# 80486/80386
# 系统原理与接口大全(下)
## —80486系统原理

艾德才 陆 明 李文彬 编著

清华大学出版社

# 80486/80386
# 系统原理与接口大全(下)
## ——80486 系统原理

艾德才　陆明　李文彬　编著

清华大学出版社

(京)新登字 158 号

## 内 容 提 要

本书由上、中、下三册组成，分别为 80386 系统原理、接口技术、80486 系统原理。

下册介绍 80486 系统，对 80486 芯片内组织结构及其系统原理给以详细论述，内容包括：80486 CPU、流水线操作原理、片内 Cache、二级 Cache、浮点部件、存储管理技术及部件、总线、调试和自测试、多任务处理、初始化处理、保护机制、输入/输出、中断等。

本系列书内容全面、实用，是计算机应用人员必备的工具书，也可作为高等院校教学用书以及计算机培训教材。



**图书在版编目(CIP)数据**

80486/80386 系统原理与接口大全(下)：80486 系统原理/艾德才等编著. —北京：清华大学出版社，1996
ISBN 7-302-02137-6

Ⅰ. 80… Ⅱ. 80… Ⅲ. 微型计算机，80486-系统理论 Ⅳ. TP 36

中国版本图书馆 CIP 数据核字（96）第 04105 号

**出 版 者**：清华大学出版社（北京清华大学校内，邮编 100084）
**责任编辑**：焦金生
**印 刷 者**：通县大中印刷厂
**发 行 者**：新华书店总店北京科技发行所
**开　　本**：787×1092 1/16　**印张**：23　**字数**：544 千字
**版　　次**：1996 年 7 月 第 1 版　1996 年 7 月 第 1 次印刷
**书　　号**：ISBN 7-302-02137-6/TP・1008
**印　　数**：0001—8000
**定　　价**：24.00 元

# 前 言

90年代是微型计算机技术空前发展的10年。90年代是32位微型计算机时代。32位微型计算机是当今微型机的杰出代表。不论是速度还是性能，32位微型计算机系统都超过了70年代末期、80年代初期中小型计算机的水平。因为32位微型机已显露出前辈机所没有的机器视觉、人工智能和声音识别的特征，所以它在专家系统、机器人以及控制系统、工程工作站、办公室自动化、事务处理、科学计算和工程计算、人工智能、过程控制、软件开发、CAD/CAM、公共服务、教育和训练等各方面的应用潜力已初见端倪。在本世纪内，32位微型计算机将会占领整个微型机市场。由于80386、80486是当代32位微处理机中的杰出代表之一，微型计算机的主流产品——膝上机的CPU将会全部采用80386、80486甚至是80586。为紧紧盯住90年代微型计算机的潮流，紧紧盯住国际微型计算机潮流，我们不失时机地编写本书，奉献给急需了解80386、80486以及微型计算机接口的广大用户和科技人员。

本书共分三册。上册描述的是80386系统原理。围绕80386系统原理，主要介绍了80386CPU、存储管理方式、数值协同处理器80387、高性能的DMA控制器82380、高速缓冲存储器Cache及其控制器82385、图形协同处理器82786等外围芯片的体系结构和工作原理等。同时也列专题讨论了与其系列机兼容的有关问题。

中册描述的是与我们用户密切相关的微处理机外围设备及其接口。其中包括我们用户须臾离不开的键盘、显示器、打印机、磁盘等看得见摸得着的外围设备。同时也对支持显示器工作的MDA、CGA、MCGA、EGA、VGA、8514/A等适配器以及各种视频服务器原理和功能给以剖析解释。对80386、80486常采用的Multibus Ⅰ、Multibus Ⅱ总线以及IBM的最新专利技术微通道都作了详尽剖析。与高级语言接口、与DOS接口等也给以适当说明。

下册介绍80486系统。80486是由提高了效率的80386微处理机、增强了性能的80387数值协同处理器、一个完整的Cache及其控制器组合而成的。根据其特征，下册着意对80486芯片内组织结构及其原理给以论述，像对80486CPU、流水线操作原理、支持部件、浮点部件等给以独到论述。对80486特有的片内Cache、二级Cache都给以详尽论述，对80486的存储管理、总线、自测试、多任务处理、初始化处理、保护机制、输入/输出、中断等方面也进行了有针对性的详细剖析。

本书力求全面、具体、丰富实用，力求成为一本有关80386、80486及其接口的微型计算机系统大全和工具书。在编写过程中，得到王俊珍、刘文萃、刘文瑜、张兰月、朱淑文、刘晓月、陈毓弘、苗君秋、胡琳、周士松、孙丙国、刘秀云、程家莲、艾菲等同志的帮助，在此深表谢意。

由于编者水平所限，时间仓促，新技术新词汇难于仔细推敲，书中不足和谬误在所难免，敬请计算机界老前辈、同仁和读者不吝指正。

编　者

1995年10月于天津大学

# 目 录

# 第31章 80486 CPU

Intel 公司于 1989 年 4 月推出了 80486 芯片。同年 IBM 就生产出以 80486 芯片为基础的 IBM PS/2 机器。

Intel 80486 微处理机是由主处理机部件、数值协同处理器部件、一个拥有 8K 字节的高速缓冲存储器部件 Cache 构成的。在 80386 微处理机上，这 3 个部件是各自独立的芯片。在 80486 中，Intel 采用了先进的集成电路工艺，把这 3 个部件集中到一块芯片上。

80486 借用 80386 指令流水线、RISC 的设计思想，继续保持与早期的 80x86 软件的兼容。80486 芯片内含有 120 万只晶体管，比 80386 的 4 倍还多。80486 用静态 RAM 作为指令数据共用的 Cache。它的目的是当处理机忙于对指令和数据重复检索时，减少必须耗用的等待状态的数量。当 CPU 需要指令和数据时，它首先在片内 Cache 中检索。静态 RAM 的 Cache 能在 25ns 内给以响应，这比动态 RAM 需 70－100ns 响应时间快许多。80486 还采用了成组传送方式，能在一个时钟周期内传送 32 位数据。这种成组传送速度为 80386 的 2 倍。

Intel 把浮点部件 FPU 也集中到 80486 芯片上。这个方案的性能比把 80386 芯片与独立的 80387 数值协同处理器芯片装配起来好得多。这是由于改进了部件之间的接口，缩短了处理机各部件间的距离所致。这样，80486 能在芯片内实现全部浮点运算。这种方案不仅维持了与 80387 的软件兼容，而且还使 80486 的计算能力远远超过 Weitek3167 数值协同处理器。

第一篇曾提到，80386 采用的是 CMOS Intel Ⅲ 1.5μm 生产工艺，CPU 芯片内集成了大约 27 万只晶体管。而新研制的 80486CPU，采用的是 CMOS Ⅲ 1μm 生产工艺，CPU 内的晶体管数量是 80386 的 4 倍还多。在这个芯片内还有一个由密集的静态 RAM 存储单元组成的片内高速缓冲存储器，为改善性能打下了良好的基础。

## 31.1 概　　论

80486 微处理机芯片内拥有一个整数处理部件、一个浮点处理部件、存储管理部件和高速缓冲存储器 Cache。之所以要把这些各自独立的部件集成到一个芯片之内，其用意十分明显，就是为了使芯片内各部件之间的信号传送是以超大规模集成电路的运行速度传送，当然要比用印刷线路板的运行速度快出许多。此举不仅增加了集成度，减少了插件板空间，而且还降低了成本，简化了设计。

根据所采用的时钟速度以及使用的环境领域不同，80486 微处理机的整体性能要比 80386 高出 2—4 倍不等。80486 也继承了 80386 的一些优点，其内也配备了一个分段存储

管理和分页存储管理的保护式的存储管理机制。由于在80486芯片内实施了指令流水线操作技术，因而有效地减少了指令处理时间。80486的片内快速的存储管理机制和配备的片内Cache，有效地满足了系统性能对存储器响应时间的需求。

80486装备的总线比80386使用的总线的运行速度要快。虽然二者所采用的总线都是32位宽。但是，由于80486微处理机引进了一系列诸如单倍频时钟(single-frequency clock)、支持奇偶校验、成组传送时钟周期、允许高速缓冲周期、无效Cache周期以及8位的数据总线等项技术，80486微处理机的总线操作速度比80386要快也在情理之中。80486微处理机采用单倍频时钟有两大好处。首先，简化了系统设计。第二，去掉了80386微处理机使用的2倍频时钟。这样就减少了80486微处理机在高速运行时的射频传播和简化了时钟的生成。

在进行读传送操作时，80486微处理机可以使用成组传送周期，而且读传送操作经常需用多个总线周期。成组传送周期是以每个时钟周期传送32位的速度连续不断地进行传送。相比较而言，80386微处理机在进行数据传送时，每传送一次至少也需要两个时钟周期时间。在成组传送期间，外部Cache、交叉存取存储体或者使用静态寻址的DRAM都可以达到零等待状态的性能。

在80486微处理机内，指令执行所需的时钟个数也比80386少。由于80486微处理机采用了指令流水线技术，它可以连续不断地依每条指令占用一个时钟的执行速率高速执行。而且80486微处理机的内部Cache也支持每个时钟周期执行一条指令的执行速率。为了更加有效地支持实时多项任务处理和多用户系统中的任务切换，80486微处理机与80386一样，也允许单条指令的执行或进行一次中断，以便能成功地完成一次任务转换。

借助于片内自测试功能，还可以对80486的部件进行测试。内部自测试对内部寄存器来说是行之有效的方法。借助于内部自测试功能，也可用汇编语言对Cache及转换旁视缓冲存储器进行测试。

**1. 80486基本特征**

80486微处理机提供了如下14个显著的特征。

① 兼容性

80486微处理机实现与8086、8088、80286、80386、80386SX等系列机在二进制代码下的全兼容。

② 全32位整数处理机

80486微处理机使用32位宽的算术逻辑运算部件ALU和8个通用32位寄存器，既可以执行8位的、16位的算术运算和逻辑运算操作，也可以执行32位的算术运算和逻辑运算操作。

③ 各自独立的32位地址通道和数据通道

可以直接对4千兆字节的物理存储器直接进行寻址操作。

④ 单周期执行

有许多种80486微处理机的指令，在一个时钟周期时间内就能完成。

⑤ 片内浮点部件

80486 微处理机芯片内集成进了一个浮点部件，它对 IEEE 标准所规定的 32 位的、64 位的以及 80 位的数据格式给以全面支持。80486 微处理机芯片内的这个浮点部件与它的前辈机 8087、80287、80387 等数值协同处理器在二进制情况下完全兼容。

⑥ 片内存储管理部件

80486 微处理机芯片内配备有一个存储管理部件 MMU，用来对存储器地址实施管理和对存储器空间给以保护，以维护存储系统的完整性。像 UNIX 和 OS/2 操作系统中常要用到的多项任务处理和虚拟存储环境等就要用到片内存储管理部件。而且 80486 微处理机芯片内的存储管理部件对存储器的分段管理和分页管理都提供全方位的支持。

⑦ 片内高速缓冲存储器 Cache

80486 微处理机芯片内装配有一个 8K 字节的片内 Cache。对 Cache 的访问像对微处理机内的寄存器进行读取操作那样迅速。总线的动作一直被跟踪，以便对 Cache 内经常被替换的内容给以检测和跟踪。根据需要既可以将片内 Cache 内容作废，也可以给以刷新处理，以使外部 Cache 控制器在多处理机环境内维持与片内 Cache 的一致性。

⑧ 外部 Cache 控制

80486 微处理机对外部 Cache 提供写回(write-back)和刷新控制，以使微处理机在多微处理机环境内维持 Cache 的一致性。

⑨ 指令流水线

在 80486 微处理机内，取指令、分析指令(对指令进行译码)、执行指令以及对指令地址的转换等操作都是重叠进行的。这种指令流水线操作方式对绝大多数指令来说，可以做到每个时钟周期内执行完一条指令的执行速率。

⑩ 成组周期

80486 微处理机采用这种成组传送周期，可以作到在每一个时钟周期内都可以从存储器读出一个新的双字。使用这种成组传送周期，80486 微处理机能非常迅速地将片内 Cache 和指令预取部件填满。

⑪ 写缓冲(write buffers)

80486 微处理机在经历了一次写操作之后，仍可以连续地执行操作，根本就不需要等待微处理机总线上的写操作。

⑫ 总线补偿(bus backoff)

在 80486 微处理机总线周期期间，若又有一个总线主控设备需要对总线实施控制，这时 80486 微处理机就会将它的总线信号悬浮起来。而当总线再次变成有效时，再重新启动它的总线周期。

⑬ 指令的再启动

在由于试图去访问存储器不成功而产生的一次异常事故之后，程序可以紧接着继续其执行。80486 微处理机的这一特征对于要求分页的虚拟存储器来说十分重要。

⑭ 动态(大小规模)总线

外部控制器能动态地变更数据总线的有效宽度。总线宽度可以是 8 位的、16 位的，也可以是 32 位的。

### 2. 操作方式

80486 微处理机可以作到与 8086、80286、80386 系列机在目标代码级上完全兼容。也就是说，为 8086、80286、80386 微处理机编写的软件不用修改即可在 80486 微处理机上运行。80486 微处理机与 80386 微处理机相似之处是，80486 也有两种操作方式，一为实方式，二为保护方式。

① 实方式

当 80486 微处理机复位时，或者在加电时总是被初始化成实方式。这种操作方式与 8086 微处理机一样拥有同样的基本结构，只不过是在这种操作方式下，80486 可以访问 32 位的寄存器。在这种操作方式下，其寻址机构最大寻址范围为 1 兆字节，其中断处理又与 80286 的实方式下的中断处理一样。在这种操作方式下，几乎所有的 80486 微处理机的指令都可以使用。但有一点不同的是，它的缺省操作数的大小是 16 位的。为了能在这种操作方式下也可以使用 32 位的寄存器和寻址方式，就必须加上指令前缀。80486 之所以要采用实方式这种操作方式，是因为它还需要建立并使用保护方式这种操作方式。

② 保护方式

保护方式也被称之为保护的虚拟地址方式。当各种应用程序在 80486 的保护方式下运行时，80486 微处理机的各种能力都可以淋漓尽致地表现出来。在保护方式下，除了可以实施分段保护之外，还可以任意选用分页保护。80486 微处理机的线性地址空间为 4 千兆字节，而虚拟的存储空间可达 64 兆兆字节。目前所有的为 8086、80286、80386 编写的软件都可以在 80486 微处理机的保护机制下运行。在保护方式下，其寻址机制比起在实方式下的寻址机制来则更加完美，其技术也更加先进。

说到底，虚拟 8086 方式只不过是 80486 微处理机保护方式下的一种子方式，它的最突出的特点是，8086 程序可以使用 80486 微处理机保护方式下的分段和分页保护机制运行。所以，这种操作方式比起在实方式下运行的 8086 程序来说其灵活性表现得更加充分。使用这种操作方式，80486 微处理机可以在所使用的 80486 的操作系统之下，既可以同时执行 8086 的操作系统和应用程序，也可以同时执行 80286 的应用程序和 80486 的应用程序。

### 3. 存储管理

80486 微处理机是一种全 32 位体系结构的微处理机，其内不仅配备有一个片内存储管理部件、一个数值协同处理器部件，而且还配备了一个规模为 8K 字节的高速缓冲存储器 Cache。80486 不仅拥有 80386 的全部 32 位微处理机的特征，更重要的是它的性能在 80386 的基础上得以全面地大幅度地提高。比如，80486 为了更好地适应新的使用环境、新的应用软件，它又在 80386 指令系统的基础上新增加 6 条指令。80486 的片内存储管理部件 MMU 与 80386 的存储管理部件完全兼容。更令人兴奋的是 80486 芯片内还集成进去一个经改进后性能增强了的 80387 数值协同处理器。从应用角度上来说，凡是为 80386 微处理机、80387 数值协同处理器，甚至为 8086/8087 系列机编写的应用软件，不用修改就可以直接在 80486 微处理机上使用。

由于80486微处理机芯片内配备有片内高速缓冲存储器Cache，80486在运行时，就可以频繁地使用存放在Cache内的数据和指令，减少了访问主存储器的次数，这样也就减少了访问外部总线的次数，从而有效地提高了80486的运行速度。在80486体系结构的设计上采用了精简指令系统计算机RISC的设计思想和技术，这样就有效地减少了指令周期时间。由于总线采用的是成组传送的方式，所以能够非常迅速地填充满Cache。

80486微处理机配备的存储管理部件MMU是由分段部件和分页部件组成的。通过分段部件可对逻辑地址空间进行分段管理，这样就可以对使用的数据和指令的地址给以再分配处理，此项措施就可以作到数据和指令共享全部80486微处理机资源的目的。分页部件是在分段部件之后的又一存储管理机构，它是在分段操作之后，再对存储器段进行分页处理，而对分段部件操作又是透明的一种存储管理过程。分页操作并非是强制性的，而是供任选的，系统软件根据需要可以进行分页处理，也可以不进行分页处理。每一个存储器段又可以进一步分成少至一个多至若干个大小规格为4K字节的段。为了在80486上能实施虚拟存储管理，80486微处理机对所有分页和分段故障都能给以重新开始处理。

80486微处理机的存储器可以被进一步分成一个或多个可变长的段，每一个段的大小在规模上可高达4千兆字节(即$2^{32}$个字节)。每一个段都拥有隶属于它的段的属性，属性内容包括段的基地址、段的规模大小、段的类型(比如说堆栈段、指令代码段、数据段)，而且段属性内容还应有保护特征等。80486微处理机上的每一项任务最多可拥有16381个段，每个段大小可高达4千兆字节。所以，每项任务可以拥有16381×4千兆＝64兆兆字节的虚拟存储空间。

分段部件是80486微处理机存储管理部件中一个很重要的组成部分，它的作用非常重要，它给操作系统和应用程序相互之间的隔离提供四级保护。执行保护的硬件可以做到使整个系统具有高度的完整性。

80486微处理机芯片内的浮点部件(即80387数值协同处理器)与算术运算、逻辑运算部件并行操作，并且给各种各样数据类型的运算操作提供相应的算术、逻辑指令。除此之外，80486微处理机芯片内的浮点部件还能执行构造在芯片内的诸如正切、正弦、余弦、对数、指数等众多的超越函数运算。而且这个浮点部件执行的浮点运算也完全符合ANSI/IEEE 754—1985标准。

80486微处理机片内高速缓冲存储器Cache大小规模为8K字节。它采用的是四路相联映象的结构和写贯穿方案。片内Cache还给芯片外部存储器系统提供了很好的灵活特性，既可以通过硬件也可以通过软件把存储器中的各个区段指定成是可以进行高速缓冲的或是不可以进行高速缓冲的。同时，对Cache而言通过硬件或软件也同样可以把空间指定为一些是允许高速缓冲的，另一些是不允许高速缓冲的。

为了改善性能和增强功能，80486微处理机相应增加了6条80386所没有的新指令。但是80486仍然维持与80386和它的协同处理器80387的绝对兼容性。在设计80486时并没有采取因小失大举措，而是转向采用RISC技术，使用流水线操作而达到高性能目的，同时也没有丧失软件的兼容性。

一般的指令在单个时钟周期内即可完成。唯一例外是控制流变化时要占用较多周期时间。有些复杂的指令在执行时要执行许多行微代码。而对那些只需一个时钟周期即可

完成操作的指令来说，是只用一行执行微代码；由硬件直接控制执行。在执行复杂的浮点指令时使用的也是同类微代码。80386 微处理机使用的存储管理方式，像分段管理和分页管理等，在 80486 微处理机芯片上都得以完全实现。

80486 微处理机最突出技术是内部实行由 5 个操作步骤构成的流水线操作。由于 80486 内部流水线操作和片内 Cache 的有效支援，由复杂指令系统计算机 CISC 所设置的每条指令执行需要两个时钟周期的障碍被突破。80486 微处理机的每条指令执行时间约为 1.7—1.8 个时钟周期。而且，80486 微处理机也是第一个允许在没有流水线延迟的情况下，在单个时钟周期内就能完成装入操作的微处理机，甚至是下一条指令就要使用的数据，也完全可以做到在一个时钟周期内将其装入指定寄存器。于是，即使是执行装入操作和存储操作，也仍然可以在单个时钟周期内完成其操作。

80486 微处理机的片内浮点部件能与整型执行部件并行运行。而且是由单个微代码机(microcode engine)来控制这种并行操作的执行。另外，80486 采用了对异常事故延迟处理技术，致使 80486 片内浮点部件的性能比 80387 数值协同处理器的性能高出 4 倍之多。

## 31.2 CISC 与 RISC

在设计 CPU 时，既要考虑到大幅度地提高性能，又要切实做到与基本设计相容。由于现今集成电路工艺的日臻完善，完全可以做到在一个芯片内集成进更大数量的器件，所以可以利用相容途径。

当今的微处理机都没有因为相容性问题而给设计增加上某些约束条件，都是根据不同的设计要求而决定如何设计。正如在许多新的微处理机结构中就广泛采用了 RISC 设计思想一样。在 80486 微处理机中就采用了 RISC 结构取代了 Intel 系列机中的 CISC 中的某些结构。

一般说来，若欲给 RISC 下一个严格的准确的定义那是比较困难的，甚至是不可能的。把注意力转移到 RISC 和 CISC 两种类型微处理机的差别上来可以说是现实的明智的态度。设计 RISC 类微型计算机的目的之一就是想使整个指令系统更加简单，具体表现在要让绝大多数指令都能够在一个时钟周期内完成其本身的操作。为达到这种目的就要求绝大多数操作使用的必须是寄存器操作数。很明显在 RISC 类微处理机上就需要大量的各种用途的寄存器。所以总是把 RISC 类微处理机结构设计成流水线操作形式。具体说来，在执行流水线操作时是在若干个执行子部件中实现的，而每一个执行子部件执行的也只能是整条指令中的某一部分操作。流水线中的若干个执行子部件按照指令执行的先后顺序各自完成了自己的操作，整条指令的流水线操作也即完成。

在设计微处理机的 CPU 时，还必须考虑到程序的整个执行环境。对绝大多数应用程序来说，都是用高级语言编写而成的。在执行这种由高级语言编写的应用程序之前，首先要用编译程序把它们翻译成硬件能直接执行的二进制机器码。对于 CISC 和 RISC 这两种类型的微处理机来说，硬件和软件之间的关系都是一样的，如图 31.1 所示。

CISC 和 RISC 类微处理机之间的一个重要的区别是它们能直接执行机器指令的能

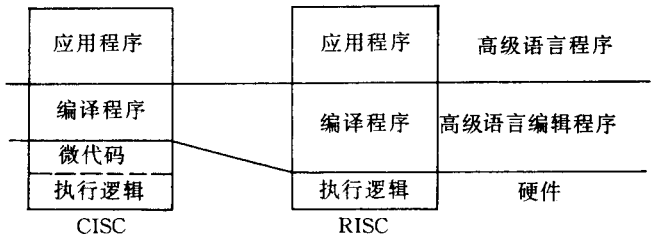


图 31.1 CISC 和 RISC 结构的软件、硬件

力。我们知道，Intel 8086 系列机都是典型 CISC 结构的微处理机，它们的机器指令有许多种类，不管是在指令大小规模上还是在操作数放置的位置上，还是在对操作数的寻址方式上都有所不同。8086 这种 16 位微处理机在执行加法指令时，使用的就是字操作数和字节操作数。这两个操作数既可以都取自寄存器，也可以一个取自寄存器，而另一个则取自存储器。在对存储器操作数进行寻址时也可以用多种方式进行。

说到灵活性问题，选择灵活性来简化编译程序的工作是要付出代价的。像 CISC 类微处理机最初那样，在对机器指令进行译码和执行机器指令时都是在 CPU 中进行的。而且在译码时会把机器指令再细分成更加简单且能直接执行的指令单位——即微代码指令再执行。

当一个寄存器操作数与一个存储器操作数进行加法运算时，微代码程序首先要计算出存储器操作数的地址，然后才能把存储器操作数取到 CPU，在 CPU 中来完成加法操作，最后再把计算结果写回到存储器存放起来。

指令的执行时间是以时钟周期个数来测量的，是由指令的微代码程序的长度决定的。这种允许指令可以变化的方法并非一无是处，它的好处是可以允许比较复杂的指令的执行。但由此也会出现这种情况，即在执行微代码程序中的某些执行步骤时，很可能会使 CPU 中的某些电路处于空闲状态。

设计 RISC 类微处理机的基本思路是，简化整个指令系统。通过进一步完善编译程序，让 CPU 中的所有电路都高效地发挥它们应起的作用，在机器运行时的所有工作时间内，让所有电路都处于有条不紊的忙状态。图 31.1 所展示的 CISC 和 RISC 软件硬件关系中就有用 RISC 类微处理机编译程序来处理由 CISC 类微处理机的微代码程序完成的工作的关系示意图部分。

根据 RISC 类微处理机的设计思想，要求指令系统中的绝大多数指令都能在一个时钟周期内完成其操作。更确切地说，那就是每一个时钟周期都要完成一条指令的操作。这就要求 RISC 类微处理机在执行操作时，所有操作数都必须取自寄存器。很显然，RISC 类微处理机必须装备有大量的通用寄存器，在存储器与寄存器间进行数据交换时，只需使用简单的存/取指令即可完成，而其使用的寻址方式也非常简单，这样也就有效地避免了由于尚需计算存储器地址而出现的浪费时间的现象。

当然，有些指令需要几个时钟周期时间才能完成。比如说一条加法指令，要用到两个

寄存器，执行的是两个寄存器中的内容相加。首先必须把两个操作数送到运算器，在运算器内完成加法运算，然后再把和存放到目的寄存器。

通过两个数相加操作我们可以看出，只有在执行加法操作的中间步骤时加法电路才是处于忙状态。为了使每一个时钟周期内都能完成一条指令操作目标，在RISC类微处理机上就采用了流水线操作技术(pipelining technique)，让几条执行指令重叠操作。由于采用了流水线操作技术，就不会再出现没有任何作用的空操作周期。但在程序执行过程中，有可能会遇到一条指令要用到前一条指令的运算结果情况、程序转移情况以及其他一些控制转移指令情况等，这些情况都会延缓流水线执行。但是，这些情况完全可以通过编译程序给以圆满解决。在编译程序对用高级语言编写的应用程序进行编译时，就把机器指令的执行顺序事先安排好，以便最大限度地挖掘流水线的使用能力。

由于RISC类微处理机采用了大量的通用寄存器，此举从根本上提高了CPU的运算速度，这是RISC的一大优点，但在多任务处理的环境下，它又成了RISC的一个缺点。这是因为在进行任务转换时，都必须把现行任务所用各寄存器内容保存起来，在下一项任务开始运行之前，又必须把下一项任务的各寄存器的内容恢复起来，由此就增加了CPU的保存信息和恢复信息的难度和操作量。

要达到每一个时钟周期完成一条指令的目标，对存储器的结构和存取速度要求也很高。对RISC类微处理机的整个系统要有一个全面考虑，既要经济又要实用。所以在RISC系统中一定要采用高速缓冲存储器Cache这种灵巧而快速的存储设备。由于使用的Cache规模都不太大(相对于主存储器而言)，这样就减少了RISC类微处理机芯片的复杂程度。同时也减小了争用RISC芯片内空间的要求。就可以在RISC芯片上配置上一个存储容量比较大的片内Cache。

## 31.3 寄 存 器

80486微处理机所具有的寄存器的种类和数量都非常多，它的寄存器既有80386微处理机中使用的全部寄存器，又有80387数值协同处理器中使用的各种寄存器。80486微处理机的寄存器按类可分为如下几类：

(1) 基本体系结构寄存器：① 通用寄存器；② 指令指针寄存器；③ 标志寄存器；④ 段寄存器。

(2) 系统级寄存器：① 控制寄存器；② 系统地址寄存器。

(3) 浮点寄存器：① 数据寄存器；② 标记字寄存器；③ 状态字寄存器；④ 指令和数据指针寄存器；⑤ 控制字寄存器。

(4) 调试和测试寄存器。

对于基本结构寄存器和浮点寄存器来说，应用程序就可以对它们进行访问。而对于系统级的寄存器来说，则只有特权级为0级的程序才可以对它们进行访问，只有系统级的程序才可以使用它们。对调试和测试寄存器而言，也同样是只有特权级为0级的程序才可以对它们进行访问。