

Principle and Application  
of INTEL 8086/8088  
Family Microcomputers

INTEL 8086/8088 系列

# 微型计算机及其应用

ZHANG MINGDA

张明达 著

中南

Principle and Application  
of INTEL 8086/8088  
Family Microcomputers



INTEL

8086/8088 系列

微型计算机及其

ZHANG MINGDA

张明达 著

, 帮助  
种实际问题  
(资料)。  
必, 下少的。第六至九  
人根据 需要跳过其中

写程 应来用

## 内 容 简 介

本书全面论述当前国内使用最广泛的以INTEL8086/8088CPU为基础的一类十六位微型计算机(例如IBM—PC及其兼容机)的原理及应用。

书中共分九章,前五章讨论该机的设计思想、体系结构、指令系统、输入/输出和中断系统。第六、七章研究该机的汇编语言程序设计及本系列微型机专用的PL/M-86高级语言。最后两章介绍可增强机器功能和处理速度的8089输入/输出处理器和8087数据处理扩充器。本书着重阐明该机设计和应用方面的根本性问题。为便于读者理解和掌握编程技巧,对每条指令的功能及信息流均有详细描述,并配有程序设计的实例。可供从事微型计算机研究、应用、软硬件开发等方面的科技人员参考,亦可作为高等院校师生教学的重要参考书。

## ER8086/8088系列微型计算机及其应用

张明达 著

责任编辑 雷丽云

中南工业大学出版社出版发行

湖南省测绘印刷厂印刷

湖南省新华书店经销

开本: 787×1092 1/16 印张: 38 字数: 975千字

1987年3月第一版 1987年3月第一次印刷

印数: 00000

ISBN

1-5/TP-002

定价: 6.20元

## 前 言

INTEL8086/8088系列是当前世界上最流行的一类十六位微型计算机。例如,美国国际商业机器公司(IBM公司)于1981年推出的IBM-PC微型计算机就是以INTEL8088微处理器为核心而设计的。当然,以INTEL8086/8088为基础的微型机可能具有各种各样的体系结构,因而它们的功能也有较大差异。但是,只要掌握它们的基本理论和性能,就不难在设计和使用方面作到举一反三、运用自如。目前,这类机器,特别是IBM-PC及其兼容机,已在我国得到广泛应用,而且使用范围正在迅速扩展;我国也自行开发了一批以INTEL8086/8088CPU为基础的微机系统。因此,掌握INTEL8086/8088系列微型计算机软硬件性能及设计使用方面的专门知识,已是我国计算机行业和广大用户的迫切要求。写一部能概括该系列微型机基本知识的专著,以满足这种需求,使之能对我国计算机事业的发展起到一定推动作用,此即撰写本书的动机。

作者假定,读者已具备微型机方面的基础知识,并对8位(特别是INTEL8080/8085系列)微型机并不陌生。因此,除个别必要的基本概念外,本书将直接研究INTEL8086/8088系列微型计算机本身的问题。初学微型机的读者,在阅读本书之前,应适当学习一些有关微型计算机的基本知识,但并不需花费很多精力。

本书内容力求准确、通俗、全面,但又不希望篇幅过长。作到这一点并非易事,因为INTEL8086/8088系列机器的软硬件资源十分丰富,功能很强,涉及的问题很多。为压缩篇幅,不得不适当缩减或舍弃一些枝节问题。即使这样,全书篇幅仍显偏长。原因是:简化或舍弃一些核心问题的阐述可能给读者带来更大困难,将使他们为了搞清一些基本问题需要去寻找更多的参考资料,花费更多的时间。

本书立足于阐述INTEL8086/8088系列微型机的基本理论和软硬件的基本性能,帮助读者掌握该系列机器最主要的基本知识。完全依靠本书来处理设计和使用中的各种实际问题是不够的,许多细节问题尚需查找其它参考资料(特别是INTEL公司的资料)。

本书共分九章。掌握这些内容对一个系统设计或使用人员来说是必不可少的。第六至九章的内容相对来说互相独立,对于某种特定机型的一些特定用户,可以根据需要跳过其中一些章节直接阅读自己感兴趣的内容。

为方便国内读者,书中程序实例的注释已全部译成中文。在实际编编程  
应采月  
注释,以方便原程序在计算机中的输入、存储、显示和打印。

由于作者水平所限,书中错误和欠妥之处在所难免,敬请读者多加批评教  
不

作 者

1986年1月

# 目 录

|                               |        |
|-------------------------------|--------|
| 第一章 绪 论                       | ( 1 )  |
| 第一节 微型机发展简史                   | ( 1 )  |
| 第二节 INTEL 8086/8088系列微型机的基本特点 | ( 4 )  |
| 第三节 微型机的应用                    | ( 7 )  |
| 第四节 数制和码制                     | ( 8 )  |
| 一、各种数制及其多项式表示法                | ( 8 )  |
| 二、各种数制的相互转换                   | ( 10 ) |
| 三、二进制运算                       | ( 14 ) |
| 四、定点数和浮点数                     | ( 16 ) |
| 五、带符号数                        | ( 16 ) |
| 六、常用的代码                       | ( 19 ) |
| 第二章 8086/8088 系列微型机的体系结构      | ( 23 ) |
| 第一节 8086/8088 微处理器的内部结构       | ( 23 ) |
| 一、8086/8088 CPU的结构特点          | ( 23 ) |
| 二、BIU 和CU 的分工和配合              | ( 24 ) |
| 三、通用寄存器                       | ( 25 ) |
| 四、段寄存器                        | ( 27 ) |
| 五、指令指针 ( IP )                 | ( 28 ) |
| 六、标志寄存器                       | ( 29 ) |
| 第二节 8086/8088 的片脚功能           | ( 30 ) |
| 第三节 8086与8088 CPU 的比较         | ( 33 ) |
| 第四节 存储器组织                     | ( 33 ) |
| 一、只读存储器 ( ROM )               | ( 34 ) |
| 二、随机存取存储器 ( RAM )             | ( 36 ) |
| 三、存储器的组成                      | ( 40 ) |
| 四、存储器的段 ( SEGMENT ) 结构        | ( 42 ) |
| 五、存储器的逻辑地址和物理地址               | ( 43 ) |
| 六、堆 栈                         | ( 45 ) |
| 七、专用和保留存储单元                   | ( 46 ) |
| 第五节 8284 A时钟信号发生器             | ( 47 ) |
| 第六节 总线操作方式                    | ( 49 ) |
| 一、8086 总线周期                   | ( 50 ) |
| 二、8088 总线周期                   | ( 50 ) |
| 三、8282/8283 地址锁存器             | ( 50 ) |
| 四、8286/8287 数据缓冲器             | ( 50 ) |
| 第七节 最小和最大模式                   | ( 50 ) |
| 一、最小模式                        | ( 50 ) |
| 二、最大模式                        | ( 50 ) |

|                           |         |
|---------------------------|---------|
| <b>第三章 8086/8088 指令系统</b> | ( 65 )  |
| 第一节 8086/8088 指令系统的基本特点   | ( 65 )  |
| 第二节 指令编码格式                | ( 68 )  |
| 一、双操作数寄存器/存储器存取指令         | ( 68 )  |
| 二、单操作数寄存器/存储器存取指令         | ( 70 )  |
| 三、与AX、AL有关的指令             | ( 71 )  |
| 四、其它指令                    | ( 71 )  |
| 第三节 寻址方式                  | ( 71 )  |
| 一、直接寻址                    | ( 72 )  |
| 二、基址寻址                    | ( 73 )  |
| 三、变址寻址                    | ( 74 )  |
| 四、基址变址寻址                  | ( 74 )  |
| 五、程序转移指令的寻址方式             | ( 75 )  |
| 六、字符串寻址                   | ( 76 )  |
| 七、I/O端口寻址                 | ( 76 )  |
| 第四节 8086/8088 指令          | ( 77 )  |
| 一、数据传送指令                  | ( 77 )  |
| 二、算术运算指令                  | ( 87 )  |
| 三、位操作指令                   | ( 110 ) |
| 四、字符串指令                   | ( 119 ) |
| 五、程序转移指令                  | ( 123 ) |
| 六、CPU控制指令                 | ( 137 ) |
| 第五节 8086/8088 指令详解        | ( 139 ) |
| <b>第四章 输入/输出</b>          | ( 207 ) |
| 第一节 输入/输出控制方式             | ( 207 ) |
| 一、程序控制方式                  | ( 207 ) |
| 二、中断控制方式                  | ( 211 ) |
| 三、中断源识别                   | ( 212 ) |
| 四、直接存储器存取                 | ( 215 ) |
| 第二节 8212通用端口              | ( 216 ) |
| 一、功能说明                    | ( 217 ) |
| 二、应用举例                    | ( 217 ) |
| 第三节 8257 可编程DMA控制器        | ( 219 ) |
| 一、8257的结构特点               | ( 220 ) |
| 二、8257的编程和读寄存器操作          | ( 224 ) |
| 三、通道操作                    | ( 226 ) |
| 四、硬件环境                    | ( 227 ) |
| 五、串行端口—8251-A USART       | ( 228 ) |
| 六、8251-A的基本结构             | ( 228 ) |
| 七、工作方式和控制字                | ( 231 ) |

|                             |         |
|-----------------------------|---------|
| 第五节 并行端口—8255A PPI .....    | ( 234 ) |
| 一、8255 A的基本结构 .....         | ( 235 ) |
| 二、8255 A的工作方式和控制字 .....     | ( 236 ) |
| 第六节 8253可编程序间隔计数器/定时器 ..... | ( 243 ) |
| 一、8253的结构特点 .....           | ( 243 ) |
| 二、8253的工作方式和控制字 .....       | ( 245 ) |
| 三、8253的读/写操作 .....          | ( 249 ) |
| 第五章 中断系统 .....              | ( 250 ) |
| 第一节 8086/8088中断结构 .....     | ( 250 ) |
| 一、中断指针表 .....               | ( 251 ) |
| 二、各类中断的优先权级别 .....          | ( 252 ) |
| 第二节 外部中断 .....              | ( 253 ) |
| 第三节 8259 A可编程序中断控制器 .....   | ( 256 ) |
| 一、结构原理 .....                | ( 256 ) |
| 二、中断引导方式 .....              | ( 259 ) |
| 三、中断优先权管理 .....             | ( 259 ) |
| 四、中断触发 .....                | ( 265 ) |
| 五、中断级联 .....                | ( 267 ) |
| 第四节 8259 A程序设计 .....        | ( 269 ) |
| 一、预置命令字 ( ICW ) .....       | ( 270 ) |
| 二、操作命令字 ( OCW ) .....       | ( 273 ) |
| 第五节 定时器控制的中断 .....          | ( 275 ) |
| 第六章 8086/8088汇编语言程序设计 ..... | ( 282 ) |
| 第一节 ASM—86语句 .....          | ( 282 ) |
| 第二节 ASM—86数据 .....          | ( 284 ) |
| 一、常数项 .....                 | ( 284 ) |
| 二、变    量 .....              | ( 285 ) |
| 三、标    号 .....              | ( 289 ) |
| 四、LABEL伪指令 .....            | ( 290 ) |
| 五、结构—STRUC/ENDS伪指令 .....    | ( 292 ) |
| 六、结构的预置和存储分配 .....          | ( 293 ) |
| 七、对结构的访问 .....              | ( 295 ) |
| 八、记录—RECORD 伪指令 .....       | ( 297 ) |
| 九、记录的预置和存储分配 .....          | ( 298 ) |
| 十、表达式中的记录 .....             | ( 299 ) |
| 第三节 ASM—86表达式 .....         | ( 300 ) |
| 一、运算对象和运算符 .....            | ( 300 ) |
| 二、算术运算符 .....               | ( 300 ) |
| 三、条件运算符 .....               | ( 300 ) |
| 四、逻辑运算符 .....               | ( 300 ) |

|                                       |         |
|---------------------------------------|---------|
| 五、属性修改运算符.....                        | ( 303 ) |
| 六、数值返回运算符.....                        | ( 306 ) |
| 七、结构和记录专用运算符.....                     | ( 309 ) |
| 八、圆 括 号.....                          | ( 310 ) |
| 九、方 括 号.....                          | ( 310 ) |
| 十、运算符的优先级.....                        | ( 312 ) |
| 十一、EQU 伪指令 .....                      | ( 313 ) |
| 第四节 ASM—86 指令操作数 .....                | ( 313 ) |
| 一、立即操作数.....                          | ( 314 ) |
| 二、寄存器操作数.....                         | ( 314 ) |
| 三、存储器操作数 I —访问数据.....                 | ( 317 ) |
| 四、存储器操作数 II —访问代码.....                | ( 321 ) |
| 第五节 程序分段—SEGMENT/ENDS及ASSUME伪指令 ..... | ( 324 ) |
| 一、SEGMENT/ENDS伪指令 .....               | ( 324 ) |
| 二、ASSUME伪指令.....                      | ( 326 ) |
| 三、段的嵌套.....                           | ( 327 ) |
| 第六节 群—GROUP伪指令 .....                  | ( 328 ) |
| 第七节 段 控 制.....                        | ( 329 ) |
| 一、段寄存器的装填.....                        | ( 330 ) |
| 二、段 前 缀.....                          | ( 333 ) |
| 三、段隐含访问和段修改.....                      | ( 334 ) |
| 四、程序计数器( \$ )和ORG伪指令.....             | ( 335 ) |
| 五、PUBLIC和EXTRN伪指令.....                | ( 336 ) |
| 六、NAME和END伪指令.....                    | ( 337 ) |
| 七、简单程序实例.....                         | ( 337 ) |
| 第八节 过程—PROC/ENDP伪指令.....              | ( 342 ) |
| 一、用PROC/ENDP伪指令定义“过程”.....            | ( 342 ) |
| 二、过程的递归、嵌套和直接插入.....                  | ( 343 ) |
| 三、过程实例.....                           | ( 344 ) |
| 第九节 段 设 置.....                        | ( 361 ) |
| 一、代码、数据和堆栈都不超过 64k .....              | ( 361 ) |
| 二、代码超过 64kB, 数据和堆栈不 超过64kB.....       | ( 364 ) |
| 三、数据超过 64kB, 代码和堆栈 不超过 64kB.....      | ( 365 ) |
| 四、代码 数据均超过 64kB, 堆栈不超过 64kB.....      | ( 367 ) |
| 第七章 P1 99 高级语言 .....                  | ( 368 ) |
| .....                                 | ( 368 ) |
| .....                                 | ( 368 ) |
| .....                                 | ( 369 ) |
| .....                                 | ( 369 ) |
| .....                                 | ( 370 ) |

|                                   |         |
|-----------------------------------|---------|
| 五空符号的用法.....                      | ( 370 ) |
| 六注 释.....                         | ( 372 ) |
| 第二节 PL/M—S6 语句.....               | ( 372 ) |
| 第三节 数据类型,说明语句.....                | ( 373 ) |
| 一、纯数常数.....                       | ( 373 ) |
| 二、浮点常数.....                       | ( 374 ) |
| 三、字符串常数.....                      | ( 374 ) |
| 四、标量及其说明.....                     | ( 375 ) |
| 五、数组及其说明.....                     | ( 376 ) |
| 六、数组元素的引用 一下标变量.....              | ( 376 ) |
| 七、结 构.....                        | ( 377 ) |
| 八、有基变量.....                       | ( 380 ) |
| 九、变量的定位 —— AT 属性.....             | ( 381 ) |
| 十、LITERALLY说明.....                | ( 382 ) |
| 十一、赋 初 值.....                     | ( 383 ) |
| 十二、复合说明语句.....                    | ( 385 ) |
| 第四节 表 达 式.....                    | ( 385 ) |
| 一、运算对象.....                       | ( 385 ) |
| 二、算术运算符.....                      | ( 386 ) |
| 三、关系运算符.....                      | ( 337 ) |
| 四、逻辑运算符.....                      | ( 387 ) |
| 五、运算符的优先级.....                    | ( 388 ) |
| 第五节 赋值语句.....                     | ( 389 ) |
| 一、隐式类型转换.....                     | ( 389 ) |
| 二、显式类型转换.....                     | ( 390 ) |
| 三、多次赋值.....                       | ( 392 ) |
| 四、嵌入赋值.....                       | ( 392 ) |
| 第六节 流程控制语句.....                   | ( 393 ) |
| 一、DO 程序块.....                     | ( 393 ) |
| 二、IF 语 句.....                     | ( 399 ) |
| 三、标号和GOTO语句.....                  | ( 403 ) |
| 四、暂停语句.....                       | ( 404 ) |
| 五、过程调用和过程返回语句.....                | ( 404 ) |
| 第七节 说明对象的作用域及其扩展.....             | ( 404 ) |
| 一、说明对象的作用域.....                   | ( 405 ) |
| 二、作用域的扩展——PUBLIC和EXTERNAL 属性..... |         |
| 第八节 标号说明.....                     |         |
| 第九节 过 程.....                      |         |
| 一、过程说明.....                       |         |

|                  |      |
|------------------|------|
| 二、过程调用           | 14)  |
| 第十节 输入和输出        | 18)  |
| 第十一节 模块化程序设计     | 19)  |
| 第八章 8089输入/输出处理器 | 21)  |
| 第一节 8089的硬件环境    | 21)  |
| 一、本地方式           | 21)  |
| 二、远程方式           | 22)  |
| 三、总线操作           | 27)  |
| 四、存储器组织          | 28)  |
| 第二节 8089的体系结构    | 30)  |
| 一、片脚功能           | 31)  |
| 二、公用控制单元 (CCU)   | 32)  |
| 三、算术逻辑运算单元 (ALU) | 32)  |
| 四、装/拆寄存器         | 32)  |
| 五、取指令单元          | 43)  |
| 六、总线接口单元 (BIU)   | 34)  |
| 七、通道             | 36)  |
| 第三节 CPU和IOP的通讯   | 39)  |
| 一、预置             | 40)  |
| 二、通道命令           | 43)  |
| 第四节 8089的DMA 传送  | 45)  |
| 一、外围芯片的预置        | 46)  |
| 二、通道准备           | 46)  |
| 三、DMA 传送         | 452) |
| 第五节 8089指令系统     | 457) |
| 一、8089指令的编码格式    | 457) |
| 二、8089寻址方式       | 459) |
| 三、8089指令         | 463) |
| 四、8089指令编码表      | 472) |
| 第六节 8089汇编语言程序设计 | 476) |
| 一、语 句            | 476) |
| 二、常 数            | 477) |
| 三、定义变量           | 477) |
| 四、结 构            | 478) |
| 五、操 作 数          | 479) |
| 六、过 程            | 481) |
| 七、程序分            | 481) |
| 八、与CPU之间的通讯      | 482) |
|                  | 483) |

|                              |       |
|------------------------------|-------|
| <b>第九章 8087数据处理扩充器 (NPX)</b> | (486) |
| <b>第一节 8087的体系结构</b>         | (487) |
| 一、控制部件 (CU)                  | (488) |
| 二、数值执行单元 (NEU)               | (488) |
| <b>第二节 8087的片脚功能</b>         | (491) |
| <b>第三节 8087中的数值计算</b>        | (493) |
| 一、8087数字系统                   | (493) |
| 二、数据的类型和格式                   | (494) |
| 三、舍入控制                       | (497) |
| 四、精度控制                       | (498) |
| 五、无穷大控制                      | (498) |
| 六、事故和响应                      | (498) |
| <b>第四节 存储器操作</b>             | (500) |
| 一、数据存储格式                     | (501) |
| 二、存储器存取                      | (501) |
| <b>第五节 8087指令系统</b>          | (502) |
| 一、数据传送指令                     | (502) |
| 二、算术运算指令                     | (504) |
| 三、比较指令                       | (508) |
| 四、超越函数运算指令                   | (510) |
| 五、常数指令                       | (512) |
| 六、处理器控制指令                    | (512) |
| 七、指令用法                       | (517) |
| 八、8087指令编码表                  | (517) |
| <b>第六节 处理器的控制和监视</b>         | (521) |
| 一、初始化                        | (521) |
| 二、中断请求                       | (522) |
| 三、中断优先权管理                    | (523) |
| <b>第七节 8087程序设计简介</b>        | (524) |
| 一、寄存器堆栈的使用                   | (524) |
| 二、NPX与CPU之间的同步               | (525) |
| 三、状态环境切换                     | (527) |
| <b>参考文献</b>                  | (532) |
| <b>附录A 8086/8088指令系统摘要</b>   | (533) |
| <b>附录B 8086/8088指令一览表</b>    | (540) |
| <b>附录C 8086/8088机器指令译码指南</b> | (560) |
| <b>附录D 8089指令一览表</b>         | (572) |
| <b>附录E 8089机器指令译码指南</b>      | (59)  |

# 第一章 绪 论

## 第一节 微型机发展简史

1946年,世界上第一台电子计算机在美国诞生,这台计算机包含18000个电子管,功率高达100千瓦,机房长度超过30米。这个庞然大物被称作第一代电子计算机。1958年开始出现的第二代电子计算机,用晶体管取代了电子管,不仅大大降低了计算机成本和体积,而且将其运算速度提高近百倍。1965年,集成电路问世,迎来了以小规模集成电路为主体的第三代计算机。随着集成电路技术的迅猛发展,1970年出现以大规模集成电路为主体的第四代计算机。作为大规模集成电路发展的必然结果,美国INTEL公司于1971年研制出一种别具一格的微型计算机INTEL4004,这是一种4位中央处理器。次年,该公司又推出8位微处理器INTEL8008。从此为计算机的发展开辟了一条崭新途径,这是计算机科学的划时代进步。

由于微型计算机具有体积小、能耗低、重量轻、价格低、可靠性高、使用方便等一系列优点,因此获得极广泛的应用。这样,反过来又推动微型机的发展。在短短的十几年时间内,微型计算机就经历了五个主要阶段,平均每2~3年换代一次。

1971~1973年为微型机发展的萌芽阶段,以INTEL4004和INTEL8008为代表,采用P-MOS工艺制成,集成度为每块芯片含约两千个晶体管,芯片引出线为16~24条,时钟频率为1MHz,平均指令周期为20 $\mu$ s,数据总线为4位,地址总线为4~8位。这类微处理器的功能尚不完善,而且当时组成微型机系统所需的外围芯片也不齐全。此即所谓的第一代微型机。

1973~1975年为微型机发展的第二个阶段。在该阶段中,不仅微处理器本身的设计和生已经相当成熟,而且外围器件也已相当齐备。此即所谓的第二代微型计算机。其典型产品是INTEL8080和MC6800。该产品是8位机,采用N-MOS工艺制成,集成度为每片五千个晶体管,具有40条引出线,时钟频率为2MHz,平均指令周期为2 $\mu$ s,数据和地址总线为8~16位。

1975~1977年为微型机发展的第三个阶段。其特点是进一步提高微型机的集成度和运算速度、减少所用芯片数目,增强微处理器及外围芯片功能。此即所谓的第三代微型计算机。其代表产品是INTEL8085、Z-80、MC8089及与之伴生的外围器件。这种计算机基本上也是8位的,采用E/D MOS工艺制成,集成度达每片一万个晶体管,芯片引出线为40条,时钟频率达2.5~5MHz,平均指令周期为1 $\mu$ s,数据及地址总线为8~16位。

随着超大规模集成电路工艺的不断完善,在一个芯片上能容纳的晶体管数量不断增加,1978~1980年微型计算机进入了第四个发展阶段。典型产品为INTEL8086/8088, MC68000和Z-8000微处理器及与之配套的各种高性能外围芯片。这样,就构成了16位微型计算机。与此同时,还出现了多种类型的单片微型计算机。此即微型机的第四代产品。它们是采用N/H MOS工艺制成的,集成度达每片2~6万个晶体管,芯片引出线为40~64条,时钟频率10MHz,平均指令周期减少到0.5~0.1 $\mu$ s,数据总线达16位,地址总线达20~24位。微型计算机的功能和运算速度已经达到小型计算机的水平,因而使小型机市场已受到严重冲击。

由于集成电路技术的进一步发展,在单块芯片上所能容纳的晶体管数目越来越多,因此INTEL公司在70年代末和80年代初相继推出性能更加完善,功能更强的高级16位微处理器80186/80188和超级16位微处理器80286。

80186/80188及80286与8086/8088具有相同的基本结构,它们之间是完全目标代码相兼容的。

80186/80188的执行速度比8086/8088快得多,这是因为80186/80188的总线接口部件和执行部件都在硬件上得到加强。例如8086/8088采用内部微程序计算存储器的物理地址,而80186/80188则采用专门的硬件加法器来实现物理地址的计算;80186/80188用硬件方法提高乘除指令的执行速度;80186/80188通过流水线工作方式使递减CX和标志检查的操作能并行执行,从而大大提高字符串的处理速度;80186/80188多位移位和循环移位的速度可达1位/时钟周期。此外,80186/80188还提供一些新指令,从而可简化汇编语言程序的编制过程,增强高级语言编译性能及缩短目标代码的长度。

80186/80188本身还包括一个DMA部件、一个计时器部件、一个中断控制器、片选及准备就绪信号发生部件、以及时钟发生器。由此可见,80186/80188把8086/8088的许多外围芯片的功能都集成在CPU内部,从而使CPU的功能显著增强。

超级16位微处理器80286是INTEL公司推出的新一代16位微处理器。它的问世,标志着16位微型机的发展进入一个崭新阶段,引起各方面的关注。其性能已超过一些典型的小型计算机。

80286 CPU包括执行部件、地址部件、总线部件和指令部件四个部分。这四个部分可以并行工作,从而提高吞吐率。

80286具有集成在芯片中的存储器管理机构,它能以四层特权来支持操作系统和任务的分离,并能使程序和数据保密。其整体性能比8086/8088提高六倍。

80286具有很强的寻址能力,并能以两种不同的方式运行。在实地址方式下,具有1MB寻址能力。在虚地址保护方式下,能将每个任务的 $2^{30}$ 字节虚拟地址映射到 $2^{24}$ 字节的物理地址,即具有16MB的寻址能力。

从1981年开始,微型机进入第五个发展阶段,很多公司纷纷研制32位微型计算机,即所谓第五代微型机。这是超大规模集成电路工艺的新成就,预示着微型机新时代的到来。INTEL公司推出的iAPX432型32位微型机特别引人注目。它是用HMOS工艺制成的,集成度为每片10万个晶体管,芯片引出线64条,时钟频率为10MHz左右,平均指令周期达 $0.1\mu s$ 左右,具有32位数据总线和24~32位地址总线。iAPX432从提高软件生产效率的观点出发,采用全新的系统结构体系。用5台iAPX432组成的多处理机系统的处理速度达200万条/秒,其性能超过VAX-11/780,接近IBM370/158。由此可知,第五代微型机已达到中型计算机的水平,必将明显冲击中型机市场,同时也给中型机的发展指明新的方向。iAPX432微型机的指令变化可由2位至200位以位为单位,可以对8、16、32、64和80位字长的数据进行处理,可以直接寻址 $2^{24}$ (16兆)字节的物理(实际)空间, $2^{32}$ 字节的逻辑空间和 $2^{40}$ 字节的虚拟空间,同时可对数据进行保护。它的操作系统大部分已经固化。iAPX432机的指令系统与8086/8088不兼容,指令格式变化为2~200位,以位为单位。由于指令采用多字段结构,因此需从指令列中选取所需的字段。微指令采用流水线执行方式,数据操作的特征是采用双端排队寄存器。

iAPX432的内部总线是16位的,但可以处理32、64、80位可变长度的数据。iAPX432内虚拟存储器,在寻址过程中,需要把40位的虚拟地址转换为24位实际地址。iAPX432

系统含有一个I/O处理器，以便控制I/O子系统与432机之间的数据传送。

由此可见，在微型机世界，INTEL公司一直处于特别重要的地位，它始终以开拓者的姿态引导着微型机发展的方向。INTEL系统产品的发展历史实际上代表着世界微型机发展历史。图1-1描绘INTEL系列微型机的发展情景。INTEL公司产品的销售量也一直处于首位。

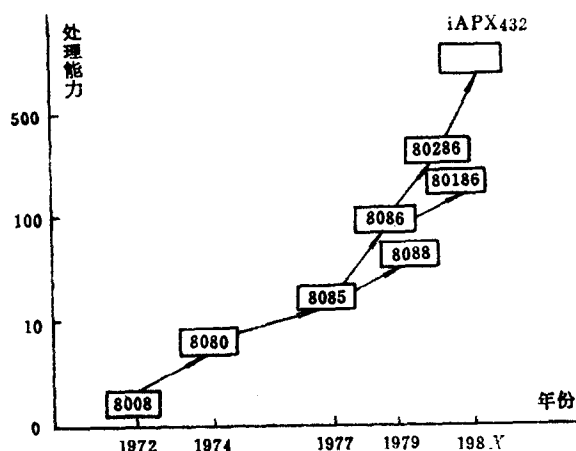


图 1-1 INTEL系列微型机的发展过程

另外，INTEL公司还得到了诸如美国、日本、西德等国家很多大公司（例如 Semiconductor, Siemens, National, NEC, Toshiba, 富士通等）的支持。畅销全球的IBM个人计算机采用INTEL 8088作为CPU，第二商用计算机厂家NCR采用INTEL 8086作为CPU。这些因素也加强INTEL产品的竞争能力。

目前，计算机正沿着两个不同的方向发展。一方面是小巧、价廉的微型机，另一方面是性能强、价格贵的巨型机。微型机不仅可以自成体系，单独使用，而且还可能成为未来计算机世界中各种

巨型机的基本组成单元。用微型机来组成巨型机将是计算机发展的一个重要途径。

从微型机的发展过程来看，一方面是小微型机的微型化。由于可以继承小型机比较成熟的软件和硬件资源，老用户的设备更新比较方便，可以节省系统开发和人员培训的开支。因此，目前仍不失为微型机发展的一个主要方向。另一方面，根据超大规模集成技术的发展，吸取以往系统结构设计的优点，果断地抛弃旧机器设计中为了考虑系列化而背上的沉重包袱，使微型机的发展能轻装前进，这也是目前得到广泛注意的一个发展趋势。当然，这种发展方向需要在硬件及软件开发方面付出一定代价，需要一个完善的阶段。

从另一个角度观察，目前微型机正沿着系列化、多机化、软件固化、虚拟存储系统、功能分布和网络计算机的方向发展。

系列化产品具有向上兼容性，例如INTEL公司的8008→8048→8080→8085→8086/8088→80186/80188→80286和MOTOROLA公司的6801/5→6800→6809→68000等就是系列化产品的典型例子。

在多机系统方面，Z8000的多机调用功能较强，ZiLOG公司采用的功能分布网络计算机适用于事务处理，用五台iAPX432组成的多机系统具有很强的处理能力。可以预计，用多机系统和功能分布网络计算机可以取代中型计算机系统，而价格可以降低一个数量级，并能克服集中控制中的“瓶颈”弱点。

由于微型机的不少系统软件逐渐成熟，为降低系统成本、提高系统运行速度和可靠性，实行软件固化是一个很重要的发展方向。

INTEL公司推出的80286和iAPX432是向虚拟存储系统发展的有代表性产品。

目前，微处理器正处于从16位向32位发展的过渡阶段。32位微型机是80年代初期国际固体电路会议上的中心议题。从1981年起，INTEL、HP、IBM、BELL及美国半导体公司都先后宣布试制出了32位微型计算机。这些32位微型机的主要特点是采用流水线控制，流水线语言的系统结构，支持高级语言调度和调试，具有开发操作系统的专用指令。

软件的生产率。让我们满怀信心来迎接以32位机为标志的微型机新时代的到来。

## 第二节 INTEL8086/8088系列微型机的基本特点

从结构体系来看,8086/8088是8080/8085对称扩展的产物。它们的地址总线为20位,具有寻址1MB存储空间和64KB I/O空间的能力。8086的内外数据总线都是16位的。由于8088是按照面向字符串处理系统来设计的,所以它采用16位内部数据总线和8位外部数据总线。但8086和8088软件完全兼容,都能处理8位或16位数据。这两种CPU的内部结构十分相似,都是由执行部件和总线接口部件两部分组成的。二者的执行部件完全相同,负责全部指令的执行,并向总线接口部件提供数据和地址。其中包含算术逻辑部件、暂存器、标志寄存器和一组通用寄存器。四个16位通用寄存器AX、BX、CX和DX可以直接用于处理16位数据,也可以把其中任一寄存器分为两个8位寄存器(即可把AX分为AH和AL, BX分为BH和BL, CX分为CH和CL, DX分为DH和DL)用来处理8位数据。另外四个16位通用寄存器SP(堆栈指针)、BP(基址指针)、SI(源变址)和DI(目标变址)只能作为16位寄存器使用。除了少数几条片脚外,执行部件与外界是隔绝的。总线接口部件由段寄存器、内部通信寄存器、指令指针和指令队列等组成,负责执行所有外部总线周期。它可以把段寄存器的内容和由执行部件提供的偏移地址合成为20位物理地址。指令队列用于暂时存放从存储器中取来的指令代码,以供执行部件逐条取出执行。执行部件和总线接口部件所进行的操作是相互独立的,从而可以提高指令的执行速度。

8086/8088包含了8080/8085的全部指令,较好地处理了向上兼容问题。8080/8085系列8位机用户的应用软件可以很容易地移植到8086/8088系列16位微型机上。此外,8086/8088又增加一组功能很强的指令,既可以处理8位数据,也可以处理16位数据,既可以进行不带符号的算术运算,也可以进行带符号的算术运算,并具有有效的可中断字节串操作和位操作功能,还专门设置协调多处理器工作的指令。

8086/8088采用变字节指令格式。指令长度为1~6个字节。与8080/8085相比,更加丰富了指令的功能和类型。数据的长度、来源及操作方式也有所不同。根据不同需要对指令中每个字段的功能做了具体规定。这些功能字段的不同组合,形成具有各种功能的变字节指令。这样,不仅可节省占用的空间和提高指令执行的速度,而且大大提高指令的灵活性,便于使用、阅读和记忆。

8086/8088具有六种寻址方式:直接寻址、基址寻址、变址寻址、基址变址寻址、字符串寻址和相对寻址。

为加强8086/8088系统微型机系统的功能,INTEL公司相继推出8089输入/输出处理器和8087数值数据处理器。8086/8088与8089及8087配合组成的多处理器系统的功能相当于PDP-11/45和IBM 370/145等中小型计算机。从结构上看,这种多处理器系统采用功能分布方式,克服集中控制方式下可能出现的阻塞和卡死现象。它把各种不同性质的任务分派给可并行工作的各个专用处理器,使任务的执行过程既简单又迅速,且可根据不同需要,在系统中采用不用配置方式,从而提高系统硬件的利用率。多处理器组合方式不仅具有功能极并行处理能力,而且可使系统具有功能隔离能力,当某一部分功能失效时,不致破坏整行。

在计算机发展史上,输入/输出一直是计算机系统的“瓶颈”。目前,这一问题仍在不少计算机系统中严重存在。追根溯源,这是由于外设的工作速度往往远低于计算机的处理速度。为了解决这一问题,已经研制出各种专用器件,取得不少经验。

最初的计算机是按单用户设计的。那时,输入/输出系统的设计任务主要是要解决好运算器、内存和I/O设备在速度上存在着巨大差距,而输入/输出过程则由程序员自行安排。由于分时系统的出现,就需要输入/输出系统具备对各用户程序的输入及运算结果输出的管理能力。这样,I/O系统就显得格外重要,其功能的强弱不仅会影响各用户输入/输出所占用的时间,而且还会影响CPU和内存的利用率。随着计算机系统不断发展和应用领域日益扩大,系统输入/输出的数据量迅速增加,输入/输出设备的种类也明显增多,从而加强I/O系统的重要性。

微型机输入/输出系统的发展大概可以分为三个阶段。图1-2绘出INTEL系列微型机I/O端口的发展过程。第一代I/O端口是由TTL门电路组成的,即采用TTL随机逻辑控制方式。用这种方式管理输入/输出时,需要CPU对外设的每一个输入/输出电平进行控制。第二代I/O端口是一些专用的外围端口芯片,例如通用的并行端口8255、串行端口8251、定时器8253、DMA控制器8257等。这些芯片比TTL随机逻辑控制电路优越,但其输入/输出仍在相当大程度上依赖CPU的支持,每当它们读入/发送一个数据元时,几乎都需要请求CPU中断当前的任务而为之服务。

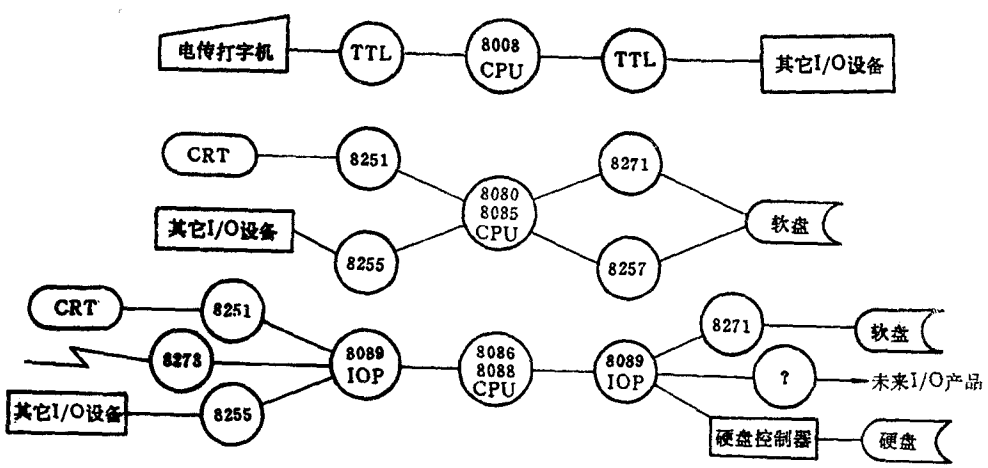


图 1-2 INTEL 系列 I/O 端口的发展过程

8089输入/输出处理器是解决CPU和I/O端口之间关系的第三代产品,它把CPU的处理功能和DMA存取功能有机地联系在一起。它不仅具有高性能的DMA(直接存储器存取)功能,且可在数据传送过程中对数据进行处理。在一定条件下,8089所进行的数据传送和处理操作可与CPU的操作并行,从而可以节省CPU有关I/O操作的开销,明显提高系统的运行速度。

由于8089输入/输出处理器同系统的其余部分可并行操作,因而为发展模块化系统奠定了基础。8089可以采用模块化的I/O系统、操作系统和应用系统。改变个别模块,不会对其它模块产生影响。

8089不仅使8086/8088容易与8位或16位外围设备实现接口,还可使I/O操作的干预。这时需要频繁I/O操作是很重要的。8089具有两个I/O通道,每个

据传送速率为2MB/s。8089有面向I/O操作的五十多条指令以及灵活的总线结构，从而组成一个多用途强功能的I/O处理系统。

8087数值数据处理器是专为提高8086/8088系统微型机的数值运算能力而设计的。它是8086/8088CPU在最大模式下的协处理器，在指令一级提供高精度的整数运算和浮点运算能力，其中既包括加、减、乘、除等普通运算，也包括一般计算机所不具备的平方根、乘幂、对数、三角函数等运算。

在科学研究、工程设计、事务管理、过程控制、航海、航空、航天等许多领域中，常需对大量数据进行实时处理，并进行三角函数、对数函数、指数函数等超越函数在内的各种算术计算。在以往的微型计算机中，甚至一般中、小型乃至大型计算机中，这些计算大多是凭借软件来实现的。每进行一次函数计算就需要调用一段子程序。在调用请求不太频繁的情况下，采用这种处理方式是完全合理的。但当数值运算子程序的调用过份频繁时，就会明显影响计算机的运行速度。因此，对于复杂的数据处理，应寻求更有效的方法。

提供硬件支持是解决上述问题的有效途径。8087正是为了适应这一需要而诞生的。INTEL公司在不断改进数值运算器件的努力中，8087是一个成功而且比较先进的发展成果。

早在70年代初期，INTEL公司就作出了把微型机计算能力从整数加减扩展到对实数进行处理的许诺。图1-3说明开发8087器件所经历的过程。1977年，该公司采纳了表示实数

的浮点标准格式，所设计的FPAL浮点运算库就是利用这一标准格式的第一个成果。实际上，FPAL是8080/8085的一组子程序，用单精度的实数（32位）实现算术运算和有限的几个标准函数的运算。一次FPAL乘法大约需要1.5ms。高速运算部件iSBC 310<sup>TM</sup>实际上是实现FPAL功能的一块iSBC<sup>TM</sup>单板机，它可以把执行一次单精度乘法的时间压缩到100 $\mu$ s左右。INTEL公司制造的8080/8085系列单片算术处理器8232不仅可接受单精度（32位）且可接受双精度的操作数。它执行一次单精度乘法的时间与iSBC 310<sup>TM</sup>大致相同，而执行一次双

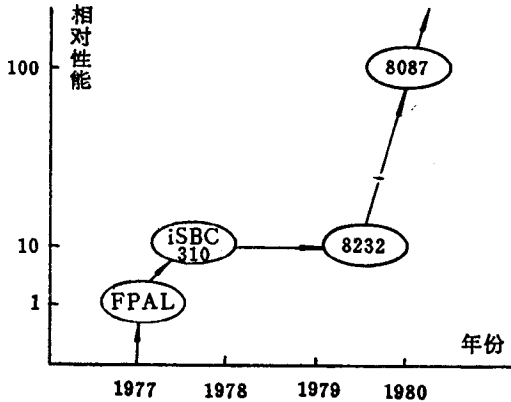


图1-3 8087开发过程

精度乘法所需的时间约为875 $\mu$ s。1979年，电气电子工程师协会IEEE推荐一种小型及微型机进行浮点运算的工业标准，其宗旨是便于各类计算机之间数值处理程序的移值，并为开发正确可靠的软件提供统一的编程环境。8087就是在这种环境中对单精度及双精度等各种类型的数据进行数值运算的单片硬件。由于前面提到的几种产品所采用的数据格式也符合IEEE标准，因而8087同INTEL公司的早期数据处理产品是兼容的。由图1-3可看出，8087的执行速度约为FPAL的100倍，为8232的10倍，8087执行一次单精度乘法运算只需约19 $\mu$ s，执行一次双精度乘法运算也只需要27 $\mu$ s。8087对系统性能的改善不仅仅表现在指令一级运行速度加快方面，而且8087可并行地与CPU一起对指令进行译码，在8087执行数值指令时，CPU仍可并行地执行别的操作，因而明显提高整个系统的吞吐能力。

8087虽然具有很强的数值和数据处理能力，但它必须同8086或8088一起使用，即只能作为8086或8088的协处理器。加上8087后，可使8086/8088的数值处理能力提高100倍左右。

8087的系统，在8086/8088指令的基础上，又增加了几十条指令，而且8087内部包含8个