

本节内容

# 带符号整数

表示和运算

(原/反/补)

王道考研/CSKAOYAN.COM

1

## 带符号整数在计算机中的应用

带符号整数，即“整数”，-2、-1、0、1、2、3、4...

C 语言中的带符号整数：

```
short a=1;    //带符号整数（短整型，2B）  
int b=-2;     //带符号整数（整型，4B）
```

位数不同，可表示数值范围不同



- 带符号整数，在计算机硬件内，如何表示？
- 带符号整数的加法、减法运算是怎么用硬件实现的？

王道考研/CSKAOYAN.COM

2

### 本节总览

带符号整数的表示

原码

补码

反码

同一个含义，用不同的编码方式表示

我爱你

我是普通人



520

请问520打算怎么过?



别爱我，没结果




王道考研/CSKAOYAN.COM

3

### 带符号整数的表示

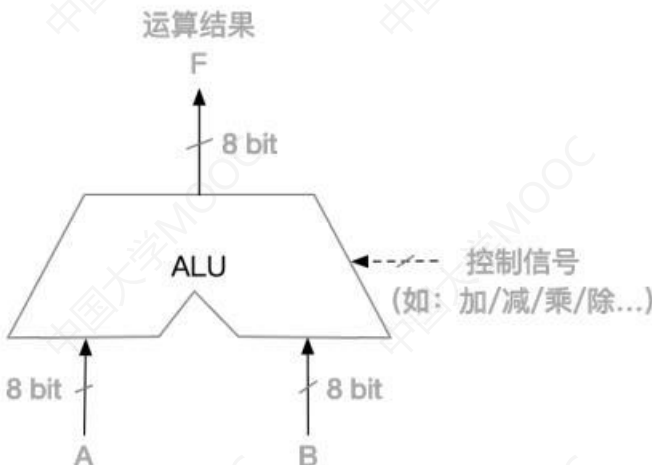
该计算机硬件能支持的带符号整数位数有上限



机器字长8位

通用寄存器只能存8位

**Tips:** 现在的个人计算机机器字长通常是64位，或至少32位



最多只能同时进行8位运算

王道考研/CSKAOYAN.COM

4

## 原码表示



常见书面写法:  $x = -19$   $[x]_{\text{原}} = 1, 0010011$

若未指明机器字长, 也可写为:  $[x]_{\text{原}} = 1, 10011$

- 原码:**
- ① 符号位“0/1”对应“正/负”, 剩余的数值位表示真值的绝对值
  - ② 若机器字长 $n+1$ 位, 带符号整数的原码表示范围:  $-(2^n-1) \leq x \leq 2^n-1$
  - ③ 真值0有两种形式: +0 和 -0,  $[+0]_{\text{原}} = 0, 0000000$ ;  $[-0]_{\text{原}} = 1, 0000000$

王道考研/CSKAOYAN.COM

5

## 原码的缺点



原码的缺点: 符号位不能参与运算, 需要设计复杂的硬件电路才能处理, 费钱! 贵!

用补码表示真值——符号位可以参与运算

王道考研/CSKAOYAN.COM

6



## 原码→反码→补码的转换（机算）



王道考研/CSKAOYAN.COM

7

敲黑板

## 原码、补码快速转换技巧（手算）



王道考研/CSKAOYAN.COM

8

### 补码的加法运算（例1）

符号位  
0正1负

← 符

数值位

← 2<sup>6</sup> 2<sup>5</sup> 2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

补码数值位不能  
解读为“位权”

A: +19 → 补码    

0	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

    8bit寄存器

B: -19 → 补码    

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

    8bit寄存器

↓ 按位相加

A+B=0 → 补码    

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

    8bit寄存器

计算机硬件如何做补码的加法：从最低位开始，按位相加（符号位参与运算），并往更高位进位

王道考研/CSKAOYAN.COM

9

### 补码的加法运算（例2）

A: -19 → 补码    

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

    8bit寄存器

B: -19 → 补码    

1	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

    8bit寄存器

↓ 按位相加

A+B=-38 → 补码    

1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

    8bit寄存器

↓ 负数补码转原码

[A+B]<sub>原</sub> = 1,0100110 → 真值 = -38

计算机硬件如何做补码的加法：从最低位开始，按位相加（符号位参与运算），并往更高位进位

王道考研/CSKAOYAN.COM

10



## 补码的减法运算

$$[A]_{\text{补}} - [B]_{\text{补}} = ?$$

能不能减法变加法？

$$A - B = A + (-B)$$

$$[A]_{\text{补}} - [B]_{\text{补}} = [A]_{\text{补}} + [-B]_{\text{补}}$$

接下来要解决的问题：已知“减数”的补码，如何求其负值的补码表示？



x = 19    [x]<sub>补</sub> = 0,0010011    全部位按位取反 → 1,1101100    末位+1 → 1,1101101

-x = -19    [-x]<sub>补</sub> = 1,1101101    全部位按位取反 → 0,0010010    末位+1 → 0,0010011



是什么蒙蔽了我的双眼



贫穷是理智消费的基础

王道考研/CSKAOYAN.COM

11

## 对比：无符号整数的减法运算

A : 99 → 二进制: 1100011

B : 9 → 二进制: 1001

A : 99 → 二进制: 1100011

A-B=90 → 二进制: 1011010

0 1 1 0 0 0 1 1

8bit寄存器

0 0 0 0 1 0 0 1

8bit寄存器

0 1 1 0 0 0 1 1

8bit寄存器

1 1 1 1 0 1 1 1

“减数”B的变形

减法变加法

1 0 1 0 1 1 0 1 0

8bit寄存器

计算机硬件如何做无符号整数的减法：

- ① “被减数”不变，“减数”全部位按位取反、末位+1，减法变加法
- ② 从最低位开始，按位相加，并往更高位进位



王道考研/CSKAOYAN.COM

12

### 补码的减法运算（例3）

A: +19 → 补码 0 0 0 1 0 0 1 1 8bit寄存器

B: -19 → 补码 1 1 1 0 1 1 0 1 8bit寄存器

A: +19 → 补码 0 0 0 1 0 0 1 1 8bit寄存器

-B: +19 → 补码 0 0 0 1 0 0 1 1 [-B]的补码

↓  
按位相加

A-B=38 → 补码 0 0 1 0 0 1 1 0 8bit寄存器

优点：用同一套电路即可处理所有的加减法，省钱！

“减数”全部位按位取反、末位+1

$[A]_{补} - [B]_{补} = [A]_{补} + [-B]_{补}$

计算机硬件如何做带符号数补码的减法：

- ① “被减数”不变，“减数”全部位按位取反、末位+1，减法变加法
- ② 从最低位开始，按位相加，并往更高位进位

王道考研/CSKAOYAN.COM

13

### 知识回顾与重要考点

**Tips:** 计算机内部，所有带符号整数的加、减法都要先转化为补码

快速转换技巧：从右往左找到第一个1，这个1左边的全部位按位取反

计算机硬件如何做带符号数补码的加法：从最低位开始，按位相加（符号位参与运算），并往更高位进位

计算机硬件如何做带符号数补码的减法：

- ① “被减数”不变，“减数”全部位按位取反、末位+1，减法变加法
- ② 从最低位开始，按位相加，并往更高位进位

王道考研/CSKAOYAN.COM

14





15