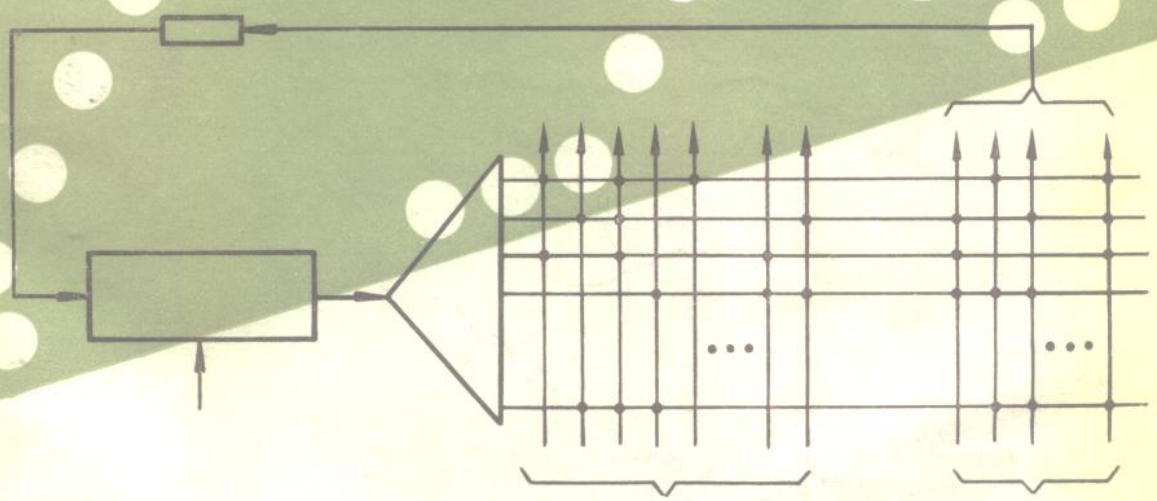
电子计算机
微程序设计技术

陈炳从 编著



国防工业出版社

电 子 计 算 机

微程序设计技术

陈 炳 从 编著

内 容 简 介

本书全面地阐述了微程序设计技术的概念、原理、方法和具体应用。从实用角度出发，讨论了十多种微指令的编译法和微程序的两大类顺序控制方式。从系统、逻辑设计的角度出发，分析了串行、并行微程序设计，毫微程序设计，以及动态微程序设计，介绍了它们的具体实现方法。讨论了微程序设计语言的发展概况，并重点阐明了框图语言、CDL语言和水平型微汇编语言的具体应用。讨论了各种微程序控制存贮器的现状和具体应用。介绍了微诊断技术及其具体应用。从计算机内部操作的本质特征出发，介绍了微程序设计的具体方法。从用户的角度出发，结合运用水平型微汇编语言，介绍了微程序仿真技术，等等。最后，对微程序设计技术的优、缺点及其发展前景，作了一定的评价。

本书可供具有计算机基本原理知识的读者作为自学微程序设计技术的入门书。也可供大、专院校计算机科学系师生以及从事计算机的设计、生产和使用的专业人员参考。

电子计算机
微程序设计技术
陈 炳 从 编著
*
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
*
787×1092 1/16　印张28　650千字
1981年7月第一版　1984年11月第二次印刷　印数：8,201-25,200册
统一书号：15034·2145　　定价：2.85元

前　言

在计算机系统中，微程序设计技术是介于系统硬件技术和系统软件技术之间的一门重要技术，有“固件技术”之称。十多年来，这门技术已经越来越广泛地、深入地应用于计算机的设计、使用和维护等领域，越来越广泛地应用于其它的数字系统。微程序设计技术同大规模、超大规模集成电路相结合，还可能会对今后计算机的系统结构产生深远的影响。

针对具有计算机原理知识的读者自学微程序设计技术的需要，本书由浅入深，全面地、严谨地阐述了微程序设计技术的概念、原理、方法和应用；针对从事于计算机的设计、生产和使用等专业技术人员的需要，本书在写作上区分技术问题的类型，着重提供尽量多的、有实用价值的微程序设计技巧和方法。此外，本书始终注意从计算机的系统概念出发，来阐述微程序设计技术的各种主要问题，使读者在学完本书的内容之后，也能在一定程度上掌握计算机系统的概念。为此，本书特别从计算机内部操作的最本质的特征（例如第八章）和从用户的观点（例如第十三章）这两方面，来阐述这门技术的概念、原理和应用。

本书是在陈力为同志的指导下写作的。

赵云彪、张根度和王敬治同志分别协助提供第十一、十三和第十二章的初稿，最后由作者统一改编和定稿。于绍镛同志协助提供第七章稿件。李基义等同志帮助审阅了大部分章节。赵云彪等同志还协助做了其它一些工作。在写作过程中，曾经得到有关领导部门的鼓励和支持，得到一些同志的帮助。在此谨表示衷心感谢。

限于经验和水平，特别是写作时间十分短促，本书难免存在着缺点和错误，敬请读者批评指正，以便将来再版时进一步修改完善。

作者　1978年

目　录

第一章　绪论 …… 1

1.1　引言 …… 1

1.2　微程序设计技术发展的几个阶段 …… 2

1.2.1　确立概念的阶段（1951～1964年）…… 2

1.2.2　普及应用的阶段（1964～1970年）…… 3

1.2.3　飞速发展的阶段（1970～19××年）…… 5

1.3　电子计算机的数据通路和控制逻辑 …… 9

1.3.1　电子计算机的基本结构和数据通路 …… 9

1.3.2　繁琐复杂的硬件组合控制逻辑 …… 11

1.3.3　直观规整的存贮控制逻辑 …… 11

1.3.4　两种数据通路的基本一致性 …… 12

1.4　电子计算机中信息传送的基本原理 …… 13

1.4.1　控制信息传送 …… 13

1.4.2　控制信息传送的先后顺序 …… 15

1.4.3　控制信息传送之间的相互联系 …… 17

1.5　微程序设计的基本概念 …… 17

1.5.1　微命令和微命令系统 …… 17

1.5.2　微操作和微周期 …… 18

1.5.3　微指令 …… 18

1.5.4　微程序和微程序设计 …… 19

1.5.5　微程序计算机的基本工作原理 …… 20

第二章　微指令的编译法 …… 23

2.1　引言 …… 23

2.2　微指令的几种基本编译法 …… 24

2.2.1　不译法 …… 24

2.2.2　最短编译法 …… 25

2.2.3　字段直接编译法 …… 26

2.2.4　字段间接编译法 …… 28

2.2.5　字段重叠编译法 …… 29

2.2.6　常数源字段 E …… 30

2.2.7　可解释的字段编译法 …… 31

2.3　微指令的分类编译法 …… 31

2.3.1　适用于中型机的分类编译法 …… 31

2.3.2　适用于小型机的分类编译法 …… 32

2.3.3　适用于微处理机的分类编译法 …… 35

2.4　微指令的其它编译法 …… 37

2.4.1　长短周微指令编译法 …… 37

2.4.2　插入微指令编译法 …… 38

2.4.3 双顺序控制编译法……39
第三章 微程序的顺序控制……42
3.1 引言……42
3.2 后继微地址的增量方式……43
3.2.1 什么是增量方式……43
3.2.2 “计数器”方式……43
3.2.3 “计数器·转移地址”方式……44
3.2.4 “计数器·转移地址·堆栈”方式……48
3.3 后继微地址的断定方式……52
3.3.1 什么是断定方式……52
3.3.2 后继微地址的来源……53
3.3.3 后继微地址非因变分量的断定方式……56
3.3.4 后继微地址因变分量的断定方式……57
3.3.5 微子程序和微程序循环……64
3.4 其它的顺序控制方式……67
3.5 微中断系统……69
3.5.1 微中断源……69
3.5.2 微中断处理微程序的入口……70
3.5.3 微中断处理微程序的出口……71
第四章 微指令格式和实例……72
4.1 引言……72
4.2 水平微指令……72
4.2.1 水平微指令的基本概念……72
4.2.2 水平微指令的特点……73
4.2.3 水平微指令的缺点……74
4.2.4 水平微指令的实例……74
4.3 垂直微指令……79
4.3.1 垂直微指令的基本概念……79
4.3.2 垂直微指令的特点……79
4.3.3 垂直微指令的缺点……80
4.3.4 垂直微指令的实例……80
4.4 混合型微指令实例……93
4.4.1 RCA Spectra 70/45计算机的数据通路……94
4.4.2 微指令格式……95
第五章 微程序设计分析……100
5.1 引言……100
5.2 串行微程序设计……103
5.2.1 基本概念和特点……103
5.2.2 速度分析……105
5.2.3 提高速度的途径……105
5.3 并行微程序设计……105

5.3.1 基本概念和特点 …… 105
5.3.2 微周期不重叠的并行微程序设计 …… 106
5.3.3 微周期相重叠的并行微程序设计 …… 108
5.3.4 快速微程序转移 …… 110
5.4 毫微程序设计 …… 114
5.4.1 毫微程序设计的基本原理 …… 114
5.4.2 ROM-V 中的垂直微程序 …… 114
5.4.3 ROM-H 中的毫微程序 …… 116
5.4.4 毫微程序设计的顺序控制 …… 116
5.4.5 关于毫微程序设计的评价 …… 118
5.5 动态微程序设计 …… 119
5.5.1 动态微程序设计的类型 …… 120
5.5.2 控制存贮器的结构 …… 121
第六章 微程序设计语言 …… 123
6.1 引言 …… 123
6.2 微程序流程图 …… 127
6.3 水平型微程序设计语言 …… 128
6.3.1 微汇编语言 …… 128
6.3.2 框图语言 …… 130
6.4 硬件描述语言 …… 137
6.4.1 CDL 语言 …… 137
6.4.2 用 CDL 语言描述计算机操作 …… 144
6.5 微程序设计自动化的一般过程 …… 148
第七章 微程序控制存贮器 …… 150
7.1 引言 …… 150
7.2 ROM 的一般原理和发展概况 …… 150
7.2.1 ROM 的逻辑结构格式 …… 150
7.2.2 ROM 的主要性能指标 …… 152
7.2.3 ROM 的各种存贮单元 …… 153
7.2.4 几种典型系统中的 ROM …… 156
7.3 半导体大规模集成电路只读存贮器 …… 156
7.3.1 双极熔丝型 PROM …… 157
7.3.2 浮栅雪崩注入 FAMOS (EPROM) …… 160
7.3.3 MNOS 型 EAROM …… 161
7.3.4 玻璃半导体 EPROM (Amorphous) …… 163
7.3.5 大规模集成只读存贮器典型产品 …… 164
7.4 ROM 在其它方面的应用 …… 166
7.5 ROM 的测试和检验 …… 167
第八章 微程序在系列机中的应用 …… 171
8.1 引言 …… 171
8.2 DJS 200 系列机简介 …… 172

8.2.1 DJS200系列机的特点……172
8.2.2 若干系统概念……175
8.2.3 指令系统……180
8.2.4 数据格式……183
8.3 DJS 200-20 型的微程序设计……184
8.3.1 主要性能……184
8.3.2 数据通路……187
8.3.3 微程序控制器……204
8.3.4 微程序设计实例……233
8.4 DJS 200-60 型的微程序设计……284
8.4.1 对运算器三个指令站操作的控制……287
8.4.2 对微程序的执行顺序的控制……289
8.5 DJS 200-40 型的微程序设计……289
第九章 微程序在微处理机中的应用……295
9.1 引言……295
9.2 微处理机和微型机的结构……296
9.2.1 基本结构……296
9.2.2 位片结构……300
9.3 微程序设计……301
9.3.1 指令和程序……301
9.3.2 微指令格式……302
9.3.3 微程序设计示例……304
9.3.4 关于微程序设计的说明……306
9.4 可编程序逻辑阵列与微处理机……307
第十章 微程序在巨型机中的应用……310
10.1 引言……310
10.2 微程序在ILLIAC Ⅳ中的应用……310
10.2.1 ILLIAC Ⅳ的阵列结构……310
10.2.2 控制器 CU……312
10.2.3 指令终点站微程序控制的原理……313
10.2.4 ILLIAC Ⅳ微程序控制的特点……315
10.2.5 ILLIAC Ⅳ采用微程序控制的理由……317
第十一章 微诊断……319
11.1 引言……319
11.2 微诊断原理……320
11.2.1 独立编制微诊断程序的原理……320
11.2.2 借用 ROM 中固有微程序的原理……329
11.3 DJS 200-20 型的微诊断……331
11.3.1 设计目标和要求……331
11.3.2 微诊断的数据通路和硬核……331
11.3.3 故障字典与故障定位……333

11.3.4 微诊断程序的编制 ……336
11.3.5 微诊断程序实例 ……337
第十二章 微程序在外围控制中的应用 ……346
12.1 引言 ……346
12.2 DJS 200 系列机磁带控制器的微程序设计 ……347
12.2.1 磁带控制器的基本性能 ……347
12.2.2 磁带控制器能执行的命令 ……348
12.2.3 磁带控制器的基本通路 ……350
12.2.4 磁带控制器的时序 ……351
12.2.5 磁带控制器的微指令格式 ……354
12.2.6 磁带控制器微程序设计实例 ……354
第十三章 微程序仿真 ……359
13.1 引言 ……359
13.2 微程序仿真的基本过程 ……360
13.2.1 仿真微程序的调入和读出 ……360
13.2.2 用户扩充指令的仿真过程 ……364
13.2.3 系统仿真过程 ……368
13.2.4 中断仿真过程 ……370
13.3 假想目标机和宿主机 ……373
13.3.1 假想目标机 ……373
13.3.2 假想宿主机 ……376
13.3.3 目标机到宿主机上的映象 ……395
13.3.4 目标机与宿主机的相配 ……396
13.4 信息区的抽取和分离 ……397
13.4.1 字段屏蔽和多路转移 ……397
13.4.2 逻辑乘和“执行”微指令 ……400
13.5 微地址堆栈与微子程序的调用 ……403
13.5.1 微地址堆栈 ……403
13.5.2 微子程序的调用 ……403
13.5.3 入口通信和出口通信 ……405
13.6 仿真微程序示例 ……408
13.6.1 扩充双倍字长（定点）加法指令的仿真微程序 ……408
13.6.2 扩充十进制加法仿真微程序 ……416
13.6.3 假想目标机的中断仿真微程序 ……416
13.6.4 假想目标机的仿真微程序 ……425
第十四章 微程序设计技术评价 ……432
14.1 微程序设计技术的优缺点 ……432
14.1.1 微程序设计技术的优点 ……432
14.1.2 微程序设计技术的缺点 ……434
14.2 微程序设计技术的前景 ……434
14.2.1 计算机系统的发展 ……435
14.2.2 微软件的发展 ……436
14.2.3 计算机的动态结构 ……436
14.2.4 微程序在其它数字系统中的应用和前景 ……438

第一章 绪 论

1.1 引 言

当代社会正在经历一场伟大的科学技术革命。这场革命是以应用电子计算机和原子能等新兴技术科学为标志的。如果说十八世纪以应用蒸汽机等为标志的工业革命曾经把人的体力放大了千百万倍，那么就可以说，今天的电子计算机又把人的智力放大了千百万倍。

1946年2月15日，电气工程师普雷斯波·埃克特（J·Prespen Eckert）和物理学家莫克利（J·W·Mauchly）等人发明的"电子数值积分器和计算器"ENIAC，在美国正式用于军事目的。尽管这台30吨重的庞然大物还不如现在的一台微型机，但是，它是世界上第一台电子计算机，起了开辟新的科学技术领域的作用。由于ENIAC的主存贮器仅仅由二十个寄存器组成，所以它不是真正存贮程序式的电子计算机。德国著名数学家冯·诺依曼（J·Von·Neumamm，1903～1957年）曾经参观过ENIAC的研制工作。1945年，冯·诺依曼发表了著名论文"非连续变量电子自动计算机"EDVAC，第一次提出了"存贮程序"概念，并且应用了二进制原理。根据这种概念和原理，冯·诺依曼等人研制了一台称为"IAS"的电子计算机。尽管IAS后来未能付诸实用，但是它却真正奠定了尔后电子计算机概念和原理的基础。这些概念和原理主要有：

（1）采用存贮程序概念，应用二进制原理；

（2）程序和数据都存放在同一个存贮器中；

（3）只用一个顺序计数器（即指令地址计数器）来指示程序的执行顺序；

（4）根据地址决定一切指令和数据。

后来，人们把具有上述概念和原理的电子计算机，称为"冯·诺依曼型结构"计算机。

为了适应当代各种科技计算和信息处理领域的迫切需要，从五十年代以来，电子计算机以三十多年改换三代的高速度向前发展。在这期间，电子计算机大约每隔五至八年，速度提高十倍，体积缩小十倍，成本降低十倍。当前，计算机的应用已经遍及各个领域。

电子计算机之所以能够得到如此迅速的发展，是与计算机设计者在设计概念、原理、方法和技术上的不断标新立异分不开的。在这些标新立异中，微程序设计就是一门重要的技术。看来，微程序设计技术有发展成为一门独立学科的趋势。

本书将比较全面地叙述和讨论微程序设计技术的概念、原理、方法和具体应用。在整个叙述和讨论中，将遵循下列原则：

（1）在叙述和讨论中，将力求从简到繁、深入浅出。

例如，在叙述微程序设计的基本概念时，既不从定义出发、也不从逻辑推理出发，而是从计算机内部操作的最本质的特征（信息传送）出发的。这样，我们就能够迅速地从本质上领会和掌握微程序设计的基本概念。又如，在讨论微指令编译法时，并不立足于形式推理，而是立足于系统和逻辑设计上的实用方法。换言之，我们是从系统和逻辑设计的实

用角度来讨论微指令的各种编译法的。这样，我们就能迅速地掌握和应用多种有实用价值的微指令编译法，等等。

（2）在叙述和讨论中，将力求阐明技术问题的来龙去脉。

为了阐明微程序设计技术各个重要概念、原理和方法的来龙去脉，使我们不仅能知其然，而且能知其所以然，本书力求从计算机系统结构及其内部操作过程的角度出发，来叙述和讨论各个重要技术问题，而避免从形式上进行孤立的讨论。

（3）在叙述和讨论中，力求提供尽量多的实用方法。

本书在对微程序设计技术进行有系统的叙述和讨论的同时，力求提供尽量多的实用方法。例如：第二章提供了近十种有实用价值的微指令编译法；第三章提供了有实用价值的两大类微程序顺序控制方式和多种具体方法；第五章提供了串行、并行微程序设计和毫微程序设计的实用方法，等等。

（4）在叙述和讨论中，力求兼顾多方面的需要。

为了使本书既可以作为初学者的入门书，也可以作为各种有关技术人员和高等院校师生的参考书，在叙述和讨论中，在内容的选择上，力求兼顾多方面的需要。第一章到第六章，集中地叙述和讨论微程序设计技术的共同性问题，可供范围更广泛的有关人员参考；第七章介绍和讨论微程序控制存贮器 ROM，可供电路设计人员参考；从第八章到第十三章，分别介绍微程序设计技术在系列机、微处理机、巨型机、微诊断、外围控制和微程序仿真中的应用，其重点是系列机（包括微诊断）和微程序仿真。第八章着重从计算机内部操作的本质特征来介绍微程序在系列机中的应用。第十三章着重从用户的观点来叙述和讨论微程序仿真技术。读者可以根据需要，选读适当的章节。

最后，本书还尽可能地从一定的历史发展阶段来叙述和讨论微程序设计技术的有关问题，使我们在理解概念、原理、方法和具体应用时，也能了解：微程序设计技术是伴随着计算机的诞生而诞生的，反过来，它又促进了计算机系统的进一步的发展。

本章将着重叙述微程序设计技术发展的几个阶段和微程序设计的基本概念。

1.2 微程序设计技术发展的几个阶段

三十多年来，微程序设计技术经历了确立概念、普及应用，以及飞速发展等三个阶段。现在简要叙述如下。

1.2.1 确立概念的阶段（1951～1964 年）

在第一台电子计算机 ENIAC 诞生五年后的 1951 年，英国剑桥大学数学教研室教授威尔克斯（M. V. Wilkes），在曼彻斯特大学计算机的会议上，第一次提出了微程序设计的概念和原理。威尔克斯在题为“设计自动化计算机的最好方法”的报告[1]中指出，一条机器指令可以分割为许多更基本的操作序列，计算机的操作可以归结为信息传送，而信息传送的关键是控制门，这些门可以用存放在某种存贮阵列中的信息位来控制。因此他认为，可以用一种有规则的、类似于程序设计的方法，来设计计算机繁杂的控制逻辑，这种方法就是属于存贮控制逻辑概念的微程序设计方法。

威尔克斯还亲自指导了剑桥大学的微程序设计实践，他的方案首先在剑桥大学的微程

序设计的 EDSAC$_2$[2~4]系统中实现。威尔克斯的方案亦称“威尔克斯模型”，我们将在第三章介绍。

“威尔克斯模型”是很初始的。微程序存放在磁芯存贮器阵列之中，每条微指令的每一位都直接用来控制某个门，每条微指令还要给出一个完整的后继微地址。这样的微程序缺乏灵活的转移能力，离实用尚远。后来，威尔克斯继续参加各种研究和实践，并作了重大改进。

1951 年到 1964 年间，正是计算机经历第一代（电子管计算机）和第二代（晶体管计算机）的发展阶段。在这个阶段中，计算机设计者发明了变址器、堆栈结构、程序中断系统，同时，出现了计算机族；开始研究分时系统；初步确立了系统软件；……。这个时期的存贮器，从水银延迟线、光屏管、磁鼓（作主存用），发展到全面采用磁芯存贮器。本来，这个时期计算机技术的发展是需要微程序设计技术的，但是，这个时期并没有为实现微程序设计技术创造客观的条件。这是因为，上述存贮器价格贵、速度慢。如果把上述存贮器用作微程序控制存贮器，就体现不出微程序设计的优越性。然而，其它类型的存贮器这时还处在研究阶段。所以，这个时期的微程序设计技术，由于缺乏廉价、可行的控制存贮器而发展缓慢。只有极少数单位做了一些很原始的微程序计算机模型。比较有名的有：如上所述的 EDSAC$_2$（用 1K 容量的只读型磁芯矩阵做控制存贮器）；美国麻省工学院林肯实验室的 CG$_{24}$[5]（用 1K 容量的只读型二极管矩阵做控制存贮器）等。美国 IBM 公司是最早重视微程序设计技术的大公司之一，在 1961 年的 ACM 全国会议上，曾经提出过有关 IBM 7950 机微程序设计的论文[6]。但是，在这个时期，IBM 公司也还不能解决控制存贮器的问题。

尽管在这个时期微程序设计技术发展缓慢，但是，威尔克斯的概念已经引起人们的重视，英、美、日、意、法和苏联等国，已经逐渐有人开始从理论上对微程序设计技术进行探讨。值得指出的是，威尔克斯一开始只是注重于用规整的存贮控制逻辑，来代替繁杂的组合控制逻辑，微程序设计也只是被看作为实现硬件设计的一种手段。格兰茨（Glantz，1956 年[7]），默塞尔（Mercer，1957 年[8]），范德普尔（Vanderpoel. W. L，1960 年和 1962 年[9][10]）等人，曾先后著文，发表了通过修改或改编微程序来改变指令系统，或者给用户提供一个更大更好的指令系统的见解。这种见解颇有远见。

1.2.2 普及应用的阶段（1964～1970 年）

1964 年到 1970 年间，正是计算机经历以采用中、小规模集成电路为主要标志的第三代发展阶段。在这个阶段中，计算机技术取得了重大的进展，从而使系统硬件达到了阶段性的相对稳定，使系统软件付诸实用。在系统硬件方面的主要进展有：确立了“计算机结构”的概念；采用了通道、虚拟存贮器或虚拟机等概念和技术；开始研究并初步实践计算机网；采用了先行控制、高速缓冲存贮器、流水线、阵列式结构等技术；等等。这样一来，便使计算机结构突破了串行式的冯·诺依曼结构，大大地提高了计算机的速度和效率（例如巨型机 ILLIAC Ⅳ的速度已经突破一亿次大关）等等。在这个阶段中，计算机技术的另一方面的重大进展就是：产生了系列机；发展了仿真和模拟技术；提高了机器的可靠性、可利用率和可维护性（即 RAS 技术）。这个方面的重大进展，是导致普及应用微程序设计

技术的直接因素。而这个阶段存贮技术的进展，则是普及应用微程序设计技术的客观基础。下面具体分析一下促使微程序设计技术进入普及应用阶段的主要原因。

1．研制系列机的需要

系列机的主要标志之一，就是在系列内的各个型号的计算机都具有程序兼容性。这就要求各个型号都要遵循统一的结构格式，特别是要求实现统一的指令系统。由于系列机往往要兼顾科技计算、数据处理和过程控制等多方面的用途，所以，这个统一的指令系统也就必然设计得很庞大和复杂。面对这个统一的结构格式和指令系统，系列机中的各个型号都必须有适当的性能/价格比，这样才能确保各种型号都有生存力。例如 IBM360 系统，一开始共分 10 个型号，从小型计算机到大型计算机的性能比是 1 比 300，价格比则是 1 比 100。对于这种系统，若用常规硬件组合逻辑的设计方法，则难以确保既定的性能/价格比，若采用微程序设计技术，则能确保这个性能/价格比。这是由实践所证实了的。另一方面，微程序设计还使系列机具有如下潜力：为了适应各种用途，可以灵活地扩充性能。

2．仿真和模拟技术的需要

计算机越来越多，其设计过程也越来越复杂，这就促使人们去寻求新的途径，以便解决如下两方面的问题：

（1）希望过去的程序能在新一代的机器中运行；

（2）希望借助计算机来分析、评价系统的设计方案，判断设计的正确性，以缩短研制周期。

要解决上述两方面的问题，其主要办法就是采用仿真和模拟技术。在微程序设计的计算机中，由于微程序把指令操作步骤分割得更独立、更细小，所以用微程序来仿真或模拟，就比用程序仿真或模拟更容易、更准确、更快速。若模拟的对象是研制中的机器，则通过模拟便可以评价设计方案、判断设计的正确性、指出修改的途径，从而大大缩短研制周期。若仿真的对象是已有的计算机，则已有的计算机的程序就可以在进行仿真的机器（仿真机）中运行，即便是第二代的程序，也可以很容易地在第三代的微程序仿真机中运行。

3．RAS 技术的需要

微程序设计的计算机具有如下优点：结构规整；更加可靠；设计灵活；修改方便；学习容易；便于诊断。例如，在微程序设计的计算机中，无需增加太多的硬件就能进行微诊断。这种微诊断，能够把故障定位在很小的范围内。这些优点正是 RAS 技术所要求的。

正是由于上述三方面的需要，才导致了微程序设计技术迅速进入普及应用阶段。从 1964 年起陆续投产的 IBM360 系统的各种型号，除超大型机以外，都普遍地、成功地采用了微程序设计技术。这是微程序设计技术进入普及应用阶段的主要标志。IBM370 系统则全部采用微程序设计。1970 年，哈森（S. S. Husson）在他所著的“微程序设计——原理和实践”[11]一书中，曾经比较全面地介绍了 IBM360 系统的微程序设计。

必须指出，威尔克斯本人和在 IBM 等公司工作的许多人，都对威尔克斯最初的方案进行了许多重大改进和发展，从而才使得微程序设计技术走上了实用的轨道，迎来了普及应用的新阶段。这些改进和发展，已经反映在哈森的著作[11]中。就其基本方面而言，有如下这些：

1．确立了微指令的基本编译法

在IBM360系统中，除20型、25型等小型机采用垂直微指令以外，一般都采用水平微指令。在水平微指令中，采用了既能大大缩短微指令字长、又能达到一定执行速率的字段直接和间接编译法。此外，IBM360系统还采用了许多技巧（如40型的间接操作控制，等等）。

2．确立了微程序的顺序控制方式

一方面，IBM360系统通过采用增量方式和分段断定方式，来确定后继微地址，使现行微指令一般不必像威尔克斯最初的方案那样，为后继微指令提供一个完整的后继微地址，从而缩短了微指令的顺序控制字段的长度。另一方面，IBM360系统的微指令，普遍设置了灵活的“转移条件选择测试字段”和“转移控制字段”等，使微程序具有如下功能：能够很灵活地顺序执行；可以进行无条件转移；可以进行2路到64路的条件转移；可以方便地构成循环和微子程序；等等。这样一来，就可以大大提高微程序的执行速度。

3．初步实现微程序设计自动化

IBM360系统的微程序设计，采用了微汇编语言和框图语言，建立了“控制自动化系统”CAS，初步实现了微程序设计自动化，并开始研究高级微程序设计语言。

4．初步实践了仿真和模拟技术

5．解决了微程序控制存贮器的问题

IBM360系统普遍采用廉价可行的变压器型和静电耦合型只读存贮器，以作为微程序控制存贮器。换句话说，IBM公司解决了微程序设计技术赖以进入普及应用阶段的控制存贮器的问题。

从六十年代中期起，微程序设计的计算机已经成批成批地投入使用了。这个阶段有代表性的微程序设计的计算机，除IBM360系统外，还有RCA Spectra 70/45[11]，H 4200/8200[11]，以及MELCOM1530，等等。

1.2.3 飞速发展的阶段(1970～19××年)

从二十世纪七十年代开始，电子计算机跨进了第四代发展阶段。这一代计算机的重要标志之一就是采用大规模集成电路LSI。LSI不仅具有成本低、体积小、功耗小，以及可靠性高等多方面的优点，而且它开拓了计算机系统结构进一步创新的前景。

这个阶段的LSI存贮器，已经或者必将最后淘汰变压器型和静电耦合型等只读存贮器ROM。实践已经表明，只读型的LSI存贮器，由于它有廉价、快速、可靠和灵活等许多优点，所以它大大地促进了微程序设计技术的发展和应用。而可写可读型的LSI存贮器，还为这门技术的进一步创新和应用，提供了物质基础。采用可写可读型的控制存贮器的机器，具有可扩展性和能由用户选择机器性能的结构。未来的这种机器，甚至可望具有动态的结构。

微程序设计技术飞速发展的另一个重要方面，就是研究活动蓬勃开展。从1968年起，“IEEE计算机协会微程序设计技术委员会”（IEEE Computer Society Technical Committee on Microprogramming）和“ACM微程序设计特设专业组”（ACM Special Interest Group on Microprogramming）每年联合举办一次微程序专业会议，并出版会刊“微会议录”（Proceeding of Micro）。1970年以来，ACM还创立季刊“SIG微新消息”（SIG

Micro Newsletter)。此外，许多期刊和会议录，例如“IEEE 计算机会报”（IEEE Trasaction on Computer），“计算机设计”（Computer Design），“AFIPS 会议录”（AFIPS Conference Proceedings）等等，也都不定期地刊登了微程序设计专集或者文章。在涉及微程序设计技术的领域内，迄今为止发表过的文献已有上千篇。在这些文献中，更引人注目的课题就是：高级微程序设计语言；用固件（即微程序）来代替软件或硬件；微程序设计技术和大规模集成电路对未来计算机结构的影响；等等。

微程序设计技术飞速发展的又一个重要方面，就是开始使用高级微程序设计语言。例如：Eckhouse 的 MPL 语言；日电的 MPGS；Intel 的 PL/M；等等。

七十年代以来的实践表明，计算机系统（包括系统硬件和系统软件、应用软件）越发展，就越需要微程序设计技术。七十年代出现的应用微程序设计技术的热潮，已足已证实了这一观点。这门技术的应用之广，已经大大地超出了威尔克斯最初预想的范围。下面我们就来简单地列举一些主要应用范围。

1．在计算机系统中的应用

（1）巨型计算机和大型计算机

由于巨型机和大型机要求速度快，所以除了“取指令”、“计算操作数地址并取操作数”等控制过程暂时仍需采用组合控制逻辑以外，已经在指令的执行部件，特别是在运算控制部件中，成功地采用了微程序设计。例如，巨型机 ILLIACⅣ，超大型机 Star 100，大型机 IBM 370/168，以及日本M系列中的大型机等，都采用了微程序设计。

（2）中型计算机

中型计算机可以整机性地，或者局部性地采用微程序设计。例如，国产 DJS200 系列机中的40型计算机，就采用了既各自独立、又有相互联系的四个微程序控制器。

（3）小型计算机

小型计算机一般都已整机性地采用微程序设计，无需一一枚举。

（4）微处理机或微型计算机

七十年代大量出现的微处理机和微型机，有许多都采用微程序设计。特别是位片结构的微处理机，大多采用微程序设计。例如，LSI-11，Itel300，日电的 NEAC System100，等等。

（5）虚拟计算机

在虚拟计算机中，从虚资源到实资源的变换，过去一般是用软件（虚拟机操作系统）实现的。由于这种软件很庞大，占用主存的很大一部分容量，而且效率低，所以降低了机器的实际使用效率。如果把这种软件部分地改用固件（即微程序），就可以大大地减少程序量、节省主存容量、提高效率，从而提高机器的性能。在这方面，IBM公司的商用机 VMA（Virtual Machine Assist）即是一例。IBM370 系统的虚拟存贮器和虚拟机的操作系统，已经部分地固件化了，特别是实现了以特权指令为中心的固件化。

（6）高级语言处理机

人们从六十年代就开始研究高级语言处理机，但到七十年代才开始实现。然而，目前仍处于实践阶段。

有的高级语言处理机，需把高级语言源程序变换成中间语言程序，然后再变换成机器

语言程序。由于微程序同各种中间语言程序相近，所以高级语言处理机采用微程序设计就能取得高效率。例如，布劳斯（Burroughs）公司的商用高级语言处理机B1700，就能使高级语言源程序变换成与事务管理、科学计算或者过程控制相应的三种中间语言的一种。

有的高级语言处理机，在采用微程序设计后，还能使机器直接执行高级语言。例如Burroughs公司的B6500就是这种机器。

（7）系统软件的辅助手段

操作系统和编译程序的共同缺点就是，占用主存和外部存贮器相当大的容量，而且效率低、灵活性差。如果用固件来部分地（将来可望大部分或全部地）取代系统软件，则能大大地克服上述三大缺点。在1967年，奥普勒(A.Opler)第一次提出了用固件来沟通硬、软件的见解[13]。在1970年，美国纽约大学的沃克赫斯(Werkheise)，首先提出了操作系统固件化的设想。此后，人们陆续进一步地进行了研究和实践。众所周知的IBM370系统，已用微程序设计来辅助虚拟存贮器的管理系统。当然，应用软件的固件化也是一种趋势。总之，在第四代计算机的发展阶段中，“软件硬化”的实现，有赖于微程序设计技术。

（8）外围控制系统

七十年代，微程序设计已成为设计外围控制系统或者外围控制器的一门流行的技术。像磁带控制器、磁盘控制器、键-带控制器、通信控制器，以及各种终端(特别是智能终端)等等，在采用微程序设计后，都能简化控制、提高外围控制的通用性、灵活性，以及性能的可扩展性。

2．仿真和模拟

微程序仿真和模拟技术是当前的又一重要研究方向(尽管已经取得了很多成果)。

前面已经指出：人们希望过去的程序能在新一代的机器中运行。这不仅是省去用户重编程序的工作量的问题，而且是解决非兼容的计算机系统之间的程序兼容性的重要问题。用户总是希望新一代机器的指令系统和结构，能够包含老一代机器的指令系统和结构，以便使过去的程序能在新一代机器中运行，从而能利用新一代机器的更好的性能、设备和经济性。但是，实际上往往做不到这一点。于是，人们早就在研究解决这个问题的途径。在微程序仿真技术出现以前，大致有三个途径：

（1）重编程序——这确实太浪费人力物力了；

（2）用新一代机器模拟老的机器——这样做效率太低了；

（3）把老的程序翻译成新的程序——如果新、老机器的结构格式相差较大，则这种翻译的繁琐程度是不言而喻的。

有了微程序仿真技术之后，利用微程序设计来实现非兼容系统的程序兼容性，有下述一些优越性：

（1）不需要重编程序。这是因为，微程序仿真机能够直接运行各种用户程序；

（2）只要用户有一台比较通用的微程序仿真机，那么它就可以仿真其它机器；

（3）微程序仿真的效率高。

微程序仿真从IBM360系统开始，就陆续见诸实例了。DEC公司的PDP-11/60系统在有用户微程序设计支持的情况下，能仿真IBM370系统。标准计算机公司的MLP-900能

仿真许多已有的和正在研制中的计算机。GPMS，EMMY、B1700、QM-1，以及D机器等等，也都是仿真机。

前面还曾经指出：人们希望借助计算机来分析、评价系统的设计方案，判断设计的正确性，以缩短研制周期。这就需要使用模拟手段。过去的模拟手段，是靠已有机器的机器语言程序，像IBM360的控制自动系统CAS就属于这一类。现在的另一种手段是靠已有机器的微程序来模拟研制中的机器的微程序和结构。由于微指令能直接访问计算机的各种寄存器，由于微程序的效率比程序的效率高，所以用微程序进行的模拟就能达到高效率。像IBM公司和Allen-Babock公司共同研制的会话程序设计系统CPS，PDP-11/40E的模拟器，都是实用的例子。不仅硬件能被模拟，而且软件也能被模拟。用微程序设计来模拟研制中的软件，实际上就等于机器尚未研制好之前，就在进行软件调试了。

3．RAS技术

微程序设计促进了RAS(可靠性、可利用率、可维护性)技术的发展。例如，微程序控制器可利用奇偶校验或冗余码校验（一般能自动纠正单错）来提高可靠性；微程序计算机的规整结构，使得数据通路的校验系统可以进一步完善；微程序设计的灵活性，使得指令复执和微指令复执得以实现；微命令译码器的可检测性，确保了硬件控制线路的可检测性；微诊断定位故障的有效性，缩短了维修机器的时间、提高了机器的可利用率，等等。上述的RAS措施，一般只适合于微程序设计的计算机。在这些方面，组合控制逻辑是很难办到的。

4．在其它方面的应用

（1）在改善机器性能和实现高速计算方面的应用

同样一个计算过程或者一次计算，当用子程序之类的方法来实现时，必须耗费不少取指令、计算操作地址、取操作数或者存中间结果等时间。而当用微程序来实现时，不仅可以节省这些时间，而且微程序的效率比程序的效率更高。早在IBM360系统的研制阶段，IBM公司的Hans Jeans在报告[14]中，就研究了IBM360系统的40、50和65型通过微程序设计可以改善的性能，并且得出结论：对于矩阵乘法和多项式等算术运算，其速度可以普遍提高50%左右，对于查表和字节测试等逻辑运算，其速度可以提高5～10倍。1974年在Interdata85型机上也做了类似的比较研究，并得出结论：对于双字长的乘法，其速度可提高3倍以上；对于求$\sin x$，其速度可提高2倍以上；对于数据处理，其速度可提高4倍以上，等等。

总之，诸如双倍字长运算、初等函数的计算、矩阵或行列式的计算、数制的转换，以及傅里叶变换等等，都能通过微程序设计来实现高速的计算。

（2）在其它数字系统中的应用

微程序设计已经越出了计算机系统这个应用范围，它在许多数字系统领域也得到了越来越广泛的应用。例如：

（a）图形或图像处理机；

（b）信号处理机；

（c）电子交换机；

（d）各种实验装置或设备；

（e）雷达数据处理装置；