**机器语言**：机器指令是CPU能直接识别和执行的指令。

机器指令 = 操作码（功能） + 操作数（对象）

不同种类的CPU对应的指令是不同的，而且它们的指令系统往往有很大的差别

对CPU的要求：新一代的指令系统必须包含之前同一系列的CPU的指令系统

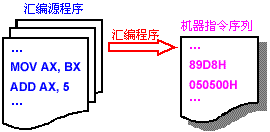
机器语言是CPU能直接识别的唯一一种语言，也就是说CPU能直接执行用机器语言描述的程序

机器语言的缺点：不易读、出错率高、难以维护

优点：执行效率高，没有额外的操作

**汇编语言：**用指令助忆符、符号地址等组成的符号指令称为汇编格式指令

把用汇编语言编写的源程序翻译成CPU能够识别的机器指令序列，把这种翻译程序称为汇编程序



常用的汇编程序有：MASM TASM DEBUG

**汇编语言的特点**：

1. 与机器相关性：汇编语言指令是机器指令的一种符号表示，而不同的CPU有不同的机器指令系统，也就有不同的汇编语言，所以汇编语言与机器有着密切的相关性

所以除了同系列不同型号的CPU之间的汇编语言有一定的可移植性外，其它不同类型的CPU之间的汇编语言是不具备可移植性的。

所以汇编语言的通用型和可移植性比一般的高级语言要低很多。

1. 执行效率高：正是由于汇编语言是机器相关的，所以程序员写汇编源程序时基本相当于直接写可执行代码，程序员可以在每个细节上进行优化，所以其执行效率很高。
2. 编写程序的复杂性
3. 调试的复杂性：现在可以用CV(CodeView)、TD(Turbo Debug)等软件在源程序级进行符号跟踪了。

**数值进制：**

二进制 1010B,1011B

八进制 1233Q

十进制 2323D

十六进制 0FF78 1234H (以字母开头时，必须以0开头，表示其实16进制数)

数值表示及其扩展：

在计算机内，为了表示正负数，并便于进行各种算数运算，所有符号数均采用二进制补码的形式进行表示。

正数的补码是其自身，负数的补码是将其正数的二进制码变反再加1

**二进制符号的扩展**

例：将8位补码 01111101、10111111扩展成16位

符号位像高位扩展 00000000 01111101 11111111 10111111

n位二进制的表示范围：

无符号整数的范围： 0~2n-1;

有符号整数（补码数）的范围：-2n-1~2n-1-1

n通常取值位8或16

当n= 8 时，范围是：-128~127

当 n=16 时，范围是：-32768~32767

BCD码：我们将每位10进制数转化为2进制编码最后得到的编码就是BCD码

计算机除了进行数值计算的能力之外还有进行非数值计算的能力。ASCII是目前应用极广的一种编码，它采用7位编码。

汇编语言中的常见数据类型：字节（8位）、字（16位）、双字（32位）

计算机的硬件资源：CPU内部资源，存储器资源，I/O端口

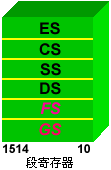
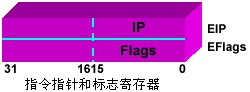
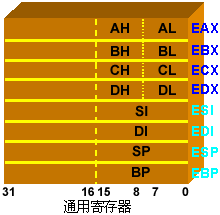
寄存器是CPU内部重要的数据存储资源

16位CPU所包含的寄存器有：

4个16位数据寄存器（AX BX CX DX） 2个变址和指针寄存器（SI BI） 2个指针寄存器（SP BP） 4个段寄存器（DS CS SS ES）一个指令寄存器（IP） 一个标志寄存器（Flags）

32位CPU所包含的寄存器有：

4个32位的通用寄存器（EAX EBX ECX EDX） 2个32位的变址和指针寄存器（ SI BI）6个段寄存器（DS CS SS ES FS GS）一个指令寄存器（EIP） 一个标志寄存器（EFlags）



**寄存器AX和AL通常被称为累加器（Accumulator）：**累加器可用于乘、除、输入输出等操作

**寄存器BX被称为基地址寄存器（Base Register）：**作为存储器指针来使用

**寄存器CX被称为计数寄存器（Count Register）:** 再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来说明移位的位数

**寄存器DX被称为数据寄存器（Data Register）:** 在进行乘除运算时，它可以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 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，寄存器EAX,EBX,ECX,EDX不仅可以传送数据、暂存数据保存算数逻辑运算结果，也可以作为指针寄存器，所以32寄存器更具通用性