

623183

83750-3

'84 INTEL

微型计算机系统器件手册

(三)

上海交通大学 微机研究所
科技交流室

1986.11



本册译者：李世祥、韩朔瞭、徐子亮、王洪澄、赵正校、

张光耀、王建中

总 审 校：徐子亮、杜毅仁

目 录

第四章 iAPX286微处理器	(4-1)
数据图表	
iAPX 286/10 带有存储器管理和保护的高性能微处理器	(4-1)
80287 80 位 HMOS 数值处理器扩展部件	(4-62)
82284 iAPX286 处理器的时钟发生器和准备就绪接口	(4-84)
82288 iAPX286 处理器的总线控制器	(4-91)
第五章 iAPX 432 微主机系统	(5-1)
数据图表	
iAPX 43201/43202 通用容错数据处理器	(5-1)
iAPX 43203 容错接口处理器	(5-40)
iAPX 43204/43205 容错总线接口部件和存储器控制部件	(5-71)
第六章	
一、存储器控制器	(6-1)
应用篇	
使用 Intel 8202A 和 8203 的动态 RAM 与 iAPX 86,88 系统的接口	(6-1)
8203/8206/2164A 存储器设计	(6-39)
8207 动态 RAM 控制器与 iAPX186 的接口	(6-43)
8207 超前动态 RAM 控制器与 iAPX286 的接口	(6-48)
转载文章	
动态 RAM 控制器协调存储器系统	(6-55)
技术文章	
一种面向系统的 RAM 控制器	(6-63)
一个 NMOS 动态 RAM 控制器	(6-73)
数据图表	
8202A 动态 RAM 控制器	(6-76)
8203 64K 动态 RAM 控制器	(6-91)
8206/8206-2 检错和纠错部件	(6-107)
8206-2 16 位检错和纠错部件	(6-129)
82C03 CMOS 64K 动态 RAM 控制器	(6-140)
8207 超前状态 RAM 控制器	(6-155)
8208 动态 RAM 控制器	(6-202)
用户手册	

可编程序的 3207	(6-222)
RAM 接口.....	(6-227)
微处理器接口.....	(6-234)
带有 ECC 的 8207 (8206)	(6-241)
附录.....	(6-244)

第四章 iAPX 286 微处理器

iAPX 286/10

带有存贮器管理和保护的高性能微处理器

(80286, 80286-6, 80286-4)

- 高性能处理器(比 iAPX86 高六倍)
- 大的地址空间
 - 16兆字节物理空间
 - 每个任务1千兆字节虚存
- 集成的存贮器管理, 四级存贮器保护及支持虚拟存贮器和操作系统
- 两种 iAPX86 向上兼容的操作方式:
 - iAPX86 实地址方式
 - 保护的虚地址方式
- 时钟速率的范围
 - 80286 为 8MHz
 - 80286-6 为 6MHz
 - 80286-4 为 4MHz
- 可选的协处理器:
 - iAPX286/20 高性能80位数值数据处理
- 完整的系统开发支持:
 - 开发软件: Assembler, PL/M, Pascal, FORTRAN, 和系统实用程序
 - 内部电路仿真程序(ICE™-286)
- 高带宽总线接口(8 兆字节/秒)
- 可在 EXPRESS 中应用:
 - 标准的温度范围

iAPX286/10(80286 部件编号)是一种先进的、高性能的微处理器, 具有适用于多用户和多任务的特别完善的能力。80286 具有内在的存贮器保护机构, 它能支持操作系统和任务的隔离以及在任务之间的程序和数据的保密。8MHz 的 iAPX286/10 能提供比标准的 5MHz iAPX86/10 高达六倍的整体功能。80286 具有存贮器管理能力, 它能把每个任务的 2^{30} (1 千兆字节)虚地址空间映照到 2^{24} 字节(16 兆字节)的物理存贮器。

iAPX286 是和 iAPX86 及 88 的软件兼容的。使用 iAPX286 的实地址方式, 80286 是和现存的 iAPX86, 88 软件目标代码相兼容的。在保护虚地址方式下, 80286 与 iAPX86, 88 软件是源代码相兼容, 并且可以请求升级来使用由 80286 的集成的存贮器管理和保护机构支持的虚地址。两种方式的操作均以 80286 的全部功能来进行并且执行一个 iAPX86 和

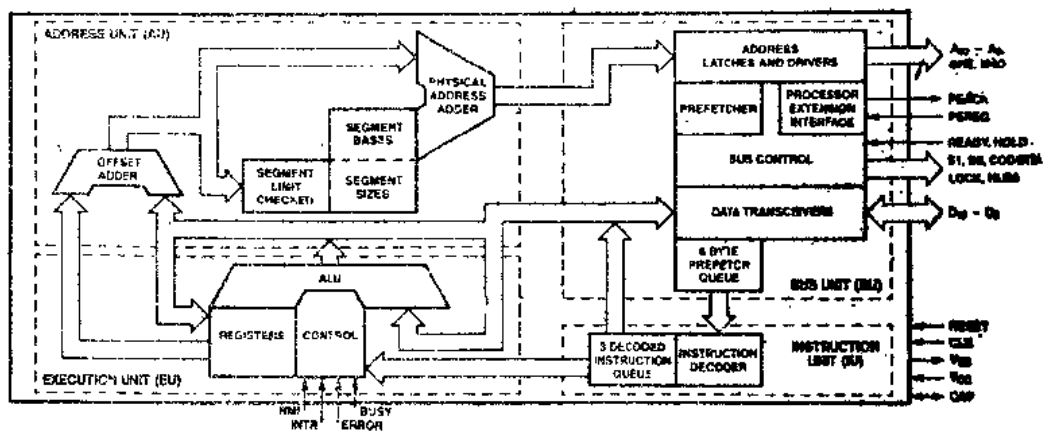
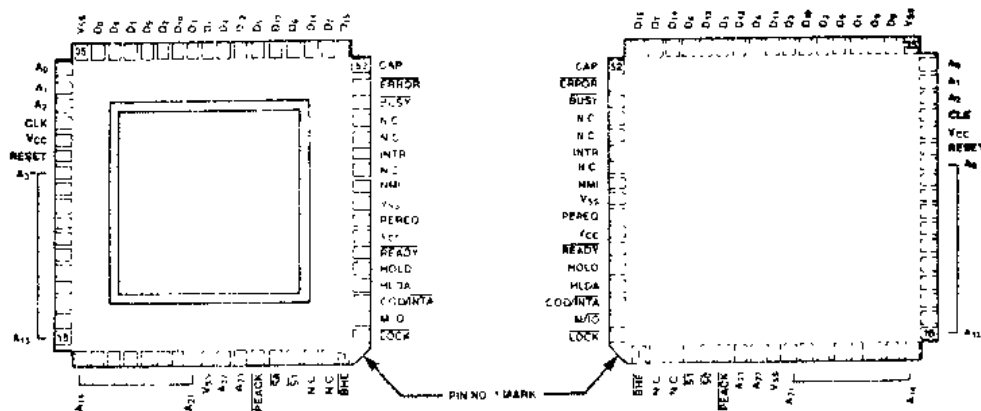


图 1 80286内部框图

Component Pad View—As viewed from underside of component when mounted on the board.

P.C. Board View—As viewed from the component side of the P.C. board



NOTE: N.C. pads must not be connected

图 2 80286引脚结构

88指令的高级集。

80286 为了支持操作系统的有效实现和运行, 提供了一些特殊的操作。例如, 一条指令可以结束一个任务的运行, 保护它的状态, 转入一个新的任务, 装入它的状态, 并开始新任务的执行。80286 还通过提供一种段不在场(Segment-not-present)异常和一些重新启动指令来支持虚拟存储器系统。

表 1 引脚说明

下面是关于 80286 微处理器的引脚功能说明:

符 号	类 型	名 称 和 功 能																																													
CLK		(System Clock)系统时钟提供 iAPX286 系统的基本定时。它在80286 内部被除以 2 以产生处理器时钟。内部的除以 2 电路能由 RESET 输入的低电平到高电平的变换来同步于一个外部时钟发生器。																																													
D ₁₅ -D ₀	I/O	(Data Bus)数据总线在存储器、输入/输出和中断响应周期期间输入数据, 在存储器, 输入/输出周期期间输出数据。数据总线是高电平有效, 而在总线保持响应期间浮空至三态OFF。																																													
A ₂₃ -A ₀	0	(Address Bus)地址总线输出物理地址和输入/输出转接口地址。当数据在引脚 D ₇₋₀ 上传送时, A ₀ 是低电平。在输入/输出传送时, A ₂₃ -A ₁₆ 是低电平。地址总线是高电平有效并且在总线保持响应期间浮空至三态 OFF。																																													
BHE	0	(Bus High Enable)总线高位允许指示在数据总线的高字节 D ₁₅₋₈ 上的数据传送。指定在数据总线高字节的 8 位设备通常使用 BHE 来决定片选功能。BHE 是低电平有效并且在总线保持响应期间浮空至三态 OFF。																																													
<table><tr><th colspan="3">BHE 和 A₀ 的编码</th></tr><tr><th>BHE 值</th><th>A₀ 值</th><th>功 能</th></tr><tr><td>0</td><td>0</td><td>字传送</td></tr><tr><td>0</td><td>1</td><td>在数据总线的高半部(D₁₅₋₈)进行字节传送</td></tr><tr><td>1</td><td>0</td><td>在数据总线的低半部(D₇₋₀)进行字节传送</td></tr><tr><td>1</td><td>1</td><td>保留</td></tr></table>			BHE 和 A ₀ 的编码			BHE 值	A ₀ 值	功 能	0	0	字传送	0	1	在数据总线的高半部(D ₁₅₋₈)进行字节传送	1	0	在数据总线的低半部(D ₇₋₀)进行字节传送	1	1	保留																											
BHE 和 A ₀ 的编码																																															
BHE 值	A ₀ 值	功 能																																													
0	0	字传送																																													
0	1	在数据总线的高半部(D ₁₅₋₈)进行字节传送																																													
1	0	在数据总线的低半部(D ₇₋₀)进行字节传送																																													
1	1	保留																																													
S ₁ , S ₂	0	(Bus Cycle Status)总线周期状态指示一个总线周期的初始状态, 并且和 M/ $\overline{\text{IO}}$ 和 $\overline{\text{COD/INTA}}$ 一起定义总线周期的类型。一旦两者之一或两者都是低电平, 总线就处于 T _s 状态。S ₁ 和 S ₀ 是低电平有效并且在总线保持响应期间浮空至三态。																																													
<table><tr><th colspan="5">80286 总线周期状态定义</th></tr><tr><th>COD/INTA</th><th>M/$\overline{\text{IO}}$</th><th>S₁</th><th>S₀</th><th>开始的总线周期</th></tr><tr><td>0 (低电平)</td><td>0</td><td>0</td><td>0</td><td>中断响应</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>保留</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>保留</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>无; 不是一个状态周期</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>若 A₁ = 1 则暂停; 否则停机</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>读存储器数据</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>写存储器数据</td></tr></table>			80286 总线周期状态定义					COD/INTA	M/ $\overline{\text{IO}}$	S ₁	S ₀	开始的总线周期	0 (低电平)	0	0	0	中断响应	0	0	0	1	保留	0	0	1	0	保留	0	0	1	1	无; 不是一个状态周期	0	1	0	0	若 A ₁ = 1 则暂停; 否则停机	0	1	0	1	读存储器数据	0	1	1	0	写存储器数据
80286 总线周期状态定义																																															
COD/INTA	M/ $\overline{\text{IO}}$	S ₁	S ₀	开始的总线周期																																											
0 (低电平)	0	0	0	中断响应																																											
0	0	0	1	保留																																											
0	0	1	0	保留																																											
0	0	1	1	无; 不是一个状态周期																																											
0	1	0	0	若 A ₁ = 1 则暂停; 否则停机																																											
0	1	0	1	读存储器数据																																											
0	1	1	0	写存储器数据																																											

续 表 1

		<table border="1"> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>无; 不是一个状态周期</td></tr> <tr><td>1 (高电平)</td><td>0</td><td>0</td><td>0</td><td>保留</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>读 I/O</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>写 I/O</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>无; 不是一个状态周期</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>保留</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>读存储器指令</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>保留</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>无; 不是一个总线周期</td></tr> </table>	0	1	1	1	无; 不是一个状态周期	1 (高电平)	0	0	0	保留	1	0	0	1	读 I/O	1	0	1	0	写 I/O	1	0	1	1	无; 不是一个状态周期	1	1	0	0	保留	1	1	0	1	读存储器指令	1	1	1	0	保留	1	1	1	1	无; 不是一个总线周期
0	1	1	1	无; 不是一个状态周期																																											
1 (高电平)	0	0	0	保留																																											
1	0	0	1	读 I/O																																											
1	0	1	0	写 I/O																																											
1	0	1	1	无; 不是一个状态周期																																											
1	1	0	0	保留																																											
1	1	0	1	读存储器指令																																											
1	1	1	0	保留																																											
1	1	1	1	无; 不是一个总线周期																																											
M/IO	0	(Memory/IO Select) 存储器/I/O 选择区分存储器访问和 I/O 访问。若在 T_s 期间是高电平, 则正在进行一个存储器周期或一个暂停/停机周期。若是低电平, 则正在进行一个 I/O 周期或中断响应周期。M/IO 在总线保持响应期间浮空到三态 OFF。																																													
COD/INTA	0	(Code/Interrupt Acknowledge) 代码/中断响应区别指令周期和读存储器数据周期。同时还区别中断响应周期和 I/O 周期。COD/INTA 在总线保持响应期间浮空到三态。它的定时和 M/IO 相同。																																													
LOCK	0	(Bus Lock) 总线封锁表示在当前总线周期之后, 其它的系统总线主设备不能获得系统总线的控制权。LOCK 信号可以由“LOCK”指令前缀直接激活, 或在存储器的 XCHG 指令、中断响应或描述于表访问期间由 80286 的硬件自动地激活。																																													
READY	1	(Bus Ready) 总线准备好结束一个总线周期。总线周期将无限制地延长, 直到 READY 变成低电平为止。READY 是一个低电平有效的同步输入, 它请求与系统时钟有关的建立和保持时间, 以满足正确操作的需要。在总线保持响应期间 READY 被忽略。																																													
HOLD HLDA	1 0	(Bus Hold Request and Hold Acknowledge) 总线保持请求和保持响应控制 80286 局部总线的所有权。HOLD 输入允许另外一个局部总线主设备请求对局部总线的控制。当控制被转让时, 80286 将它的总线驱动器浮空到三态 OFF, 然后激活 HLDA, 这样就进入了总线保持响应状态。该局部总线将保持转让给请求主设备, 直到 HOLD 变成无效, HOLD 的这种变化使 80286 撤销 HLDA, 并重新取得局部总线的控制权。这样, 就终止了总线保持响应状态。HOLD 可以异步于系统时钟。这些信号是高电平有效。																																													
INTR	1	(Interrupt Request) 中断请求请求 80286 挂起当前的程序执行, 并且为一个外部的急迫的请求服务。在标志字中的允许中断位一旦被清除, 中断请求就被屏蔽。当 80286 响应一个中断请求时, 它执行两个中断响应总线周期以读得一个识别中断源的 8 位中断向量。为了确保程序中断, INTR 必须保持有效到第一个中断响应周期完成。INTR 在每个处理器周期的开头被取样, 并且在当前指令结束之前至少保持高电平有效 2 个处理器周期, 以便能在下一条指令之前中断。INTR 是电平敏感 (level Sensitive) 的, 高电平有效, 并且可以异步于系统时钟。																																													
NMI	1	(Non-maskable Interrupt Request) 不可屏蔽中断请求用一个内部提供的向量值 2 中断 80286。不执行中断响应周期。在 80286 标志字中的中断允许位不影响这种中断。NMI 输入是高电平有效, 可以异步于系统时钟, 并且是在内部同步后被边缘触发的。为了正确地识别, 该输入必须至少前 4 个系统时钟周期是低电平并且保持至少 4 个系统时钟周期的低电平。																																													

续 表 1

PEREQ PEACK	I O	(Processor Extension Operand Request and Acknowledge) 协处理器操作数请求和响应把 80286 的寄存器管理和保护能力扩展到协处理器。 PEREQ 输入请求 80286 为一个协处理器执行一个数据操作数传送。 PEACK 在被请求的操作数正被传送时通知协处理器。 PEREQ 是高电平有效, 并且在总线保持响应期间浮空至三态 OFF 。 PEACK 可以异步于系统时钟。 PEACK 是低电平有效。										
BUSY ERROR	I I	(Processor Extension Busy and Error) 协处理器忙和出错给 80286 指示某一个协处理器的操作情况。一个有效的 BUSY 输入停止 80286 程序在 WAIT 和一些 ESC 指令上的执行, 直到 BUSY 变成无效为止。在等待 BUSY 变无效期间 80286 可以被中断。有效的 ERROR 输入使 80286 在执行 WAIT 或一些 ESC 指令时实现协处理器中断。这些输入是低电平有效, 并且能异步于系统时钟。										
RESET	I	(System Reset) 系统复位清除 80286 的内部逻辑并且是高电平有效。80286 在任何时候都可以用在 RESET 引脚上从低电平到高电平的转换来重新初始化, 这个高电平应该保持有效多于 16 个系统时钟周期。在 RESET 有效期间, 80286 的输出将进入下列状态:										
<table border="1" style="margin: 10px auto; width: 80%;"> <tr> <th colspan="2">80286 在复位期间的引脚状态</th> </tr> <tr> <th>引脚值</th> <th>引脚名称</th> </tr> <tr> <td>1(高电平)</td> <td>S0, S1, $\overline{\text{PEACK}}$, A23-A0, BHE, $\overline{\text{LOCK}}$</td> </tr> <tr> <td>0(低电平)</td> <td>M/$\overline{\text{IO}}$, $\text{COD}/\overline{\text{INTA}}$, HLDA</td> </tr> <tr> <td>三态OFF</td> <td>D15-D0</td> </tr> </table>			80286 在复位期间的引脚状态		引脚值	引脚名称	1(高电平)	S0, S1, $\overline{\text{PEACK}}$, A23-A0, BHE, $\overline{\text{LOCK}}$	0(低电平)	M/ $\overline{\text{IO}}$, $\text{COD}/\overline{\text{INTA}}$, HLDA	三态OFF	D15-D0
80286 在复位期间的引脚状态												
引脚值	引脚名称											
1(高电平)	S0, S1, $\overline{\text{PEACK}}$, A23-A0, BHE, $\overline{\text{LOCK}}$											
0(低电平)	M/ $\overline{\text{IO}}$, $\text{COD}/\overline{\text{INTA}}$, HLDA											
三态OFF	D15-D0											
80286 的操作在 RESET 脚上由高电平到低电平的转换之后开始。RESET 的高电平到低电平的变换必须同步于系统时钟。在执行第一个从接通电源后的运行地址中取代码的总线周期之前, 80286 为了内部初始化大约需要 50 个时钟周期。 RESET 的从低电平到高电平的变换同步于系统时钟, 将在系统时钟的下一个高电平到低电平的变换时开始一个新的处理器周期。RESET 的低电平到高电平的变换可以异步于系统时钟; 然而, 在这种情况下, 它不能预先决定在下一个系统时钟期间处理器时钟将会出现哪一个相位。RESET 的低电平到高电平的同步变换只为系统所需要, 系统的处理器时钟必须相位同步于另一个时钟。												
V_{SS}	I	(System Ground:) 系统接地: 0 伏。										
V_{CC}	I	(System Power:) 系统电源: +5 伏电源。										
CAP	I	(Substrate Filter Capacitor:) 衬底滤波电容器: 一个 $0.047\mu\text{f} \pm 20\%$ 12V 电容器必须接在这个引脚和地之间。该电容器用来滤内部衬底偏压发生器的输出。该电容器允许有最大值为直流 $1\mu\text{A}$ 的漏电流。为了使 80286 能正确操作, 衬底偏压发生器必须把这个电容充电到它的工作电压。这个电容在 V_{CC} 和 CLK 到达它们规定的 AC 和 DC 参数以后的充电时间是 5 微秒(最大)。 RESET 可以用来防止在这段时间中由 CPU 产生的假动作。在这之后, 通过加上同步于系统时钟的 RESET 低电平, 80286 处理器时钟可以相位同步于另一个时钟。										

功能说明

引言

80286 是一种先进的, 高性能的微处理器, 它具有支持多用户和多任务系统的特别完善的能力。以应用为标准, 80286 的性能要比标准的 5MHz 8086 的性能高六倍, 同时还提供了和 Intel 的 iAPX86, 88 和 186 系列的 CPU 完全向上软件兼容的能力。

80286 以两种方式进行操作: iAPX86 实地址方式和保护虚地址方式。两种方式都运行 iAPX86 和 88 指令集的一个高级集。

在 iAPX86 实地址方式下, 程序使用多达 1 兆字节的地址空间。在保护虚地址方式下, 也称保护方式下, 程序使用虚拟地址。在保护方式下, 80286 CPU 自动地把每个任务的 1 千兆字节地址映照到一个 16 兆字节的实地址空间。这种方式还提供了存储器保护以隔离操作系统并确保每个任务之间的程序和数据的保密。两种方式提供相同的基本指令集, 寄存器和寻址方式。

下面的功能说明, 首先, 描述对于两种方式来说是共同的基本 80286 结构, 其次, 介绍 iAPX86 实地址方式, 第三, 介绍保护方式。

iAPX286/10 基本结构

iAPX86, 88, 186 和 286 CPU 系列都含有相同的基本寄存器集, 指令集和寻址方式。80286 处理器与 8086, 8088 和 80186 CPU 是向上兼容的。

寄存器集

80286 的基本结构具有如图 3 所示的 15 个寄存器。这些寄存器被分成下列四种类型:

通用寄存器:

8 个 16 位通用寄存器用来存放算术和逻辑操作数。其中 4 个寄存器 (AX, BX, CX 和 DX) 既可作为 16 位的字寄存器, 也可作为两个独立的 8 位寄存器。

段寄存器:

4 个 16 位专用寄存器, 可在任何给定的时刻选择可立即寻址的代码、堆栈和数据段。(关于用法, 参见存储器的组成)

基址和变址寄存器:

通用寄存器中的 4 个, 也可以用来决定操作数在存储器中的偏移地址。这些寄存器可以含有在一个段中特定单元的基地址或变址。寻址方式决定用来计算操作数地址的具体寄

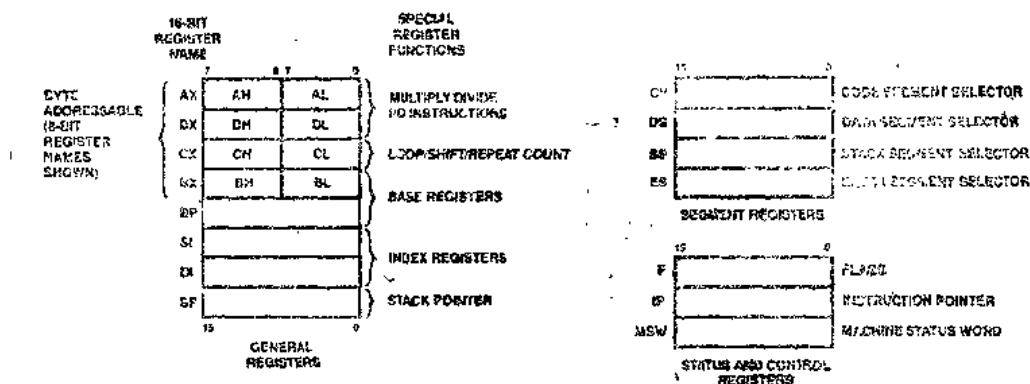


图 3 寄存器集

寄存器。

状态和控制寄存器：

在图 3a 中的三个 16 位专用寄存器，用来记录或控制 80286 处理器某些方面的状态，其中包括指令指示器，它含有下一条顺序执行的指令的偏移地址。

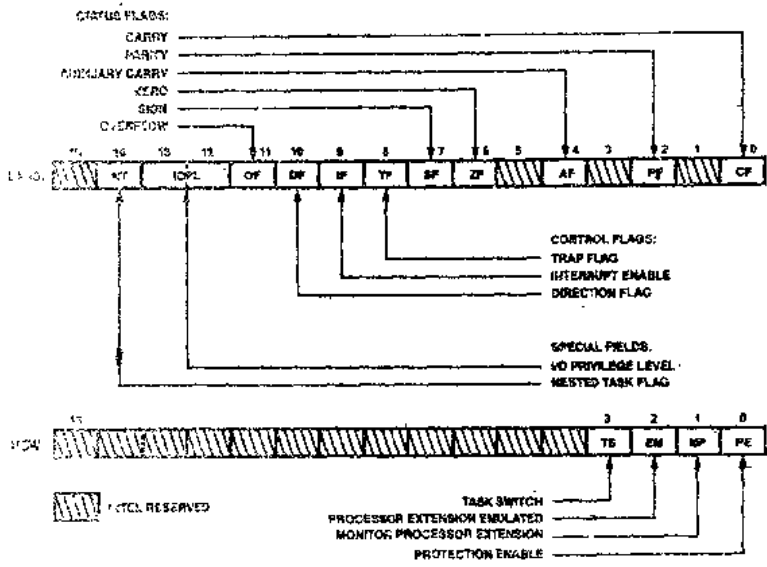


图 3a 状态和控制寄存器的位功能

标志字说明

标志字记录了逻辑指令和算术指令的结果的特点(0, 2, 4, 6, 7和 11 位)并且控制 80286 在给定操作方式下的操作(8 和 9 位)。标志字是一个 16 位寄存器。标志位的功能给出在表 2 中。

表 2 标志字位功能

位	名称	功能
0	CF	进位标志—当最高位进位或借位时建立此标志，否则清 0。
2	PF	奇偶校验标志—当结果的低 8 位含有偶数个 1 位时，设置此标志，否则清 0。
4	AF	当 AL 的低 4 位向高位进位或高位借位给低 4 位时，设置此位，否则清 0。
6	ZF	零标志—当结果为 0 时设置此位，否则清 0。
7	SF	符号标志—设置成与结果的最高位相等(0 表示正，1 表示负)。
11	OF	溢出标志—当结果对于目操作数是一个太大的正数，或太小的负数(包括符号位)时设置此位，否则清除。
8	TF	单步标志—一旦此标志设置，则在下一条指令执行完后，产生单步中断。TF 标志由单步中断清除。
9	IF	中断允许标志—当设置此标志时，可屏蔽中断将使 CPU 把控制转移到中断向量所指定的单元。
10	DF	方向标志—当设置此标志时，使串指令自动增量某个变址寄存器，DF 为 0 时则作自动减量。

指令集

指令集可分为七类：数据传送；算术；移位/环移/逻辑；串处理；控制转移；高级指

令和处理器控制。这些类型总结在图 4 中。

一条 80286 指令可以引用 0 个, 1 个或 2 个操作数。这些操作数可以在寄存器中, 也可以在存储器中或在指令中。0 个操作数的指令(如 NOP 和 HLT)通常是 1 字节长。单操作数指令(如 INC 和 DEC)通常是 2 字节长, 不过其中有一些仅在 1 字节中编码, 单操作数指令可以引用一个寄存器或存储器单元。双操作数指令可以有下列六种类型的指令操作数:

- 寄存器到寄存器
- 存储器到寄存器
- 立即数到寄存器
- 存储器到存储器
- 寄存器到存储器
- 立即数到存储器

双操作数指令(如 MOV 和 ADD)通常是 3 到 6 字节长。存储器到存储器的操作是由需要 1 到 3 字节的专用类型的串指令提供的。指令的详细格式和编码请参见本文末尾的指令集总结。

关于每条指令的详细操作和用途请参见 iAPX286 程序员参考手册 (编号 No. 210498)

通 用 指 令	
MOV	传送字节或字
PUSH	把字推入堆栈
POP	把字弹出堆栈
PUSHA	把所有的寄存器内容推入堆栈
POPA	把所有的寄存器内容从堆栈中弹出
XCHG	交换字节或字
XLAT	字节翻译
输入/输出指令	
IN	输入字节或字
OUT	输出字节或字
地 址 指 令	
LEA	装入有效地址
LDS	使用 DS 装入指示器
LES	使用 ES 装入指示器
标 志 传 送 指 令	
LAHF	把标志装入 AH 寄存器
SAHF	把 AH 寄存器内容存入标志
PUSHF	把标志推入堆栈
POPF	把标志从堆栈中弹出

图 4a 数据传送指令

加 法 指 令	
ADD	字节或字相加
ADC	字节或字带进位相加
INC	字节或字加 1
AAA	加法 ASCII 调整
DAA	加法 10 进制调整
减 法 指 令	
SUB	字节或字相减
SBB	字节或字带借位相减
DEC	字节或字减 1
NEG	取补
CMP	字节或字比较
AAS	减法 ASCII 调整
DAS	减法 10 进制调整
乘 法 指 令	
MUL	无符号字节或字相乘
IMUL	整数字节或字相乘
AAM	乘法 ASCII 调整
除 法 指 令	
DIV	无符号字节或字相除
IDIV	整数字节或字相除
AAD	除法 ASCII 调整
CBW	字节转换为字
CWD	字转换为双字

图 4b 算术指令

MOVS	传送字节或字串
INS	输入字节或字串
OUTS	输出字节或字串
CMPS	比较字节或字串
SCAS	扫描字节或字串
LODS	装入字节或字串
STOS	存放字节或字串
REP	重复
REPE/REPZ	相等/为零时重复
REPNE/REPNZ	不相等/不为零时重复

图 4c 串指令

逻辑指令	
NOT	字节或字取反
AND	字节或字相“与”
OR	字节或字相“或”
XOR	字节或字相“异或”
TEST	字节或字测试
移位指令	
SHL/SAL	字节或字逻辑/算术左移
SHR	字节或字逻辑右移
SAR	字节或字算术右移
循环移位指令	
ROL	字节或字左环移
ROR	字节或字右环移
RCL	字节或字通过进位左环移
RCR	字节或字通过进位右环移
图 4d 移位/环移/逻辑指令	
条件转移指令	
JNBE	高于或不低于等于跳转
JAE/JNB	高于等于或不低于等于跳转
JBE/JNAE	低于或不高于等于跳转
JBE/JNA	低于等于或不高于等于跳转
JC	有进位跳转
JE/JZ	相等/为零跳转
JG/JNLE	大于/不小于等于跳转
JGE/JNL	大于等于/不小于等于跳转
JL/JNGE	小于/不大于等于跳转
JLE/JNG	小于等于/不大于等于跳转
JNC	无进位跳转
JNE/JNZ	不等于/不为零跳转
JNO	无溢出跳转
JNP/JPO	不校验/校验为奇跳转
JNS	无符号跳转
JO	溢出跳转
JP/JPE	校验/校验为偶跳转
JS	有符号跳转

无 条 件 转 移 指 令

CALL	调用过程
RET	从过程返回
JMP	跳转

环 循 控 制 指 令

LOOP	循环
LOOPE/LOOPZ	相等/为零循环
LOOPNE/LOOPNZ	不相等/不为零循环
JCXZ	若寄存器CX = 0 就跳转

中 断 指 令

INT	中断
INTO	若溢出就中断
IRET	中断返回

图 4e 程序转移指令

标 志 操 作 指 令

STC	设置进位标志
CLC	清除进位标志
C	进位标志取反
STD	设置方向标志
CLD	清除方向标志
STI	设置中断允许标志
CLI	清除中断允许标志

外 部 同 步 指 令

HLT	暂停直到中断或复位
WAIT	等待 $\overline{\text{BUSY}}$ 失效
ESC	交权给协处理器
LOCK	在下一条指令执行期间封锁总线

空 操 作 指 令

NO	空操作
----	-----

执行环境控制指令	
LMSW	装入机器状态字
SMSW	存放机器状态字

图 4f 处理器控制指令

ENTER	为过程入口格式化堆栈
LEAVE	为过程结束存放堆栈
BOUND	检测在指定范围之外的值

图 4g 高级指令

存储器组成

存储器被组织成为可变长度的段的集合。每个段是可多至64K(2^{16})的线性相邻字节序列。存储器使用具有两个分量的地址(一个指示器)来寻址,它由一个16位的段选择子(selector)和一个16位的偏移量组成。段选择子指示在存储器中所要求的段。偏移量分量指示在该段内所要求的字节地址。

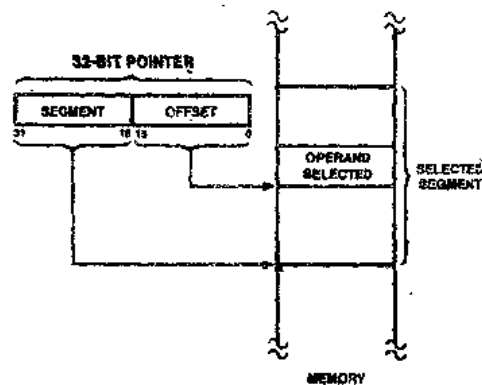


图 5 两个地址分量

所有在存储器中寻址操作数的指令必须指定段和偏移量。为了指令的执行速度和指令编码的兼容,段选择子通常是存放在高速的段寄存器中的。寻址一个存储器操作数,指令只要说明所需的段寄存器和偏移量就可以了。

大多数指令不需要直接指明要使用哪一个段寄存器,根据表 3 中的规则,会自动地选择正确的段寄存器。根据这些规则,程序被写成需要代码和数据,堆栈区域和访问外部数据区域的独立模块。

专用的段超越(override)指令前缀,允许在某些特殊情况下超越某些段寄存器选择规则。对于简单的程序,堆栈,数据和附加段可以是重合的。访问不驻留在 4 个立即可用的段中的操作数时,就必须装入一个 32 位的指示器或一个新的段选择子。

寻址方式

80286 共提供了 8 种用于指令指定操作数的寻址方式。有 2 种方式是提供给那些对寄存器或立即操作数进行操作的指令的,它们是:

表 3 段寄存器选择规则

需要的存储器引用	使用的段寄存器	隐含的段寄存器选择规则
指令	代码段寄存器(CS)	随指令的预取而自动选择
堆栈	堆栈段寄存器(SS)	所有堆栈的推入和弹出。任何把 BP 作为基地址寄存器的存储器引用
局部数据	数据段寄存器(DS)	除了与堆栈或串操作的目标操作数有关的操作之外的所有数据引用
外部(全局)数据	附加段寄存器(ES)	改变数据段和串操作的目标操作数

寄存器操作数方式：操作数被放在一个 8 位或 16 位的通用寄存器中。

立即操作数方式：操作数是包含在指令中。

其它六种方式用来指定操作数在一个存储器段中的地址。一个存储器操作数的地址是由 2 个 16 位分量所组成：段选择子和偏移量。段选择子是由段寄存器提供的，该段寄存器既可由寻址方式隐含地选择，也可由段超越前缀直接选择。偏移量是三个地址分量的任意组合加起来而计算得到的，这三个分量如下：

位移量(包含在指令中的 8 位或 16 位立即值)

基地址(BX 或 BP 基址寄存器的内容)

变址值(SI 或 DI 变址寄存器的内容)

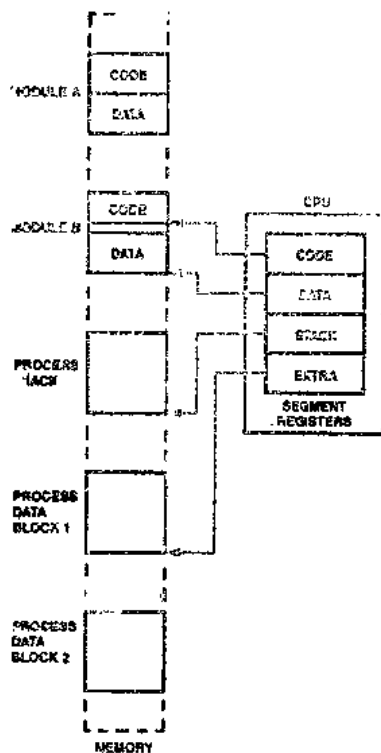


图 6 分段的存储器有利于结构化软件