第二章 寄存器

寄存器是CPU中进行**指令读写**的部件，也是汇编程序员最关心的器件。我们通过改变寄存器中的内容实现对CPU的控制。

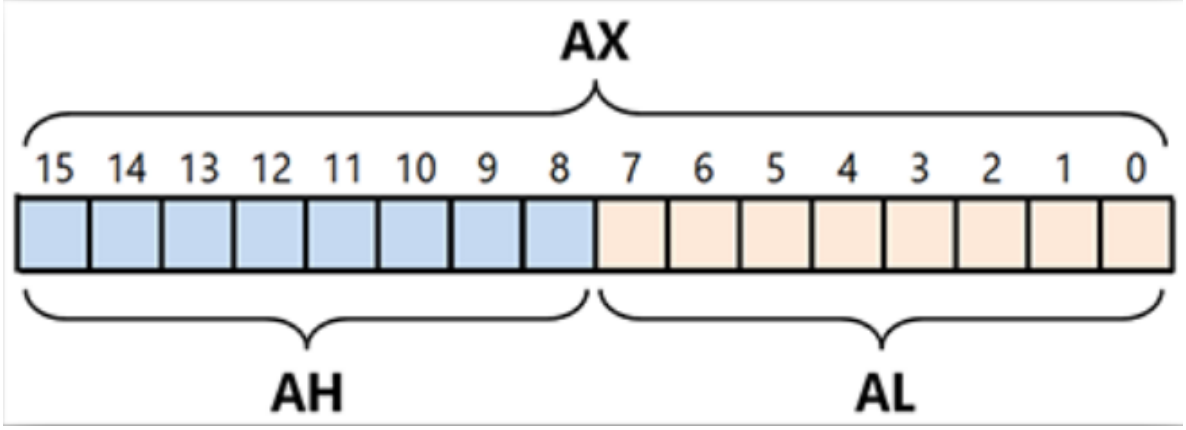
8086CPU中共有以下的14个寄存器，AX, BX, CX, DX, SI, DI SP, BP, IP, CS,SS, DS, ES, PSW。

每个寄存器都有其相应的功能，这里不一一介绍。

通用寄存器

8086CPU中所有的寄存器都是16位的，可以存放两个字节。其中**AX, BX, CX, DX**通常用来存放一般性的数据，因而被称为通用寄存器。

以AX寄存器为例，其逻辑结构如下图所示。



一

上图中将AX分成了AH，AL两个部分。上一代的8088CPU的所有的寄存器都是8位的，为了保证兼容，8086CPU中的**四个通用寄存器都可以分为两个8位的寄存器来使用。**

AX的低八位是AL（Low），AX的高八位是AH（High）。BX，CX，DX同理。

其余的寄存器都是不可拆分的。

寄存器	寄存器中的内容	所表示的值
AX	0100111100010000	4E20H
AH	01001110	4EH
AL	00100000	20H

几个简单的汇编指令

mov 指令

mov ax, 4E20H， 将4E20H送如ax寄存器中， 相当与 ax = 18H

mov ax, bx， 将bx中的内容送到ax寄存器中， 相当与 ax = bx

注意：在进行数据传送的时，**两个操作对象的位数应当是一致的**。

mov ax, bl (X)

mov al, 4E20H (X)

mov ah, bx (X)

add 指令

add ax, 10H，将ax寄存器中内容加10H，相当于 ax += 10H

add ax, bx，将ax寄存器中的内容加bx寄存器中的内容，相当与 ax += bx

注意：

add指令可能会**存在溢出的情况**。如ax中的数据为8226H。

```
mov bx, ax      ; 此时bx中的数据也是8226H
add ax, bx      ; 结果为1044CH，但是AX中只能存放16位。
what is ax now ?
```

此时ax中的数据为**044CH**，溢出的1并没有丢失，而是存放到了其余的地方中，这里不做讨论。

注意：如果**al寄存器中因为add产生了溢出，溢出的不会存放到ah中**。此时al是作为一个单独的8位寄存器来使用的，不会影响到ah中的值。

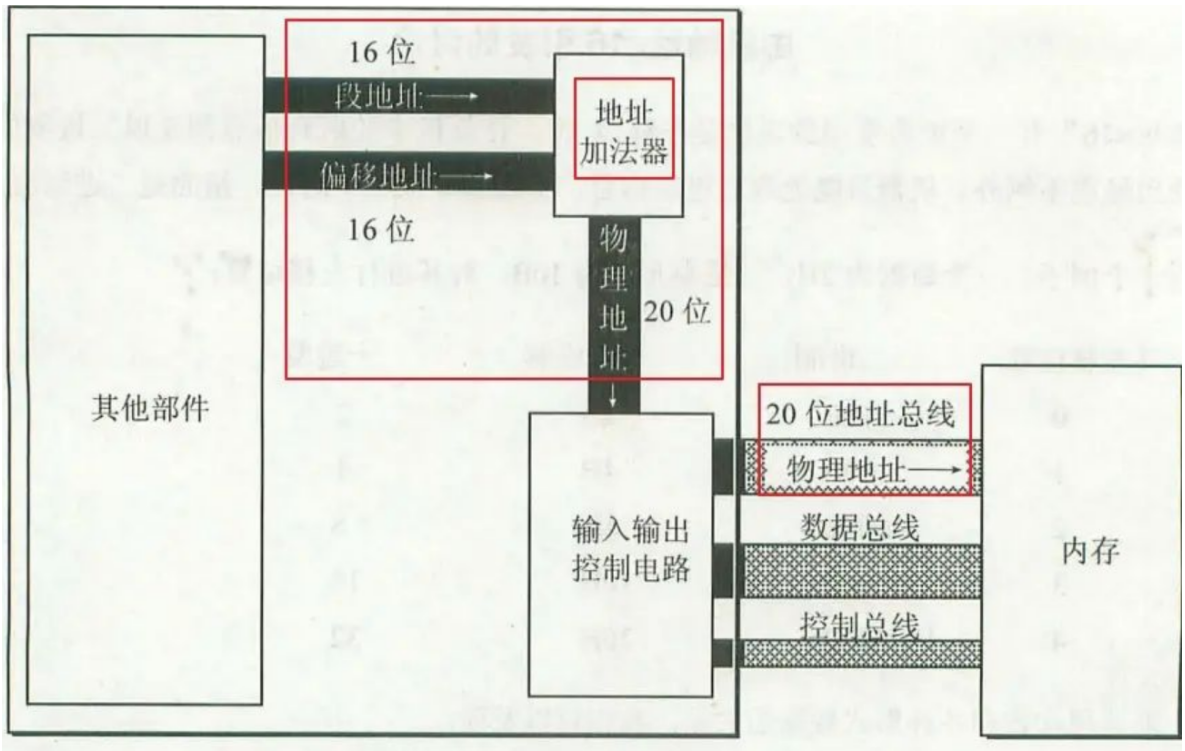
8086CPU物理地址

8086CPU是16位机。所谓的16位机。

1. 运算器一次最多可以处理16位的数据。
2. 寄存器的最大宽度为16位。
3. 寄存器和运算器之间的通路为20位。

但是8086CPU有**20根地址总线**，可以达到1MB的寻址能力。

8086CPU采用一种内部用**两个16位的地址合成**的方法来形成一个20位的物理地址。



如图所示，地址加法器使用**段地址 * 10H + 偏移地址**的方法合成物理地址。

如果想要访问物理地址123C8H，可以使用段地址为1230H，偏移地址为00C8H。

地址加法器通过运算**1230H*10H + 00C8H**得到真实的物理地址123C8H。

很容易可以看出来，同一个物理地址，可以**使用不同的段地址和偏移地址**来得到，所谓的段地址*10H就是将段地址的二进制左移4位（16进制左移1位），通过这种方法，我们可以得到表示一个物理地址为段地址和偏移地址的最简单的方式。

如12345H可以表示为段地址1234H，偏移地址5H，也可以表示为段地址1230H，偏移地址45H.....

什么是段

CPU在内存中并没有分段，但是我们根据实际编程的需要可以将一段连续的内存单元看成一个段，其中**段地址*10H**作为这个段是起始地址（基础地址），使用**偏移地址**来定位段中的内存单元。段的起始地址一定是16的倍数，偏移地址是16位，寻址能力为64KB，因而**一个段的最大的大小为64KB**。

段寄存器

8086CPU中有4个段寄存器，分别为CS, DS, SS, ES。当8086CPU要访问内存时由这四个段寄存器提供内存地址。

CS 和 IP

功能介绍

CS和IP这两个寄存器是8086寄存器中两个最关键的寄存器，它们**指示了CPU当前要读取指令的地址。**

CS为代码段寄存器，IP为指令指针寄存器。

在8086机中，任何时刻，CPU将CS:IP所指向的内容当作指令执行。

上面提到在机器中指令和数据本质上是一样的，如果一段数据被CS:IP指向，那么它就是作为指令来执行的。

8086CPU的工作流程

1. 从CS:IP指向的内存单元中读取指令，读取的指令进入内存单元。
2. IP += 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,重复执行。

如果修改CS和IP寄存器中的内容

之前学过的是一个mov指令，mov指令是可以修改寄存器中的内容的，其实mov指令可以修改8086CPU中的大部分的寄存器，但是无法修改CS和IP寄存器。

mov指令被称为**传送指令**。

如果要修改CS和IP寄存器需要使用**转移指令**。

一个最简单的方法就是jmp指令。

可以使用**jmp 段地址：偏移地址**的方法修改CS和IP的值，也可以使用**jmp 偏移地址**之修改IP的值。

如: jmp 2000H:38H执行之后CS = 2000H, IP= 38H

如果我们**只想修改IP寄存器的值**，可以使用**jmp 某一合法寄存器**的方法。

如果 jmp ax执行之后 IP = AX，这条指令在含义上和mov ip, ax类似。(mov ip, ax是非法的)

DosBox的使用

常用功能

命令	功能
R	查看，改变CPU寄存器的内存
D	查看内存中内容
E	改写内存中的内容
U	将内存中的内容翻译为汇编指令
T	执行一条机器指令
A	以汇编语言的格式在内存中写入一条机器指令