

PDP—11

小型计算机及其应用

上 册

〔美〕雅曼斯W. 科贝 著 刘甲耀 译

华侨大学电脑系

1981.9

内 容 提 要

本书分上、下两册,上册介绍了小型计算机体系结构基本知识,详述了PDP-11汇编语言,同时讨论了适用于实验室计算机的各种型式的程序设计技术或算法。下册介绍了小型计算机的应用,包括诸如数据外围设备、一般信号、显示、测绘、峰值检测以及付里叶变换应用等许多特别复杂的算法,并用框图详细描述了这种算法。

本书可供各行各业从事小型计算机应用的人员及大专院校师生参考。也可作为实验室小型计算机应用短训班学习教材。

译者序

PDP-11 系列电子计算机是目前世界上最先进的小型计算机。其设计思想新颖；功能较强，便于扩充，不仅具有科学计算、实时控制和数据处理功能，而且带有各种扩展选件、外围设备、通讯选件和系统软件以供选用，因此，PDP-11 系列机在科研、工业、商业、交通运输等各部门得到广泛应用，成为科研、生产、教学不可缺少的工具。为了早日实现四个现代化，为了促进小型计算机在我国推广应用，特将此书译出。

本书分上、下两册，上册为：PDP-11 小型计算机概论。共十三章，内容包括计算机概述，数的基本概念，PDP-11 基本体系结构，寻址方式，编写简单程序，汇编程序，逻辑和移位指令以及乘除选择指令，输入/输出与子程序，陷阱和中断，浮点程序包FPMP11，高级程序设计概念与技巧，使用ODT-11调试程序。下册为：实验室小型计算机应用。共六章，内容包括LSP-11 数据外围设备，实验室数据的一般信号，数据显示绘图的方法，峰值印出，付里叶变换在实验室应用，以及偶极子干扰作用和相位修正方法。

本书几乎各章都提供有大量习题，书末並附有答案，同时还附有变址指令和各种条件码、ASCII码等若干有关附录及英汉对照。

本书可供各行各业从事计算机应用的人员和大专院校师生参考，也可作为实验小型计算机应用短训班学习教材。

为使译文更通俗地介绍给我国广大读者，译者在忠实原

文基础上，对原文个别段句作了意译与补充。由于目前国内计算机的名词术语尚未统一，所以只能按照译者习惯译法译出，由于译者水平所限，译文可能会有许多错误或不妥之处，敬请读者批评指正。

译者 1981.3.10. 于华侨大学

目 录

第一章 计算机概述	1
计算机存储器	1
固态存储器	2
计算机程序设计	3
第二章 数的基本概念	7
计算机字	7
数制	8
引言	8
二进制数表示法	8
八进制数	9
八进制数运算	10
换算	11
八进制数化十进制数	11
十进制数化八进制数	11
习题	12
PDP-11 程序设计所用的有关数的概念	13
反码	13
补码	14
减法	16
逻辑乘	17
或运算	18
异或运算	18

习题	19
第三章 PDP-11 基本体系结构	20
PDP-11 控制台用法	22
状态指示灯	23
控制台开关	24
由纸带装入程序	25
利用绝对地址装入程序输入程序	28
绝对地址装入程序带格式	29
第四章 PDP-11 指令系统概述	31
处理机状态寄存器	32
单操作数指令	34
双操作数指令	35
一个简单程序	37
汇编程序分析	39
习题	41
第五章 PDP-11 寻址方式	43
0 方式: 寄存器方式 R_n	44
1 方式: 寄存器间接方式 (R_n)	44
2 方式: 寄存器间接自动增量方式 $(R_n) +$	45
3 方式: 双间接自动增量方式 $@(R_n) +$	47
4 方式: 间接予减量方式 $-(R_n)$	47
5 方式: 双间接予减量方式 $@-(R_n)$	48
6 方式: 变址方式 $X(R_n)$	49
7 方式: 间接变址方式 $@X(R_n)$	50
寻址方式举例	50
习题	51
程序计数器寻址方式	52

PC方式2: 立即方式 #n	53
PC方式3: 绝对方式 @ #n	54
PC方式6: 相对方式 A	54
标号用法	55
PC方式7: 相对间接方式 @ A	57
转移指令	58
JMP 指令	62
习题	62
第六章 编写简单程序	66
把10个数相加的一个程序	66
不使用R0—R6的程序设计	71
专门予置	73
SOB 指令	74
习题	75
第七章 汇编程序	77
文本修改命令	80
检索命令	80
输入与输出命令	81
举例	82
汇编程序	86
汇编程序专用字符	92
ASCII转换	94
汇编程序指令	94
习题	95
第八章 逻辑和移位指令以及乘除选择指令	97
条件码操作	100
移位指令	101

	算术移位与算术移位兼容.....	105
	乘除指令.....	106
	SWAB指令	109
	习题.....	109
第九章	输入/输出与子程序	110
	状态寄存器.....	111
	数据缓冲寄存器.....	111
	ASCII 码	111
	读和打印字符.....	113
	电传打印机特情.....	115
	汇编程序惯例.....	116
	子程序和存储栈.....	117
	JSR指令.....	119
	一个比较简单的JSR形式.....	123
	高级子程序概念.....	125
	习题.....	128
	其它设备.....	129
	打印出八进制数.....	130
	打印出十进制数.....	134
	习题.....	134
第十章	陷阱和中断.....	135
	等待时间.....	138
	编写中断程序.....	138
	禁止中断.....	140
	中断驱动带复制.....	141
	陷阱.....	150
	软件陷阱.....	150

硬件陷阱.....	150
PS的T位和RTT指令	151
清除指令.....	152
习题.....	152
第十一章 浮点数学程序包FPMP11	154
予置.....	159
规定汇编程序的FPM11码	160
寻址方式详细说明.....	161
0方式: 存貯栈方式.....	161
100方式: @R0方式.....	163
200方式: 立即方式	164
300方式: 相对方式	164
装入和存貯FLAC.....	168
浮点输入和输出.....	170
FLIP和FLOP	170
用波兰方式调用子程序.....	176
浮点数.....	178
定点数.....	179
带有小数部分的定点数.....	179
GLOBL值用法.....	181
汇编指令概述和在PDP-10上使用FPMP11的	
程序连接.....	183
使用RT-11汇编和连接.....	184
整数乘除运算.....	186
直接通过R5调用子程序	186
缓冲输入与输出子程序.....	188
FPMP11错误处理.....	191

11.1.1	FORTTRAN 格式限定	193
11.1.2	PDP-11/40 浮点硬件	195
11.1.3	PDP-11/40 浮点硬件错误	196
11.1.4	修改FPMP11来使用PDP-11/40硬件指令	197
11.1.5	FPMP11错误	198
11.2	习题	198
第十二章	高级程序设计概念与技巧	199
12.1	使用 PDP-10	199
12.2	PDP-10 交叉汇编指令摘要	200
12.3	MACDLX和MACY11 错误信息摘要	201
12.4	PDP-11 汇编程序指令摘要	201
12.5	LNKX11 用法	203
12.6	存貯栈操作	204
12.7	浮动地址代码	207
12.8	重入程序	209
12.9	增量和减量	211
12.10	递归子程序调用的存貯栈用法	212
第十三章	使用ODT-11调试程序	215
13.1	检查和改变存貯单元	218
13.2	断点	219
13.2.1	检查寄存器	220
13.2.2	断点特性使用示例	220
13.2.3	改变偏移	222
13.2.4	字节偏移	223
13.2.5	假断点	224
13.3	检索存貯器中的各值	225
13.4	中断	226

ODT 补充	227
ODT-11X 命令	228
单指令方式	229
参考文献	231
附条一 变址指令和各种指令对条件码的影响	232
附条二 2 的幕次	235
附条三 十八进数换算表	237
附条四 ASCII 字符代码	250
附录五 存贮器映象	252
习题解答	255
英汉对照	297

第一章 计算机概述

计 算 机 存 贮 器

计算机存贮器可以用能迅速检测和改变的、具有两种不同状态的任何一种材料制成。历史上的大多数普通存贮器都是用一种铁磁合金所构成的很小的圆环，称之为磁芯。每个磁芯在不大于一个大周期就能磁化为两种状态的任一状态，也就是说，既可适时针方向磁化，也可顺时针方向磁化。这两种状态可随意规定为1或0，正如以下所述，并可置“1”或读出。

若干导线连成一串穿过每个磁芯，称之为X驱动线，Y驱动线，读出线和禁止线。在某些存贮器中，每个磁芯只有三根线，而读出线和禁止线的功能多多少少是兼容的。在磁芯存贮器中，可以用一个小电流通过X和Y两根驱动线来改变磁芯的磁化。只在X和Y线上两个电流相交的地方才发生磁化变化。这可使X和Y驱动线穿过若干磁芯，並可选择电流大小来改变磁芯磁化。小型计算机的磁芯，在每块平板上的标准排列为 64×64 或4096。每个存贮器堆积的平板数则根据所需要的字长来确定。

从X和Y驱动线的一个方向或另一个方向加入电流就可实现磁芯写1或0。显然，把磁芯的内容读出是不难理解的。把0写入所有磁芯，然后用读出线检测它的磁通是否变化。凡是磁通变化的地方则检测出相当磁芯原来等于1，而

磁通没有变化的磁芯则观察到等于 0。这就是图 1:1 所描述的情况。

注意，不管怎样，读出磁芯则破坏其内容。为此，结果必须存入比较恒定的触发器，以便读出后立即写入磁芯。这种操作是所有具有磁芯存储器的计算机的基础，并把它叫做读/写周期。多年来由于磁芯设计改进，这一操作时间已从毫秒降低到 1 微秒。

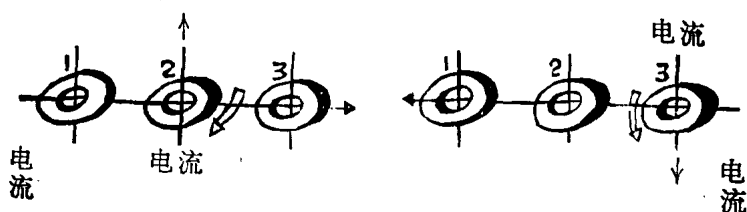


图1.1 (a) 2号磁芯写入1。(b) 3号磁芯写入0。

固 态 存 贮 器

有许多固态存储器的技术设计都有类似的特性。每一二进制位都可以看作等效于一个小电容器。只要外加电压，这一状态就保持置“1”。然而，在电压发生故障时如果不提供备蓄电池，具有固态存储器的计算机就“丧失记忆”。

另一方面，固态存储器存取速度非常快。没有执行任何磁化操作的各二进制位均可读出信息，并且是非破坏性读出。这种非破坏性读出方法，访问存储器所花费的时间很少，因为不需要重写。那完全可能造成读出时间为 70—100 毫微秒或者甚至更快的存储器。

固态存贮器也是相当小的，一种1024位存贮器片比通常 0.5×2 厘米集成电路块还小。这就使得设计格外经济。

计算机程序设计

在初期的计算机，逻辑和算术功能是组成在一块叫做中央处理机的组件上，並包含诸如加法、减法、乘法和除法，保存计算结果以及测试正号和负号这样一种简单的功能。这些功能可以通过利用接线板顺序安排，很象是在现在模拟计算机上所看到的情况一样。因此，程序设计相当于将插入各种指令变成所需要的命令。

虽然在存贮器中增加或增多存贮行列数的程序很容易插入，但将来执行复杂数据作的能力仍然多，以至需要很复杂的插件或硬件设计。

正如我们所知，考虑到目前已有的计算机研制的重大技术进展，插件一被废弃就用安放在存贮器中的计算机指令概念（完全是数据）来代替。这就是通过分配存贮器部分程序的一些变量以及存贮器部分指令译码器寄存器的通道数来完成。例如，一个简单的四位计算机，可以容纳从存贮器取来的数，将这些数相加，並將结果送回存贮器保存，增加数，便停机。指令译码逻辑可以把二进制数译码来完成以下各种操作：

0001 取数

0010 加

0100 存数

1000 乘

0000 停机

将两个数相加,再与第3个数相乘的程序,可表示如下:

0001 由存贮器取一个数

0010 将一个数与它相加

1000 与第三个数相乘

0100 存贮计算结果

0000 停机

这是简单化的例子,假定数是存贮在存贮器中,用若干预定的指令,使计算机知道对这些数进行自动运算。实际上,某些通用计算机可能规定四位访问地址号码,这些地址只不过是插述存贮单元号的一种有顺序的二进制数。在存贮器中,字的总编号通常是由0编到高。

在我们所列举的简单的一些计算机,我们便认为在程序

十进数 地 址	二进数 地 址	指令	地址	含 义
0	0000	0001	0101	从0101(5)单元取数
1	0001	0010	0110	0110单元内容与0101单元 内容相加
2	0010	1000	0111	相加所得结果乘以0111单 元内容
3	0011	0100	1000	将结果存放在1000号单元
4	0100	0000	0000	停机
5	0101	xxxx	xxxx	
6	0110	yyyy	yyyy	
7	0111	zzzz	zzzz	
8	0000	0000	0000	

中涉及到诸如5、6、7、8号地址。

显然，连以上程序都是麻烦的，由于这个缘故，在最初的程序中间，为计算机编写的是把简单的字母符号翻译成二进制数这样一种程序。象这样一种程序叫做汇编程序。汇编程序执行两种主要操作：（a）把各种计算机指令翻译成二进制数，（b）允许用用户选择的名称而不是用绝对的二进制地址去访问各种变量。因此，上述程序可以更简单地用汇编语言分出如下：

```
FET  N1  将N1地址的内容取出。
ADD  N2  将N2的内容与N1的内容相加。
MUL  N3  将N1与N2内容相加所得之值乘以N3
        的内容。
STO  N4  将结果存放在N4中。
STOP    停机。
N1, xxxx
N2, yyyy
N3, zzzz
N4, 0
```

因为汇编语言是小型计算机的主要支撑，虽然它的指令基本是冗长的，但它提供了最高的速度和多功能性。

大多数科应用读者可能更熟悉所谓“高级语言”用高级语言可以把上述表示为：

$$Y = M * X + B$$

或

$$N4 = (N1 + N2) * N3$$

这种表达式是用一个名叫编译程序的十分复杂的程序经过若干步翻译成二进制数机器语言。通常，我们可通过经验

法把编译程序与汇编程序区别开来，汇编程序是对各个汇编语言语句产生二进制机器指令，而编译程序则对各高级语言语句产生许多这样那样的指令。

对于数据收集和处理的计算机程序多半是用汇编语言编写。为什么必须用汇编语言编程序，有许多原因。首先，用最通用的高级语言例如FORTRAN和BASIC不容易有效地产生脉冲或者不容易收集和显示数据点。这样一些特点需要增加许多制造厂的BASIC解释，但是通常只做相当缓慢而低效的数据收集用。其次，大多数收集数据的专用语言简直不容易对数据进行处理，因此，即使可能比较容易进行数据收集，但是，例如付里叫叶交换计算以及相位校正就几乎是不可能的。其三，要求数据收集和处理两者都容易的混合语言总是受到语言设计者予想范围的限制。混合语言通常也是很缓慢的。最后，通常由于容易对计算机字的特殊位进行操作，实验室某方面的数据处理结果证明用汇编语言处理更为有效。