

杜毅仁 李慕靖
王建中 陈启帆

编

16位微型计算机

SHI LIU WEI WEI
XING JI SUAN JI

—— 中 册

上海交通大学出版社

十六位微型计算机

(中 册)

杜毅仁 李慕靖 王建中 陈启帆编

上海交通大学出版社

内 容 简 介

本书从微型计算机的硬件、软件、系统结构等方面出发,通过Intel公司的8086系列以及IBM公司的个人计算机作为典型,进行系统的介绍和剖析,使读者获得有关微机的基础和应用知识。

全书共分上、中、下三册。上册通俗易懂地介绍了8086的硬件结构、指令系统及其程序设计语言,是全书的基础和核心部分;中册就整个8086系列,分别介绍了输入/输出处理器8089、专用数值处理器8087、操作系统固件80130、8086的改进型80186和80286,同时,还对MULTIBUS这一标准的系统总线以及8086的其它配套系列芯片作了介绍;下册比较系统地分析了IBM公司的个人计算机,内容包括PC的硬件结构、系统固件及操作系统等。

本书具有硬件、软件相结合的特点,而且既能了解8086,又能由此了解十六位微型计算机的一般原理及应用。因此,既可作为高等院校的教学参考书,也可作为研究院所、厂矿有关研究人员、工程技术人员乃至一般用户的工作手册。

十六位微型计算机

(中册)

杜毅仁 李慕端 王建中 陈启帆 编

上海交通大学出版社出版
上海淮海中路1984弄19号

新华书店上海发行所发行

常熟文化印刷厂排版

交通大学印刷厂印刷

开本: 787×1092 毫米 1/16 印张: 31.75 字数: 768,000
1984年10月第一版 1986年1月第二次印刷
印数: 10,000—20,000

定价: 5.80元

中 册 目 录

第二篇 iAPX 86/88 微处理器系列

第八章 iAPX 86/88 系列的形成和发展	1
§ 8.1 概述	1
§ 8.2 iAPX 86/88 系列的发展	1
§ 8.2.1 iAPX 86/88 系列对 8 位机的性能提升	2
§ 8.2.2 iAPX 86/88 系列性能的横向提升	3
一、数值数据协处理器 8087 (NPX)	3
二、输入/输出协处理器 8089 (IOP)	4
三、操作系统固件 80130 (OSF)	5
§ 8.2.3 iAPX 86/88 系列性能的纵向提升	5
一、高性能的 16 位微处理器 80186	6
二、超级 16 位微处理器 80286	7
§ 8.3 iAPX 86/88 系列的总结	7
第九章 8089 输入/输出处理器	8
§ 9.1 8089 I/O 处理器概述	8
一、程序控制输入输出的方式	8
二、DMA (直接存储器访问)方式	9
三、I/O 处理机方式	9
§ 9.2 8089 的工作原理	12
§ 9.2.1 CPU 与 8089 之间的通信	12
§ 9.2.2 8089 的 DMA 传送	15
§ 9.2.3 总线和系统结构	15
§ 9.3 8089 的体系结构	25
§ 9.3.1 框图及引脚功能介绍	25
§ 9.3.2 公用控制部件	27
§ 9.3.3 算术逻辑部件	28
§ 9.3.4 装配/拆卸寄存器	28
§ 9.3.5 取指令单元	28
§ 9.3.6 总线接口部件	29
§ 9.3.7 通道	30
一、I/O 控制部件	30

二、通道寄存器组	31
§ 9.3.8 8089 的存储器结构	35
§ 9.3.9 输入/输出机构	38
§ 9.4 DMA 传送	39
§ 9.4.1 外设控制器的初始化	40
§ 9.4.2 通道的准备工作	40
一、源指针和目标指针	40
二、翻译表指针	41
三、字节数	41
四、屏蔽/比较数值	41
五、总线的逻辑宽度	41
六、通道控制寄存器	42
§ 9.4.3 DMA 传送	45
§ 9.5 8089 处理器的控制和监督	47
§ 9.5.1 8089 的初始化	47
§ 9.5.2 通道命令	50
§ 9.5.3 几个控制信号	52
§ 9.6 8089 对多处理器的支持	53
§ 9.7 8089 的指令系统与寻址方式	56
§ 9.7.1 8089 的指令系统	56
一、数据传送指令	56
二、算术运算指令	57
三、逻辑和位操作指令	58
四、程序转移指令	60
五、处理器控制指令	61
§ 9.7.2 8089 的寻址方式	62
§ 9.7.3 8089 指令系统小结	65
§ 9.8 8089 的程序设计	78
§ 9.8.1 8089 汇编语言 ASM-89	78
一、语句	78
二、常数	80
三、数据定义	80
四、寻址方式	82
五、过程	82
六、分段控制	83
七、模块间的通讯	83
八、程序设计举例	85
§ 9.8.2 iAPX 86/11、iAPX 88/11 的程序设计及实例	90

第十章 IAPX 86/20、88/20 数值处理机的硬件与软件	98
§ 10.1 概述	98
§ 10.2 8087 数值处理器的结构	103
§ 10.2.1 8087 的结构概貌	103
§ 10.2.2 8087 的引脚功能介绍	104
§ 10.2.3 8087 处理器的结构	107
一、控制部件	108
二、数值运算执行部件	109
§ 10.2.4 8087 的数字系统	114
§ 10.3 8087 的指令系统	121
§ 10.3.1 数据传送指令	122
§ 10.3.2 比较指令	123
§ 10.3.3 算术运算指令	125
§ 10.3.4 超越函数计算指令	129
§ 10.3.5 常数指令	131
§ 10.3.6 处理器控制指令	132
§ 10.3.7 8087 指令系统小结	137
§ 10.4 数值处理机的体系结构	140
§ 10.4.1 8087 与 8086/8088 之间的连接	141
§ 10.4.2 IAPX 86/88 的协处理器接口	144
§ 10.4.3 应该考虑的几个问题	150
§ 10.5 软件基础及程序设计技术	158
§ 10.5.1 并发性	158
§ 10.5.2 同步控制	159
§ 10.5.3 程序设计技术	164
§ 10.5.4 IAPX 86/20、88/20 的程序设计	165
第十一章 IAPX 86/30、iAPX 88/30 操作系统处理机	233
§ 11.1 引言	233
§ 11.2 操作系统固件 80130	234
§ 11.3 操作系统处理机 IAPX 86/30、88/30 的结构	238
§ 11.4 操作系统处理机的管理功能	241
§ 11.4.1 作业与任务管理	244
§ 11.4.2 中断管理	247
§ 11.4.3 存贮器管理	249
§ 11.4.4 任务间的通信、同步与互斥	249
§ 11.4.5 其它控制功能	251
§ 11.5 应用举例	252

第十二章 iAPX 86/88 系列	254
§ 12.1 iAPX 88微处理器	254
§ 12.1.1 概述	254
§ 12.1.2 8088 的结构	254
一、寄存器组	255
二、8088 中的流水操作	256
三、存储器结构及其寻址方式	256
四、输入/输出	260
五、中断机构	260
六、对实现多处理机系统的支持	260
七、8088 的指令系统	261
§ 12.1.3 8088 与 8086 的比较	261
§ 12.2 8284 A 时钟发生和驱动器	262
§ 12.2.1 概述	262
§ 12.2.2 8284 A 的引脚结构	262
§ 12.2.3 8284 A 的功能概述	264
一、晶体振荡器	264
二、时钟发生器	264
三、复位逻辑	265
四、READY 信号同步机构	265
§ 12.3 8288 总线控制器	266
§ 12.3.1 概述	266
§ 12.3.2 8288 的引脚结构	266
§ 12.3.3 8288 的功能	268
§ 12.4 8289 总线裁决器	270
§ 12.4.1 概述	270
§ 12.4.2 8289 的引脚结构与功能	270
§ 12.4.3 8289 的工作过程	273
§ 12.5 8282/8283 8 位锁存器	278
§ 12.6 8286/8287 8 位总线收发器	280
第十三章 MULTIBUS 系统总线	282
§ 13.1 概述	282
§ 13.2 MULTIBUS 系统总线的结构	283
§ 13.3 MULTIBUS 系统总线的操作原理	287
§ 13.3.1 数据传送操作	287
§ 13.3.2 中断操作	290
§ 13.3.3 总线交换技术	291

第十四章 高性能的 16 位微处理器——80186/80188	295
§ 14.1 引言	295
§ 14.2 80186 概况	296
§ 14.2.1 CPU	296
§ 14.2.2 得到增强的 80186 CPU	297
§ 14.2.3 DMA 部件	297
§ 14.2.4 计时器	297
§ 14.2.5 中断控制器	298
§ 14.2.6 时钟发生器	298
§ 14.2.7 片选和准备就绪信号发生部件	298
§ 14.2.8 集成的外围电路访问	298
§ 14.3 80186 的使用	299
§ 14.3.1 总线与 80186 的接口	299
一、概述	299
二、物理地址的产生	300
三、数据总线操作	301
四、80188 数据总线操作	302
五、通用数据总线操作	302
六、控制信号	303
七、暂停定时	306
八、8288 和 8289 接口	306
九、准备就绪接口	307
十、总线特性总结	309
§ 14.3.2 存储器系统举例	310
一、2764 接口	310
二、2186 接口	311
三、8203 DRAM 接口	312
四、8207 DRAM 接口	314
§ 14.3.3 HOLD/HLDA 接口	314
一、HOLD 响应	315
二、HOLD/HLDA 定时和总线等待时间	315
三、退出 HOLD	317
§ 14.3.4 8086 总线和 80186 总线的区别	317
§ 14.4 DMA 部件接口	319
§ 14.4.1 DMA 的特性	319
§ 14.4.2 DMA 部件的编程	320
§ 14.4.3 DMA 传送	321
§ 14.4.4 DMA 请求	322
§ 14.4.5 DMA 响应	323

§ 14.4.6 内部产生的 DMA 请求	323
§ 14.4.7 外部同步的 DMA 传送	324
一、源同步的 DMA 传送	324
二、目同步的 DMA 传送	325
§ 14.4.8 DMA 暂停和 NMI	325
§ 14.4.9 DMA 接口举例	326
一、8272 软磁盘接口	326
二、8274 串行通信接口	327
§ 14.5 计时器部件接口	328
§ 14.5.1 计时器操作	328
§ 14.5.2 计时器寄存器	329
§ 14.5.3 计时器事件	330
§ 14.5.4 计时器输入引脚操作	331
§ 14.5.5 计时器输出引脚操作	332
§ 14.5.6 80186 计时器应用实例	332
一、80186 计时器实时时钟	332
二、80186 计时器波特率发生器	332
三、80186 计时器事件计数器	333
§ 14.6 80186 中断控制器接口	333
§ 14.6.1 中断控制器模块	333
§ 14.6.2 中断控制器操作	334
§ 14.6.3 中断控制器寄存器	334
一、控制寄存器	334
二、请求寄存器	335
三、屏蔽寄存器和优先级屏蔽寄存器	335
四、正被服务寄存器	336
五、查询和查询状态寄存器	336
六、中断结束寄存器	336
七、中断状态寄存器	337
八、中断向量寄存器	337
§ 14.6.4 中断源	337
一、内部中断源	337
二、外部中断源	338
三、IRM X TM 86 方式中断源	339
§ 14.6.5 中断响应	339
一、内部导向、主控制器方式	340
二、内部导向、IRM X TM 86 方式	340
三、外部导向	341
§ 14.6.6 中断控制器的外部连接	342

一、直接输入方式	342
二、级联方式	342
三、特殊的完全嵌套方式	343
四、iRMX 86 方式	343
§ 14.6.7 8259 A/级联方式接口举例	344
§ 14.6.8 80130 iRMX™ 86 方式接口举例	344
§ 14.6.9 中断等待时间	345
§ 14.7 时钟发生器	346
§ 14.7.1 晶体振荡器	346
§ 14.7.2 使用外部振荡器	346
§ 14.7.3 时钟发生器	346
§ 14.7.4 产生准备就绪信号	346
§ 14.7.5 复位	347
§ 14.8 片选	347
§ 14.8.1 存储器片选	348
§ 14.8.2 外围片选	349
§ 14.8.3 准备就绪信号的产生	349
§ 14.8.4 片选用法举例	350
§ 14.8.5 重叠的片选区域	350
§ 14.9 在 80186 系统中的软件	350
§ 14.9.1 80186 系统初始化	350
§ 14.9.2 iRMX™ 86 系统初始化	351
§ 14.9.3 8086 和 80186 执行指令的区别	351
§ 14.10 结论	354
§ 14.11 80186 技术资料	354
§ 14.11.1 外围控制块	354
一、设置外围控制块的基地址	354
二、外围控制块寄存器	355
§ 14.11.2 80186 同步信息	356
一、为什么需要同步装置	356
二、80186 同步装置	356
§ 14.11.3 80186 DMA 接口程序举例	357
§ 14.11.4 80186 计时器接口程序举例	361
§ 14.11.5 80186 中断控制器接口程序举例	367
§ 14.11.6 80186/8086 系统初始化程序举例	369
§ 14.11.7 80186 等待状态特性	371
§ 14.11.8 80186 的新指令	376
§ 14.11.9 80186/80188 的区别	378
§ 14.11.10 80186 特性	379

第十五章 超级 16 位微处理器 iAPX 286/10	385
§ 15.1 概述	385
§ 15.2 iAPX 286/10 概况	386
§ 15.2.1 iAPX 286/10 CPU	386
§ 15.2.2 80286 芯片引脚介绍	388
§ 15.3 iAPX 286/10 基本结构	392
§ 15.3.1 寄存器集	393
§ 15.3.2 iAPX 286/10 标志字和机器状态字	394
§ 15.3.3 iAPX 286/10 指令集	396
§ 15.3.4 存储器组成	401
§ 15.3.5 寻址方式	403
§ 15.3.6 数据类型	403
§ 15.3.7 I/O 空间	405
§ 15.3.8 中断	405
§ 15.3.9 中断优先级	406
§ 15.3.10 初始化和处理器复位	407
§ 15.3.11 暂停	407
§ 15.3.12 扩展能力	407
§ 15.4 iAPX 286 实地址方式	410
§ 15.4.1 存储器容量	410
§ 15.4.2 存储器寻址	410
§ 15.4.3 保留的存储器单元	411
§ 15.4.4 中断	411
§ 15.4.5 保护方式初始化	412
§ 15.4.6 停机	412
§ 15.5 iAPX 286 保护虚地址方式	412
§ 15.5.1 存储器容量	412
§ 15.5.2 存储器寻址	413
一、选择子	414
二、描述子	414
三、代码段和数据段描述子	414
四、系统控制描述子	416
五、段描述子 CACHE 寄存器	420
六、局部和全局描述子表	421
七、中断描述子表	422
八、存储器寻址小结	423
§ 15.5.3 特权	425
一、任务特权	426
二、描述子特权	426

三、选择子特权	427
§ 15.5.4 描述子访问和特权检查	427
一、数据段访问	427
二、控制转移	428
三、特权层的改变	430
§ 15.5.5 保护	430
一、对保护机构的要求	430
二、保护的实现	430
三、异常	431
§ 15.5.6 特殊操作	432
一、中断	432
二、任务转换操作	432
三、虚拟存贮器	435
四、可恢复的堆栈故障	436
五、协处理器上下文转换	436
六、指示器测试指令	437
七、双重错误和停机	437
八、保护方式初始化	437
§ 15.5.7 保护虚地址方式的总结	438
§ 15.6 系统接口	439
§ 15.6.1 总线接口信号和定时	439
§ 15.6.2 物理存贮器和 I/O 接口	440
§ 15.6.3 总线操作	440
§ 15.6.4 总线状态	440
§ 15.6.5 流水线寻址	441
§ 15.6.6 总线控制信号	441
§ 15.6.7 命令定时控制	442
§ 15.6.8 总线周期结束	443
§ 15.6.9 READY 操作	443
一、同步准备就绪	443
二、异步准备就绪	443
§ 15.6.10 数据总线控制	444
§ 15.6.11 总线用途	446
一、HOLD 和 HLDA	447
二、取指令	447
三、协处理器传送	447
四、中断响应序列	448
五、局部总线使用优先权	449
六、暂停或停机周期	449

§ 15.7	系统结构	449
§ 15.8	iAPX 286/10 技术资料	452
§ 15.8.1	D. C. 特性	452
§ 15.8.2	A. C. 特性	453
§ 15.8.3	波形	454
§ 15.8.4	80286 指令集总结	456
一、	指令定时注释	457
二、	关于指令时钟计数的假定	457
三、	指令集总结注释	457
第十六章	iSBC 286/10 产品系列	474
§ 16.1	iSBC 286/10 单板计算机	475
§ 16.1.1	iSBC 286/10 功能概述	475
§ 16.1.2	中央处理部件	476
§ 16.1.3	指令集	476
§ 16.1.4	结构特点	476
一、	向量中断控制	477
二、	中断源	477
三、	存储器容量	477
四、	串行 I/O	477
五、	可程序计时器	478
六、	行式打印机接口	478
§ 16.1.5	MULTIBUS 系统结构	479
一、	概述	479
二、	系统总线——IEEE796	479
三、	系统总线扩展能力	480
四、	系统总线——多总线主设备能力	480
五、	iLBX™ 总线——局部协总线	480
六、	iSBX™ 总线 MULTIMODULE™ 在板扩展	481
§ 16.1.6	软件支持	481
§ 16.2	iSBC 028 CX, 056CX 和 012 CX iLBX™ RAM 板	482
§ 16.2.1	功能概述	482
§ 16.2.2	双向端口特性	482
§ 16.2.3	系统存储器容量	482
§ 16.2.4	检错和纠错	483
一、	ECCI/O 地址选择	483
二、	控制状态寄存器	483
§ 16.2.5	电池后备电源/存储器保护	485
§ 16.3	iSBC 428 通用插座存储器扩展板	485

§ 16.3.1	功能概述	485
§ 16.3.2	iLBX™ 总线	486
§ 16.3.3	存贮体	486
§ 16.3.4	存贮器寻址	486
§ 16.3.5	操作方式	486
§ 16.3.6	存贮器访问	487
§ 16.3.7	中断	487
§ 16.3.8	禁止	487
§ 16.3.9	电池后备电源	487
§ 16.3.10	支持器件	487
§ 16.4	iSBC 580 MULTICHANNEL™ 总线到 iLBX™ 总线的接口	488
§ 16.4.1	MULTICHANNEL™ 接口能力	488
§ 16.4.2	iLBX™ 总线接口能力	490
§ 16.5	iRMX™ 286R 操作系统	490
§ 16.5.1	概述	490
§ 16.5.2	功能描述	491
§ 16.6	iSDM™ 286 iAPX 286 系统调试监督包	491
§ 16.6.1	功能概述	492
§ 16.6.2	通用开发接口	492
§ 16.6.3	有效的调试命令	492
§ 16.6.4	格式化显示	492
§ 16.6.5	对数值数据处理器的支持	492
§ 16.6.6	高速串行连接	493

第 二 篇

iAPX 86/88 微处理器系列

第八章 iAPX 86/88 系列的形成和发展

§ 8.1 概 述

随着社会对信息处理能力需求的不断增长,16位微型机的研制和开发,成了微型计算机应用的主流。这种趋势无论在国外还是国内,都已经变得越来越明显。由于16位微型机同8位微型机相比,具有更大的寻址空间和更强的运算、处理能力,所以在复杂的控制和诊断、字处理、通讯终端、图象终端以及在原先由小型机占据的那些领域中,16位微型机已经并且会继续开辟出更为广阔的天地。

自1978年以来,国际市场上各种16位微处理器纷纷涌现,但在所有的16位微处理器中,最早推出微处理器的Intel公司的产品仍然以其独特的风格占据着领先地位。特别是IBM公司宣布在该公司生产的个人计算机(PC)上采用Intel的产品以后,Intel的16位微处理器日益得到各方面的重视。Intel公司从构成系统的高度出发,提供了一系列向上兼容的16位CPU,并且还提供了种类繁多的支持芯片和各种提升系统功能的协处理机,从而在8086的基础上形成了16位微机的iAPX 86/88系列。由于Intel公司是最早推出16位微处理器的厂家,它拥有大量可以直接提供给用户的成熟的软件。例如80130集成的操作系统固件芯片等。

为了把握住16位微型计算机的发展趋势,对Intel公司的iAPX86/88系列的形成和发展作一番研究,对有志于研究、开发和应用16位微型计算机的人来说是会有所裨益的。本章将对iAPX86/88系列的形成和发展作一些探讨,以便使读者对这个完整的系列有一个全面的认识,在这个基础上再分章地进行详细介绍。

§ 8.2 iAPX 86/88 系列的发展

iAPX86/88系列的基础是8086(即iAPX86或iAPX86/10),它是一种16位的微处理机CPU。8086的主机设计对于8位机有了很大的突破,它的性能较8位机提高了大约10倍,达到了一般小型计算机的水平。由于8086是Intel16位微型机的基础,我们在上册中对8086的结构、指令系统及程序编制作了极为详尽的论述。从中我们知道8086用4个段寄存器(CS, DS, SS, ES)作为基础,能直接寻址1兆字节。我们还知道8086的指令系统设计是很有特色的,在8位字长的处理方面,比其他的16位机更为灵活和方便。在寻址方面,由于采用了mod, r/m方式,在最简单的情况下,仅使用一个字节中的5位就能产生出20位物理地址。

8086 的指令短小精悍,能产生简短高效的目的代码,这些都是 8086 的成功之处。但是,由于 8086 出现得最早,因而不可避免地在某些方面存在着缺陷。这主要表现在寻址空间、双字长运算及存储器保护等方面。针对这些存在的问题以及为了适应日新月异的发展需要,Intel 公司在 8086 的基础上,对 8086 进行了横向和纵向的性能提升,同时利用了 16 位微型机的优势,对 8 位机的性能也进行了大幅度的提升,从而形成了一个完整的 16 位微型机系列。

在 iAPX 86/88 系列中,把单纯的 8086 命名为 iAPX 86/10,当与 8087 配接后称为 iAPX 86/20,当再与 8089 配接后就称为 iAPX 86/21,当 8086 与 80130 配接后就称为 iAPX 86/30。这种命名法对于以 8088, 80186, 80286 作 CPU 的系统也适用。表 8.1 列出了各种处理器的构成。

表 8.1 iAPX 系列的命名

器件组合		8086 80186 80286 8087 80287 8089 80130						器件组合		8088 80188 8087 80287 8089 80130					
命 名								命 名							
iAPX 86/10		1						iAPX 88/10		1					
iAPX 86/11		1 1						iAPX 88/11		1 1					
iAPX 86/20		1 1						iAPX 88/20		1 1					
iAPX 86/21		1 1 1						iAPX 88/21		1 1 1					
iAPX 86/30		1 1						iAPX 88/30		1 1					
iAPX 186/10		1						iAPX 188/10		1					
iAPX 186/20		1 1						iAPX 188/20		1 1					
iAPX 186/30		1 1						iAPX 188/30		1 1					
iAPX 286/10		1													
iAPX 286/20		1 1													

§ 8.2.1 iAPX 86/88 系列对 8 位机的性能提升

早期的微型计算机系统都是 8 位的, iAPX 86/88 系列采取了向上兼容的策略,使原来为 8 位机编制的软件仍能为新一代的 16 位机所使用。另一方面,由于在许多应用中大量存在着以字节为单位进行的信息交换,这类应用需要大的存储容量和较高的运算速度,但在外部却只要 8 位数据总线就够了。Intel 的 8088 就是专门设计来满足这种要求的。iAPX 88 (8088) 具有与 iAPX 86 相同的寄存器结构,相同的寻址空间,相同的指令系统,因此它内部就是一个完整的 16 位微处理器。iAPX 88 对外的数据总线是 8 位,从而能与 8 位的微型机系统匹配。显然,由于 iAPX 88 内部是 16 位的 CPU,所以把它作为 8 位的微处理器来应用时,自然会比其他的 8 位机具有更强的指令功能,更大的寻址空间,并且具有更快的处理速度。由此可见, iAPX 88 在 8 位机领域中有着强大的生命力。采用 8088 作 CPU,并不会影响整个系统的造价,但由此而带来的性能提升却是巨大的。这一点在字符处理和事务处理等应用中特别明显。关于 8088 的详细情况可参见第十二章。

§8.2.2 iAPX 86/88 系列性能的横向提升

iAPX 86/88 系列, 作为 16 位微型计算机的 CPU 对 8 位微处理器作为 CPU 构成的微型计算机来讲, 性能有了很大的提高。但是, 若 CPU 既要完成各种控制、管理, 又要执行各种数值计算, 还要处理输入/输出, 这就难以使整个系统的速度的提高和性能的增长有所突破。Intel 对此采取的策略并不是马上着手去制造速度更高、功能更强的新的微处理器, 而是针对着某几方面存在的缺陷, 设计了一系列可供用户选择的, 在某一方面功能特别强的协处理器。加上这些协处理器以后, 8086 就能和这些协处理器并行操作, 从而充分发挥了 8086 的潜力。8087, 8089, 80130 等是使 iAPX 86/88 系列的性能在横向得以提升的主要协处理器。

一、数值数据协处理器 8087 (NPX)

8087 是一种专门用于处理数值数据运算的协处理器。8087 内部含有 8 个 80 位字长的数据寄存器, 4 个专用寄存器, 它们分别是控制寄存器 (16 位), 状态寄存器 (16 位), 指令指示器 (32 位) 和数据指示器 (32 位)。8087 具有 62 条指令, 分为传送、比较、算术、取常数、超越函数及控制共 6 种类型。表 8.2 列出了其中有关数值运算的指令。

表 8.2 8087 的有关数值运算指令

算术运算	+, -, *, /, $\sqrt{\quad}$, 比例、求余、舍入、抽选、求绝对值、改变符号					
超越函数	tg、	arctg、	2^N-1 、	$M \cdot \log_2 N$	$M \cdot \log_2(N+1)$	
取常数	0.0、	1.0、	π	$\log_2 10$ 、	$\log_2 e$ 、	$\log_{10} 2$ 、 $\log_{10} e$

8087 的指令结构是十分独特的, 它借用了 iAPX 86/88 的 ESC 指令中的六个用户可编程序位来构成 (见图 8.1)。因此, 8087 的数值运算指令能与原 iAPX 86/88 的指令混合编制在一个完整的程序中, 并由 iAPX 86/88 的 CPU 来控制整个程序, 但由 CPU 和 8087 分别执行各自的指令操作。

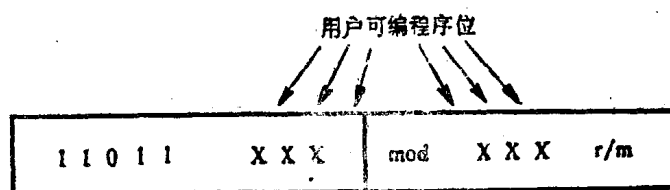


图 8.1 ESC 指令的格式

8087 与 iAPX 86/88 系统的配接是十分方便的。在使用 8087 时, iAPX 86/88 需采用最大模式工作, 两种芯片的请求/允许 (RQ/GT) 端连在一起, 并将 8086/88 的队列状态信号 QS_0 , QS_1 接入 8087, 将 8087 的 BUSY 信号引入 8086 的 $\overline{TEST}$ 端。运行时, 8087 通过 QS_0 , QS_1 两个引脚来监视 CPU 中指令队列的状态, 并由此对 ESC 指令进行检测。一旦 8087 在数据总线上发现 ESC 指令后, 它也同 8086/88 一样将 ESC 指令读入。由于 ESC 中包含了 8087 的数值运算指令, 这一过程也就完成了 CPU 向 8087 的指令传递。

8087 的指令都是双存储器操作数指令, 其中一个在系统堆栈的栈顶 (或栈顶下第 n 个