|  |  |
| --- | --- |
| **基本寄存器** | |
| **通用寄存器**  通用寄存器是那些你可以根据自己的意愿使用的寄存器，修改他们的值通常不会对计算机的运行造成很大的影响 | |
| EAX/AX/AL/AH | [通用寄存器](http://baike.baidu.com/view/1418486.htm)。相对其他[寄存器](http://baike.baidu.com/view/6159.htm)，在进行运算方面比较常用。在保护模式中，也可以作为[内存](http://baike.baidu.com/view/1082.htm)偏移指针（此时，DS作为段 [寄存器](http://baike.baidu.com/view/6159.htm)或选择器） |
| EBX/BX/BL/BH | [通用寄存器](http://baike.baidu.com/view/1418486.htm)。通常作为[内存](http://baike.baidu.com/view/1082.htm)偏移[指针](http://baike.baidu.com/view/159417.htm)使用（相对于EAX、ECX、EDX），DS是默认的[段寄存器](http://baike.baidu.com/view/364403.htm)或选择器。在保护模式中，同样可以起这个作用。 |
| ECX/CX/CL/CH | [通用寄存器](http://baike.baidu.com/view/1418486.htm)。通常用于特定[指令](http://baike.baidu.com/view/178461.htm)的计数。在保护模式中，也可以作为[内存](http://baike.baidu.com/view/1082.htm)偏移指针（此时，DS作为[寄存器](http://baike.baidu.com/view/6159.htm)或段选择器）。 |
| EDX/DX/DL/DH | [通用寄存器](http://baike.baidu.com/view/1418486.htm)。在某些运算中作为EAX的溢出[寄存器](http://baike.baidu.com/view/6159.htm)（例如乘、除）。在保护模式中，也可以作为[内存](http://baike.baidu.com/view/1082.htm)偏移指针（此时，DS作为段 [寄存器](http://baike.baidu.com/view/6159.htm)或选择器）。 |
| **用作内存指针的特殊寄存器** | |
| ESI/SI | 通常在[内存](http://baike.baidu.com/view/1082.htm)操作[指令](http://baike.baidu.com/view/178461.htm)中作为“源地址指针”使用。当然，ESI可以被装入任意的数值，但通常没有人把它当作[通用寄存器](http://baike.baidu.com/view/1418486.htm)来用。DS是默认段寄存器或选择器。 |
| EDI/DI | 通常在内存操作指令中作为“目的地址[指针](http://baike.baidu.com/view/159417.htm)”使用。当然，EDI也可以被装入任意的数值，但通常没有人把它当作[通用寄存器](http://baike.baidu.com/view/1418486.htm)来用。ES是默认段寄存器或选择器。 |
| EBP/BP  ESP/SP | 作为指针的[寄存器](http://baike.baidu.com/view/6159.htm)，也可作为16位寄存器BP, SP使用，常用于椎栈操作。通常，它被高级语言[编译器](http://baike.baidu.com/view/487018.htm)用以建造‘[堆栈](http://baike.baidu.com/view/93201.htm)帧'来保存[函数](http://baike.baidu.com/view/15061.htm)或过程的[局部变量](http://baike.baidu.com/view/552847.htm)，不过，还是那句话，你可以在其中保存你希望的任何数据。SS是它的默认[段寄存器](http://baike.baidu.com/view/364403.htm)或选择器。 |
| **段选择器**  [实模式](http://baike.baidu.com/view/404433.htm)下的[段寄存器](http://baike.baidu.com/view/364403.htm)到保护模式下摇身一变就成了选择器。不同的是，实模式下的“[段寄存器](http://baike.baidu.com/view/364403.htm)”是16-bit的，而保护模式下的选择器是32-bit的。 | |
| CS | [代码](http://baike.baidu.com/view/41.htm)段，或代码选择器。同IP[寄存器](http://baike.baidu.com/view/6159.htm)(稍后介绍)一同指向当前正在执行的那个地址。处理器执行时从这个[寄存器](http://baike.baidu.com/view/6159.htm)指向的段（实模式）或[内存](http://baike.baidu.com/view/1082.htm)（保护模式）中获取[指令](http://baike.baidu.com/view/178461.htm)。除了跳转或其他分支[指令](http://baike.baidu.com/view/178461.htm)之外，你无法修改这个[寄存器](http://baike.baidu.com/view/6159.htm)的内容。 |
| DS | [数据段](http://baike.baidu.com/view/1005328.htm)，或[数据选择器](http://baike.baidu.com/view/990200.htm)。这个[寄存器](http://baike.baidu.com/view/6159.htm)的低16 bit连同ESI一同指向的[指令](http://baike.baidu.com/view/178461.htm)将要处理的[内存](http://baike.baidu.com/view/1082.htm)。同时，所有的[内存](http://baike.baidu.com/view/1082.htm)操作[指令](http://baike.baidu.com/view/178461.htm)默认情况下都用它指定操作段(实模式)或内存(作为选择器，在保护模式。这个[寄存器](http://baike.baidu.com/view/6159.htm)可以被装入任意数值，然而在这么做的时候需要小心一些。方法是，首先把数据送给AX，然后再把它从AX传送给DS(当然，也可以通过[堆栈](http://baike.baidu.com/view/93201.htm)来做). |
| ES | 附加段，或附加选择器。这个[寄存器](http://baike.baidu.com/view/6159.htm)的低16 bit连同EDI一同指向的[指令](http://baike.baidu.com/view/178461.htm)将要处理的[内存](http://baike.baidu.com/view/1082.htm)。同样的，这个[寄存器](http://baike.baidu.com/view/6159.htm)可以被装入任意数值，方法和DS类似。 |
| FS | F段或F选择器(推测F可能是Free?)。可以用这个[寄存器](http://baike.baidu.com/view/6159.htm)作为默认[段寄存器](http://baike.baidu.com/view/364403.htm)或选择器的一个替代品。它可以被装入任何数值，方法和DS类似。 |
| GS | G段或G选择器(G的意义和F一样，没有在Intel的文档中解释)。它和FS几乎完全一样。 |
| SS | [堆栈段](http://baike.baidu.com/view/76043.htm)或堆栈选择器。这个[寄存器](http://baike.baidu.com/view/6159.htm)的低16 bit连同ESP一同指向下一次[堆栈](http://baike.baidu.com/view/93201.htm)操作(push和pop)所要使用的堆栈地址。这个[寄存器](http://baike.baidu.com/view/6159.htm)也可以被装入任意数值，你可以通过入栈和[出栈](http://baike.baidu.com/view/346791.htm)操作来给他赋值，不过由于[堆栈](http://baike.baidu.com/view/93201.htm)对于很多操作有很重要的意义，因此，不正确的修改有可能造成对堆栈的破坏。 |
| **指令指针寄存器** | |
| EIP/IP | 这个[寄存器](http://baike.baidu.com/view/6159.htm)非常的重要。这是一个32[位宽](http://baike.baidu.com/view/1128.htm)的[寄存器](http://baike.baidu.com/view/6159.htm)，同CS一同指向即将执行的那条[指令](http://baike.baidu.com/view/178461.htm)的地址，存放指令的[偏移地址](http://baike.baidu.com/view/883224.htm)。[微处理器](http://baike.baidu.com/view/1125.htm)工作于实模式下，EIP是IP（16位）[寄存器](http://baike.baidu.com/view/6159.htm)。不能够直接修改这个[寄存器](http://baike.baidu.com/view/6159.htm)的值，修改它的唯一方法是跳转或分支[指令](http://baike.baidu.com/view/178461.htm)。(CS是默认的段或选择器) |
| [**标志寄存器**](http://baike.baidu.com/view/1845107.htm)  EFR（extra flags register）包括状态位、控制位和系统标志位，用于指示[微处理器](http://baike.baidu.com/view/1125.htm)的状态并控制微处理器的操作。 | |
| [状态标志](http://baike.baidu.com/view/2919142.htm)位 | 包括进位标志CF、奇偶标志PF、[辅助进位标志](http://baike.baidu.com/view/1330728.htm)AF、零标志ZF 、[符号](http://baike.baidu.com/view/115742.htm)标志SF和溢出标志OF。 |
| 控制标志位 | 包括陷阱标志（单步操作标志）TF、中断标志IF和方向标志DF。80486 CPU[标志寄存器](http://baike.baidu.com/view/1845107.htm)中的状态标志位和控制标志位与8086 CPU标志寄存器中的[状态标志](http://baike.baidu.com/view/2919142.htm)位和控制标志位的功能完全一样，这里就不再赘述。 |
| 系统标志位和IOPL字段 | 在EFR[寄存器](http://baike.baidu.com/view/6159.htm)中的系统标志和IOPL字段，用于控制[操作系统](http://baike.baidu.com/view/880.htm)或执行某种操作。它们不能被[应用程序](http://baike.baidu.com/view/330120.htm)修改。 |
| IOPL | 输入/输出[特权级](http://baike.baidu.com/view/2188561.htm)标志位。它规定了能使用I/O[敏感指令](http://baike.baidu.com/view/4849713.htm)的[特权级](http://baike.baidu.com/view/2188561.htm)。在保护模式下，利用这两位[编码](http://baike.baidu.com/view/237708.htm)可以分别表示0, 1, 2, 3这四种[特权级](http://baike.baidu.com/view/2188561.htm)，0级特权最高，3级特权最低。在80286以上的处理器中有一些I/O敏感[指令](http://baike.baidu.com/view/178461.htm)，如CLI（关[中断指令](http://baike.baidu.com/view/6402585.htm)）、STI（开中断[指令](http://baike.baidu.com/view/178461.htm)）、IN（输入）、OUT（输出）。IOPL的值规定了能执行这些指令的[特权级](http://baike.baidu.com/view/2188561.htm)。只有特权高于IOPL的程序才能执行I/O敏感[指令](http://baike.baidu.com/view/178461.htm)，而特权低于IOPL的程序，若企图执行敏感指令，则会引起异常中断。 |
| NT(nested task flag) | 任务嵌套标志。在保护模式下，指示当前执行的任务嵌套于另一任务中。当任务被嵌套时，NT=1，否则NT=0。 |
| RF(resume flag) | 恢复标志。与调试[寄存器](http://baike.baidu.com/view/6159.htm)一起使用，用于保证不重复处理[断点](http://baike.baidu.com/view/278359.htm)。当RF=1时，即使遇到断点或故障，也不产生异常中断。 |
| VM(virtual 8086 mode flag) | 虚拟8086模式标志。用于在保护模式系统中选择虚拟操作模式。VM=1，启用虚拟8086模式；VM=0，返回保护模式。 |
| AC(alignment check flag） | [队列](http://baike.baidu.com/view/38959.htm)检查标志。如果在不是字或双字的边界上寻址一个字或双字，队列检查标志将被激活。 |
| **其他寄存器** | |
| CR0～CR3 ([控制寄存器](http://baike.baidu.com/view/4092816.htm))。举一个例子，CR0的作用是切换实模式和保护模式。  D0, D1, D2, D3, D6和D7(调试寄存器)。他们可以作为[调试器](http://baike.baidu.com/view/185339.htm)的硬件支持来设置条件断点。  TR3, TR4, TR5, TR6 和TR? [寄存器](http://baike.baidu.com/view/6159.htm)(测试寄存器)用于某些条件测试。 | |