

1、数据寄存器

数据寄存器主要用来保存操作数和运算结果等信息，从而节省读取操作数所需占用总线和访问存储器的时间。

32位CPU有4个32位的通用寄存器EAX、EBX、ECX和EDX。对低16位数据的存取，不会影响高16位的数据。这些低16位寄存器分别命名为：AX、BX、CX和DX，它和先前的CPU中的寄存器相一致。

4个16位寄存器又可分割成8个独立的8位寄存器(AX：AH-AL、BX：BH-BL、CX：CH-CL、DX：DH-DL)，每个寄存器都有自己的名称，可独立存取。程序员可利用数据寄存器的这种“可分可合”的特性，灵活地处理字/字节的信息。

- 寄存器AX和AL通常称为累加器(Accumulator)，用累加器进行的操作可能需要更少时间。累加器可用于乘、除、输入/输出等操作，它们的使用频率很高；
- 寄存器BX称为基址寄存器(Base Register)。它可作为存储器指针来使用；
- 寄存器CX称为计数寄存器(Count Register)。在循环和字符串操作时，要用它来控制循环次数；在位操作中，当移多位时，要用CL来指明移位的位数；
- 寄存器DX称为数据寄存器(Data Register)。在进行乘、除运算时，它可作为默认的操作数参与运算，也可用于存放I/O的端口地址。

在16位CPU中，AX、BX、CX和DX不能作为基址和变址寄存器来存放存储单元的地址，但在32位CPU中，其32位寄存器EAX、EBX、ECX和EDX不仅可传送数据、暂存数据保存算术逻辑运算结果，而且也可作为指针寄存器，所以，这些32位寄存器更具有通用性。详细内容请见第3.8节——32位地址的寻址方式。

2、变址寄存器

32位CPU有2个32位通用寄存器ESI和EDI。其低16位对应先前CPU中的SI和DI，对低16位数据的存取，不影响高16位的数据。

寄存器ESI、EDI、SI和DI称为变址寄存器(Index Register)，它们主要用于存放存储单元在段内的偏移量，用它们可实现多种存储器操作数的寻址方式(在第3章有详细介绍)，为以不同的地址形式访问存储单元提供方便。

变址寄存器不可分割成8位寄存器。作为通用寄存器，也可存储算术逻辑运算的操作数和运算结果。

它们可作一般的存储器指针使用。在字符串操作指令的执行过程中，对它们有特定的要求，而且还具有特殊的功能。具体描述请见第5.2.11节。

3、指针寄存器

32位CPU有2个32位通用寄存器EBP和ESP。其低16位对应先前CPU中的SBP和SP，对低16位数据的存取，不影响高16位的数据。

寄存器EBP、ESP、BP和SP称为指针寄存器(Pointer Register)，主要用于存放堆栈内存单元的偏移量，用它们可实现多种存储器操作数的寻址方式(在第3章有详细介绍)，为以不同的地址形式访问存储单元提供方便。

指针寄存器不可分割成8位寄存器。作为通用寄存器，也可存储算术逻辑运算的操作数和运算结果。

它们主要用于访问堆栈内的存储单元，并且规定：

- BP为基指针(Base Pointer)寄存器，用它可直接存取堆栈中的数据；
- SP为堆栈指针(Stack Pointer)寄存器，用它只可访问栈顶。

4、段寄存器

段寄存器是根据内存分段的管理模式而设置的。内存单元的物理地址由段寄存器的值和一个偏移量组合而成的，这样可用两个较少位数的值组合成一个可访问较大物理空间的内存地址。

CPU内部的段寄存器：

- CS——代码段寄存器(Code Segment Register)，其值为代码段的段值；
- DS——数据段寄存器(Data Segment Register)，其值为数据段的段值；
- ES——附加段寄存器(Extra Segment Register)，其值为附加数据段的段值；

- SS——堆栈段寄存器(Stack Segment Register)，其值为堆栈段的段值；
- FS——附加段寄存器(Extra Segment Register)，其值为附加数据段的段值；
- GS——附加段寄存器(Extra Segment Register)，其值为附加数据段的段值。

在16位CPU系统中，它只有4个段寄存器，所以，程序在任何时刻至多有4个正在使用的段可直接访问；在32位微机系统中，它有6个段寄存器，所以，在此环境下开发的程序最多可同时访问6个段。

32位CPU有两个不同的工作方式：实方式和保护方式。在每种方式下，段寄存器的作用是不同的。有关规定简单描述如下：

实方式：前4个段寄存器CS、DS、ES和SS与先前CPU中的所对应的段寄存器的含义完全一致，内存单元的逻辑地址仍为“段值：偏移量”的形式。为访问某内存段内的数据，必须使用该段寄存器和存储单元的偏移量。

保护方式：在此方式下，情况要复杂得多，装入段寄存器的不再是段值，而是称为“选择子”(Selector)的某个值。段寄存器的具体作用在此不作进一步介绍了，有兴趣的读者可参阅其它科技资料。

5、指令指针寄存器

32位CPU把指令指针扩展到32位，并记作EIP，EIP的低16位与先前CPU中的IP作用相同。

指令指针EIP、IP(Instruction Pointer)是存放下次将要执行的指令在代码段的偏移量。在具有预取指令功能的系统中，下次要执行的指令通常已被预取到指令队列中，除非发生转移情况。所以，在理解它们的功能时，不考虑存在指令队列的情况。

在实方式下，由于每个段的最大范围为64K，所以，EIP中的高16位肯定都为0，此时，相当于只用其低16位的IP来反映程序中指令的执行次序。