

Z-80汇编语言实用程序

天津科学技术出版社

73.87221
195.1

Z-80汇编语言实用程序

刘全忠 编

334/07



Z-80汇编语言实用程序

刘全忠 编

责任编辑：宋淑萍

•

天津科学技术出版社出版

天津市赤峰道124号

天津新华印刷一厂印刷

新华书店天津发行所发行

•

开本 787×1092毫米 1/32 印张 14.25 字数 300,000

一九八四年八月第一版

一九八四年八月第一次印刷

印数：1—35,000

书号：15212·126 定价：2.25元

内 容 提 要

本书是一本Z-80微型计算机汇编语言实用程序汇集。第一部分较系统地介绍了汇编语言程序设计的方法和技巧,包括简单程序,转移和循环,码的转换,表格,算术运算,子程序,输入/输出,中断和计数定时电路等程序。第二部分全面地介绍了适用于工业控制和检测仪表中常用的数据运算,码制转换,控制操作和显示,以及数/模、模/数转换等实用子程序。全书的特点是通俗易懂,实用性强。

本书可供从事微型计算机应用的科技人员阅读,也可供高等院校有关专业的师生参考。



序 言

早在四、五十年代计算机刚刚问世时，就有人预料它将会引起一次新的工业革命。不过，几十年来计算机一直是一件庞大而又昂贵的设备，人们不得不把它安置在少数科研和教学单位的专门实验室中，而在更广阔的领域对它却是望而生畏或认为高不可攀，因而，计算机的出现给人们带来的冲击也是有限的。但自七十年代初出现微型计算机以来，使计算机的面貌焕然一新。微型计算机不但功能齐全，体积微小，使用方便，而且价格逐年下降，目前，Z-80单板微型计算机在国内售价仅两千多元，和一台普通示波器的价格相当，但它给人们带来的经济效益要远远地超过一台示波器。例如，它可以实现一个完整的实时控制系统，可以组装一台智能仪器并形成自动检测系统等。至于一个微型计算机系统给人们带来的效益更要远远地超过前者。十几年来，微型计算机技术得到迅速的发展，它的应用范围正在深入到各个领域，一个大量应用计算机的时代正在到来。

但是，应用计算机的一个很大的难题是要正确无误地实现人一机对话，即人们要把自己的意图用准确的、计算机能够识别的语言表达出来，而计算机必须严格地按照这些语言工作。这里就隐藏着一种危险，当人们编制的语言稍有差错，计算机就会按照这种错误的语言工作，以致给人们造成不堪设想的恶果。因而，帮助程序设计人员掌握编程技能，

节省编程时间，充分发挥计算机效能已提到议事日程上来了。

对于微型计算机系统，一般可使用专门的高级语言进行源程序的设计，系统编译程序将源程序翻译成机器能够识别的目标程序，并且能帮助程序设计人员调试程序。然而，由于高级语言编程效率低，编译程序本身又要占用大量存储器，这就限制了它在微型机小系统中的使用。当微型机在具有频繁输入输出的实时控制系统中应用时，使用汇编语言比使用高级语言设计程序要优越得多。同时，目前国内大量微型单板计算机用户，由于缺少微型机系统所能提供的任何支援，无疑要使用汇编语言来设计程序。汇编语言是一种面向机器的语言，也就是说，汇编语言程序是不能互相移植的，这就要求程序设计人员在已有资料的基础上自己动手编制程序。实践证明，掌握汇编语言程序设计，是微型计算机应用的一个基本技能。

为了便于读者学习，本书的全部 Z-80 汇编语言程序分两部分介绍。第一部分给出了九个实验程序，着重典型程序的分析，并研究程序内部的执行过程，以利培养设计、找错和修改程序的能力，也为使用第二部分介绍的实用子程序打下必要的基础。第二部分介绍了自动检测仪表和工业控制中常用的数据运算、控制操作和显示，以及普遍使用的模/数、数/模转换等实用程序，全部以子程序的形式给出，便于用户直接调用。

书中第一部分的部分程序是参考《Z80 Assembly Language programming》一书编写而成，这些程序作为编程基本训练是很有价值的。其余所有程序都是作者自己编制

并经上机调试通过的。由于编制一个高质量的实用程序所涉及到的知识领域十分广泛，而作者水平有限，因此，所列程序不一定是最优程序，只供读者使用或编程时参考。对其中不妥或错误之处，敬请读者批评指正。

在编写过程中，得到天津大学杨志泽副教授的指导和关心，陈碧莲同志、李芝英同志提供了大量的资料和数据，也得到《计算机世界》编辑部和全国各地同行们的鼓励和支持，在此一并表示谢意。

作 者

一九八三年七月于天津大学

目 录

第 一 部 分

第一章 简单程序	(3)
§ 1-1 8 位数运算	(4)
§ 1-2 16位数运算	(14)
§ 1-3 数据块传送	(18)
第二章 转移和循环程序	(20)
§ 2-1 标志寄存器	(21)
§ 2-2 数列求和	(25)
§ 2-3 拣选	(32)
第三章 码的转换	(41)
§ 3-1 十六进制到ASCII码	(42)
§ 3-2 十进制到七段码	(47)
§ 3-3 ASCII码到十进制	(54)
§ 3-4 BCD码到二进制	(58)
第四章 表格	(61)
§ 4-1 算术运算	(61)
§ 4-2 拣选和排序	(66)
§ 4-3 转移表	(82)

第五章	算术运算程序	(87)
§ 5-1	多精度加法	(88)
§ 5-2	乘法	(94)
§ 5-3	除法	(99)
第六章	子程序	(107)
§ 6-1	码的转换	(109)
§ 6-2	算术运算	(116)
第七章	输入和输出	(121)
§ 7-1	开关与微型计算机连接	(127)
§ 7-2	发光二极管与微型计算机连接	(133)
§ 7-3	数据输入/输出	(136)
§ 7-4	七段LED数码管	(138)
§ 7-5	键盘	(145)
第八章	中断	(151)
§ 8-1	外设请求输入/输出	(152)
§ 8-2	逻辑控制	(170)
§ 8-3	多重中断和优先中断	(175)
第九章	计数定时电路	(183)
§ 9-1	定时器	(187)
§ 9-2	计数器	(194)
§ 9-3	系统中断链	(198)

第 二 部 分

第十章 多精度四则运算	(207)
§ 10-1 加减法	(208)
§ 10-2 乘法	(214)
§ 10-3 除法	(228)
§ 10-4 十进制乘除法	(260)
§ 10-5 带符号加减法	(270)
§ 10-6 浮点运算	(278)
第十一章 BCD码和二进制数的相互转换	(298)
§ 11-1 整数二进制到BCD码的转换	(298)
§ 11-2 小数二进制到BCD码的转换	(304)
§ 11-3 整数BCD码到二进制的转换	(312)
§ 11-4 小数BCD码到二进制的转换	(316)
第十二章 键盘和拨盘	(326)
§ 12-1 键盘输入数据	(326)
§ 12-2 拨盘输入数据和命令	(334)
第十三章 显示	(340)
§ 13-1 串行扫描显示	(340)
§ 13-2 并行锁存显示	(348)
第十四章 数/模和模/数转换	(352)
§ 14-1 数/模转换器	(353)

§ 14-2 模/数转换器(368)

§ 14-3 双斜积分模/数转换(377)

附录一实用子程序一览表(397)

附录二 Z-80指令系统表(按内部逻辑功能划分)(404)

附录三 Z-80指令系统表(按字母顺序排列)(432)

参考文献

第一部分

这部分包括前九章的内容，全部以实验的形式给出，以便于对典型程序的分析和研究，它包括简单程序，转移和循环程序，码的转换，表格，算术运算程序，子程序，输入和输出，中断以及计数定时电路等程序。这部分内容是学习 Z-80 汇编语言程序设计，掌握接口技术必不可少的基本训练。

所有程序可以在 DBJ—Z80 或 TP—801 型单板微型计算机上执行，这两种型号的单板机在硬件结构上完全一致，并具有基本相同的 ZBUG 监控程序，适用于简单的数据处理和中、小型实时控制系统的开发，只是 TP—801 较前者在开发功能上稍有改进，板上的存储器容量也增加了一些。

在编制、调试或使用汇编语言程序时，必须明了计算机存储器和外设的地址分配，否则，程序将难以执行，甚至会破坏已编好的程序。DBJ—Z80 和 TP—801 单板微型机的地址分配见表 1 和表 2。

使用具有汇编功能的 Z-80 微型计算机系统时，要对程序中所使用的地址根据具体机型作相应的修改，并在程序（包括子程序、服务程序和数据区）的开始加上 ORG 伪指令，而在程序的最后加 END 伪指令；数据输入使用 DEFB 伪指令，同样可以执行这部分的所有程序。当然，如果程序中使用了单板机 ZBUG 监控程序中的子程序和单板机上的六位数显示器，就要对程序作适当的修改和增加必要的外设。

表 1

存储器地址分配

地 址 (Hex)	容量	类 型	用 途	
			DBJ—Z80	TP—801
0000~07FF	2 K	PROM或 EPROM	监控程序ZBUG	监控程序ZBUG
0800~0FFF	2 K	EPROM	安放用户程序	安放用户程序
1000~17FF	2 K	EPROM	EPROM编程器	EPROM编程器
1800~1FFF	2 K		未用	未用
2000~238F	912	RAM	用户使用	用户使用
2390~23FF	112	RAM	监控程序使用	用户使用
2400~27FF	1 K	RAM	待扩充(用户使用)	用户使用
2800~2F8F	1936	RAM	未用	用户使用
2F90~2FFF	112	RAM	未用	监控程序使用
3000~3FFF	4 K		未用	未用

表 2

通道地址分配

地址(Hex)	用 途
80~83	并行输入输出接口PIO
84~87	计数定时电路CTC
88~8B	字模锁存器(用于显示器)
8C~8F	位选锁存器(用于显示器、键盘)
90~93	键盘
94~97	待扩充
98~9B	待扩充
9C~9F	待扩充

第一章 简单程序

简单程序一般是指整个程序从始至终顺序执行，在执行过程中不会出现分支或转移，因而又称作直线程序。当然，任何实用程序绝非单纯由简单程序组成，但初学编程从简单程序入手却很有必要。

在简单程序中，大量使用的是数据传送指令，这是计算机最频繁和最基本的操作。Z-80指令系统具有强有力的数据传送指令，它包括寄存器之间，寄存器和存储器之间以及存储器之间的互相传送，不但有8位送数指令，也有16位送数指令。在存储器寻址上，有指令本身包含存储器地址的直接寻址，也可以用IX、IY和HL寄存器建立存储器指针，实现变址寻址和间接寻址。此外，它还有数据块传送和寄存器组互换的指令，这两类指令大大提高了传送速度和效率。

使用数据传送指令最要紧的一点，是数据从源地址传送到目的地址，目的地址的内容被源地址的内容更新，而源地址的内容并不改变。

在简单程序中，一般还要用到单一的算术、逻辑指令，它们完成某一种算术或逻辑运算，这在以后程序中再作分析。

简单程序并不意味着计算机执行过程会一目了然，对初学者尤其是这样。因而，有必要充分利用单板机开发装置的单步和断点功能，来研究和观察程序执行的细节，以逐步培养编程人员的调试、找错和修改程序的技能。

§1-1 8 位数 运算

程序1-1 求一个数的 1 的补码

目的 将存储器2040H单元的数求 1 的补码，并送入 2041H单元。

【实验步骤】

(1) 送入表1-1所列程序。

表1-1 直接寻址求一个数的 1 的补码程序

地 址 (Hex)	内 容 (Hex)	助 记 符	注 解
2000	3A	LD A, (2040H)	, GET DATA
2001	40		
2002	20		
2003	2F	CPL	, COMPLEMENT
2004	32	LD (2041H), A	, STORE RESULT
2005	41		
2006	20		
2007	76	HALT	
2040	6A		

(2) 置程序计数器PC到2000H，单步执行程序，观察显示器，并检查存储器2041H单元：

单步	PC	A	2041
1	2003	6A	XX
2	2004	95	XX
3	2007	95	95
4	2008	95	95

【程序分析】

该程序中数据传送是用直接寻址实现的，第1步操作的结果是将地址为2040H的存储器单元中的内容送入累加器A；第2步完成对A中数据求1的补码的运算；第3步将A的内容再送回地址为2041H的存储器单元。在两次送数操作中，存储器的地址在指令中都直接给出，这种直接寻址方式在编程时会经常用到。第4步为动态停机。

程序1-2 求一个数的1的补码

目的 将存储器2040H单元的数求1的补码，并送入2041H单元。

【实验步骤】

(1) 送入表1-2所列程序。

表1-2 间接寻址求一个数的1的补码程序

地 址 (Hex)	内 容 (Hex)	助 记 符	注 解
2000	21	LD HL,2040H	, POINT TO OPERAND
2001	40		
2002	20		
2003	7E	LD A,(HL)	, GET DATA
2004	2F	CPL	, COMPLEMENT
2005	23	INC HL	, POINT TO DESTINATION
2006	77	LD (HL),A	, STORE RESULT
2007	76	HALT	
2040	6A		

(2) 按复位键，置程序计数器PC到2000H，单步执行程序，检查下列各寄存器内容，应为：

单步	PC	A	H	L
1	2003	XX	20	40
2	2004	6A	20	40
3	2005	95	20	40
4	2006	95	20	41
5	2007	95	20	41
6	2008	95	20	41

(3) 检查存储器2041H单元的内容为95H。

【程序分析】

该程序是用间接寻址实现数据传送的，程序中使用 HL 寄存器对作为寻址指针。第 1 步建立指针初值，使 HL = 2040 H；第 2 步用间接寻址方式将指针所指的存储器单元 2040H 的内容送入累加器 A；在第 4 步，指针加 1，即 HL = 2041 H，以便使程序执行第 5 步时将 A 的内容再送入存储器的 2041 H 单元。可见，在间接寻址时，参加操作的存储器地址是由指针确定的。在编程中，这种间接寻址方式比直接寻址方式使用得更为普遍。

在 Z-80 指令系统中，可以作为指针的寄存器除 HL 外，还有 IX、IY 寄存器，如对累加器 A 进行操作，指针也可以用 BC、DE 寄存器对。可见，作为指针的寄存器，必须是 16 位寄存器或两个 8 位寄存器组成的寄存器对。使用指针是编程的重要技巧，后面还要做深入地研究。

程序 1-3 两数相加

目的 将存储器 2040H 和 2041H 单元的内容相加，并将其结果送入 2040H 单元。

【实验步骤】