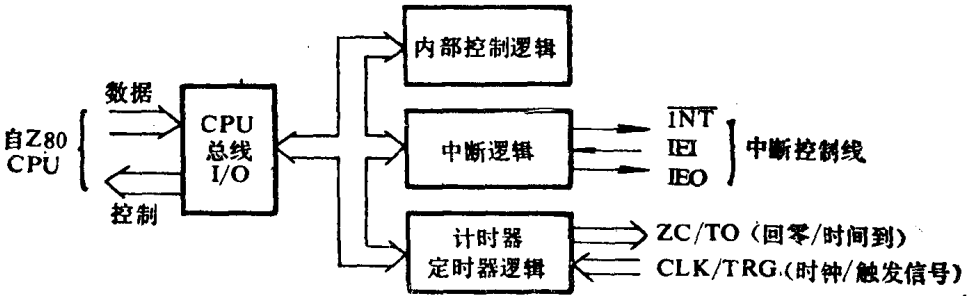

图 1-2-3 Z80 CPU内部结构

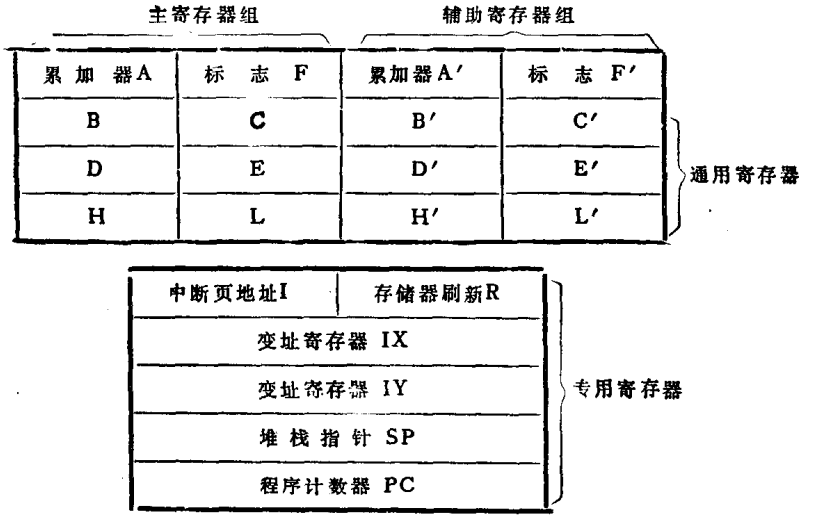

图 1-2-4 Z80 内部寄存器

程序计数器又称指令计数器或指令地址计数器，用以存放正在执行的指令地址，一般随着指令的执行计数器自动加1，指示出接着要执行的下条指令地址。因此，在执行目的程序前，把程序的起始地址置入程序计数器，程序便能从开始地址顺序执行。

指令计数器中指令的地址除自动加1外，在执行程序过程中，当遇到转移指令，调用子程序指令或响应中断时，能自动地装入转移地址或中断服务程序入口地址。从而改变程序执行顺序，实现程序的转移。

Z80程序计数器是16位计数器，所以，装在64K存储器任何地方的程序都可以控制执行。

2．堆栈指针（Stack Pointer）

堆栈是在RAM存储器中按照后进先出的规则开辟的一个存储区域。SP是这个存储区