

数据的机器级表示与运算

整数之间的转换

主讲人: 邓倩妮

上海交通大学

部分内容来自：

1. 《深入理解计算机系统》第三版，机械工业出版社，作者：Bryant,R.E.等
2. Computer Organization and Design, 4th Edition, Patterson & Hennessy



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY



对整数的操作



- 有符号、无符号数之间的转换





无符号整数

- 某些应用中，不可能出现负数，整型数中有一半的数值范围是被浪费的。
- 因此在C/C++中可以将所有的数都看成正整数，称为无符号数

无符号数：在各整数类型前加上关键词 **unsigned**

unsigned int	$0 \sim 2^{32}-1$
unsigned short	$0 \sim 65535$
unsigned long	$0 \sim 2^{32}-1$ (32 位机器)



Signed vs. Unsigned in C语言



■ 常量

- 缺省情况下为 带符号整数
- 如果 “U” 作后缀，就是无符号整数 例如：**0U, 4294967259U**

■ 转换 Casting

- 显式转换

- **int tx, ty;**

unsigned ux, uy;

tx = (int) ux;

uy = (unsigned) ty;

- 隐式转换：赋值、过程调用时

tx = ux;

uy = ty;

- 表达式中如果有 unsigned :

- **signed 会自动转换为 unsigned**

- **包括比较操作 <, >, ==, <=, >=**

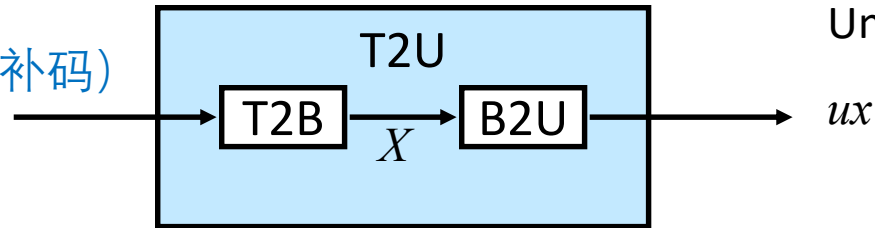


Signed & Unsigned 之间的转换



带符号整数

Two's Complement (补码)
 x



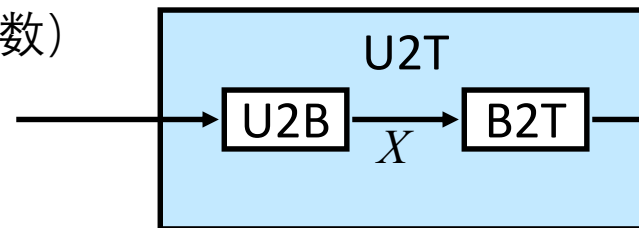
Unsigned (无符号数)

ux

Maintain Same Bit Pattern

Unsigned (无符号数)

ux



带符号整数

Two's Complement (补码)
 x

Maintain Same Bit Pattern

- unsigned 与带符号整数 之间的转换：
- 维持机器数的格式不变，重新解释



举例



以下C语言代码：

```
short si=-8196; // 1101 1111 1111 1100
```

```
unsigned short usi=si; // 1101 1111 1111 1100
```

执行以上程序后，usi的值为：

A.8196 B.34572 C.57339 D.57340

答案：D



举例

- 同时有无符号数和有符号数参加运算时，C编译器会隐含的将带符号整数强制类型转换为无符号数。
- 例如：以下程序试图求所有数组元素的和，元素的个数由length 表示。

```
1  /* WARNING: This is buggy code */
2  float sum_elements(float a[], unsigned length) {
3      int i;
4      float result = 0;
5
6      for (i = 0; i <= length-1; i++)
7          result += a[i];
8      return result;
9  }
```

当length 的参数值设置为0， 出现什么问题？



怎样正确使用 Unsigned?

- C 语言：unsigned 加、减法 是模为Umax 的算术运算

$0 \rightarrow 1 \rightarrow \dots \rightarrow U_{\max} \rightarrow 0$

- 不要轻易使用，否则程序容易出错

```
unsigned i;  
for (i = cnt-2; i >= 0; i--)  
    a[i] += a[i+1];    //cnt: 数组长度
```

- 正确的方法

```
▪ unsigned i;  
for (i = cnt-2; i < cnt; i--)  
    a[i] += a[i+1]; // 即使 cnt = Umax 代码仍然有效
```



什么时候使用unsigned ?



- 需要正整数的模运算系统时
- When Performing Modular Arithmetic
 - C 语言的标准实现：unsigned 加法 类似于取模算术运算
 - $0 \rightarrow 1 \rightarrow \dots \rightarrow U_{max} \rightarrow 0$
- 需要用各个数的各位编码来表示集合时（下一节讨论）
- When Using Bits to Represent Sets
 - 逻辑右移 Logical right shift,
 - 无符号位扩展 no sign extension



小结



- 介绍了无符号数是通过 unsigned 来定义的
- 编程时要注意有符号数和无符号数之间的转换
- C语言支持隐含的强制类型转换，可能会带来意想不到的问题



谢谢！

