第二章 数据的表示与处理

主讲：李博涵
bhli@nuaa.edu.cn

数据的机器级表示
数据的存储与运算

数据的存储问题

- 比特（bit）是计算机中处理、存储、传输信息的最小单位
- 二进制信息的计量单位是“字节”(Byte)，也称“位组”
  - 现代计算机中，存储器按字节编址
  - 字节是最小可寻址单位 *(addressable* unit *)*
  - 如果以字节为一个排列单位，则LSB表示最低有效字节，MSB表示最高有效字节
- 除比特和字节外，还经常使用“字”(word)作为单位
- 注意“字”和 “字长”的概念不同

字长反映 CPU 处理信息的一种能力。也指数据通路宽度。

字是计算机进行数据处理时，一次存取、加工和传送的数据长度，也用来度量数据类型的宽度。

数据的基本宽度

数据通路指CPU内部数据流经的路径以及路径上的部件，主要是CPU内部进行数据运算、存储和传送的部件，这些部件的宽度基本上要一致，才能相互匹配。因此，”字长”也等于CPU内部总线的宽度、运算器的位数、通用寄存器的宽度等。）

- 字和字长的宽度可以一样，也可不同。
  例如，x86体系结构定义“字”的宽度为16位，但从386开始字长就是32位了。

数据量的度量单位

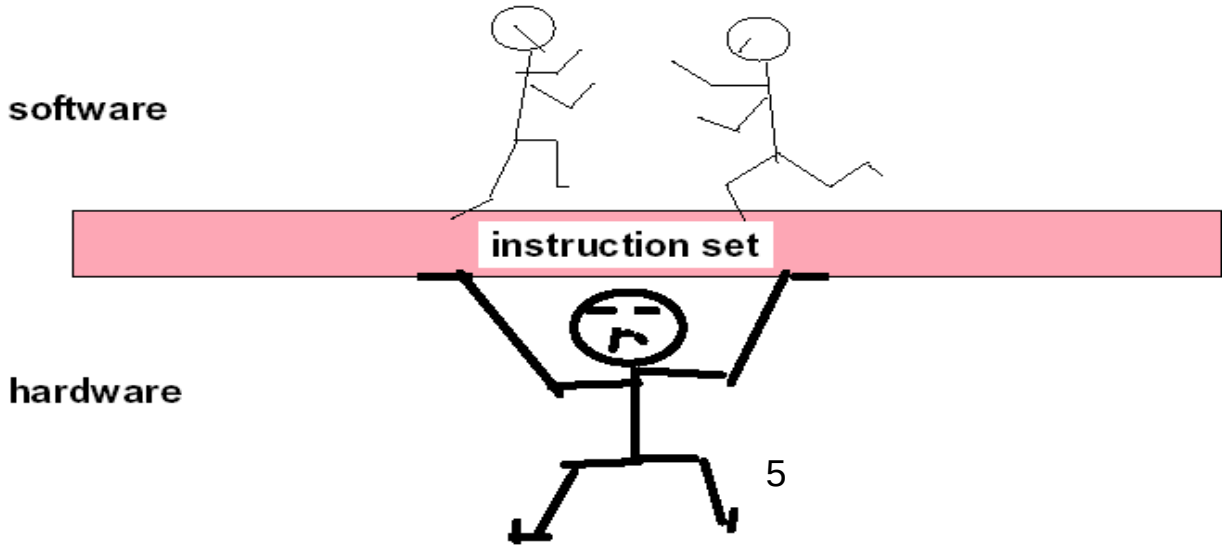
- 存储二进制信息时的度量单位要比字节或字大得多
- 容量经常使用的单位有：
  - “千字节”(KB)，1KB=2^{10}字节=1024B
  - “兆字节”(MB)，1MB=2^{20}字节=1024KB
  - “千兆字节”(GB)，1GB=2^{30}字节=1024MB
  - “兆兆字节”(TB)，1TB=2^{40}字节=1024GB
- 通信中的带宽使用的单位有：
  - “千比特/秒”(kb/s)，1kbps=10^3 b/s=1000 bps
  - “兆比特/秒”(Mb/s)，1Mbps=10^6 b/s =1000 kbps
  - 。。。

如果把b换成B，则表示字节而不是比特（位）

例如，10MBps表示 10兆字节/秒；如果是万兆，就是**10G bps**

存储和排列顺序

- 80年代开始，几乎所有机器都用字节编址
- ISA设计时要考虑的两个问题：
  - 如何根据一个字节地址取到一个32位的字？ - 字的存放问题
  - 一个字能否存放在任何字节边界？ - 字的边界对齐问题



字的存放问题

若 int i = -65535，存放在100号单元（占100～103），则用“取数”指令访问100号单元取出 i 时，必须清楚 i 的4个字节是如何存放的。

$65535=2^{16}-1$

$[-65535]_{补}$ =FFFF0001H

Word:

FF	FF	00	01
103	102	101	100
msb			lsb
100	101	102	103

little endian word 100#

big endian word 100#

大端方式（Big Endian）：MSB所在的地址是数的地址

e.g. IBM 360/370, Motorola 68k, MIPS, Sparc, HP PA

小端方式（ Little Endian）：LSB所在的地址是数的地址

e.g. Intel 80x86, DEC VAX

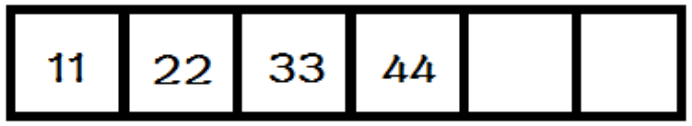
有些机器两种方式都支持，可通过特定控制位来设定采用哪种方式。

Tip：另一个易记的大小端的方法

11 22 33 44

尾端 44

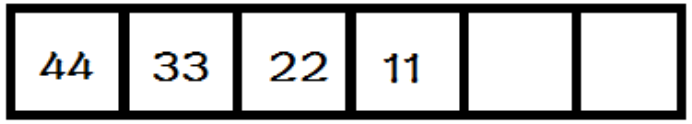
高尾端（大端）



0 1 2 3

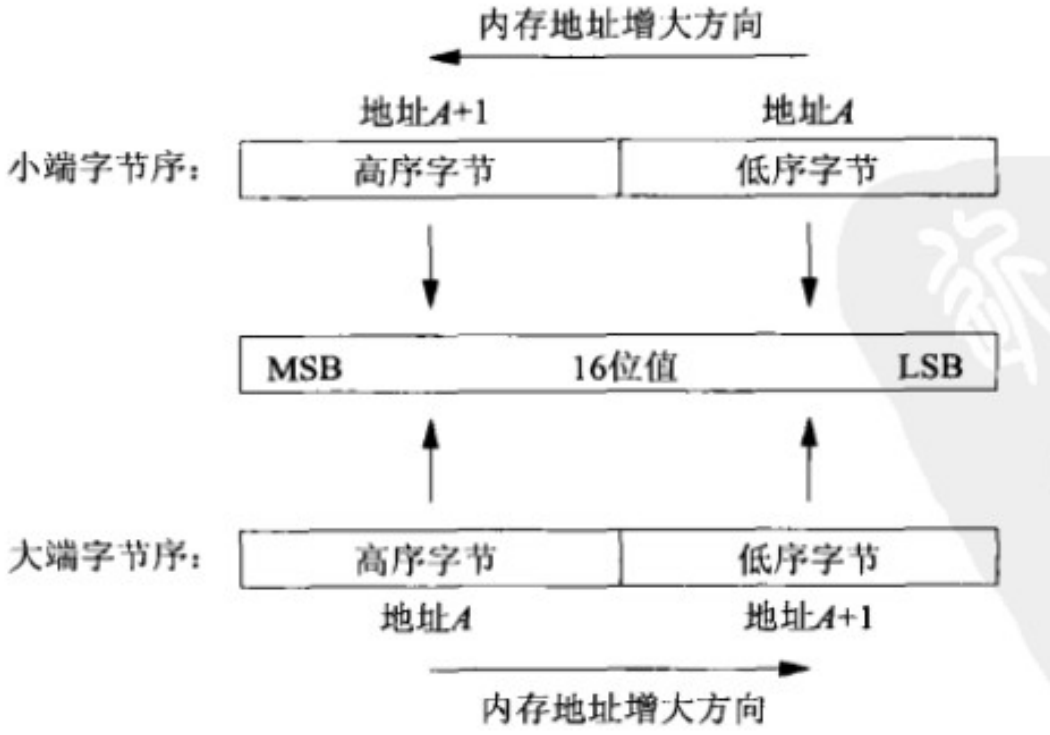
低地址 ----->高地址

低尾端（小端）



0 1 2 3

低地址 ----->高地址



对齐方式

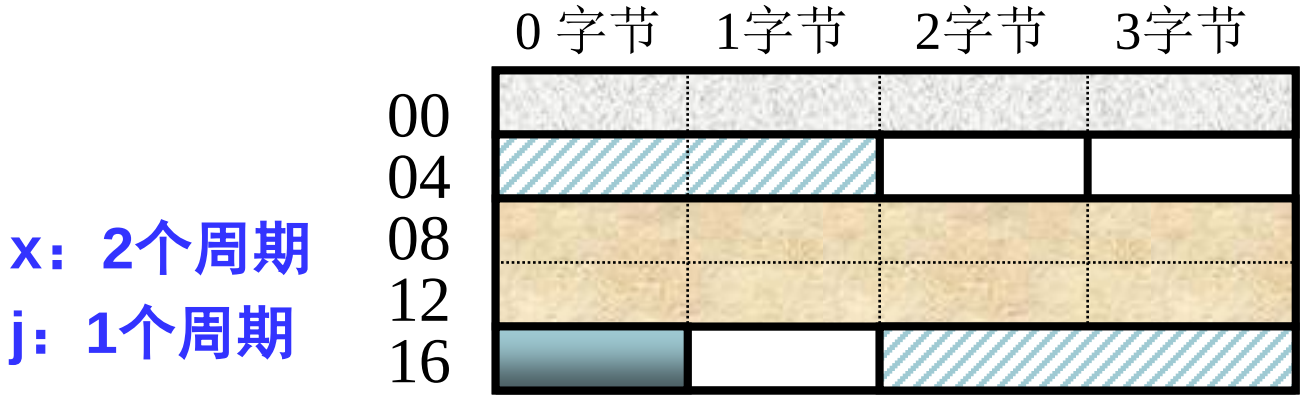
存储器按字节编址

每次只能读写某个字地址开始的4个单元中连续的1个、2个、3个或4个字节

虽节省了空间，但增加了访存次数！

需要权衡，目前来看，浪费一点存储空间没有关系！

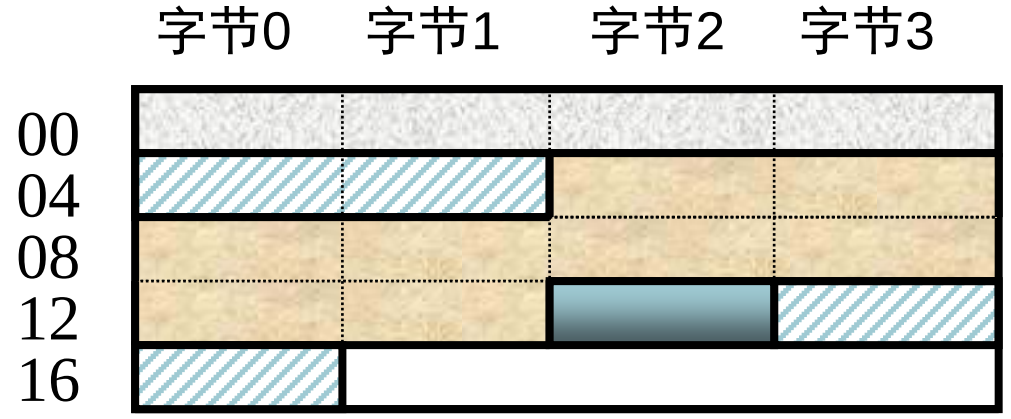
如：int i, short k, double x, char c, short j,……



x：2个周期

j：1个周期

则：&i=0; &k=4; &x=8; &c=16; &j=18;……



x：3个周期

j：2个周期

则： &i=0; &k=4; &x=6; &c=14; &j=15;……