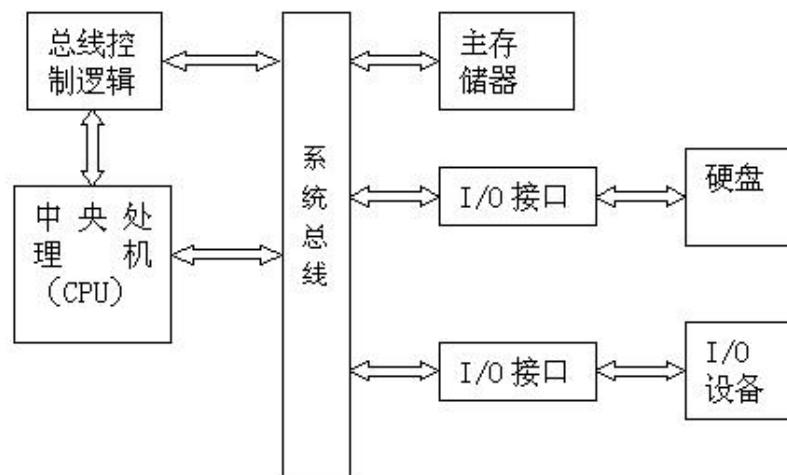


第2章 计算机基本原理

- 2.1 计算机系统组成
- 2.2 存储器
- 2.3 中央处理器（CPU）中的寄存器
- 2.4 外部设备和接口
- 2.5 32位80x86 CPU的工作模式

2.1 计算机系统组成

- 计算机的基本工作原理是存储程序和程序控制。
- 冯诺依曼原理的计算机结构



2.2 存储器

- 2.2.1 16位结构的CPU
- 2.2.2 存储器
- 2.2.3 存储器分段
- 2.2.4 逻辑地址
- 2.2.5 CPU对内存的读写操作

2.2.1 16位结构的CPU

- 8086是16位结构的CPU。16位结构的CPU具有以下几方面的结构特征：
 - (1) 数据总线为16位；
 - (2) 运算器一次最多可以处理16位的数据；
 - (3) 寄存器的最大宽度为16位；
 - (4) 寄存器和运算器之间的通路为16位。

2.2.1 16位结构的CPU

1、总线接口单元 BIU (Bus Interface Unit)

功能：负责与 存储器或者I/O 端口传送数据， 完成内外信息交换。

- ① 形成物理地址
- ② 取指令送到指令队列
- ③ 传送数据

2、执行单元EU (Execution Unit)

功能：负责指令执行。

- ① 指令译码
- ② 执行指令
- ③ 向BIU传送偏移地址信息
- ④ 管理通用寄存器和标志寄存器

2.2.1 16位结构的CPU

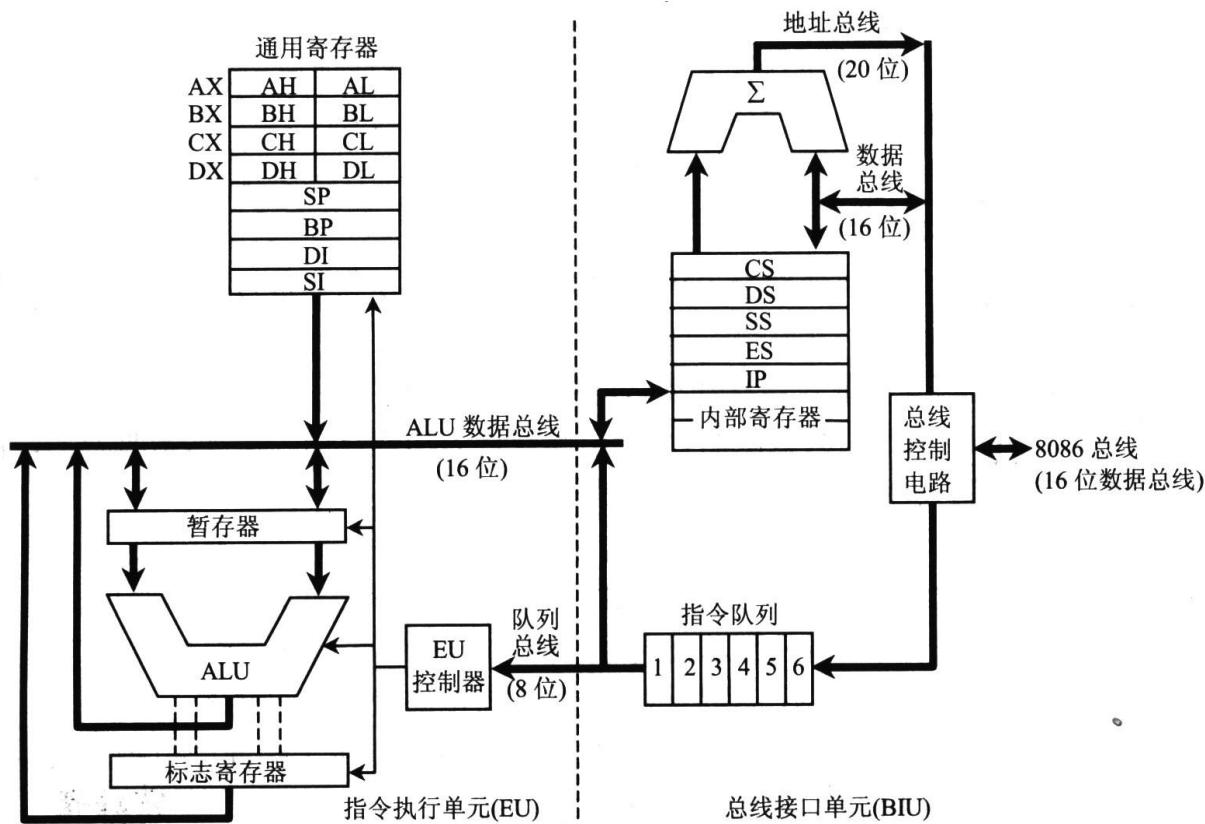


图 2-1 8086 CPU 内部结构框图

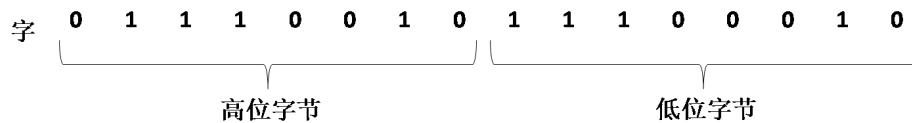
2.2.2 存储器

- 1. 基本存储单元
- 计算机存储信息的最小单位是一个二进制位 (bit)
- 8位二进制位组成一个字节(Byte)
- 2个字节 (16位) 组成一个字(Word)
- 2个字 (32位) 称为双字。
- 80x86微机的内存储器以字节为基本存储单位 , 或叫基本存储单元

2.2.2 存储器

● 2. 内存中字的存储

字与字节的对应关系



从31200H单元开始存放的字数据为A28FH，从31202H单元开始存放的字数据为1234H，分别记为：

(31200H) 字 = A28FH

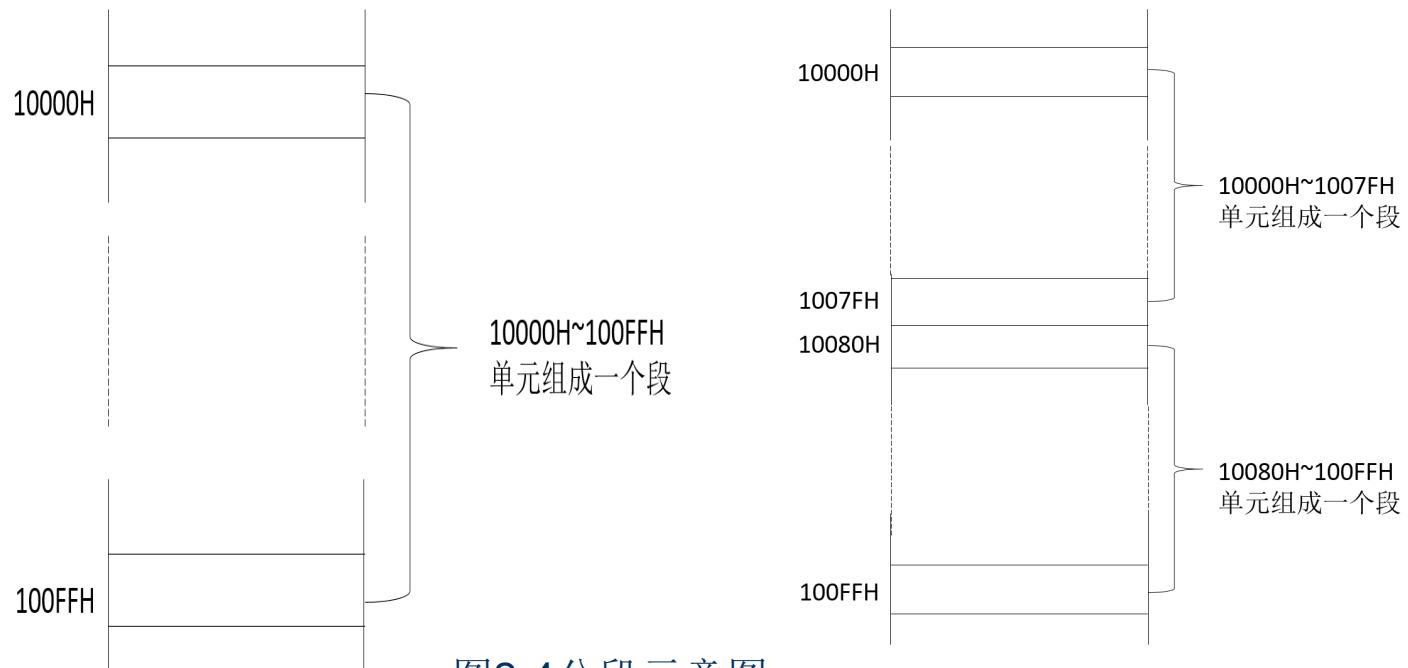
(31202H) 字 = 1234H

| 物理地址 | 存储单元 |
|--------|-------|
| 30000H | B8 |
| 30001H | 23 |
| 30002H | 72 |
| 30003H | E2 |
| | |
| 31200H | 8F |
| 31201H | A2 |
| 31202H | 34 |
| 31203H | 12 |
| | |
| 3FFFFH | C9 |

2.2.3 存储器分段

1. 分段的概念

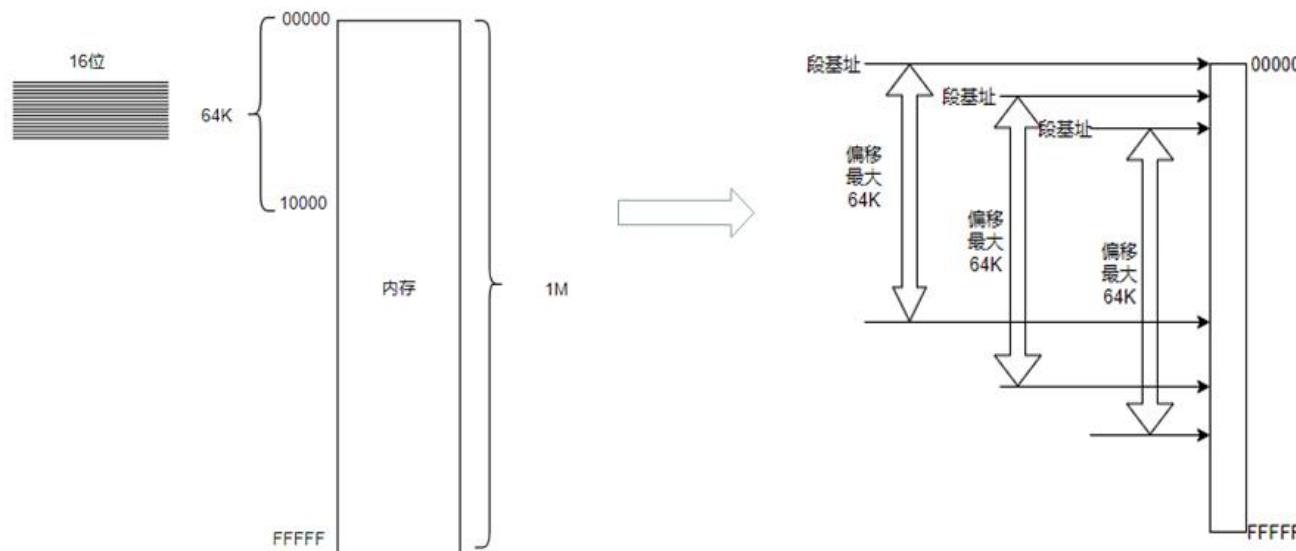
内存并没有分段，分段只是**CPU**管理内存的方式。



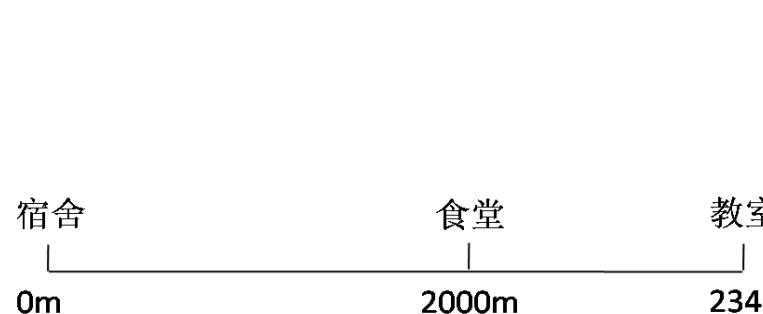
2.2.3 存储器分段

1. 分段的概念

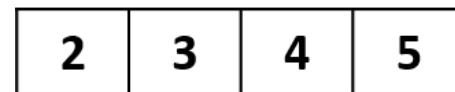
将内存划分成多段，通过段基址+段偏移的方式访问



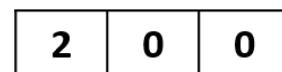
2.2.3 存储器分段



如何描述从宿舍到教室的距离？



可以写下4位数据的纸条



可以写下3位数据的纸条

2.2.3 存储器分段

1. 分段的概念

内存地址 = 段基址 * 10H + 段偏移, 因为低四位被舍弃了, 变成了段偏移 所以需要 *10H

段基址: 073F
段偏移: 0100



内存地址: $73F \times 10 + 100 = 74F0$

段基址: 063F
段偏移: 1200



内存地址:
 $63F \times 10 + 1200 = 75F0$

段基址: 043F
段偏移: 0120

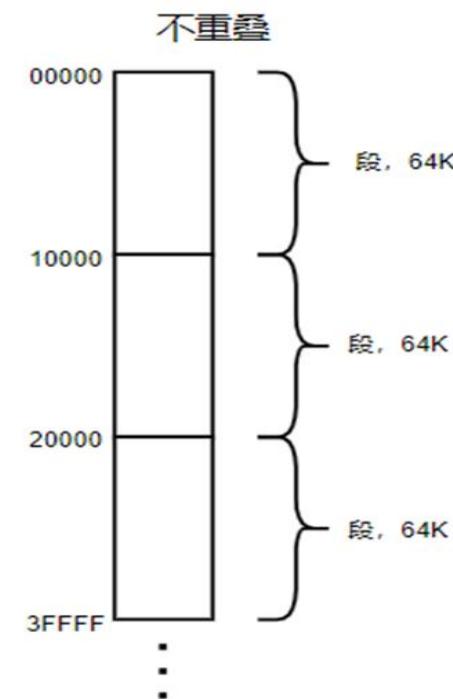
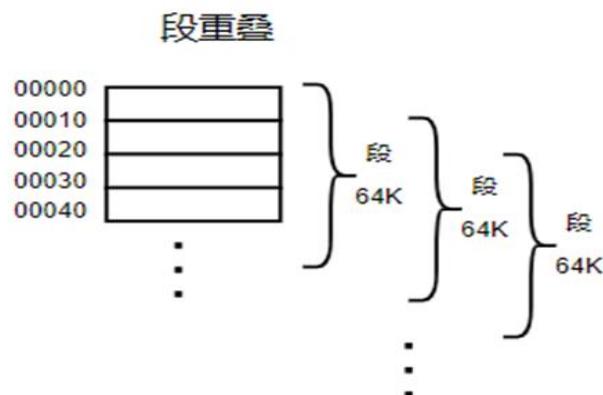


内存地址:
 $43F \times 10 + 120 = 4510$

2.2.3 存储器分段

1. 分段的概念

分段 - 内存分布



2.2.3 存储器分段

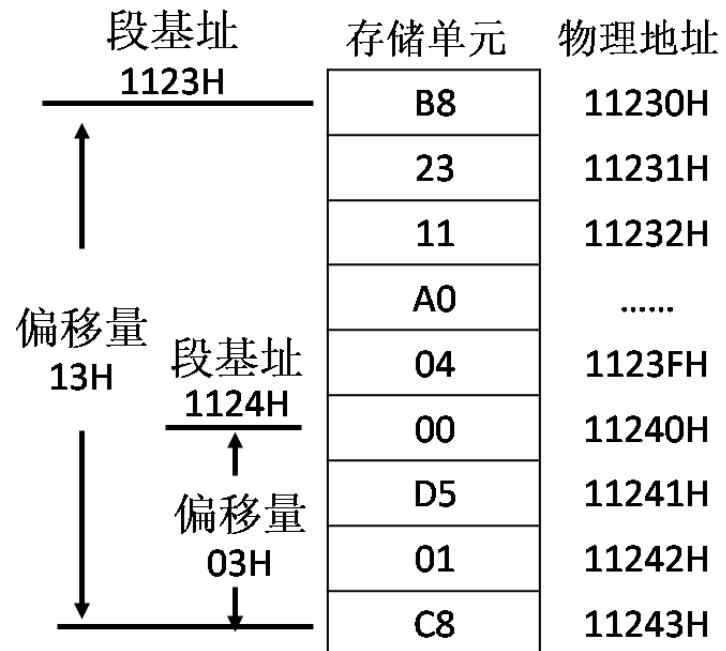
2.段的类型

- 代码段—用于存放指令，代码段段基址存放在段寄存器CS
- 数据段—用于存放数据，数据段段基址段地址存放在段寄存器DS
- 附加段—用于辅助存放数据，附加段段基址存放在段寄存器ES
- 堆栈段—是重要的数据结构，可用来保存数据、地址和系统参数，堆栈段段基址存放在段寄存器SS

2.2.4 逻辑地址

逻辑地址是用户编程时使用的地址，分为段地址和偏移地址两部分。

- 段地址：偏移地址



PA与LA的对应关系

2.2.4 逻辑地址

例题2-1 段基址为1896H，偏移地址为1655H。
其物理地址为多少？

$$18960H + 1655H = 19FB5H$$

习题

给定段地址为0001H，仅通过变化偏移地址寻址
，CPU的寻址范围从 () 到 ()

要求：填空1和填空2需要用完整的五位十六进制来表示，末尾还要加上十六进制数的标志H.

习题

给定段地址为0001H，仅通过变化偏移地址寻址，CPU的寻址范围从 (00010H) 到 (1000FH)

要求：填空1和填空2需要用完整的五位十六进制来表示，末尾还要加上十六进制数的标志H.

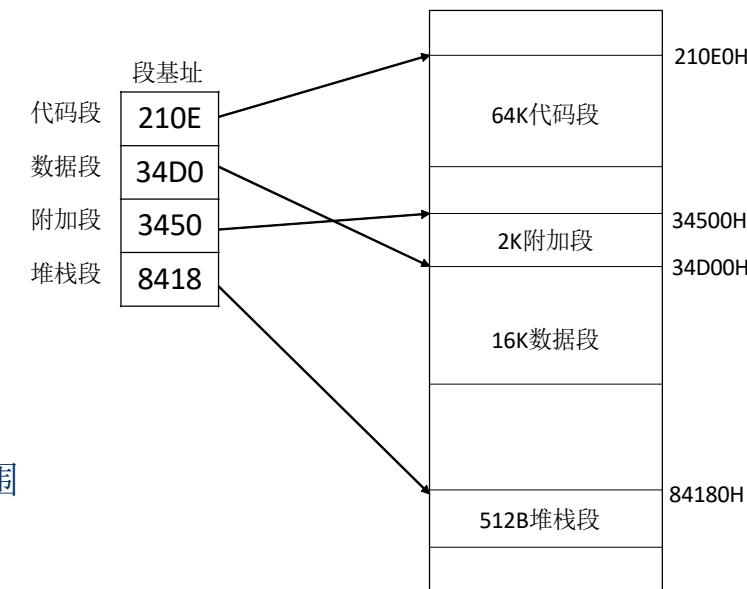
2.2.4 逻辑地址

- 例题2-2 段基址与内存分段情况如图2-9所示，观察各个段的大小与分布，判断其地址范围标出每个段首地址和末地址。

从图中看出如下内容。

- 代码段有64KB，它的地址范围在210E0H~310DFH，已经达到段的最大范围。
- 附加段只有2KB，地址范围在34500H~34CFFH之间。
- 数据段为16KB，其地址范围为34D00H~38CFFH。可知数据段紧接着附加段的最后单元存放，而不必在附加段的64KB最大区域之外设置其他段。此方式也称为段重叠，可充分利用现有的存储空间。

堆栈段的空间最小，只有512个字节单元，它的地址范围是84180H~8437FH。



习题

已知 8086系统某存储单元物理地址为:52506H,
段基址的最大值是 [填空1] , 最小值是 [填空2]
。

要求：填空1和填空2需要用完整的五位十六进制来表示，末尾还要加上十六进制数的标志H.

习题

已知 8086系统某存储单元物理地址为:52506H,
段基址的最大值是 [5250H] , 最小值是 [4251H]
。

要求：填空1和填空2需要用完整的五位十六进制来表示，末尾还要加上十六进制数的标志H.

2.2.5 CPU对内存的读写操作

CPU要想进行数据的读写，必须和外部器件（芯片）进行下面3类的信息交互。

- (1) 存储单元的地址（地址信息）
- (2) 器件的选择，读或写命令（控制信息）
- (3) 读或写的数据（数据信息）

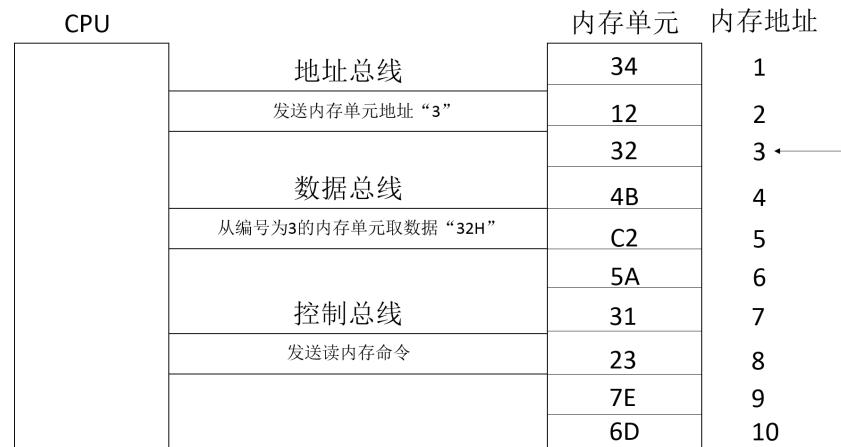
2.2.5 CPU对内存的读写操作

CPU从地址为3的内存单元中读取数据的过程

- (1) CPU通过地址线将要进行操作的内存单元地址“3”发出；
- (2) CPU通过控制线发出内存读命令，选中存储器芯片，并通知它，将要从中读取数据
- (3) 存储器将内存单元地址为“3”中的数据“32H”通过数据线送入CPU

写操作与读操作的步骤相似，如向地址为“3”的单元写入数据“FFH”。

- (1) CPU通过地址线将要进行操作的内存单元地址“3”发出；
- (2) CPU通过控制线发出内存写命令，选中存储器芯片，并通知它，要向其中写入数据。
- (3) CPU通过数据线将数据“FFH”送入内存的地址为“3”的单元中。



2.2.5 CPU对内存的读写操作

写操作与读操作的步骤相似，如向地址为“3”的单元写入数据“FFH”。

- (1) CPU通过地址线将要进行操作的内存单元地址“3”发出；
- (2) CPU通过控制线发出内存写命令，选中存储器芯片，并通知它，要向其中写入数据。
- (3) CPU通过数据线将数据“FFH”送入内存的地址为“3”的单元中。



2.2.5 CPU对内存的读写操作

- 要让一个计算机或微处理器工作，应向它输入能够驱动它进行工作的电平信息（即机器码）。

对于8086 CPU，下面的机器码，能够完成从内存地址为“3”的单元读数据。

- 机器码：101000010000001100000000
- 含义：从内存地址为“3”的单元读取数据送入寄存器AX
- 机器码是01串，难以记忆和书写，用汇编指令来表示，情况如下。
 - 机器码：10100001 00000011 00000000
 - 对应的汇编指令：Mov AX, [3]
 - 含义：从内存地址为“3”的单元读取数据送入寄存器AX

2.3 中央处理器（CPU）中的寄存器

- 2.3.1 寄存器介绍
- 2.3.2 CS和IP
- 2.3.3 堆栈

2.3.1 寄存器介绍

1. 通用数据寄存器

- 8086 CPU的所有寄存器都是16位的，可以存放两个字节。**AX**、**BX**、**CX**、**DX**这4个寄存器通常用来存放一般性的数据，有时候也可以存放地址，被称为通用数据寄存器。
- ①**AX**: 累加器，运算时较多使用这个寄存器，有些指令规定必须使用它。
- ②**BX**: 基址寄存器，除了存放数据，它经常用来存放一段内存的起始偏移地址。
- ③**CX**: 计数寄存器，除了存放数据，它经常用来存放重复操作的次数。
- ④**DX**: 数据寄存器，除了存放数据，它有时存放32位数据的高16位。

2.3.1 寄存器介绍

2. 地址寄存器

- 16位的8086处理器有4个16位的通用地址寄存器。它们的主要作用是存放数据的所在偏移地址，也可以存放数据。这4个寄存器不能再拆分使用。
- ①**SP**: 堆栈指针，这是一个专用的寄存器，存放堆栈栈顶的偏移地址。
- ②**BP**: 基址指针，可以用来存放内存中数据的偏移地址。
- ③**SI**: 源变址寄存器，它经常用来存放内存中源数据区的偏移地址，所谓变址寄存器，是指在某些指令作用下它可以自动地递增或递减其中的值。
- ④**DI**: 目的变址寄存器，它经常用来存放内存中目的数据区的偏移地址，并在某些指令作用下可以自动地递增或递减其中的值。

2.3.1 寄存器介绍

3.段寄存器

- 16位80x86处理器有4个16位的段寄存器，分别命名为CS， SS， DS， ES。它们用来存放4个段的段基址。
- ①**CS**: 代码段寄存器，用来存放当前正在执行的程序段的段基址。
- ②**SS**: 堆栈段寄存器，用来存放堆栈段的段基址。
- ③**DS**: 数据段寄存器，用来存放数据段段基址。
- ④**ES**: 附加段寄存器，用来存放另一个数据段的段基址。

2.3.1 寄存器介绍

4. 指令指针寄存器

- IP：指令指针寄存器，存放即将执行指令的偏移地址。

5. 指令指针寄存器

- FLAGS：存放CPU的两类标志。
- 状态标志：反映处理器当前的状态，如有无溢出，有无进位等。
- 状态标志有6个：CF、PF、AF、ZF、SF和OF。见表2-2
- 控制标志：用来控制处理器的工作方式，如是否响应可屏蔽中断等
- 控制标志有3个：TF、IF和DF

2.3.2 CS和IP

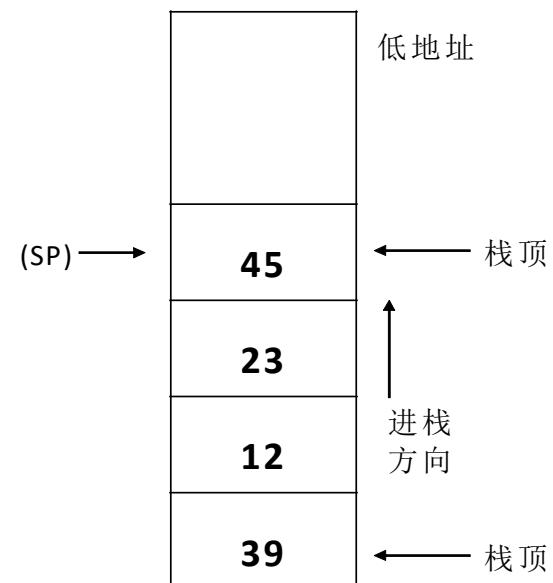
8086 CPU的工作过程可以简要描述如下。

- (1) 从CS:IP指向的内存单元读取指令，读取的指令进入指令缓冲器；
- (2) $IP=IP+所读取指令的长度$ ，从而指向下一条指令；
- (3) 执行指令，转到步骤(1)，重复这个过程。

CS和IP的内容提供了CPU要执行指令的地址。

2.3.3 堆栈

- 堆栈区就是这样一个特殊的存储区，它的末单元称为栈底，数据先从栈底开始存放，最后存入的数据所在单元称为栈顶。当堆栈区为空时，栈顶和栈底是重合的。数据在堆栈区存放时，必须以字存入，每次存入一个字，后存入的数据依次放入栈的低地址单元中。栈指针**SP**每次减2，由栈指针**SP**指出当前栈顶的位置，数据存取时采用后进先出的方式



2.4 外部设备和接口

- 外部设备是计算机系统的不可缺少的重要组成部分。通过输入设备把程序和数据输入计算机主机（CPU和内存），通过输出设备把结果输出给用户或保存起来。
- 根据不同用途，接口中的寄存器（端口）分为以下3类。
 - (1) 数据端口
 - (2) 控制端口
 - (3) 状态端口
- CPU与I / O接口中端口的信息传输也都是通过**数据总线**进行的。

2.5 32位80X86 CPU 的工作模式

- 实模式
- 保护模式
- 虚拟8086模式