计算机技术丛书

8087应用和程序设计

《计算机技术》编辑部

8087应用和程序设计

(适用于IBMPC和其他PC)

译者　马启文　王　宏

校者　苏　文

致　谢

我已经断定，一本书在任何时候获得出版的唯一原因是，作者控制某个人出现精力的时间和给此计划一种推动力的时间。至少这一点对于本书是肯定真实的。

罗伯特J.勃累蒂（RobertJ.Brady）公司的人员、编辑和职员共同愉快地在一起工作。

在付诸出版之前对本书的体裁和正确性进行了审定。我有幸拜托三位审稿者，他们对最后的定稿作出了重要贡献。非常感谢彼得.诺顿（Peter Norton）、里.司坎龙（Leo Scanlon）、和安蒂.浮哈伦（Andy Verhalen）。

Intel公司十分慷慨地允许使用已获版权的材料和提供技术信息。有关8087的权威性来源是Intel的iAPX86、88用户手册，从手册中我获益非浅。

我的妻子，谢莉.伦得贝格（Shelly Lundberg），对原稿作了最后的检查。当然，我从最好的朋友那里得到了最宝贵的建议。

在我让别人看到此初始原稿之前，我将原稿送交我姊姊巴勃拉.司塔兹（Barbara Startz）那里进行修改（编辑）。巴勃拉熟悉一些计算机知识及大量写作知识。我并不十分清楚，她在其本身工作每周90小时之后还能挤出时间来修改我的作品，但是我对她的帮助极感愉快。

勿庸说可得出结论，以上提到名字的所有人都有资格对本书中的内容分享功劳，但是，我个人对任何保留的不正确性或粗劣性负有责任。

狄克·司塔兹（Dick）

斯垣福德，加利福尼亚

引　言

让我介释一下，为什么我会受到8087的激励。我已经使用多年的大型计算机作为专业研究的工具，当我购买了个人计算机，我发现，我可以远为比大型机更方便地做许多工作。但又很快发现，我的机器对于大规模数值计算尚嫌太慢。

拥有8087意味着，我现在可在个人计算机上解决许多大问题。当某些问题仍应赋于大型机（往往如此）时，我的个人计算水平已经扩展了10倍。

8087不仅速度快，而且非常易于使用。无论你主要是一个“程序用户”还是主要是一个“程序编写员”，你将发现，8087是一种不寻常的装置。希望你会感到，“8087应用及程序设计”是一本既有趣而又有教育性的入门书。

一、本书为谁服务？

·希望了解8087能做什么工作的读者（尤其第Ⅰ部分，1—4章）；

·希望学习如何对8087进行程序设计的读者（尤其是第Ⅱ部分，5—8章以及第Ⅲ部分中的12章）；

·希望在个人计算机上配置数值演算（Crunching）应用程序的读者（尤其第Ⅲ部分，9—15章）

第Ⅰ部分在完全非技术水平上来描述8087的能力。若你正考虑购买8087，并希望了解有关8087相兼容的硬件和软件，第Ⅰ部分愿为您服务。

第Ⅱ部分和第Ⅲ部分服务于更有技术倾向的读者。当我们“从头开始”时，某些先验的程序设计经验是有用的。你完全可以不是一位专家，而且本书并非是一本计算机导论。

第Ⅱ部分（5—8章）给出8087指令的深刻描述。我们也讨论8088汇编语言程序设计的某些基础。我们特别注重于汇编语言和BASIC程序的连接，包括一种极为详细的交互式会晤，利用这种会晤，我们把一个汇编语言程序同解释型和编译型BASIC两者连结一起。

第Ⅲ部分集中在应用。我们在第Ⅲ部分中开发许多有用的8087汇编语言子程序。你可使用这些程序作为例子来学习更多的8087程序设计技术，或者以“烹调全书”（Cookbook）方式来使用这些程序（第Ⅲ部分12章也包括某些8087高级指令的说明）。

二、如何阅读本书

我已在编写时谨慎从事，使你能按照自己的心愿从一部分跳到另一部分。请不要感到拘泥于从头到尾的阅读。

大多数读者会发现，第Ⅰ部分是有益的和容易阅读的。若你希望编写8087汇编语言程序，集中在第Ⅱ部分（若你是一名有经验的8087程序员，你可跳过第Ⅱ部分转到第Ⅲ部分中的应用方面）。假若你对应用感兴趣，但并不注重于详细的程序设计细节，请阅读第Ⅲ部分。若你需要检查某些内容，往往可返回到第Ⅰ部分。

最后，你可按“烹调全书”方式来使用本书的这些程序。你不需要了解为什么一个程序用某种方法编写或内部如何操作，若你希望得到迅速的回答。如果一个程序是足以有用的，你希望自己修改它或编写一个类似的程序，你可稍后转回到“如何与为什么”。

三、烹调全书（Cookbook）

好几章是由引言开始的，然后说到一个符号

The Cookbook

在这个符号下，你将找到出现在本章中的程序的清单，并附带有程序目标的简要说明以及所要求的输入和输出。当你希望迅即查到一个程序，就使用cookbook。

我们化费相当多的时间来讨论为什么用某些方式来做某些事情。若你希望运行这些程序，但并不建立自己的程序，你就不需要阅读“如何与为什么”的内容。请扫描一下这些内容，它描述你通向这些程序所需求的信息。

四、关键的数字演算

除了有关8087和有关数值编程的大量细节之外，本书提出一种关键的方法来对付繁琐的计算工作。我们的对策渐渐从两种编程准则来形成：

· 10%的程序代码引起90%的程序执行时间；

· 建立一个工作程序的费用正比于代码长度的平方，而不考虑所用程序设计语言的能力。

认真的程序员们有时付出巨大的努力，错误地编写“有效的程序”。一种更好的对策是识别具有90%计算重担的10%代码。重新编写此10%以达到最大有效性，编写其余的90%以达到最大清晰性。

寻求有效性经常导致用汇编语言来编写程序。由于汇编代码可以10倍于等效BASIC代码的长度，汇编语言程序的调试就要困难 100 倍。使用汇编语言来编写一个完整的程序几乎没什么意义，而用汇编语言来编写关键的子程序代码就具有意义。采用这种方式，我们以汇编语言编程成本的一小部分获得几乎汇编语言速度的全部优点。

实际上，当我们认识到许多数值编程问题使用相同的基本子程序，就会做得更好。对数组求和的一个8087汇编语言程序是比BASIC中的FOR／NEXT循环更复杂一些。但我们只需编写和调试8087程序一次。若这样，重复使用子程序可能比每次对数组求和来编写一个FOR／NEXT循环更容易。计算机科学家将这种设想的子程序再使用称呼为“模块化程序设计”。作为模块化8087子程序的方便性和能力的一个例子，看一看第14章中的统计软件包。

实际上，你可做得更好。对于许多数值计算需求的8087子程序按排在本书中（并在可选的软盘片上）。当我们希望你决定学习8087的所有功能以及编写你自己的专用子程序时，更欢迎你这样开始，从这些书页内完整提取这些子程序，并将它们放置到配备8087的个人计算机中来使用。

五、硬件和软件要求

本卡中的这些程序运行在以Intel8087数值数据处理机和Intel微处理机的 8088／8086系列为基础的计算机上。除了8088和8086之外，这个系列包括188，186和286微处理机及相关的8087型号。这些程序全部在IBMPC机上进行开发和测试。所有的定时都假定处理机是一个

5MHz的8088。定时只是近似的（例如，IBMPC的运行大约比标准定时要慢5％。以8086为基础的机器将稍微快些）。对BASIC程序给出的定时参照不带8087的解释型BASIC，除非另外授予资格。

8087汇编语言程序能从解释型Microsoft BASIC或编译型Microsoft BASIC所编写的程序中作为子程序被调用，正如在IBMPC上所利用的那样。这些程序假定，以8087兼容格式来存贮数据（参看第3章—8087兼容软件的广泛讨论）。这些程序将在8087不兼容（Pre—8087）的BASIC版本下运行，但你将需要加上附录中的Microsoft—Intel转换程序。

为了汇编本书中的程序，你将需要一个汇编程序，它认识整个Intel助记符指令系统（提出警告，IBMPC宏汇编程序不认识8087指令助记符，虽然它将产生8087指令。若你乐意重新编码8087助记符，使其进入8088ESCape指令，你仍能使用这个汇编程序。在可选软盘片上，我们早已对助记符重新编码，所以你能使用IBM宏汇编程序）。因为BASIC是占优势的个人语言，我们已经编写出所有的由BASIC调用的程序，以替代在大型机上更通用的FORTRAN或某些其他语言。若你希望由一种使用不同内部惯例的BASIC语言来组合这些程序，你就必须重新编写若干指令。这些程序全部运行在Microsoft BASIC下面，它使用IBMPC上的1.1版PC—DOS操作系统。假若你正使用另一台计算机或不同的软件，某些次要细节可能有所不同。

六、关于责任的否认和限制

1.合法的申明

本书作者和出版者以及任何伴随的软件在此否认有关本书中的程序和信息所表达的及意含的任何保证与保用期。作者和出版者对故障（无论是直接的还是间接的）都不负责任。这些产品在一种“好像是”的基础上进行销售；既不表达或意含对无论什么目标的适应性和商品性保证。

2.人员的申明

我们已经尽最大努力确信，给出的所有信息是正确的以及所有的程序可工作起来。然而，在几百页手稿中以及成千行代码中可能潜伏着错误。本书及软件的目的是教授。当你使用本书的程序时，你必须完整地测试它们。假若你的程序设计的大多数已使用BASIC，请专门留意本书中有关错误处理的部分。汇编语言由于其那种特点要比用高级语言编写的程序更难防止错误。

尽管我处处小心，若你认为发现了错误，请写信通知我(C/O RobertJ.Brady Company, Bowie, Maryland, 20715)，让我能修改将来的版本。

3.商　　标

本书中使用如下的商标：

· IBM, IBM Personal Computer, IBMPC和PC—DOS是国际商业机器公司的商标。

· 8086、8087、8088、186、188、286、Numeric Data Processor和iAPX是Intel公司的商标。8087和8088指令助记符由Intel拥有版权。

· Microsoft和MSDOS是Microsoft公司的商标。

Apple Ⅱ + 是Apple计算机公司的商标。

· DEC—2060和VAX是Digital Equipment公司的商标。

· IMSL是IMSL公司的商标。

目　　录

第一章　分转到秒

配有8087的个人计算机有三个引人注目的特性。即，方便、精确和高速。本书包括了把8087用于实际计算时的每一件有关事项。本章叙述8087及其最主要的应用。

为了理解8087，我们用科学的和分析的方法贯穿全书。无论在哪里，只要可能，我们都从一般原理进行分析。即，为什么这样编程——包括使计算机工作的各个技术细节。为使讨论具体化，每一个一般原理都结合实际应用加以说明。8087功能很强，而且易于使用。我们希望，本书会使你思路开阔，也给你带来兴趣。

§1. 方便性如何

8087设计的重点是使用方便。使用8087，就好象原始运算一样方便。作为8087的用户，第一步就特别方便。只要在个人计算机上加上8087，就能照常运行程序（想要使用8087，还需要它的BASIC版本或其它软件），你不必做任何进一步工作，就能期望得到运行速度的改善，其范围在百分之二十到十倍之内。

如果要高效率地利用8087硬件能力，你就需要专门为8087设计软件。第三章将进一步讨论如何物色和挑选8087软件。

有关8087兼容软件最重要的实际知识是对硬件的了解。8087扩充了现有处理器能力，而且不妨碍通常的处理器操作。因此，存在8087时，任何在8087之前设计的软件(Pre—8087）都能正常连续运行。

百分之一百“向上兼容性”是一大优点，但也有不足之处。在加上8087的系统中，采用8087不兼容软件（Pre—8087）编程完全不能提高速度。例如，早期IBM个人计算机的运行速度为一分钟或二分钟。如果把8087加在这种机器上（为了写这本书，在我的机器上加了一块8087）。那末，所有的解释BASIC程序和编译BASIC程序都能正确运行。但是，并不见得快。因此，当我们作出关于加上8087以后速度效益的说明时，也隐含使用8087兼容软件的说明。

（对于少量重新编程工作，你可以使用8087不兼容BASIC版本和其它软件。第三章将讨论这个问题。但是，如果你愿放心一点，本书中的所有程序就使用8087不兼容软件来编写）。

购买用于8087的软件时，你会意识到一个潜在危险。尽管不一定，但是，很可能陷于困境，你会分不清具有8087优点的软件和8087不兼容软件之区别。更多的讨论见第三章。

如果具有打算使用8087的BASIC版本或其它程序设计语言版本，你就能运行所有通常程序。当与Pre—8087情况相比较，做大量数值计算的程序会加速运行。如果确实担负着繁重的数字演算，你最终还应使用一个专门的高速8087子程序组成的程序库。

本书第三部分（第九章～第十五章）包括最重要的数值计算子程序。你应阅读有关如何

使用每一个子程序的说明，把子程序输入计算机，并把它们与你的BASIC程序合并在一起（随本书提供的磁盘上，已经为你写入和汇编好子程序）。与8087不兼容BASIC相比较，这些子程序把运行速度提高了十倍到二百倍（在少数情况下，速度改善可高达惊人的五百倍）。

本书第二部分（第五章～第八章）为更高级地使用8087（写自己的子程序）做准备。通过贯穿全书的例子，你将认识到，由于8087的周密设计，用汇编语言对8087编程比较方便。当看了本书的例子和指令以后，你将能方便地编写自己的专用程序。

§2. 精确性如何

除非你对正确答案无所谓，否则，方便地编写程序和高速地执行程序不是主要手段。8087最重要的标志是极其精确。8087有三个提高精确性的特点。

* 内部计算产生比BASIC双精度数多十一位的精度，即相当于三位附加十进数。

* 内部计算范围很大。8087能表示大至10^{4932}、小至10^{-4932}的数。因此，在中间运算阶段，运算很少出现“上溢出”或“下溢出”。事实上，精度和数的范围都比大多数传统主机（一般指中、大型机）更大。

* 8087设计有大范围的出错管理，并实现自动的、适度的矫正。结果，很象运行了简单的“纸和笔”算法。当出现某些错误时，8087继续正常运行，而不是产生错误的答案。

§3. 快速性如何

配有8087的个人计算机的快速性如何？你可仔细地将其与标准主机或没有8087的微机作一比较。

8087的速度确实能与价值几十万美元的主机相媲美。这也许是有关8087的最值得注意之点。8087的速度比五十万美元的机器慢若干倍，但是，它便宜得多（超过若干倍）。

精确的比较总是冒险的。但是，若干数字会使你体会到8087的速度。进行两个数字的乘法，中速的主机大约需要1～5微秒；超级小型机需要1微秒；五万美元的台式小型机大约需要3微秒。高效率的8088软件做两个数字的乘法大约化400微秒（用双精度大约要900微秒）。在个人计算机上加一块便宜的8087以后，同样工作只要20～30微秒。

最初，廉价的微型机也可代替大型机实现数字演算。但带有8087的个人计算机是大型机价格的1／10～1／100，它的速度是大型机的1／4～1／20。当大型机对某些任务总是比微型机有更高的性能价格比时，配有8087的个人计算机是第一台与大型机家族竞争的实用微型机。

大多数个人计算机用户可能更关心如何使8087加速自己的运算，而不是与大型计算机中心相比较。个人计算机加上8087以后，要根据应用情况和使用8087的方法来决定速度效益（读了本书以后，你将懂得怎样最大限度地得到效益）。根本之点，你要认识到，8087是一个数值数据处理器，8087仅加速包含数值计算的程序。如果个人计算机仅用作字处理，那末，大约百分之九十九与8087不相干。但是，如果你演算临时的数字，那末，加上8087就象在七月四日的焰火展览上经销焰火一样。

8087的许多速度效益都由8087的使用方法所决定。因此，本书的第一章为：

8087把分转到秒

§4. 专门的速度比较

8087能否得到充分利用，这取决于所使用的软件及8087的硬件速度。第四章将进一步讨论速度问题。这里，仅作初步的探讨。

8087为你做些什么，这根据软件执行各种“开销”任务所化的时间与进行数字计算所化的时间之比来决定。8087加快了数字运算，但是，根本没有减少执行总开销的时间。当你结合8087运行低开销、高速度的程序时，可以期望得到表1—1所示的结果。

表1—1 BASIC—8087速度标准（时间单位：秒）

程　序	50×50矩阵乘法	5000个平方根
BASIC	1200	52
8087子程序	8	0.35

在表1—1中，时间的比较指8087不兼容BASIC和本书以后将采用的专用8087程序。这里是8087结合高效软件所能得到的典型改善。根据应用情况，8087硬件增加了十倍到五十倍的速度（其余让给低开销软件）。如果你把8087与高开销软件一起使用，你几乎不会看到任何改善（建立在计算机中的解释BASIC一定是高开销软件）。由于8087仅加快数值运算。因此，相对来说，这样的软件在数值运算上化费较少的时间，开销时间和数值运算时间之和几乎不会低于上表所示的总数。然而，得到的改善还是惊人的。

§5、使用8087需要的配置

当然，你需要一块8087。8087既可以作为个人计算机最初配置的一部分，也可以把它加在现有的机器上。或许，你可以把8087加在任何以Intet8088或8086系列为基础的个人计算机上。加上一块8087的难度由制造商提供的机器是否设计有8087安装位置所决定。即使没有预备好位置，也许也能加上一块8087。但是，这样做需要一些相当的专门技术。

一些制造商愿意提供8087的位置确实是好消息。尤其是当IBM（以及那些制造兼容个人计算机的公司）推出它的第一台个人计算机时，它就在主板上专门为8087留了一个空插座。插入8087很方便（没有任何人帮助，我就装上了8087）。事实上，这样做比在计算机内部的扩充槽上加一块印刷电路板还要方便（如果你确实对计算机内部一无所知，那末，请某个人来帮助你。因为，计算机毕竟是件十分贵重的设备）。

一旦打开机器以后，只要一分钟就能插上8087。可是，同时还应做其它的硬件调整。你的计算机也许已把8087不兼容软件（如解释BASIC）编进了只读存贮器（ROM）里。如果来自制造商的新的8087兼容软件是有效的，那末，你同时要改变ROM芯片。

拥有非Intel8088、8086芯片系列计算机的人怎么办？他们能够得到8087的速度效益吗？不幸，回答是有保留的“否”。8087只能与Intel系列一起工作。但是，由于Intel系列是如此普及，以致现在许多有进取精神的公司出售带有Intel处理器的电路板。它们能适用于Apple和其它计算机。其中，一些板带有8087，一些则备有加上8087的插座。在原来的处理器下，这些板不会加快程序的运行，但是，允许你利用本书的程序和其它8087兼容软件。

第二章 Intel 8087 蕊 片

§1 处理器与协处理器

计算机的“大脑”是CPU（中央处理器部件），IBM个人计算机和许多其它“第二代”个人计算机的“大脑”是Intel 8088。8088是建立在单片上的、完整的、通用的中央处理器部件。它有一套完整的指令。它们是，八位和十六位整型运算指令，编程逻辑指令和输入／输出指令。象大多数微处理器一样，8088没有那种以大型机为基础的高级数学指令。

通过增加新的高级数学指令，Intel 8087数值数据处理器扩展了8088的指令系统。8087高速硬件所做的数学操作，如果用软件完成的话，则需要数千条指令，8087硬件操作比相应的软件操作快十倍到二百倍。

根据程序员的观点，8087在8088指令系统之外增加了附加的指令，并使附加的处理器寄存器有效。为什么不在一个蕊片上包括全部功能，反而产生一个附加器件呢？有这样一些原因：

＊8087是一种非常高级的计算部件，在一块单片上包括有75000个晶体管。因此，8087即使限于数值处理。它也比一般的8088复杂得多（而且贵得多）。制造二块单独的蕊片，可以降低开发成本，并使用户及系统制造商能根据不同用途编排出相应的系统。

＊在8087使用之前，许多年来，8088（及其16位总线系列蕊片8086）就可在普通市场买到。当有了8087以后，为了使机器能方便地升级，一些制造商在设计16位个人计算机时，都留有一个空插座。在IBM个人计算机中，插座上贴有“协处理器插座（Co—Processor socket）”的标签。

＊由于8087和8088是两个器件，因此，它们应同时执行指令。作为一个实际运行的程序，这就意味着，在8087完成一次数值运算期间，8088就准备下一个运算。

在本章的其余部分，我们一般地叙述8087的功能，。第五章提供更为详细的技术论述。

§2 8087 概 述

作为8088的协处理器，在8088接收指令时，8087“监视”指令。在允许8088指令通过期间，8087处理它自己的指令。在允许8087指令通过期间，8088也监视所有的指令，同时处理它自己的指令。8088为8087提供一个重要的服务，在看到8087指令时，8088计算每一个必要的存贮器地址，并使地址对8087有效。然后，8088立即转到下条指令。这样的协处理器设计使得8087和8088能同时执行指令。可以认为，它提高了整个系统的性能。

8087结构的主要特点是八个80位的数据寄存器。这些寄存器组成一个传统的“后进先出堆栈”。这种构造技术既能导致快速向量操作，又能使高级编译语言生成高效代码（第五章包含对后进先出堆栈操作的进一步讨论）。80位的寄存器使8087能进行极其精确的运算。在

8087指令系统识别内存中七种不同数据类型期间，所有数据都转换成8087运行时的80位内部表示法。这样。程序员避免了两种数据类型转换所带来的麻烦。

§3 指令分类

8087的68条指令分成六类（分类是描述8087功能的一种简便方法，使用8087，不一定要记住这些分类），这六类是：

* 数据传输类（在第六章中讨论）

这些指令来回传输8087和存贮器之间的数据，并使8087内部寄存器之间的数据慢慢地流动。

* 算术类（在第六章中讨论）

8087指令系统中心是加、减、乘、除。再加上一些其它运算，例如，平方根和绝对值。

* 超越类（在第十二章中讨论）

8087硬件具有计算对数和三角函数的功能（甚至大型机也很少有这些指令）。

* 常数（在第六章和第十二章中讨论）

8087内部有最常使用的七个常数，例如0，1，π 等。

* 比较（在第六章和第七章中讨论）

这些指令用于进行小于、等于、大于和其它类似的测试。

* 处理器控制（在第十二章中讨论）

这类指令使程序员能控制8087的全部动作。为了控制程序流动。其中一些指令还用于把比较指令和8088的跳转指令连接在一起。

§4 数据类型

在第五章中，将深入研讨七种正式的8087数据类型。可是，对于大多数一般的8087编程事项，只有一些事实是真正重要的。在BASIC中，直接有效的数据类型只有整型、单精度和双精度。通常，仅用后两种数据类型支持数值数据。如果主要把8087用于科研程序，那末，你只要记住有关数据类型的三个事实。

1.单精度数（8087术语称为短实数）精确到6位或7位（十进制数），占四个存贮器字节。

2.双精度数（8087术语称为长实数）精确到15位或16位（十进制数），占八个存贮器字节。

3.暂存实数用于所有8087内部运算，它保留18位（十进制数）以上的精确性。在存贮器中，暂存实数占十个字节。

如果主要进行数值转换，那末，以上三种数据类型的使用大概占百分之九十五，但是，8087还承认另外四种数据类型。

1.整数（8087术语称为字整数）在存贮单元中占两个字节，它主要用于检索数组和其它数据结构。BASIC和8087使用同样的表示法表示整数。

2.短整数占四个字节，当最大（带符号）字整数为32,768时，一个短整数可大到二十亿。

3.长整数占八个字节，长整数比双精度实数精确二位或三位以上。它能表示大到10^{18}的数。

4.压缩十进制数表示法用于商业和数据管理操作。一个压缩十进数使用十个存贮器字节。它能表示18位十进数。不象前面三种数据类型，压缩十进数不用二进制表示法，而用十进制表示法。每个数字（0～9）用四个二进制位表示，这些十进数每两位被压缩成一个字节。

对比之下，8088硬件承认的数据类型限于一个字节和二个字节的二进制整数以及短的压缩十进制数。8087不兼容BASIC和其它高级语言的所有数值处理软件都以整数为基础。8087取消了这样的软件要求。因此，配置8087的系统不仅速度快，而且程序占用空间少，数值运算结果更为可靠。

§5 怎样使计算机得到8087的能力

在下一章中，我们将讨论8087的软件。为什么有的软件与8087兼容，有的软件与8087不兼容。要知道这些原因，复习一下建立8088／8087协操作的基本知识将是有益的。

8088的指令系统设计允许对新数据补充。8088有一条称为escape的指令。8088知道，执行escape指令，实际需要运行8087。所以，8088必须忽略这条指令，并且让8087去处理它。8087使用的指令是种种不同的8088换码（escape）指令。

当8088和8087都装在机器中时，我们可以认为，系统具有象大型机一样的扩充能力。为了正确运行，软件使用escape指令时，系统必须存在8087。8087不兼容软件不能简单地使用escape指令。因此，它得不到新性能带来的效益。

如果用机器语言编写程序，你将知道是否使用了escape指令。可是，在使用计算机的大部分时间里，你不可能控制这样详细的内部细节问题。下一章中，我们讨论一些各种各样的8087兼容软件和不兼容软件。