第二章 80x86计算机组织

1. 80x86计算机组织
   1. 80x86微处理器
   2. 基于微处理器的计算机系统构成
   3. 中央处理机
   4. 存储器
   5. 外部设备
2. 80x86计算机组织
   1. 80x86微处理器

计算机主要由运算器、控制器、存储器和输入输出设备构成。20世纪70年代初期，由于大规模集成电路技术的发展，已经开始把运算器和控制器集成在一个芯片上，构成中央处理机（central processing unit，CPU），80x86就是这样一组微处理器系列。

很多计算机厂商把微处理器芯片作为中央处理机，再配上存储器、输入输出设备和系统软件等构成微计算机系统。如由80386微处理器芯片构成的微机称为386微机由80486微处理器芯片构成的微机称为486微机等。

表2.1 80x86微处理器概况

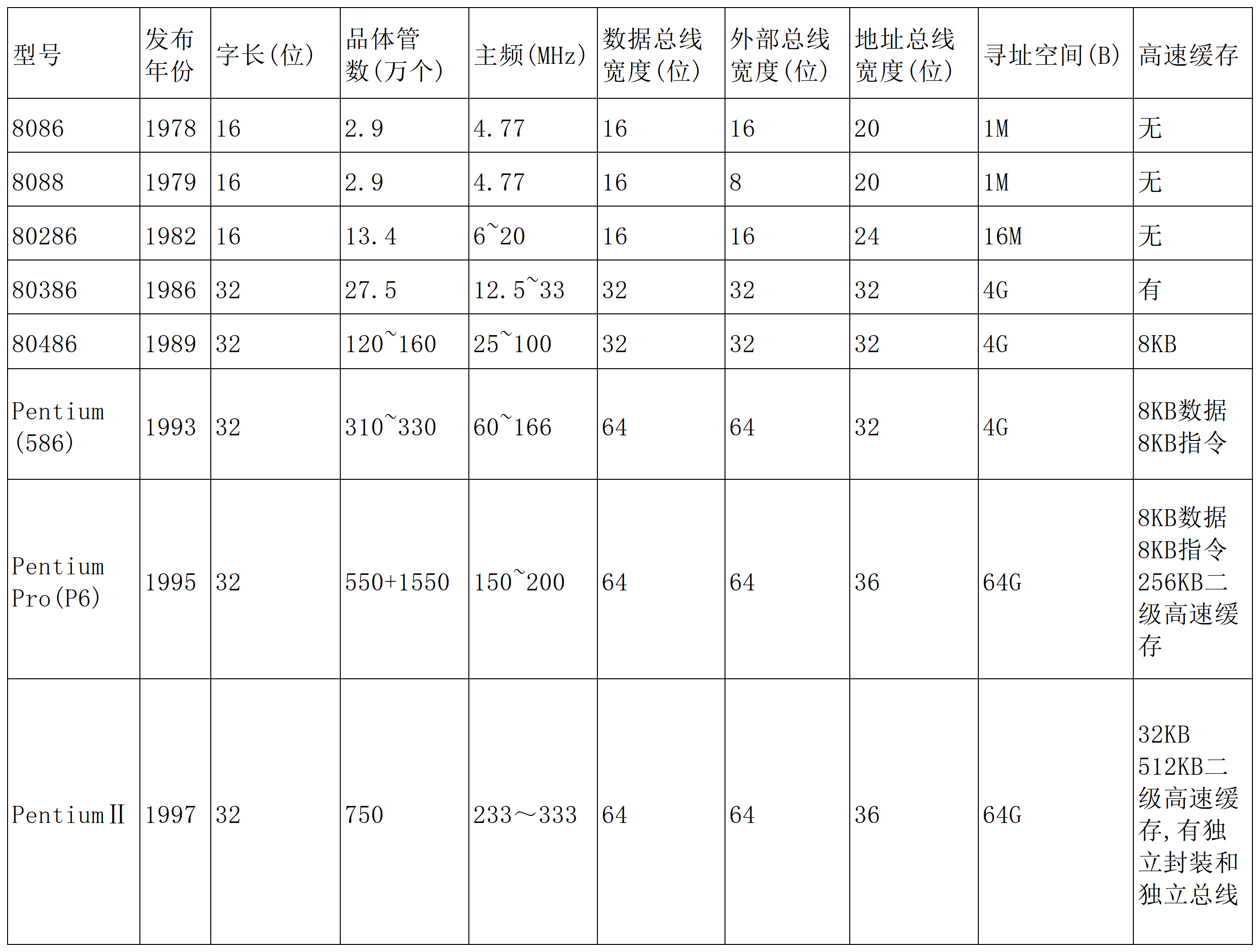


表2.1给出Intel公司生产的80x86微处理器系列的一些主要技术数据。从表2.1中可以看出这一芯片系列的发展概况。

为了能更清楚地说明问题，下面解释一些名词术语。

晶体管数是指芯片中所包含的晶体管数，它说明器件的集成度主频是指芯片所用的主时钟频率，它直接影响计算机的运行速度，由于处理器体系结构的差别，同样的主频可能产生不同的计算速度，但主频仍然是反映计算机速度的一个重要指标数据总线负责计算机中数据在各组成部分之间的传送，数据总线宽度是指在芯片内部数据传送的宽度，外部数据总线宽度则是指芯片内和芯片外交换数据的宽度地址总线宽度是指专用于传送地址的总线宽度，根据这一数值可确定处理机可以访问的存储器的最大范围（寻址空间），如 20位地址总线可访问2°=1048576个存储单元。在计算机中，为方便起见，在讨论存可访问1M个存储单元，24位地址总线可访问16M个存储单元，28=1024M称为1G，32位地址总线可访问4G个存储单元。在计算机里，8个二进制位组成一个字节（byte），储器容量时，以210=1024为基本单位，称为1K，1024K就称为1M。所以20位地址总线一般存储器以字节为存储信息的基本单位，用符号B来表示。这样，上述存储容量又称为1MB、16MB和4GB等。

* 1. 基于微处理器的计算机系统构成

计算机系统包括硬件和软件两部分。硬件包括电路、插件板、机柜等。软件则是为了运行、管理和维护计算机而编制的各种程序的总和。

* + 1. 硬件

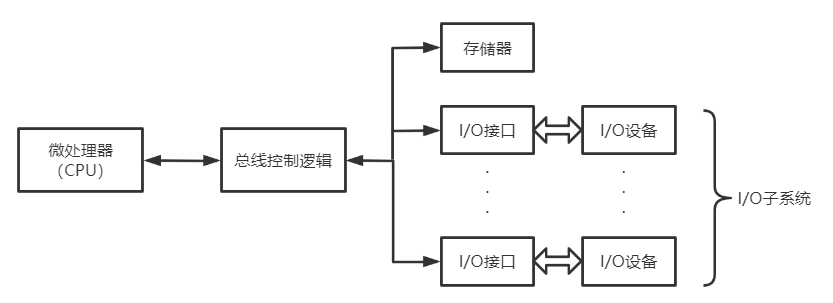


图2.1 计算机结构

典型计算机的结构可用图2.1表示。其中包括中央处理机CPU（Central Processing Unit）、存储器（Memory）和输入输出（I/O）子系统三个主要组成部分，用系统总线把它们连接在一起。

存储器是计算机的记忆部件。们编写的程序（由指令序列组成）就存放在这里，它也可以存放程序中所用的数据、信息及中间结果。

中央处理机包括运算器和控制器两部分。运算器执行所有的算术和逻辑运算指令，控制器则负责全机的控制工作，它负责把指令逐条从存储器中取出，经译码分析后向全机发出取数、执行、存数等控制命令，以保证正确完成程序所要求的功能。

I/O子系统一般包括I/O设备及大容量存储器两类外部设备，I/O设备是指负责与计算机的外部世界通信用的输入、输出设备，如显示终端、键盘输入、打印输出等多种类型的外部设备。大容量存储器则是指可存储大量信息的外存储器如磁盘、磁带、光盘等，机器内部的存储器则称为内存储器，简称内存。由于内存的容量有限，所以计算机用外存储器作为内存的后提设备，它的容量可以比内存大很多，但存取信息的速度要比内存慢得多，所以除必要的系统程序外，一般程序（包括数据）是存放在外存中的。只有当运行时，才把它从外存传送到内存的某个区域，再由中央处理机控制执行。

系统总线把CPU、存储器和I/O设备连接起来，用来传递各部分之间的信息，系统总线包括数据线，地址线和控制线三组.数据线传送信息，地址线指出信息的来源和且的地，控制线则规定总线的动作如方向等。系统总线的工作由总线控制逻辑负责指挥。

* + 1. 软件

计算机软件是计算机系统的重要组成部分，它可以分成系统软件和用户软件两大类。系统软件是由计算机的生产厂家提供给用户的一组程序，这些程序是用户使用机器时为产生、准备和执行用户程序所必需的。用户软件则是用户自行编制的各种程序。在这里我们将简要介绍系统软件的组成。

系统软件的核心称为操作系统（Operating System）。操作系统是系统程序的集合，它的主要作用是对系统的硬、软件资源进行合理的管理，为用户创造方便、有效和可靠的计算机工作环境。

操作系统的主要部分是常驻监督程序（monitor），只要一开机它就存在于内存中，它可以从用户接收命令，并使操作系统执行相应的动作。

1/o驱动程序（I/O driver）用来对1/O设备进行控制或管理。当系统程序或用户程序需要使用1/0设备时，就调用I/O驱动程序来对设备发出命令，完成CPU和I/O设备之间的信息传送。

文件管理系统（File management）用来处理存储在外存储器中的大量信息，它可以和外存储器的设备驱动程序相连接，对存储在其中的信息以文件（File）形式进行存取、复制及其他管理操作。

文本编辑程序（Text editor）用来建立、输入或修改文本，并使它存入存储器或大容量存储器中。文本是指由字母、数字、符号等组成的信息，它可以是一个用汇编语言或高级语言编写的程序，也可以是一组数据或一份报告。

翻译程序（Translator）：我们已经知道计算机是通过逐条地执行组成程序的指令来完成人们所给予的任务的，所以指令就是计算机能识别并能直接加以执行的语句，它当然是由二进制代码组成的，这种语言称为机器语言，显然它对于人们是很不方便的。既然计算机能识别的唯一语言是机器语言，而这种语言使编写程序很不方便，所以在计算机语言的发展过程中就出现了汇编语言和高级语言。汇编语言是一种符号语言，它几乎和机器语言一一对应，但在书写时却使用由字符串组成的助记符。例如，加法在汇编语言中是用助记符ADD表示的，而机器语言则用6位二进制代码来表示。显然，相对于机器语言来说，汇编语言是易于为人们所理解的，但计算机却不能直接识别汇编语言。汇编程序就是用来把由用户编制的汇编语言程序翻译成机器语言程序的一种系统程序。

高级语言脱离开机器指令用人们更加易于理解的方式来编写程序，当然它们也要翻译成机器语言才能在机器上执行。高级语言的翻译程序有两种方式：一种是先把高级语言程序翻译成机器语言（或先翻译成汇编语言，然后再由汇编程序再次翻译成机器语言）程序，然后再在机器上执行，这种翻译程序称为缠译程序（Compiler），多数高级语言如PASCAL，FORTRAN等都采用这种方式。另一种是直接把高级语言程序在机器上运行，一边解释一边执行，这种翻译程序称为解释程序（Interpreter），如BASIC就经常采用这种方式。

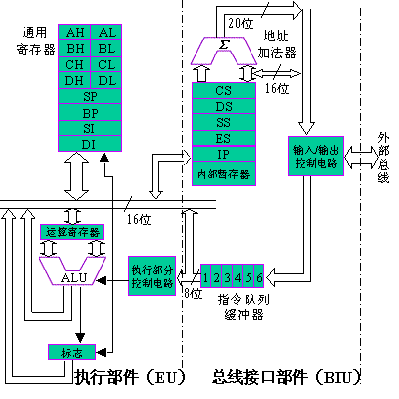
系统程序中的翻译程序包括汇编程序，BASIC解释程序及各种高级语言的编译程序. 连接程序（Linker）用采把要执行的程序与库文件或其他已经翻译的子程序（能完成一种独立功能的程序模块）连接在一起，形成机器能执行的程序。

装入程序（Loader）用来把程序从外存储器传送到内存储器，以便机器执行。例如，计算机开机后就需要立即启动装入程序把常驻监督程序装入存储器，使机器运转起来。又如，用户程序经翻译和连接后，由连接程序直接调用装入程序，把可执行的用户程序装入内存以便执行。

调试程序（Debug）是系统提供给用户的能监督和控制用户程序的一种工具，它可以装入、修改、显示或逐条执行一个程序。在IBM PC机上，简单的汇编语言程序可以通过DEBUG来建立、修改和执行。

系统程序库（System Library）和用户程序库（User Library）；各种标准程序、子程序及一些文件的集合称为程序库，它可以被系统程序或用户程序调用。操作系统还允许用户建立程序距。以提高不同类型用户的工作效率。

* 1. 中央处理机
     1. 中央处理机CPU的组成



CPU的任务是执行存放在存储器里的指令序列。为此，除要完成算术逻辑操作外，还需要担负CPU和存储器以及I/O之间的数据传送任务。早期的CPU芯片由运算器和控制器两部分组成。从80386开始，为使存储器速度能更好地与运算器的速度相匹比，已在芯片中引入高速缓冲存储器。其后生产的芯片随着半导体器件集成度的提高，片内高速缓冲存储器的容量也逐步扩大，但这部分器件就其功能而言还是属于存储器的，本节要说明的是CPU芯片中除高速缓冲存储器之外的组成，它们主要由以下三部分组成：

（1）算术逻辑部件ALU（Arith-metic Logic Unit）用来进行算术和逻辑运算。

（2）控制逻辑负责对全机的控制工作，包括从存储器取出指令，对指令进行译码分析，从存储器取得操作数，发出执行指令的所有命令，

把结果存入存储器，以及对总线及I/O传送的控制等。

（3）工作寄存器在计算机中起着重要的作用，每一个寄存器相当于运算器中的一个存储单元，但它的存取速度比存储器要快得多。

* + 1. 80x86寄存器组**（PPT介绍各类寄存器）**

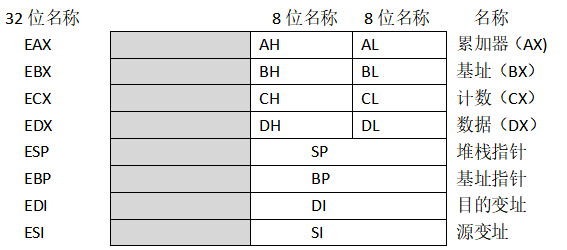
寄存器可以分为程序可见的寄存器和程序不可见的寄存器两大类。所谓程序可见的

寄存器是指在汇编语言程序设计中用到的寄存器，它们可以由指令来指定。而程序不可

见的寄存器则是指一般应用程序设计中不用而由系统所用的寄存器。本节将介绍80x86 中程序可见的那部分寄存器，而程序不可见的寄存器就不加以说明了。

程序可见寄存器可以分为通用寄存器、专用寄存器和段寄存器3类。图2.3表示了80x86的程序可见寄存器组。下面分别加以说明。

**1.通用寄存器**



表： 80x86通用寄存器

图2.3中除阴影区以外的寄存器是8086/8088和80286所具有的寄存器，它们都是16位寄存器。其中AX、BX、CX、DX可称为数据寄存器，用来暂时存放计算过程中所用到的操作数、结果或其他信息。它们都可以以字（16 位）的形式访问，或者也可以以字节（8位）的形式访问。例如，对AX可以分别访问高位字节AH或低位字节AL。这4个寄存器都是通用寄存器，但它们又可以用于各自的专用目的。

AX（accumulator）作为累加器用，所以它是算术运算的主要寄存器。在乘、除等指令中指定用来存放操作数。另外，所有的I/O指令都使用这一寄存器与外部设备传送信息。

BX（base）可以作为通用寄存器使用。此外在计算存储器地址时，它经常用作基址寄存器。

CX（count）可以作为通用寄存器使用。此外常用来保存计数值，如在移位指令、循环（loop）和串处理指令中用作隐含的计数器。

DX（data）可以作为通用寄存器使用。一般在作双字长运算时把DX和AX组合在一起存放一个双字长数，DX用来存放高位字。此外，对某些I/O操作，DX可用来存放I/O 的端口地址。

SP、BP、SI、DI四个16位寄存器可以像数据寄存器一样在运算过程中存放操作数，但它们只能以字（16位）为单位使用。此外，它们更经常的用途是在存储器寻址时，提供偏移地址。因此，它们可称为指针或变址寄存器。在这些寄存器中，SP（stack pointer）称为堆栈指针寄存器，BP（base pointer）称为基址指针寄存器，它可以与堆栈段寄存器 SS 联用来确定堆栈段中的某一存储单元的地址。SP用来指示段顶的偏移地址，BP可作为堆栈区中的一个基地址以便访问堆栈中的信息。SI（source index）源变址寄存器和DI （destination index）目的变址寄存器一般与数据段寄存器 DS联用，用来确定数据段中某一存储单元的地址。这两个变址寄存器有自动增量和自动减量的功能，所以用于变址是很方便的。在串处理指令中，SI和DI作为隐含的源变址和目的变址寄存器，此时SI和DS联用，DI和附加段寄存器 ES联用，分别达到在数据段和附加段中寻址的目的。有关段地址和偏移地址以及堆栈段的概念，将在2.4.2节作详细说明。

对于80386及其后继机型则是图2.3中所示的完整的寄存器，它们是82位的通用寄存器，包括EAX，EBX，ECX，EDX，ESP，EBP，EDI和ESI。在这些机型中，它们可以用来据，用AX保存16位数据，用AH或AL保存8位数据。在计算机中，8位二进制数可组成一个字节，8086/8088和80286的字长为16位，因此把2个字节组成的16位数称为字。这样，80386及其后继的32位机就把32位数据称为双字，64位数据称为4字。上述8个通用寄存器可以以双字的形式或对其低16位以字的形式被访问，其中EAX，EBX，ECX和EDX的低16位还可以以字节的形式被访问。当这些寄存器以字或字节形式被访问时，不被访问的其他部分不受影响，如访问AX时，EAX的高16位不受影响。

此外，这8个通用寄存器还可用于其他目的。在8086/8088和80286中只有4个指针和变址寄存器以及BX寄存器可以存放偏移地址，用于存储器寻址。在80386及其后继机型中，所有32位通用寄存器既可以存放数据，也可以存放地址。也就是说，这些寄存器都可以用于存储器寻址。在这8个通用寄存器中，每个寄存器的专用特性与8086/8088和80286的AX，BX，CX，DX，SP，BP，DI，SI是一一对应的。如EAX专用于乘、除法和I/O指令，ECX的计数特性，EDI和ESI作为串处理指令专用的地址寄存器等。

**2.专用寄存器**

8086/8088和80286的专用寄存器包括IP、SP和FLAGS3个16位寄存器。

IP（instruction pointer）为指令指针寄存器，它用来存放代码段中的偏移地址。在程序运行的过程中，它始终指向下一条指令的首地址，它与段寄存器CS联用确定下一条指令的物理地址。当这一地址送到存储器后，控制器可以取得下一条要执行的指令，而控制器一旦取得这条指令就马上修改IP的内容，使它指向下一条指令的首地址。可见，计算机就是用IP寄存器来控制指令序列的执行流程的，因此IP寄存器是计算机中很重要的一个控制寄存器。

SP为堆栈指针寄存器，它与堆栈段寄存器联用来确定堆栈段中栈顶的地址，也就是说SP用来存放栈顶的偏移地址。

FLAGS为标志寄存器，又称程序状态寄存器（program status word，PSW）。这是一个存放条件码标志、控制标志和系统标志的寄存器。

80386及其后继机型也有三个32位专用寄存器，它们是EIP、ESP和EFLAGS。它们的作用和相应的16位寄存器相同。

表：8086/8088/80286标志寄存器

Z~1B(W_BO@EZ]HB4D1OA@NJ

下面介绍标志寄存器。上表说明了80x86中标志寄存器的内容，图中未标明的暂不用。

（1）条件码标志用来记录程序中运行结果的状态信息，它们是根据有关指令的运行结果由CPU自动设置的。由于这些状态信息往往作为后续条件转移指令的转移控制条件，所以称为条件码。它包括以下6位

溢出标志（overflow flag，OF）在运算过程中，如操作数超出了机器能表示的范围称为溢出。此时OF位置1，否则置0。

号标志（sign flag，SE），记录运算结果的符号，结果为负时置1，否则置Q。

零标志（zero flag，ZF），运算结果为0时ZF位置1，否则置0。

进位标志（carry flag，CE），记录运算时从最高有效位产生的进位值。例如，执行加法指令时，最高有效位有进位时置1，否则置0。

辅助进位标志（auxiliary carry flag，AF），记录运算时第3位（半个字节）产生的进位值。例如，执行加法指令第3位有进位时置1，否则置0。

奇偶标志（parity flag，PF），用来为机器中传送信息时可能产生的代码出错情况提供检验条件。当结果操作数中1的个数为偶数时置1，否则置0。

（2）控制标志位为方向标志（direction flag，DF），在串处理指令中控制处理信息的方向用。当DF位为1时，每次操作后使变址寄存器SI和DI减小，这样就使串处理从高地址向低地址方向处理。当DF位为0时，则使SI和DI增大，使串处理从低地址向高地址方向处理。

（3）系统标志位可以用于I/O、可屏蔽中断、程序调试、任务切换和系统工作方式等的控制。一般应用程序不必关心或修改这些位的状态，只有系统程序员或需要编制低层I/O设备控制等程序时才需要访问其中的有关位。下面只简单介绍某些位的情况，其他各位由于在本书中不涉及，故不再加以说明。

陷阱标志（trap flag，TF），用于调试时的单步方式操作。当TF位为1时，每条指令执行完后产生陷阱，由系统控制计算机当TF位为0时，CPU正常工作，不产生陷阱。

中断标志（interrupt flag，IF），当IF 位为1时，允许CPU响应可屏蔽中断请求，否则关闭中断。

IIO特权级（I/O privilege level，IOPL），在保护模式下，用于控制对I/O地址空间的访问。

以上就是EFLAGS中主要标志位的含义。机器提供了设置某些状态信息的指令。必要时，程序员可使用这些指令来建立状态信息。

在调试程序DEBUG中提供了测试标志位的手段，它用符号表示某些标志位的值。下表说明这些标志位的符号表示。

表： Debug程序标志位符号表示



3. 段寄存器

段寄存器也是一种专用寄存器，它们专用于存储器寻址，用来直接或间接地存放段地址。段寄存器的长度为16位，在80286以前的处理器中，只有代码段（code segment，CS）、数据段（data segment，DS）、堆栈段（stack segment，SS）和附加段（extra segment，ES）4个寄存器。从80386起，增加了FS和GS两个段寄存器，它们也属于附加的数据段。

* 1. 存储器
     1. 存储单元的地址和内容**（PPT讲解存储器，视频，动画演示存储单元存储的过程）**

计算机存储信息的基本单位是一个二进制位，一位可存储一个二进制数：0或1，每8位组成一个字节。基于80x86CPU的IBM PC机以字节为单位存储信息，即每个存储单元存储一个字节的数据。每个存储单元被分配一个唯一的地址。物理地址从0开始，顺序编号。

若CPU地址总线宽度为N，则可访问的字节单元地址范围为 0~2N-1。8086/8088的地址总线宽度为20位，可访问的地址空间为0~FFFFFH（1MB）。80286的地址总线宽度为24位，386~Pentium地址总线宽度为32位。



一个存储器有128个存储单元，它可以存储128个Byte。微机存储器的容量是以字节为最小单位来计算的。对于拥有128个存储单元的存储器，我们可以说，它的容量是128个字节。

数据类型及存储顺序：

·位（比特、bit）

·字节：8位，可存储于任何单元。

·字：16位，2字节，低位字节在低地址单元，高位字节在高地址，起始地址最好是偶地址。

·双字：32位，低位字存入低地址，高位字存入高地址，起始地址最好是4的倍数。

·4字：64位，低位双字存入低地址，高位双字存入高地址，起始地址最好是8的倍数。

对于大容量的存储器一般还用以下单位来计量容量（以下用B来代表Byte）：

1KB = 1024B = 210B （KiloByte、千字节）

1MB = 1024KB = 220B （MegaBate、兆字节）

1GB = 1024MB = 230B （GigaByte、千兆字节）

1TB = 1024GB = 240B （TeraByte、兆兆字节）

1个二进制位：bit （位、比特）

8个二进制位：Byte （字节）

1Byte = 8bit

2个字节： Word （字）

1Word = 2Byte = 16bit

磁盘的容量单位同内存的一样，实际上以上单位是微机中常用的计量单位。

* + 1. 实模式存储器寻址**（PPT，视频讲解实模式存储器寻址的过程）**

1.存储地址的分段

段Segment是按照程序的逻辑结构划分成的多个相对独立的部分。程序段可以是主程序、子程序、数据块、数组、表格等。

实地址模式，CPU可访问的最大存储容量为1MB， 需20位地址信号。1MB的地址空间被划分为大小不等的段，每段最大长度为64KB，段基址为16的整数倍。

逻辑地址的格式：段基址:段内偏移量

物理地址 ＝ 段基址 × 16 + 偏移量 （即段基址左移4位，与段内偏移求和）

2.段寄存器

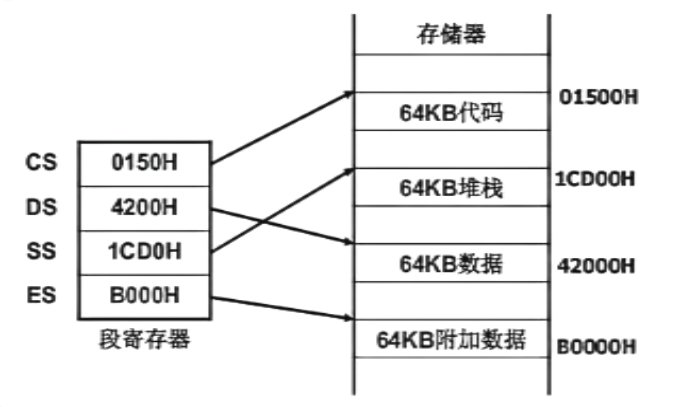
有四个专门存放段基址的寄存器，称位段寄存器。它们是CS代码段、DS数据段、ES附加数据段、SS堆栈段寄存器。

偏移量由BX、BP、IP、SP、SI、DI或根据寻址方式计算出的有效地址EA（Effective Address）提供。

段分配实例之一：

• 各段不连续

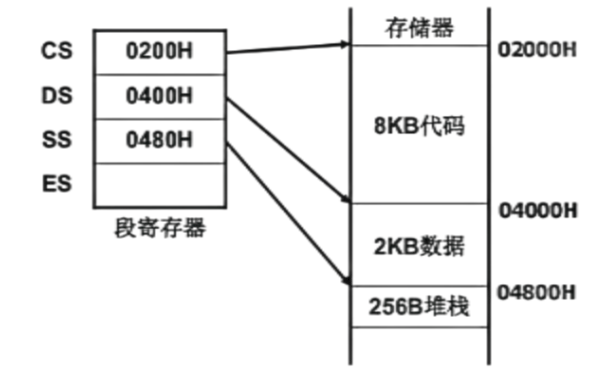
• 各段段基址不相同



段分配实例之二：（400H=1K，2000H=8K）

• 各段连续

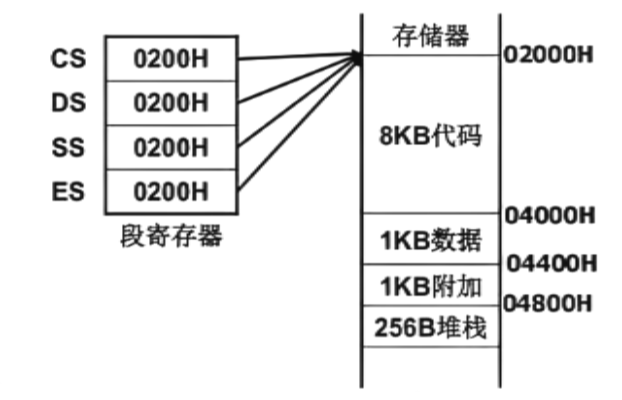
• 各段段基址不同



段分配实例之三

• 各段连续

• 各段段基址相同



* + 1. 保护模式存储器寻址**（PPT，视频讲解保护模式存储器寻址的过程）**

80x86的地址线条数及最大寻址空间

·8086/8088：20根，1MB空间

·286：24根，16MB空间

·386及后续CPU：32根，4GB

保护模式：

·解决了实模式只能寻址1MB的问题。

·段页式存储管理，每段最大4GB，每段划分为等长的页（4KB）。

·支持多任务处理。

·保护模式下的三种地址：虚拟地址（逻辑地址）、线性地址、物理地址。

1.逻辑地址

·实模式逻辑地址：

地址形式：段基址:偏移量

段基址：存放于段寄存器中。

·保护模式逻辑地址：

地址形式：选择器（选择子）:偏移量

选择子：存放于段寄存器中，16位长，不能直接表示段基地址，而是由操作系统根据复杂算法获得32位段基地址。

偏移量：32位长，即保护模式每段最长可达4GB。

2.段描述符

段描述符用来描述段的大小、位置、控制与状态信息。它由基地址、界限、访问权限、附加字段组成。基地址用来指定段的起始地址，界限用来指定段的长度。

保护模式中，每个段都有一个段描述符。

逻辑地址→线性地址：

① 根据段选择子找到段描述符；

② 从段描述符中提取段基地址与长度；

③ 段基址与偏移量求和，得线性地址。

* 1. 外部设备

计算机运行时的程序和数据都要通过输入设备送入机器，程序运行的结果要通过输出设备送给用户，所以输入、输出设备是计算机必不可少的组成部分。大容量的外存储器（如磁盘）能存储大量信息，也是现代计算机不可缺少的一部分，对于外部设备的管理是汇编语言的重要使用场合之一。

外部设备与主机（CPU和存储器）的通信是通过外设接口进行的，每个接口包括一组寄存器。一般说来，这些寄存器有三种不同的用途∶

1.数据寄存器∶用来存放要在外设和主机间传送的数据，这种寄存器实际上起缓冲器的作用。

2.状态寄存器∶用来保存外部设备或接口的状态信息，以便CPU在必要时测试外设状态，了解外设的工作情况。例如，每个设备都有忙闲位用来标志设备当前是否正在工作，是否有空接受CPU给予的新任务等。

3.命令寄存器∶CPU给外设或接口的控制命令通过此寄存器送给外部设备。例如，CPU要启动磁盘工作，必须发出启动命令等。

各种外部设备都有以上三种类型的寄存器，只是每个接口所配备的寄存器数量是根据设备的需要确定的。例如，工作方式较简单、速度又慢的键盘只有一个8位的数据寄存器，并把状态和命令寄存器合二为一个控制寄存器。又如工作速度快、工作方式又比较复杂的磁盘则需要多个数据、状态和命令寄存器。

为使主机访问外设方便起见，外设中的每个寄存器给予一个端口（port）地址（又称端口号），这样就组成了一个独立于内存储器的I/O地址空间。80x86机的I/O地址空间可达64KB，所以端口地址的范围是0000~FFFFH，用16位二进制代码来表示。端口可以是8位或16位的，386及其后继机型还可以有32位端口，但整个I/O空间不允许超过64KB。

主机与外设交换信息是通过输入、输出指令来完成的，在第三章我们还要专门说明。主机与外设的信息传送方式可有直接、查询、中断、成组传送等，将在第八章说明。实际上，对外设的管理及信息传送是汇编语言最经常使用的场合，也是最复杂的一部分程序。

为了便于用户使用外设，80x86提供了两种类型的例行程序供用户调用。一种是基本输入输出系统（basic input/output system，BIOS），另一种是磁盘操作系统（disk operating system，DOS）功能调用。它们都是系统编制的子程序，通过中断方式转入所需要的子程序去执行，执行完后返回原来的程序继续执行。这些例行程序有的完成一次简单的外设信息传递，如从健盘输入一个字符，或送一个字符至显示器等；也有的完成相当复杂的一次外设操作，如从磁盘读写一个文件等。总之，操作系统把一些复杂的外设操作编成例行程序，使用户用简单的中断指令（INT）就可以进入这些例行程序，完成所需要的外设操作，所以用户应尽量利用这些系统所提供的工具来编写自己的程序。

BIOS和DOS功能调用虽然都是系统提供的例行程序，但是它们之间又有差别。B10S存放在机器的只读存储器ROM中，所以可以把它看成是机器硬件的一个组成部分，它的层次比DOS更低，更接近硬件，因此它的语句要完成每一个对设备的直接命令，或信息传送。DOS功能调用是操作系统DOS的一个组成部分，它在开机时由磁盘装入存储器，在它的例行程序中可以一次或多次调用BIOS以完成比BIOS更高级的功能。用户需要使用外设时，应尽可能使用层次较高的DOS功能调用，但有时它不能满足你的要求，此时，就需要直接调用BIOS，如果BIOS 还不能解决你的问题，那么就只好自己编制中断处理程序了。

无实验

习题：

1 在实模式操作下，段寄存器的用途是什么？

段寄存器是用来定义数据段得起始地址的。在实模式下，它定义一个最大64KB存储段的起点。

2 实模式存储器地址允许访问低于哪个地址的存储区？

在实模式下，存储器寻址允许访问低于FFFF:FFFF地址的存储区。

3 堆栈存储器由 堆栈 段加 堆栈指针 偏移的组合来寻址。

4 80286工作在实模式下，给出下列寄存器组合所寻址的存储单元地址。

（a）DS=1000H 和 DI=2000H： 12000H

（b）DS=2000H 和 SI=1002H： 21002H

（c）SS=2300H 和 BP=3200H： 26200H

（d）DS=A000H 和 BX=1000H： A1000H

（e）SS=2900H 和 SP=3A00H： 2CA00H

5 中央处理器CPU的组成：

1.算术逻辑部件ALU :用来进行算术、逻辑运算。

2.控制器CU：负责对机器的控制工作，如，从存储器发出指令，对指令进行译码分析;从存储器取得操作数，发出执行指令的所有命令，把结果存入存储器以及对总线及IO的传送控制等。

3.寄存器，每一个寄存器相当于运算器中的一个存储单元，但它的存取速度比存储器要快得多它用来存取计算过程中所需要的或所得到的各种信息，包括操作数地址，操作数及运算结果等。

6 通用寄存器共有哪八个？它们的作用分别都是什么？

通用寄存器分为4个通用数据寄存器:AX、BX、CX、DX和4个地址指针寄存器:SP、BP、SI、Dl。

4个通用数据寄存器：

AX：累加器，主要用在算术运算中，在乘、除等指令中指定用来存放操作数，所有的I/O指令都使用它与外设传送信息。

BX：在计算存储器地址时，常用作基址寄存器。

CX：常用来保存计数值，如移位指令、循环指令和串处理指令中用作隐含的计数器。

DX：一般在作双字长运算时把DX和AX组合在一起存放一个双字长数，DX存放高位字，AX存放低位字（DX：AX）。对某些I/O操作，DX可存放I/O的端口地址。

4个地址指针寄存器:

SP：堆栈指针寄存器

BP：基址指针寄存器

SI：源变址寄存器

DI：目的变址寄存器

7 两种寄存器的区别是什么？

数据寄存器：一般用于存放数据，包括AX、BX、CX、DX，可以分别访问其低端字节AL、BL、CL、DL和高端字节AH、BH、CH、DH，从386起扩充为32位，分别是EAX、EBX、ECX、EDX

指针和变址寄存器：一般用来存放地址的偏移量，包括指针寄存器SP、BP和变址寄存器SI、DI，从386起扩充为32位，分别是ESP、EBP、ESI、EDI。

8 地址加法器合成物理地址的方法：

物理地址 = 段地址×16 + 偏移地址

9 外设接口寄存器：

外设接口是主机（CPU、内存）与外设之间通信的接口。

外设接口包括一组寄存器，用户编程时，只需与这些接口寄存器进行通信。

数据寄存器：存放主机与外设间传送的数据

状态寄存器：存放外设或接口的状态。

命令（控制）寄存器：存放主机给外设或接口的命令。