

微程序设计入门

[美] H. 卡赞 著

科学出版社

73.87221
170

微程序设计入门

〔美〕H. 卡赞 著

吴时霖 张根度 陈炳从 译

JS101/16

科学出版社



内 容 简 介

这是一本关于微程序设计的入门书。作者在书中简明地介绍了微程序设计的概念和方法以及仿真技术的基本概念和方法。全书共分十章：微程序设计的概念；计算机基础；微程序计算机的组织；微程序操作；仿真入门；翻译程序；模拟程序；仿真原理 I：算术运算；仿真原理 II：解释程序的设计；微程序设计的其他问题。每章后都有讨论题和习题，以供读者进一步理解和掌握基本概念与方法。

本书适宜于计算机专业学习使用。可供计算机设计人员、微型机工作人员、研究微程序设计的软、硬件工作者及大学有关专业高年级学生学习参考。

Harry Katzan Jr.
MICROPROGRAMMING PRIMER
McGraw-Hill, Inc. 1977

微 程 序 设 计 入 门

[美] H. 卡赞 著

吴时霖 张根度 陈炳从 译

责任编辑 那莉莉 李 立

科学出版社出版

北京朝阳门内大街 137 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1983 年 7 月 第 一 版	开本：850×1168 1/32
1983 年 7 月 第一次印刷	印张：9
印数：0001—18,650	字数：229,000

统一书号：15031·508

本社书号：3155·15—8

定 价：1.70 元

译者序

本书是 H. 卡赞教授的专著。

迄今为止,各种不同型号的计算机已大量投入应用,在应用中积累了许多丰富的软件资料。面对这种实际情况,人们在研制新的计算机系统时,就不能只是简单地淘汰老系统、老设备,装备新系统、新设备,而往往要求新系统也能运行老系统的应用程序。这就是说,往往要求新、老系统之间具有“程序兼容性”。十多年来的实践已经证实:微程序设计是解决这种程序兼容性的主要手段。这种手段的描述性术语就是微程序仿真。Burroughs 公司著名的 D 机器实际上已是一种仿真机,具有一定程度的通用仿真的能力。D 机器采用了毫微程序设计概念,具有较高的微程序设计技巧。H. 卡赞教授以 D 机器的微程序设计方法为实际背景,针对初学微程序设计和仿真技术的读者,写了这本书,想达到如下三个目标:

1. 使读者初步掌握微程序设计的基本概念;
2. 教会读者书写微程序;
3. 使读者初步掌握微程序仿真的基本方法。

译者认为,以上三个目标对我国计算机界的许多读者可能也是适用的。如果这本译著能在上述三方面取得效果,就是译者的希望所在了。

中国科学院计算所的惠毓明同志审阅了本书的译稿,在此表示衷心的感谢。

译者 于北京

• • •

序 言

本书的目的是简明地介绍微程序设计和仿真技术,并说明程序、计算机和实现方法之间的关系。我们假定大多数读者通过学习计算机课程或工作实践,对计算机、程序设计语言、操作系统、语言编译程序和程序设计等基本概念已有所了解。显然,对于大多数学过计算机结构这门课程的读者来说阅读本书将更容易,虽然这门课程的知识并不是必要的。

我们打算向读者介绍三方面的知识:(1)介绍微程序设计概念;(2)掌握编制微程序的实际方法;(3)了解仿真原理。这些知识将通过引导性的材料、实际例子以及翻译程序和模拟程序系统介绍给读者,而这些系统则使学生可以在任何一台配备有FORTRAN语言的计算机上编写并执行微程序。

本书的组织将使得编写微程序过程中所必须解决的问题按先后次序罗列出来了,其中也吸收了一些较好的学术观点和与此有关的实际经验。我们将首先向读者介绍微程序设计概念以及它与现代计算机系统结构的关系;其次,介绍仿真概念,微程序设计语言和可编微程序计算机。从而使学生能利用这些工具掌握编写微程序的方法;最后,介绍仿真原理,使学生能运用这些方法去设计和研制实际的系统。

这里所用的翻译程序和模拟程序是用标准 FORTRAN IV 写成的,这些程序能在任何一台配备有 FORTRAN IV 的计算机上执行。根据实际经验,编写这样的程序用不了一天的时间。它们已收录在 McGraw-Hill 出版社出版的教师手册中了。

非常感谢 E. W. Reigel 的帮助,是他引进了D机器的概念。也要感谢 J. T. Lynch 的支持。他们二位都是 Burroughs 公司的成员。Pratt 学院毕业的助手 Sheldon Orloff 和 Myron Sagall 在翻

译程序和模拟程序方面作了辛勤的努力。Calldata 系统的 Hall Robins 和 P.S. Young III 给予了密切的配合和技术支援，而整个工作都得到我的妻子 Margaret 的帮助。

H. 卡赞

目 录

序言

第一章 微程序设计的概念	1
1-1 现代化技术	1
1-1-1 黑箱概念	1
1-1-2 常规计算机的设计及其进展	1
1-1-3 现代计算机的设计及其进展	2
1-2 微程序设计的概念	2
1-2-1 微程序控制	3
1-2-2 微程序存储器	4
1-2-3 基本硬件的功能	4
1-2-4 微程序设计的简要历史	5
1-3 微程序系统举例	6
1-3-1 假想机器	6
1-3-2 微程序功能	8
1-3-3 机器操作	9
1-3-4 注意事项	10
1-4 微程序设计的应用	10
1-4-1 通用计算机	11
1-4-2 控制设备	13
1-4-3 专门应用	13
词汇	13
提问	14
习题	15
第二章 计算机基础	16
2-1 系统构造	16
2-1-1 计算机组织与计算机结构	16
2-1-2 计算机系统概貌	17
2-1-3 数据的表示	18
2-1-4 字与字节	19
2-1-5 字符、逻辑值与数值	20

2-1-6 计算机指令	23
2-2 处理机	25
2-2-1 寄存器	25
2-2-2 机器寄存器的实现方法	30
2-2-3 有效地址	31
2-2-4 处理机的操作	32
2-3 表示方法和术语	34
2-3-1 描述性的表示法	35
2-3-2 微程序设计术语	37
2-4 主存贮器	38
2-4-1 存取宽度	38
2-4-2 主存贮器的操作	39
2-4-3 存贮器变换	40
2-4-4 对界	41
词汇	42
提问	43
习题	44
第三章 微程序计算机的组织	45
3-1 引言	45
3-1-1 实现方法	45
3-1-2 垂直型与水平型微程序设计	45
3-2 系统构造	46
3-2-1 D 机器的组织	46
3-2-2 控制存贮器	48
3-3 微指令	48
3-3-1 分割指令的概念	48
3-3-2 机器周期	49
3-3-3 指令	50
3-3-4 关于 D 机器说明的注释	53
3-4 逻辑部件	53
3-4-1 加法器	54
3-4-2 A 寄存器	56
3-4-3 B 寄存器	57
3-4-4 环移开关	60
3-4-5 存贮器信息寄存器	60
3-4-6 关于逻辑部件操作的注释	61

3-5 存储器控制部件	61
3-5-1 微程序的寻址	61
3-5-2 S 存储器的寻址	62
3-5-3 计数寄存器	65
3-5-4 文字寄存器	67
3-6 控制器	67
3-6-1 移位系统	68
3-6-2 条件系统	70
3-6-3 命令系统	73
3-7 结束语	75
词汇	75
提问	77
习题	78
第四章 微程序操作	79
4-1 概述	79
4-1-1 程序结构	79
4-1-2 语句标号	81
4-1-3 效率	85
4-1-4 求补操作与逻辑“非”	85
4-2 逻辑部件的操作	86
4-2-1 逻辑操作	86
4-2-2 算术操作	87
4-2-3 条件	91
4-2-4 移位操作	94
4-2-5 目标描述	95
4-2-6 下一条命令控制	96
4-3 微程序设计方法	96
4-3-1 常数值的产生	96
4-3-2 交换寄存器中的数据	101
4-3-3 比较操作	102
4-3-4 循环	103
4-3-5 移位	105
4-3-6 其他技术	109
词汇	110
提问	111
习题	111

第五章 仿真入门	113
5-1 任务	113
5-2 有关仿真的微程序设计技术	113
5-2-1 基址寄存器	113
5-2-2 S 存储器的读和写操作	115
5-2-3 文字表	118
5-3 一个简单的仿真程序的设计	122
5-3-1 S 机器的特性	122
5-3-2 S 机器映象	122
5-3-3 仿真程序的微程序	123
5-4 仿真程序的微程序的分析	125
5-4-1 目录表	125
5-4-2 取指令子程序	126
5-4-3 取数子程序	128
5-4-4 存数子程序	128
5-4-5 转移子程序	128
5-4-6 测试零子程序	129
5-4-7 测试最低位子程序	129
5-4-8 减量和增量子程序	130
5-4-9 移位子程序	131
词汇	131
提问	132
习题	132
第六章 翻译程序	133
6-1 概述	133
6-2 微程序设计语言	134
6-2-1 Barkus Naur 范式	134
6-2-2 参考语言	135
6-2-3 TRANSLANG 语言	138
6-3 执行翻译程序	144
6-3-1 处理说明	144
6-3-2 运行实例	145
6-4 十六进制微编码	148
6-4-1 文字赋值语句的刻划	148
6-4-2 毫微指令语句的刻划	150
词汇	154

提问	154
习题	155
第七章 模拟程序	156
7-1 概述	156
7-2 运行模拟程序	156
7-2-1 微程序执行的例子	158
7-2-2 文件名	159
7-2-3 输出格式	161
7-2-4 S 存储器的输入	161
7-2-5 起始地址	162
7-2-6 模拟的最大时钟数	162
7-2-7 两个输出点之间的时钟数	162
7-2-8 输出行	163
7-2-9 输出说明表	163
7-2-10 S 存储器的装入	164
7-2-11 S 存储器的输出	165
7-3 在微程序执行期间的打印输出	165
7-3-1 时钟的一般描述	166
7-3-2 程序控制行	167
7-3-3 逻辑部件寄存器行	171
7-3-4 第一控制寄存器行	172
7-3-5 第二控制寄存器行	173
7-3-6 条件行	174
词汇	175
提问	176
习题	176
第八章 仿真原理 I: 算术运算	179
8-1 概述	179
8-2 定点运算——补码运算	179
8-2-1 加法和减法	179
8-2-2 乘法	181
8-2-3 除法	189
8-3 定点算术运算——原码表示法	194
8-3-1 加法和减法	194
8-3-2 乘法和除法	199
8-4 浮点运算	199

8-4-1 表示法	200
8-4-2 基本浮点算法	202
8-4-3 加法微程序	205
8-4-4 乘法微程序	210
词汇	215
提问	215
习题	216
问题	216
第九章 仿真原理 II: 解释程序的设计	218
9-1 概述	218
9-2 设计考虑	218
9-2-1 计算机的操作原理	218
9-2-2 机器寄存器	219
9-2-3 S 指令格式	219
9-2-4 S 机器操作	220
9-3 堆栈机器的设计和实现	220
9-3-1 波兰表示法	221
9-3-2 指令格式和机器操作	222
9-3-3 机器寄存器的实现	223
9-3-4 操作上的约定	223
9-3-5 堆栈机器的解释程序	223
9-3-6 运行实例	232
9-4 多寄存器计算机的设计和实现	237
9-4-1 指令格式和机器操作	237
9-4-2 机器寄存器的实现	238
9-4-3 操作上的约定	238
9-4-4 多寄存器机器的解释程序	239
9-4-5 运行实例	247
词汇	249
提问	250
习题	250
问题	251
第十章 微程序设计的其他问题	252
10-1 概述	252
10-2 定时	252

10-2-1 时钟脉冲	252
10-2-2 指令相位的描述	253
10-2-3 定时的例子	255
10-3 微程序设计的一些特殊问题	258
10-3-1 动态地址变换	259
10-3-2 表操作	261
10-3-3 矩阵操作	263
10-3-4 向量运算	264
10-3-5 图灵机器	265
参考文献	268
关键字词汇表	269

第一章 微程序设计的概念

1-1 现代化技术

现代社会正经历着一场电子革命，这场革命实际上已影响到计算技术、信息科学和通信科学的所有领域和日常生活的许多方面。所以我们能有诸如计算器、精致的收音机、电视机、录音机、监视器与监听器以及品种多样化的通信与控制设备；也正是因为这个原因，电子设备在商业、教育与政府管理部门中得到了大量的使用。

1-1-1 黑箱^{*}概念

如同开汽车一样，对许多其他电子设备或产品来说，为了有效地使用它们，并不一定需要有专门的电子学知识，而这一事实本身把我们引向所谓功能块概念。在功能块中，人们关心的主要是系统的输入、输出以及系统所完成的功能。也就是说，对于仅仅使用功能块的系统设计师、系统工程师或系统分析师来说可以不需要知道功能块内部的具体结构。

1-1-2 常规计算机的设计及其进展

对于计算机的用户来说，计算机可以看作是一个功能块，其理由很简单。因为大多数计算机用户既不是计算机工程师，也不是计算机科学工作者，他们往往是其他领域中的专业人员。事实上，计算机之所以能得到广泛的应用，最重要的一个因素就是没有专门电子学知识的人，也能有效地使用计算机。

^{*} 黑箱即通常所说的功能块。——译者注

引入功能块概念以后,使得一台计算机可通过它的指令系统、指令和数据格式、数据通路的特性以及系统结构来描述。对于常规计算机,一般认为:当将计算机中各部件组织好以后,计算机的体系结构也就确定了。第一代和第二代计算机如果不是全部、至少也是大多数是用这种方式组织起来的,而近来大部分微型机、小型机以及超高速计算机也都是如此。总之,当一台计算机组装好以后,其功能结构就由那些相互连接起来的硬件确定了;除非修改硬件,否则功能是不会改变的。

1-1-3 现代计算机的设计及其进展

对现代大、中、小型计算机的设计,要求比常规硬连部件的计算机具有更大的灵活性。无论是对计算机厂家还是用户,在考虑到经济上与技术上的原因后都希望能有相同体系结构的计算机系列,以便为用户提供不断增长的能力而又无需重编程序。也就是说,在更新计算机时,要求原来的程序能在新机器上运行。而微程序设计技术恰恰能提供这种灵活性。

因此,微程序设计的主要优点是:它使一个指令系统能在几种不同型号的计算机上实现^{*)},而在一台计算机上又能实现多个指令系统。IBM360/370 计算机系列具有前一种优点。具有后一种优点的系统则是通过在许多现代计算机中所用的兼容性特点达到的。

1-2 微程序设计的概念

通常,一个计算机系统在处理部件可以看作由两类线路组成:即数据通路和控制线路。数据通路包括代码线与存贮代码的元件数字信号在这些代码线与存贮元件中传送和保存。一般说来,数据通过算术逻辑部件后,它的值就改变了。控制线路用来解释

^{*)} 例如,由于微程序设计的优越性,使得同一个系列中的大、中、小型机都能实现同一个庞大的指令系统。——译者注

计算机指令,并在数据通路网络中规定适当的数据传送线路。

1-2-1 微程序控制

在处理部件中,控制线路是一步一步进行操作的,它很像一般计算机程序中的指令。对计算机系统中控制器进行程序设计的过程,就是所谓微程序设计。由于在机器语言的程序设计¹⁾与微程序设计之间存在某种相似性,因此,机器语言程序与微程序也有某种相似性。从概念上说,它们之间的关系如图 1-1 所示。算法是作为面向过程语言或汇编语言中的源程序描述的,然后通过汇编编译程序或其他编译程序将源程序翻译成机器语言程序^{*)}。程序中机器指令的执行是通过执行微程序来实现的,这个微程序执行由该指令所规定的操作功能。

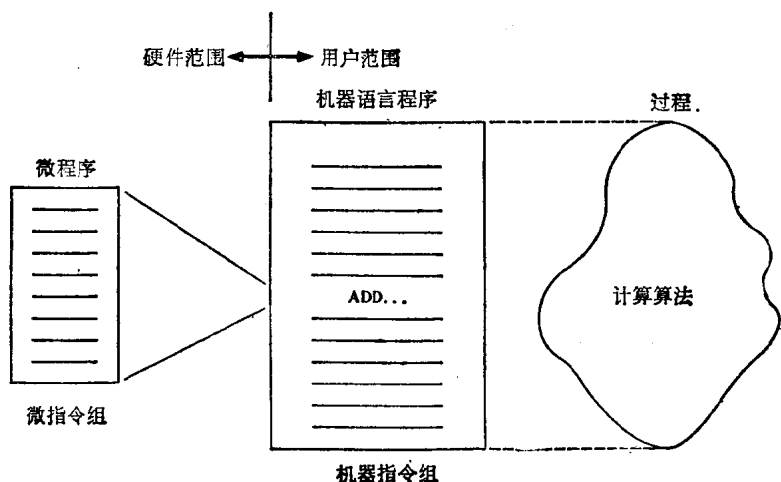


图 1-1 机器语言指令与微指令之间相互关系的概念图

1) 汇编语言程序设计与机器语言程序设计这两个术语时常用作同一意思，它们都与机器级程序有关,汇编语言是机器语言的符号形式。

*) 原文误为“机器语言指令”。——译者注

然而,这种相似性是有局限性的。因为对于每条指令,并不是都有一组微指令与之对应^{*}。在把面向过程语言中的一个语句翻译成一组机器指令时,这些指令是在语句对语句的基础上产生的。而在微程序级,微程序是通过译出每条指令,进而对其进行解释来执行该机器指令的操作功能的。因此,在操作上,微程序类似于语言解释程序。

1-2-2 微程序存贮器

在执行微程序期间,微程序存放在高速的控制存贮器中,这种控制存贮器通常有两种存取方式,即只读型与可写型。对一台计算机来说,只读存贮器(ROS)的内容是固定的,它不能在微程序控制下进行修改。修改只读存贮器的内容相当于修改计算机,虽然,修改它要比修改硬连控制部件更容易些。对一台计算机来说,可写控制存贮器(WCS)的内容也是固定的,但是它可以在微程序控制下进行修改。因此,有了可写控制存贮器,就可以动态地改变计算机的功能结构。而在微程序控制下进行操作的计算机中,常常既包含有只读型也包含有可写型控制存贮器。

通常,一台计算机系统的微程序是由计算机厂家研制的,用户不能改变它。当把可写控制存贮器加进计算机系统后,由于它允许改变操作方式而提高了系统的操作能力。研制微程序的厂家,通过指令和数据格式以及指令系统从用户角度给出了计算机的外特性,而不规定实现措施,这就使用户也能修改通用计算机系统中的微程序。

市场上出售的、用户可编微程序计算机一般都是微型机或小型机,它们是专用的或教育上用的可编微程序计算机。

1-2-3 基本硬件的功能

一台可编微程序计算机由一些基本的硬件组成。这些硬件包

^{*} 因为机器语言程序中相同的指令,而相同的指令则公用一组微指令。——译者注