

无符号数

无符号数：整个机器字长的全部二进制位均为数值位，没有符号位，相当于数的绝对值。

1001 1100B

=1 × 27 + 0 × 26 + 0 × 25 + 1 × 24 + 1 × 23 + 1 × 22 + 0 × 21 + 0 × 20

=156D

表示范围

8位二进制数： 28种不同的状态

0000 0000 ～ 1111 1111

= 1 0000 0000 - 1

0 ～ 255 =

28

- 1

n位的无符号数表示范围为：0 ～ 2n -1

王道考研/CSKAOYAN.COM



第二章 数据的表示和运算

[WWW.CSKAOYAN.COM](http://WWW.CSKAOYAN.COM/)

王道考研——组成原理

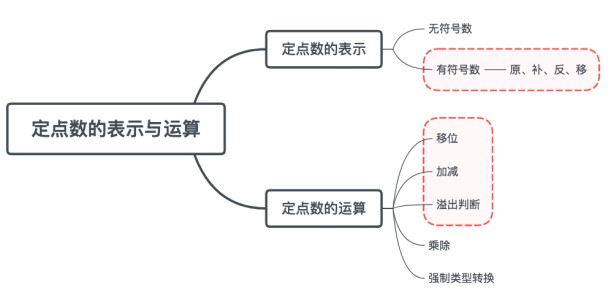


本节内容

定点数的 表示和运算

无符号数原码

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768 | 65536 |



有符号数

+ 156 D = 0 1001 1100B

- 156 D = 1 1001 1100B

真值 机器数

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

-0.75D = 1.11B 存储为111

表示范围

𝐾−1 × 2−1 + 𝐾−2 × 2−2 + … + 𝐾−𝑚 × 2−𝑚

绝对值：0.00 ～ 0.11 = 1.00 – 0.01

0 ～ 1- 2−2

有n位尾数的定点小数：-(1- 2−𝑛) ～ 1- 2−𝑛

+0.75D = 0.11B 存储为011 (未考虑位数扩展)

定点小数

定点表示

+ 156 D = 0 1001 1100B

- 156 D = 1 1001 1100B

真值 机器数

小数点：隐含存储(定点数：事先约定；浮点数：按规则浮动)



王道考研/CSKAOYAN.COM

+3D = 011.B 存储为011 (未考虑位数扩展)

-3D = 111.B 存储为111

表示范围

绝对值：0 ～ 2n -1

有n位尾数的定点整数：-(2n -1 ) ～ 2n -1

定点整数

-0.75D = 1.11B 存储为111

表示范围 -(1- 2−𝑛) ～ 1- 2−𝑛

+0.75D = 0.11B 存储为011 (未考虑位数扩展)

定点小数

定点表示

+ 156 D = 0 1001 1100B

- 156 D = 1 1001 1100B

真值 机器数

小数点：隐含存储(定点数：事先约定；浮点数：按规则浮动)



[*x*1]原=**0**.1101000 ，[*x*2]原=**1**.1101000

计算机中：**0**1101000 ，**1**1101000

若字长为*n*+1，则原码小数的表示范围为−(1−2−*n*) ≤ *x* ≤ 1−2−*n*（关于原点对称）

王道考研/CSKAOYAN.COM

0≥*x*  1

1  *x*  1 | *x* |

原 

*x*  *x*

[*x*1]原=**0**.1101 ，[*x*2]原=0.1101 + 1.0000 =**1**.1101 计算机的做法：“加”

1  *x*≥0

原码

+ 156 D = 0 1001 1100B

- 156 D = 1 1001 1100B

真值 机器数

约定：用*X*表示真值，用[*X*]原表示原码，[*X*]补表示补码，[*X*]反表示反码，[*X*]移表示移码。假设字长为8位(符号位+数值位)，最高位为符号位

纯小数原码

*x*1 = +0.8125 ， *x*2 = −0.8125 真值(十进制形式)

*x*1 = +0.1101 ， *x*2 = −0.1101 真值(二进制形式)

[*x*1]原=**0**.1101 ，[*x*2]原=**1**.1101 我们的做法：+换成0，-换成1





约定：用*X*表示真值，用[*X*]原表示原码，[*X*]补表示补码，[*X*]反表示反码，[*X*]移表示移码。

假设字长为8位(符号位+数值位)，最高位为符号位

纯整数原码

*x*1 = +14 ， *x*2 = −14 真值(十进制形式)

*x*1 = +1110 ， *x*2 = −1110 真值(二进制形式) [*x*1]原=**0**,1110 ，[*x*2]原=**1**,1110 我们的做法：+换成0，-换成1 [*x*1]原=**0**,1110 ，[*x*2]原=0,1110 + 1,0000 =**1**,1110 计算机的做法：“加”

 

*x*

0, *x*

2*n*  *x*≥0

原

 

2*n*  x  2*n*  | *x* | 0≥*x*  2*n*

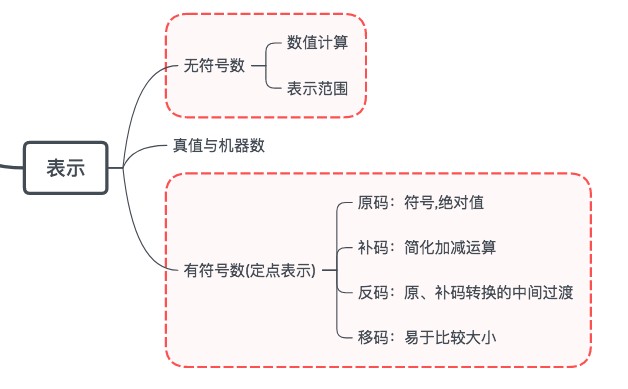
[*x*1]原=**0**,0001110 ，[*x*2]原=**1**, 0001110

计算机中： **0**0001110 ， **1**0001110

若字长为*n*+1，则原码整数的表示范围为−(2*n*−1) ≤ *x* ≤ 2*n*−1（关于原点对称）

王道考研/CSKAOYAN.COM

|  |  |  |
| --- | --- | --- |
|  | 原码 |  |
| 符号、数值分开处理 | + 156 D = 0 1001 1100B | 运算： **0**0001110 + **1**0001110 |
| [+0]原=**0**0000… | - 156 D = 1 1001 1100B | 根据最高位调整成相应的无符号数运算 |
| [−0]原=**1**0000… | 真值 机器数 |  **0**0001110 - **0**0001110 |



知识点回顾

王道考研/CSKAOYAN.COM

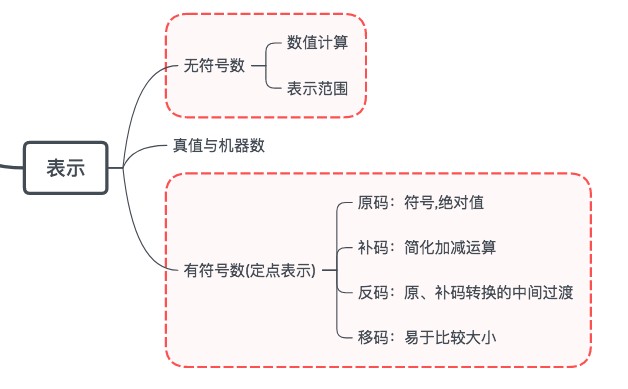


本节内容

定点数的 表示和运算

补码反码移码

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM



加减运算

**0**0001110

- **0**0001110

**0**0000000



王道考研/CSKAOYAN.COM



加减运算





**0**0001110

+ **1**0001110

**0**0001110

- **0**0001110

**0**0000000

**0**0001110

+ **1**1110010

1**0**0000000

王道考研/CSKAOYAN.COM

+ 1,00000000 - **0**0001110



加减运算

**0**0001110

- **0**0001110





正数：与原码相同

补码 负数：符号位与原码相同，数值位由原码取反加1得到

14

242

**0**0001110

+ 1,00000000 - **0**0001110

14

-14

0



+

1**0**0000000

256 mod 1,00000000 = 0



28

0,00000001+ 0,11111111 - **0**0001110 [*X*]原  [*X*]补：正数不变；负数符号位不变，数值位取反加1

加1 取反 [*X*]补  [-*X*]补：连同符号位一起取反加1

王道考研/CSKAOYAN.COM

**0**0001110

**1**1110010

**0**0000000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 有符号数 |  |  | 无符号数 |
| 14 |  | **0**0001110 | 14 |
| -14 | + | **1**0001110 | 142 |
| 0 |  | **1**0011100 | 156 |
|  |  |  |  |
| **0**0001110 |  |  |  |
| + | **1**0001110 |  |  |  |



王道考研/CSKAOYAN.COM

12 = 1 ⋯ ⋯ 8 20 mod 12 = 8

模

10 + 10 = 20 10 + 12 – 2 = 20

20

10 – 2 = 8

补码

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 有符号数 |  | 无符号数 |  |  | 有符号数 |  | 无符号数 |
| 14 | **0**0001110 | 14 |  | **0**0001110 | 14 |  | **0**0001110 14 |
| -14 | + **1**0001110 | 142 | + | **1**0001110 | -14 | + | **1**0001110 142 |
|  |  |  |  |  |  |  |  |
| 0 | **1**0011100 | 156 |  |  | 0 |  | **1**0011100 156 |



王道考研/CSKAOYAN.COM

*x*1=+1010，*x*2=−1010，字长为8位，则其补码表示为：

[*x*1]补 = **0**,0001010

[*x*2]补 = 28 − 0,0001010 = 10,0000000 – 0,0001010 = **1**,1110110 = [-*x*1]补

[*x*2]原 = **1**,0001010

若字长为*n*+1，则补码的表示范围为−2*n* ≤ *x* ≤ 2*n*−1（比原码多表示−2*n*）

[− 2*n*]补 = 10,0000 − 1,0000 = 1,0000

2n+1

( mod )

2*n*  *x*≥0

0≥*x*≥  2*n*

2*n*1  *x*  2*n*1 | *x* |

 

补

0, *x*

*x*

 

纯整数补码

**0**0001110

+ 1,00000000 - **0**0001110

**0**0001110

- **0**0001110

对于正数，补码与原码的表示相同，[*x*]补=[*x*]原。

对于负数，原码符号位不变，数值部分按位取反，末位加1（即所谓“取反加1”） 此规则同样适用于由[*x*]补求[*x*]原。

补码



王道考研/CSKAOYAN.COM

*x*1=+0.1001，*x*2=−0.1001，字长为8位，则其补码表示为：

[*x*1]补=**0**.1001000

[*x*2]补 = 2 − 0.1001000 = 10.0000000 – 0.1001000 = **1**.0111000 = [-*x*1]补

[*x*1]原 = **1**.1001000

若字长为*n*+1，则补码的表示范围为−1≤*x*≤1−2−*n*（比原码多表示−1）

[−1]补=10.0000−1.0000= 1.0000

1  *x*≥0

0  *x*≥ 1( mod 2)

2  *x*  2 | *x* |

补 

*x*  *x*

纯小数补码

**0**0001110

+ 1,00000000 - **0**0001110

**0**0001110

- **0**0001110

对于正数，补码与原码的表示相同，[*x*]补=[*x*]原。

对于负数，原码符号位不变，数值部分按位取反，末位加1（即所谓“取反加1”） 此规则同样适用于由[*x*]补求[*x*]原。

补码



王道考研/CSKAOYAN.COM

*x*1=+1011，*x*2=−1011，字长为8位，则其反码表示为：

[*x*1]反= **0**,0001011

[*x*2]反= 1,1111111− 0,0001011 = **1**,1110100

[*x*1]原 = **1**,0001011

若字长为*n*+1，则反码的表示范围为−(2*n*−1) ≤ x ≤ 2*n*−1（关于原点对称）

2*n*  *x*≥0

n+1

0≥*x*  2*n*( mod 2 -1 )



(2 1)  *x*

0, *x*

 

反 *n* 1

*x*

 

对于正数，反码与原码的表示相同，[*x*]反=[*x*]原。 对于负数，原码符号位不变，数值部分按位取反，

此规则同样适用于由[*x*]反求[*x*]原。

表示范围：与原码一样。

纯整数反码

反码



王道考研/CSKAOYAN.COM

*x*1=+0.0110，*x*2=−0.0110，字长为8位，则其反码表示为：

[*x*1]反=**0**.0110000

[*x*2]反=1.1111111 − 0.0110000 = **1**.1001111

[*x*1]原 = **1**.0110000

若字长为*n*+1，则反码的表示范围为−(1−2−*n*)≤*x*≤1−2−*n*（关于原点对称）

1  *x*≥0

−n

0≥*x*  1( mod 2 − 2 )

)  *x*



(2  2

反 *n*

*x*  *x*

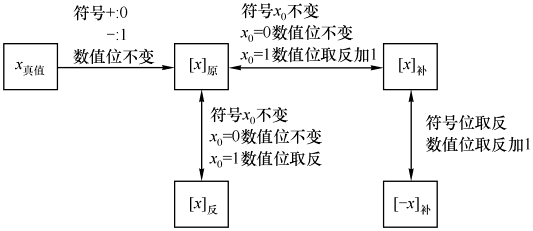
对于正数，反码与原码的表示相同，[*x*]反=[*x*]原。 对于负数，原码符号位不变，数值部分按位取反，

此规则同样适用于由[*x*]反求[*x*]原。

表示范围：与原码一样。

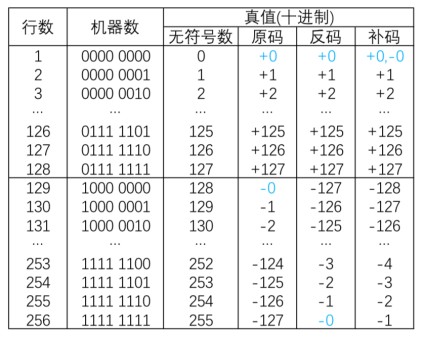
纯小数反码

反码



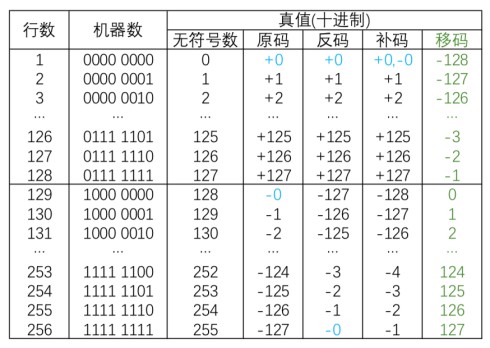
原补反相互转换

王道考研/CSKAOYAN.COM



原补反相互转换

王道考研/CSKAOYAN.COM



移码

移码就是在真值*X*上加上一个常数（偏置值），通常这个常数取2*n* 。

[*x*]移=2*n*+*x*

*x*1=+10101，*x*2=−10101，字

长为8位，则其移码表示为：

[*x*1]移=27+10101

-1 =10000000+10101

0 =1,0010101

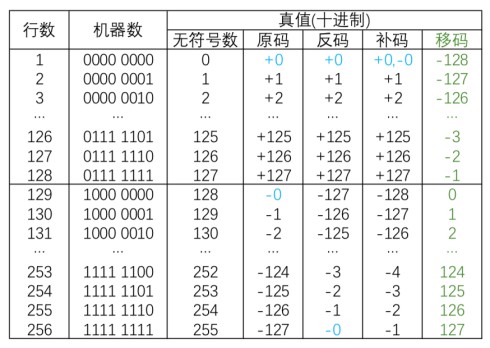
1

[*x*2]移=27+(−10101)

=10000000 +(−10101)

=0,1101011

王道考研/CSKAOYAN.COM



移码

移码0111 1110的真值：

1. 转换成无符号数真值：126

2. 减去偏置值1000 0000对应的无符号数真值128得到移码真值：

126 – 128 = -2

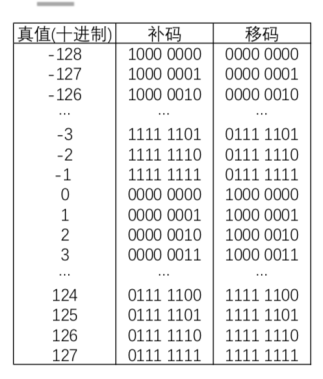
或者：

-1 0111 1110 – 1000 0000

0 = 1111 1110

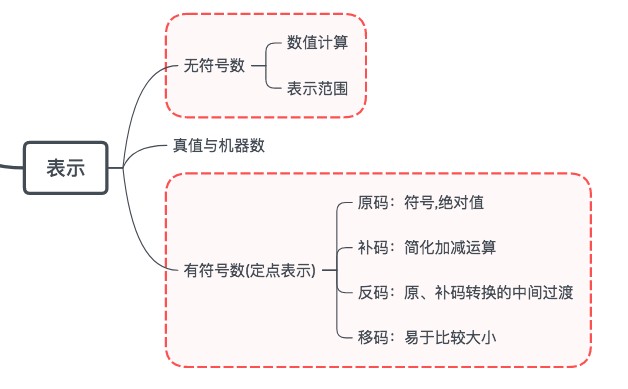
1 对应补码真值-2

王道考研/CSKAOYAN.COM



移码

王道考研/CSKAOYAN.COM



知识点回顾

王道考研/CSKAOYAN.COM

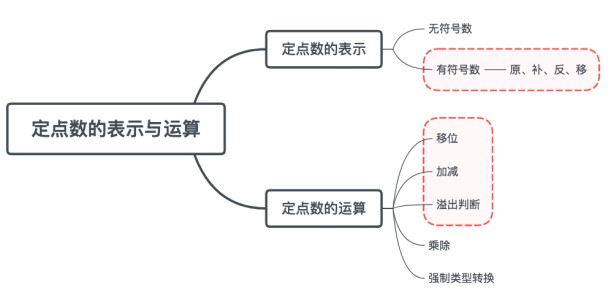


本节内容

定点数的 表示和运算

移位运算

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM



移位运算

r 进制： 𝐾n 𝐾n−1 … 𝐾2 𝐾1 𝐾0𝐾−1𝐾−2 … 𝐾−𝑚

× 𝑟0

10 进制：100.0

小数点左移2位：1.000，相当于除以100，即除以102 小数点右移1位：1000. ，相当于乘以10，即乘以101

= 𝐾n × 𝑟𝑛 + 𝐾n−1 × 𝑟𝑛−1 + ⋯ + 𝐾2 × 𝑟2 + 𝐾1 × 𝑟1 + 𝐾0 +𝐾−1 × 𝑟−1 + 𝐾−2 × 𝑟−2 + … + 𝐾−𝑚 ×

𝑟−𝑚

也可看作固定小数点： 数字右移2位

数字左移1位

右移n位： ÷ 𝑟n 左移n位： × 𝑟n

2 进制： 𝐾7𝐾6𝐾5𝐾4𝐾3𝐾2𝐾1𝐾0

= 𝐾7 × 27 + 𝐾6 × 26 + 𝐾5 × 25 + 𝐾4 × 24+ 𝐾3 × 23 + 𝐾2 × 22 + 𝐾1 × 21 + 𝐾0 × 20

机器数采用无符号数：逻辑移位

逻辑左移时，高位移丢，低位添0；逻辑右移时，低位移丢，高位添0

**1**

王道考研/CSKAOYAN.COM



算术移位

2 进制： S 𝐾6𝐾5𝐾4𝐾3𝐾2𝐾1𝐾0

=(−1)𝑆 × (𝐾6 × 26 + 𝐾5 × 25 + 𝐾4 × 24+ 𝐾3 × 23 + 𝐾2 × 22 + 𝐾1 × 21 + 𝐾0 × 20)

算术移位：机器码采用有符号数

符号位不参与移位原码：

左移、右移都补**0**

1,0110101 真值-53

左移1位(丢0)：1,1101010 真值-106

右移1位(丢1)：1,0011010 真值-26

再左移1位(丢1)：1,1010100 真值-84

再右移1位(丢0)：1,0001101 真值-13

假设不丢1： 1,0011010.1 真值-26.5

假设不丢1： 1, 11010100 真值-212

原码算术移位：左移丢**1**，运算出错；右移丢**1**，影响精度。

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

**1 0 1 1 0 1 0 1**

2 进制： 𝐾7𝐾6𝐾5𝐾4𝐾3𝐾2𝐾1𝐾0

= 𝐾7 × 27 + 𝐾6 × 26 + 𝐾5 × 25 + 𝐾4 × 24+ 𝐾3 × 23 + 𝐾2 × 22 + 𝐾1 × 21 + 𝐾0 × 20

机器数采用无符号数：逻辑移位

逻辑左移时，高位移丢，低位添0；逻辑右移时，低位移丢，高位添0

**1 0 1 1 0 1 0 1**

𝑟−𝑚

也可看作固定小数点： 数字右移2位

数字左移1位

右移n位： ÷ 𝑟n 左移n位： × 𝑟n

= 𝐾n × 𝑟𝑛 + 𝐾n−1 × 𝑟𝑛−1 + ⋯ + 𝐾2 × 𝑟2 + 𝐾1 × 𝑟1 + 𝐾0 +𝐾−1 × 𝑟−1 + 𝐾−2 × 𝑟−2 + … + 𝐾−𝑚 ×

× 𝑟0

10 进制：100.0

小数点左移2位：1.000，相当于除以100，即除以102 小数点右移1位：1000. ，相当于乘以10，即乘以101

r 进制： 𝐾n 𝐾n−1 … 𝐾2 𝐾1 𝐾0𝐾−1𝐾−2 … 𝐾−𝑚

移位运算



王道考研/CSKAOYAN.COM

**1 0 1 1 0 1 0 1**

2 进制： 𝐾7𝐾6𝐾5𝐾4𝐾3𝐾2𝐾1𝐾0

= 𝐾7 × 27 + 𝐾6 × 26 + 𝐾5 × 25 + 𝐾4 × 24+ 𝐾3 × 23 + 𝐾2 × 22 + 𝐾1 × 21 + 𝐾0 × 20

机器数采用无符号数：逻辑移位

逻辑左移时，高位移丢，低位添0；逻辑右移时，低位移丢，高位添0

**1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 1**

𝑟−𝑚

也可看作固定小数点： 数字右移2位

数字左移1位

右移n位： ÷ 𝑟n 左移n位： × 𝑟n

= 𝐾n × 𝑟𝑛 + 𝐾n−1 × 𝑟𝑛−1 + ⋯ + 𝐾2 × 𝑟2 + 𝐾1 × 𝑟1 + 𝐾0 +𝐾−1 × 𝑟−1 + 𝐾−2 × 𝑟−2 + … + 𝐾−𝑚 ×

× 𝑟0

10 进制：100.0

小数点左移2位：1.000，相当于除以100，即除以102 小数点右移1位：1000. ，相当于乘以10，即乘以101

r 进制： 𝐾n 𝐾n−1 … 𝐾2 𝐾1 𝐾0𝐾−1𝐾−2 … 𝐾−𝑚

移位运算

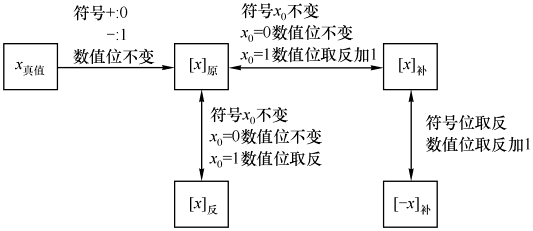
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **0** | **1** | **1** | **0** | **1** | **0** | **1** |

|  |  |
| --- | --- |
| 符号位 | 绝对值 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **0** | **1** | **1** | **0** | **1** | **0** | **1** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **1** | **0** | **1** | **0** | **1** | **0** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** |



王道考研/CSKAOYAN.COM

负数：反码**1 <—>** 原码**0**

原 码 **1,0110101**

反 码 **1,1001010**

补 码 **1,1001011**

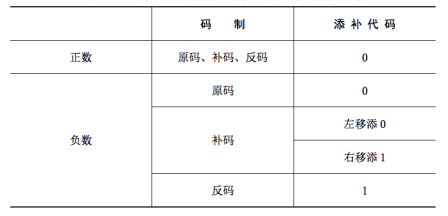
**1 0 1 1 0 1 0 1**

2 进制： S 𝐾6𝐾5𝐾4𝐾3𝐾2𝐾1𝐾0

算术移位：机器码采用有符号数符号位不参与移位

正数：原码、补码、反码一样 **** 左移、右移都补**0**

算术移位



算术移位

**1 0 1 1 0 1 0 1**

2 进制： S 𝐾6𝐾5𝐾4𝐾3𝐾2𝐾1𝐾0

算术移位：机器码采用有符号数符号位不参与移位

正数：原码、补码、反码一样

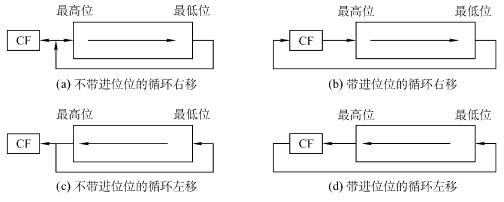
**** 左移、右移都补**0**

负数：反码**1 <—>** 原码**0** 原 码 **1,0110101**

反 码 **1,1001010**

补 码 **1,1001011**

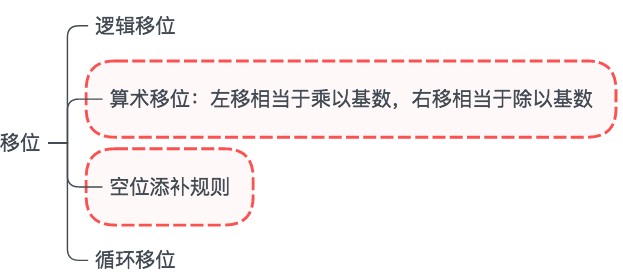
王道考研/CSKAOYAN.COM



循环移位

**1 0 1 1 0 1 0 1**

王道考研/CSKAOYAN.COM



知识点回顾

王道考研/CSKAOYAN.COM

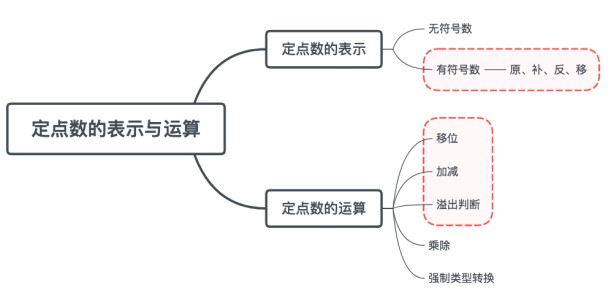


本节内容

定点数的 表示和运算

加减运算符号扩展溢出判断

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM



符号扩展

设机器字长为8位（含1位符号位），*A* = 15，*B* = -24，求[*A*+*B*]补和[*A*−*B*]补

原码

*A* = +1111  0,1111  0,0001111

*B* = -11000  1,11000  1,0011000

补码

*A* = +1111  0,1111  0,0001111

*B* = -11000  1,01000  1,1101000

王道考研/CSKAOYAN.COM



加减运算

设机器字长为8位（含1位符号位），*A* = 15，*B* = -24，求[*A*+*B*]补和[*A*−*B*]补

补码

*A* = +1111  0,1111  0,0001111

*B* = -11000  1,01000  1,1101000

[*A*+*B*]补 = [*A*]补 + [*B*]补 = 0,0001111 + 1,1101000 = 1,1110111

原码：1,0001001 真值-9

[*A-B*]补 = [*A*]补 + [-*B*]补 = 0,0001111 + 0,0011000 = 0,0100111 真值+39

[-*B*]补 ： [*B*]补连同符号位一起取反加1

*C* = 124，求[*A*+*C*]补和[*B*−*C*]补

[*A*+*C*]补 = 0,0001111 + 0,1111100 = 1,0001011 真值-117

[*B*−*C*]补 = 1,1101000 + 1,0000100 =10,1101100 真值+108

王道考研/CSKAOYAN.COM



溢出判断

设机器字长为8位（含1位符号位），*A* = 15，*B* = -24，求[*A*+*B*]补和[*A*−*B*]补

*C* = 124，求[*A*+*C*]补和[*B*−*C*]补

[*A*+*C*]补 = 0,0001111 + 0,1111100 = ,0001011 真值-117

[*B*−*C*]补 = 1,1101000 + 1,0000100 =1 1101100 真值+108

方法一：采用一位符号位

设*A*的符号为*A*S，*B*的符号为*B*S，运算结果的符号为*S*S，则溢出逻辑表达式为

王道考研/CSKAOYAN.COM

0,

1



溢出判断

设机器字长为8位（含1位符号位），*A* = 15，*B* = -24，求[*A*+*B*]补和[*A*−*B*]补

*C* = 124，求[*A*+*C*]补和[*B*−*C*]补

[*A*+*C*]补 = 0,0001111 + 0,1111100 = ,0001011 真值-117

[*B*−*C*]补 = 1,1101000 + 1,0000100 =1 1101100 真值+108

逻辑表达式

与：如ABC，表示A与B与C

仅当A、B、C均为1时，ABC为1

A、B、C中有一个或多个为0，则ABC为0

方法一：采用一位符号位

设*A*的符号为*A*S，*B*的符号为*B*S，运算结果的符号为*S*S，则溢出逻辑表达式为

或：如A+B+C，表示A或B或C

非：如 A ，表示A非若A为1，则 A 为0 若A为0，则A 为1

或

王道考研/CSKAOYAN.COM

0,

1

*A*S为0且*B*S为0且*S*S为1

*A*S为1且*B*S为1且*S*S为0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| -4 -3 -2 -1 0 1 2 3 | *V* =*A*S*B*S *S*S  *A*S *B*S | *S*S | *V* =*A*S*B*S *S*S  *A*S *B*S | *S*S |
| 100 101 110 111 000 001 010 011 | 若*V*=0，表示无溢出；  若*V*=1，表示有溢出。 | 仅当A、B、C均为0时，A+B+C为0  A、B、C中有一个或多个为1，则A+B+C为1 | 若*V*=0，表示无溢出；  若*V*=1，表示有溢出。 |  |



溢出判断

设机器字长为8位（含1位符号位），*A* = 15，*B* = -24，求[*A*+*B*]补和[*A*−*B*]补*C* = 124， 求 [*A*+*C*] 补 和 [*B*−*C*] 补[*A*+*C*]补 = 0,0001111 + 0,1111100 = 1,0001011 真值-117

[*B*−*C*]补 = 1,1101000 + 1,0000100 =10,1101100 真值+108

方法二：采用一位符号位，根据数据位进位情况判断溢出符号位的进位*C*S 最高数位的进位*C*1

上溢 0 1

下溢 1 0

即： *C*S与*C*1不同时有溢出

处理“不同”的逻辑符号：异或 异或逻辑：不同为1，相同为0 溢出逻辑判断表达式为*V*=*C*S*C*1 0  0 = 0

若*V*=0，表示无溢出；*V*=1，表示有溢出。 0  1 = 1

1  0 = 1

1  1 = 0

王道考研/CSKAOYAN.COM



采用双符号位的移位运算：低位符号位参与移位，高位符号位代表真正的符号

王道考研/CSKAOYAN.COM

11,1110111 右移1位：11,1111011

00,0100111 左移1位：00,1001110

00,0100111 左移2位：01,0011100 上溢

上溢

下溢

方法三：采用双符号位

正数符号为00，负数符号为11

[*A*+*C*]补 = 00,0001111 + 00,1111100 = 01,0001011

[*B*−*C*]补 = 11,1101000 + 11,0000100 = 10,1101100

记两个符号位为*S*S1*S*S2 ，则*V*=*S*S1*S*S2

若*V*=0，表示无溢出；若*V*=1，表示有溢出。

[*A*+*B*]补 = 00,0001111 + 11,1101000 = 11,1110111

[*A-B*]补 = 00,0001111 + 00,0011000 = 00,0100111

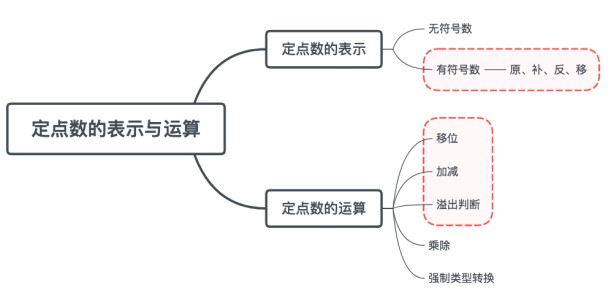
设机器字长为8位（含1位符号位），*A* = 15，*B* = -24，求[*A*+*B*]补和[*A*−*B*]补

*C* = 124，求[*A*+*C*]补和[*B*−*C*]补

[*A*+*C*]补 = 0,0001111 + 0,1111100 = 1,0001011 真值-117

[*B*−*C*]补 = 1,1101000 + 1,0000100 =10,1101100 真值+108

溢出判断



本节总览

王道考研/CSKAOYAN.COM



原码一位乘法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x* = −0.1101，*y* = +0.1011，采用原码一位乘法求*x*·*y*

符号：一正一负，结果为负，即符号位 = *x*s*y*s

原码一位乘法

|*x*|=00.1101，|*y*|=00.1011

0.1101

0.1011

ACC

MQ

00.0000

+|*x|* 00.1101

00.1101

右移 00.0110

+|*x|* 00.1101

01.0011

右移 00.1001

ACC

MQ

1

ACC

MQ

1

ACC

MQ

11

ACC MQ

王道考研/CSKAOYAN.COM

0.10001111

|  |  |
| --- | --- |
| 符号位 | 绝对值 |

|  |  |
| --- | --- |
| 00000 | 1011 |

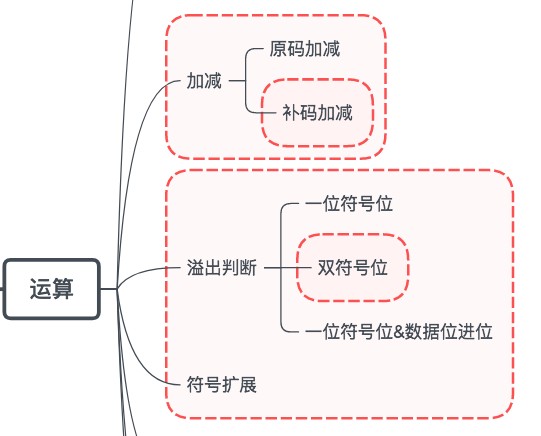
|  |  |
| --- | --- |
| 01101 | 1011 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | 1101 | | |  |
| 1101 | | |  |
|  | 0000 | | |  |
| 1101 | | |  | | |

|  |  |
| --- | --- |
| 00110 | 1101 |

|  |  |
| --- | --- |
| 10011 | 1101 |

|  |  |
| --- | --- |
| 01001 | 1110 |



知识点回顾

王道考研/CSKAOYAN.COM



本节内容

定点数的 表示和运算

乘法运算

王道考研/CSKAOYAN.COM



原码一位乘法

设机器字长为5位（含1位符号位，*n*=4），*x* = −0.1101，*y* = +0.1011，采用原码一位乘法求*x*·*y*

符号：一正一负，结果为负，即符号位 = *x*s*y*s

原码一位乘法

ACC MQ

0.1101

0.1011

符号位*P* =*x* *y* =10=1

s s s

[*x*·*y*]原 = 1.10001111

即*x*·*y*=−0.10001111

|*x*|=00.1101，|*y*|=00.1011

00.0000

+|*x|* 00.1101

00.1101

右移 00.0110

+|*x|* 00.1101

01.0011

右移 00.1001

+0 00.0000

00.1001

右移 00.0100

+|*x|* 00.1101

01.0001

右移 00.1000

1

1

11

11

011

011

1011

王道考研/CSKAOYAN.COM

0.10001111



补码一位乘法

设机器字长为5位（含1位符号位，*n*=4），*x* = −0.1101，*y* = +0.1011，采用Booth算法求*x*·*y*

[*x*]补=11.0011，[−*x*]补=00.1101，[*y*]补=0.1011

ACC MQ

00.0000

+ *[−x*] 补 00.1101

00.1101

右移

+ 0

右移

+ [*x*]补

00.0110

00.0000

00.0110

00.0011

11.0011

11.0110

11.1011

0

0

10

右移

+ *[−x*] 补 00.1101

00.1000

10

110

右 移 00.0100

+ [*x*]补 11.0011

11.0111

110

0110

0110

根据yn+1 - yn判断：

yn+1 - yn = 0，加0，右移一位yn+1 - yn = 1，加[X]补，右移一位yn+1 - yn = -1，加[-X]补，右移一位

[*x*·*y*]补 = 1.01110001

即*x*·*y* = −0.10001111

王道考研/CSKAOYAN.COM

|  |  |
| --- | --- |
| 01101 | 1011 |
| 00110 | 1101 |

|  |  |
| --- | --- |
| 10011 | 1101 |
| 01001 | 1110 |

|  |  |
| --- | --- |
| 01001 | 1110 |
| 00100 | 1111 |

|  |  |
| --- | --- |
| 10001 | 1111 |

|  |  |  |
| --- | --- | --- |
| yn（高位） | yn+1（低位） | 操 作 |
| 0 | 0 | 部分积右移一位 |
| 0 | 1 | 部分积加[X]补  右移一位 |
| 1 | 0 | 部分积加[−X]补  右移一位 |
| 1 | 1 | 部分积右移一位 |

|  |  |
| --- | --- |
| 001101 | 010110 |
| 000110 | 101011 |

|  |  |
| --- | --- |
| 000110 | 101011 |
| 000011 | 010101 |

|  |  |
| --- | --- |
| 110110 | 010101 |
| 111011 | 001010 |

|  |  |
| --- | --- |
| 001000 | 001010 |
| 000100 | 000101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | 1101 | | |  |
|  | | 1101 | | |
|  | 0000 | | |  |
| 1101 | | |  | | |

|  |  |
| --- | --- |
| 00000 | 1011 |

|  |  |
| --- | --- |
| 000000 | 010110 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 01000 | 1111 |  |

|  |  |  |
| --- | --- | --- |
|  | 110111 | 000101 |



乘法运算总结回顾

符 号 位

移 位

乘 法

类 型

参 与运 算

部 分积

累 加

次 数

乘 数

方 向 次 数 每 位

次 数

原码一位乘法

补码Booth 乘法

否

2位

0位

n

右

n

1

是

2位

1位

n+1

右

n

1

王道考研/CSKAOYAN.COM



本节内容

定点数的 表示和运算

除法运算

王道考研/CSKAOYAN.COM



除法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x*=0.1011，*y*=0.1101，求*x*/*y*

0.1101

0.1101 0.10110

0.01101

0.010010

0.001101

0.00010100

0.00001101

0.000001110

*x*/*y*结果为0.1101，余数为0.00000111

王道考研/CSKAOYAN.COM



原码恢复余数法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x*=0.1011，*y*=0.1101，采用原码加减交替除法求*x*/*y*

|*x*|=00.1011，|*y*|=00.1101，[|*y*|]补=00.1101，[−|*y*|]补=11.0011

+[−|*y*|]补

被除数

00.1011

11.0011

11.1110

00.1101

00.1011

01.0110

11.0011

00.1001

01.0010

11.0011

00.0101

…

商

0

+[|*y*|]补

左移

+[−|*y*|]补

01

左移

+[−|*y*|]补

011

…

左移n次，上商n+1次

王道考研/CSKAOYAN.COM



原码恢复余数法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x*=0.1011，*y*=0.1101，采用原码加减交替除法求*x*/*y*

|*x*|=00.1011，|*y*|=00.1101，[|*y*|]补=00.1101，[−|*y*|]补=11.0011

被除数

00.1011

+[−|*y*|]补 11.0011

商

a b

a+b

0

00.1001

左移 01.0010

+[−|*y*|]补 11.0011

00.0101

…

(a+b)×2 = 2a + 2b

(a+b)×2 - b = 2a + 2b – b = 2a + b

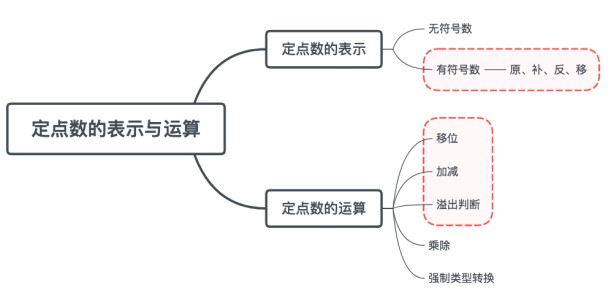
01

011

…

左移n次，上商n+1次

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM

|  |  |
| --- | --- |
| 符号位 | 绝对值 |

|  |  |
| --- | --- |
| 符号位 | 绝对值 |

|  |  |
| --- | --- |
| 符号位 | 绝对值 |

|  |  |  |
| --- | --- | --- |
|  | 11.1110 | |
| +[|*y*|]补 | 00.1101 |  |
| 左移  +[−|*y*|]补 | 00.1011 |  |
| 01.0110 |  |
| 11.0011 |  |



本节内容

定点数的 表示和运算

原码加减交替法补码加减交替法

王道考研/CSKAOYAN.COM



原码恢复余数法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x*=0.1011，*y*=0.1101，采用原码加减交替除法求*x*/*y*

|*x*|=00.1011，|*y*|=00.1101，[|*y*|]补=00.1101，[−|*y*|]补=11.0011

被除数

00.1011

+[−|*y*|]补 11.0011

a左移，加b

不恢复余数法：

被除数减去除数，即

|*x*|+ [−|*y*|]补，

若结果为正，商1，左移，再减去除数；

若结果为负，商0，左移，再加上除数。

2a + b

00.1001

左移 01.0010

+[−|*y*|]补 11.0011

00.0101

…

…

左移n次，上商n+1次

王道考研/CSKAOYAN.COM



原码不恢复余数法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x*=0.1011，*y*=0.1101，采用原码加减交替除法求*x*/*y*

|*x*|=00.1011，|*y*|=00.1101，[|*y*|]补=00.1101，[−|*y*|]补=11.0011

王道考研/CSKAOYAN.COM



原码不恢复余数法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，*n*=4），*x*=0.1011，*y*=0.1101，采用原码加减交替除法求*x*/*y*

|*x*|=00.1011，|*y*|=00.1101，[|*y*|]补=00.1101，[−|*y*|]补=11.0011

被除数

00.1011

ACC

MQ

*Q*s=*x*s*y*s=00=0 得 *x*/*y*=+0.1101 余0.0111×2−4

+[−|*y*|]

若余数为负， 需+[|*y*|]补

得到正确余数

00.0111

王道考研/CSKAOYAN.COM

|  |  |
| --- | --- |
| 符号位 | 绝对值 |

|  |  |  |
| --- | --- | --- |
|  | 11.1110 | a |
| +[|*y*|]补 | 00.1101 | b  a+b  (a+b)×2 = 2a + 2b  (a+b)×2 - b = 2a + 2b – b = |
| 左移  +[−|*y*|]补 | 00.1011 |
| 01.0110 |
| 11.0011 |

|  |  |
| --- | --- |
| 01011 | 00000 |
| 11110 | 00000 |
| 11100 | 00000 |

|  |  |
| --- | --- |
| 01001 | 00001 |
| 10010 | 00010 |

|  |  |
| --- | --- |
| 00101 | 00011 |
| 01010 | 00110 |

|  |  |
| --- | --- |
| 01011 | 00000 |
| 11110 | 00000 |
| 11100 | 00000 |

|  |  |
| --- | --- |
| 01001 | 00001 |
| 10010 | 00010 |

|  |  |
| --- | --- |
| 00101 | 00011 |
| 01010 | 00110 |
| 11101 | 00110 |
| 11010 | 01100 |

|  |  |
| --- | --- |
| 00111 | 01101 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 被除数 | ACC | MQ | 补 11.0011 |
| 00.1011 |  |  | 11.1110 |
| +[−|*y*|]补 | 11.0011 |  |  | 左移 11.1100 |
|  |  |  |  |  |
|  | 11.1110 |  |  | +[|*y*|]补 00.1101 |
| 左移 | 11.1100 |  |  | 00.1001 |
| +[|*y*|]补 | 00.1101 |  |  | 左移 01.0010 |
|  | 00.1001 |  |  | +[-|*y*|]补 11.0011 |
| 左移 | 01.0010 |  |  | 00.0101 |
| +[-|*y*|]补 | 11.0011 |  |  | 左移 00.1010 |
|  | 00.0101 |  |  | +[-|*y*|]补 11.0011 |
| 左移 | 00.1010 |  |  | 11.1101 |
| +[-|*y*|]补 | 11.0011 |  |  | 左移 11.1010 |
|  |  |  |  | +[|*y*|]补 00.1101 |

|  |  |
| --- | --- |
| 符号位 | 绝对值 |



补码加减交替法

设机器字长为5位（含1位符号位，*n*=4），*x*=+0.1000，*y*=-0.1011，采用补码加减交替除法求*x*/*y*

[*x*]补=00.1000，[*y*]补=11.0101，[−*y*]补=00.1011

被除数

00.1000

[*x*/*y*]补=1.0101，余0.0111×2−4

ACC

MQ

+[*y*]补

左移

+[-*y*]补

左移

+[*y*]补

左移

+[-*y*]补

被除数和除数同号， 则被除数减去除数； 异号则被除数加上 除数。

余数和除数同号， 商1，余数左移一位减去除数；

余数和除数异号， 商0，余数左移一位加上除数。

重复n次。

左移

+[*y*]补

11.0101

11.1101

11.1010

00.1011

00.0101

00.1010

11.0101

11.1111

11.1110

00.1011

00.1001

01.0010

11.0101

00.0111

末位恒置1

王道考研/CSKAOYAN.COM



除法运算总结回顾

王道考研/CSKAOYAN.COM

|  |  |
| --- | --- |
| 01000 | 00000 |
| 11101 | 00001 |
| 11010 | 00010 |

|  |  |
| --- | --- |
| 00101 | 00010 |
| 01010 | 00100 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 符号位参与运算 |  | 移 位 | |  |
| 除 法 类 型 | 加 减 次 数 |  |  | 说 明 |
|  |  | 方 向 | 次 数 |  |
| 原码加减交  替法 | 否 | N+1或N+2 | 左 | N | 若最终余数为负，  需恢复余数 |
| 补码加减交  替法 | 是 | N+1 | 左 | N | 商末位恒置1 |

|  |  |
| --- | --- |
| 11111 | 00101 |
| 11110 | 01010 |

|  |  |
| --- | --- |
| 01001 | 01010 |
| 10010 | 10100 |

|  |  |
| --- | --- |
| 00111 | 10101 |



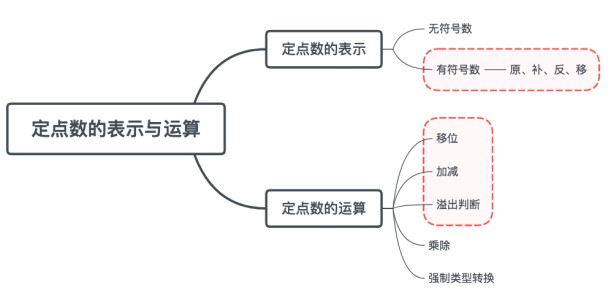
本节内容

定点数的

表示和运算

强制类型转换

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM



强制类型转换

void main(){

x：1110 1111 0001 1111 y：1110 1111 0001 1111 真值61215

无符号数与有符号数：

不改变数据内容， 改变解释方式。

长整数变短整数：

高位截断，保留低位。

a：0x000286a1 c：0x86a1 真值-31071

b：0xffff7751

d：0x7751 真值30545

} 短整数变长整数： x：1110 1111 0001 1111

符号扩展。 0xef1f

m：1111 1111 1111 1111 1110 1111 0001 1111

0xffffef1f 真值-4321

n：1110 1111 0001 1111 0xef1f 真值61215

p：0000 0000 0000 0000 1110 1111 0001 1111

0x0000ef1f 真值61215

王道考研/CSKAOYAN.COM

|  |
| --- |
| short x=-4321; //short型占用2个字节unsigned short y=(unsigned short)x; |
| int a=165537, b=-34991; //int型占用4个字节short c=(short)a, d=(short)b; //short型占用2个字节 |
| short x=-4321; int m=x;  unsigned short n=(unsigned short)x;  unsigned int p=n; |

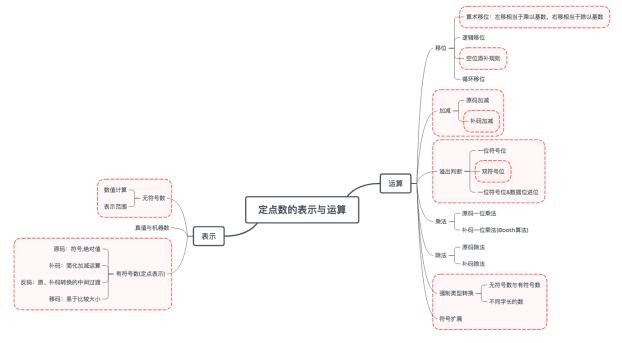


本节内容

定点数的 表示和运算

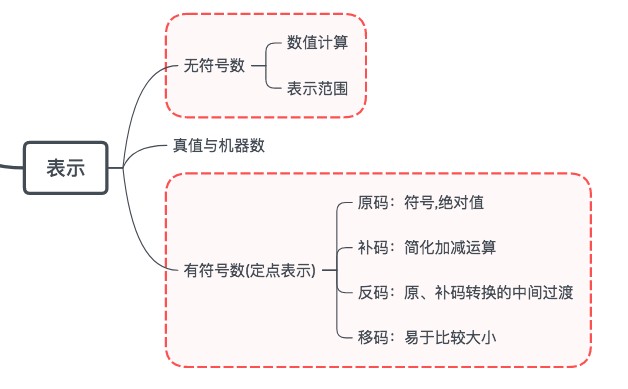
本节回顾

王道考研/CSKAOYAN.COM



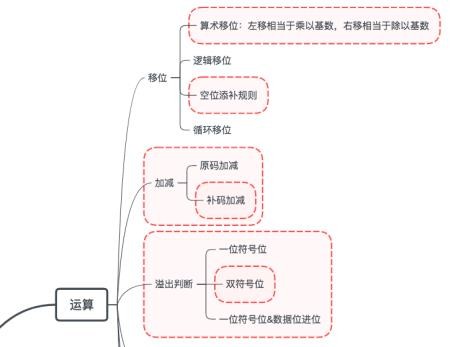
本节回顾

王道考研/CSKAOYAN.COM



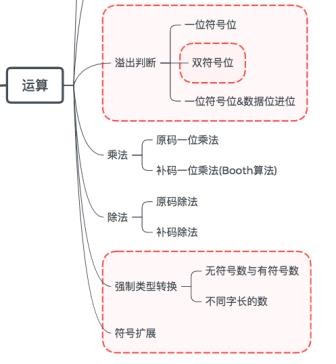
本节回顾

王道考研/CSKAOYAN.COM



本节回顾

王道考研/CSKAOYAN.COM



本节回顾

王道考研/CSKAOYAN.COM