

关于 8051 单片机“端”问题的说明

大多数的微处理器都有一个存储器结构，它是按照 8 个比特位即一个字节存储器位置构成。很多数据项，比如地址、数字和字符串，由于它们太长，所以不能用一个字节保存，而必须用连续的多个字节保存。

当使用保存在多个字节的数据时，字节的顺序就是一个很严重的问题。但是，对于多个字节的顺序并没有一个标准。目前，在计算机系统中广泛使用两种字节顺序存储方法。

(1) 小端

Intel 的存储器就是采用的小端模式。在小端模式中，先保存最低有效位，或者是低字节。比如：一个十六位的整数 0x1234（十进制表示为 4660），使用小端方式连续占用两个字节，即表示为：

地址 +0 +1

内容 0x34 0x12

对于一个 32 位的整数值 0x57415244（十进制表示为 1463898692），使用小端方式连续占用四个字节，即表示为：

地址 +0 +1 +2 +3

内容 0x44 0x52 0x41 0x57

(2) 大端

第二种访问多字节的方法称为大端模式，典型的，摩托罗拉计算机系统表示顺序。在大端模式中，先保存最高有效位，或者是高字节；最后是最低有效位，或者是低字节。比如：一个十六位的整数 0x1234（十进制表示为 4660），使用大端方式连续占用两个字节，即表示为：

地址 +0 +1

内容 0x12 0x34

对于一个 32 位的整数值 0x57415244（十进制表示为 1463898692），使用大端方式连续占用四个字节，即表示为：

地址 +0 +1 +2 +3

内容 0x57 0x41 0x52 0x44

对于 8051 单片机来说，它是 8 位 CPU，没有提供一条指令直接操作大于 8 位的数据对象。根据下面的规则，保存多字节数据

(1) 8051 的 LCALL 指令在堆栈内保存下一条指令的地址。压入堆栈时，先保存的低字节。因此，地址以小端方式保存在存储器中。

(2) 对于保存所有其它 16 位和 32 位值来说，与 Intel 处理器不同，采用大端模式，即首先保存高字节。比如：LJMP 和 LCALL 指令期望 16 位的地址是大端模式（从中断向量的分析可以看出）。

(3) 根据 IEEE-754 格式所保存的浮点数，是以大端格式保存，即先保存高字节。