# 



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

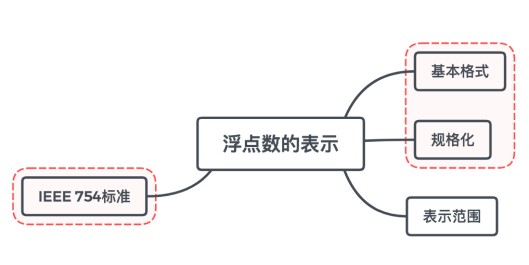


本节内容

浮点数的 表示与运算

表示

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

阶码E反映浮点数的表示范围及小数点的实际位置； 尾数M的数值部分的位数n反映浮点数的精度。

十进制：299792458m/s = 2.998× 108m/s

浮点数的真值： *N* =*r E*  *M*

阶码的底，通常为2

阶码：常用补码或移码表示

尾数：常用原码或补码表示

尾数

𝑟−𝑚

浮点数： 阶码

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

× 𝑟0

定点数：如纯小数0.1011和纯整数11110

r 进制： 𝐾n 𝐾n−1 … 𝐾2 𝐾1 𝐾0𝐾−1𝐾−2 … 𝐾−𝑚

浮点数的表示



浮点数的表示

r 进制： 𝐾n 𝐾n−1 … 𝐾2 𝐾1 𝐾0𝐾−1𝐾−2 … 𝐾−𝑚

× 𝑟0

定点数：如纯小数0.1011和纯整数11110

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

𝑟−𝑚

浮点数： 阶码

尾数

浮点数的真值： *N* =*r E*  *M*

阶码的底，通常为2

阶码E反映浮点数的表示范围及小数点的实际位置； 尾数M的数值部分的位数n反映浮点数的精度。

阶码：常用补码或移码表示

尾数：常用原码或补码表示

例：阶码、尾数均用补码表示，求a、b的真值

a = 0,01;1.1001

b = 0,01;0.01001

a： 阶码0,01对应真值+1

尾数1.1001对应真值-0.0111 = - (2 + 2 + 2 )

−2 −3 −4

或者理解为-111右移4位：- 7 = - 7

24

16

所以a = 21 × (−0.0111) = 21 × (- 7 ) = - 7

16 8

1B的存储空间

王道考研/CSKAOYAN.COM



浮点数的表示

r 进制： 𝐾n 𝐾n−1 … 𝐾2 𝐾1 𝐾0𝐾−1𝐾−2 … 𝐾−𝑚

× 𝑟0

定点数：如纯小数0.1011和纯整数11110

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

𝑟−𝑚

浮点数： 阶码

尾数

浮点数的真值： *N* =*r E*  *M*

阶码的底，通常为2

阶码E反映浮点数的表示范围及小数点的实际位置； 尾数M的数值部分的位数n反映浮点数的精度。

阶码：常用补码或移码表示

尾数：常用原码或补码表示

例：阶码、尾数均用补码表示，求a、b的真值

a = 0,01;1.1001

b = 0,01;0.01001

b： 阶码0,01对应真值+1

尾数0.01001对应真值+0.01001 = + (2 + 2 )

−2 −5

或者理解为+1001右移5位：+ 9 = + 9

25

32

所以b = 21 × (+0.01001) = 21 × 9 = 9

32 16

1B的存储空间

王道考研/CSKAOYAN.COM



浮点数的规格化

浮点数： 阶码

尾数

浮点数的真值： *N* =*r E*  *M*

阶码的底，通常为2

阶码E反映浮点数的表示范围及小数点的实际位置； 尾数M的数值部分的位数n反映浮点数的精度。

阶码：常用补码或移码表示

尾数：常用原码或补码表示

例：阶码、尾数均用补码表示，求a、b的真值

a = 0,01;1.1001

b = 0,01;0.01001

b： 阶码0,01对应真值+1

尾数0.01001对应真值+0.01001 = + (2 + 2 )

或者理解为+1001右移5位：+ 25 = + 32

−2 −5

9 9

所以b = 21 × (+0.01001) = 21 × 9 = 9

32 16

1B的存储空间

b = 21 × (+0.01001)

= 22 × (+0.10010)

王道考研/CSKAOYAN.COM



浮点数的规格化

规格化：规定尾数的最高数位必须是一个有效值 。

左规：当浮点数运算的结果为非规格化时要进行规格化处理，

将尾数左移一位，阶码减1（基数为2时）。

右规：当浮点数运算的结果尾数出现溢出（双符号位为01或10）时， 将尾数右移一位，阶码加1（基数为2时）。

例：a = 010;00.1100，b = 010;00.1000，求a+b a = 22 × 00.1100 ，b = 22 × 00.1000

a+b = 22 × 00.1100 + 22 × 00.1000

= 22 ×(00.1100 + 00.1000)

= 22 × 01.0100

= 23 × 00.1010

规格化浮点数的尾数*M*的绝对值应满足：1/r≤|*M*|≤1

如果*r*=2，则有1/2≤|*M*|≤1

王道考研/CSKAOYAN.COM

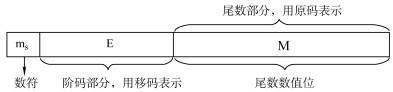
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |

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

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

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

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



**IEEE 754**标准

隐藏表示最高位1

xx…x

表示尾数1.xx…x

1000 0001 1100 1010 0101 0000 1000 0000

1000 0001 1100 1010 0101 0000 1000 0000

0000 0000 0001 1111 0000 0000 0000 0000

规格化的短浮点数的真值为：(−1)*s*×1.*M*×2*E*−127 规格化长浮点数的真值为：(−1)*s*×1.*M*×2*E*−1023

王道考研/CSKAOYAN.COM

# 



规格化浮点数的特点

规格化浮点数的尾数*M*的绝对值应满足：1/r≤|*M*|≤1 如果*r*=2，则有1/2≤|*M*|≤1

1. 原码规格化后：

正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。尾数的表示范围为1/2≤*M*≤(1−2−*n*)。

负数为1.1××…×的形式，其最大值表示为1.10…0；最小值表示为1.11…1。尾数的表示范围为−(1−2−*n*)≤*M*≤−1/2。

1. 补码规格化后：

正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。尾数的表示范围为1/2≤*M*≤(1−2−*n*)。

负数为1.0××…×的形式，其最大值表示为1.01…1；最小值表示为1.00…0。尾数的表示范围为−1≤*M*≤−(1/2+2−*n*)。

-4 -3 -2 -1 0 1 2 3

100 101 110 111 000 001 010 011

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

1.00 1.01 1.10 1.11 0.00 0.01 0.10 0.11

1/4 2/4 3/4

-4/4 -3/4 -2/4 -1/4 0

规格化浮点数的特点

规格化浮点数的尾数*M*的绝对值应满足：1/r≤|*M*|≤1 如果*r*=2，则有1/2≤|*M*|≤1

1. 原码规格化后：

正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。尾数的表示范围为1/2≤*M*≤(1−2−*n*)。

负数为1.1××…×的形式，其最大值表示为1.10…0；最小值表示为1.11…1。尾数的表示范围为−(1−2−*n*)≤*M*≤−1/2。

1. 补码规格化后：

正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。尾数的表示范围为1/2≤*M*≤(1−2−*n*)。

负数为1.0××…×的形式，其最大值表示为1.01…1；最小值表示为1.00…0。尾数的表示范围为−1≤*M*≤−(1/2+2−*n*)。



浮点数的溢出

规格化浮点数的尾数*M*的绝对值应满足：1/r≤|*M*|≤1 如果*r*=2，则有1/2≤|*M*|≤1

1. 原码规格化后：

正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。尾数的表示范围为1/2≤*M*≤(1−2−*n*)。

负数为1.1××…×的形式，其最大值表示为1.10…0；最小值表示为1.11…1。

尾数的表示范围为−(1−2−*n*)≤*M*≤−1/2。

1. 补码规格化后：

正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。尾数的表示范围为1/2≤*M*≤(1−2−*n*)。

负数为1.0××…×的形式，其最大值表示为1.01…1；最小值表示为1.00…0。尾数的表示范围为−1≤*M*≤−(1/2+2−*n*)。

当作机器0

王道考研/CSKAOYAN.COM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 偏 置 值 | |
| 类 型 | 数 符 | 阶 码 | 尾 数 数 值 | 总 位 数 | 十 六 进  制 |  |
|  |  |  |  |  | 十 进 制 |
| 短浮点数 | 1 | 8 | 23 | 32 | 7FH | 127 |
| 长浮点数 | 1 | 11 | 52 | 64 | 3FFH | 1023 |
| 临时浮点数 | 1 | 15 | 64 | 80 | 3FFFH | 16383 |



**IEEE 754**标准

一些规定(短浮点数为例)：

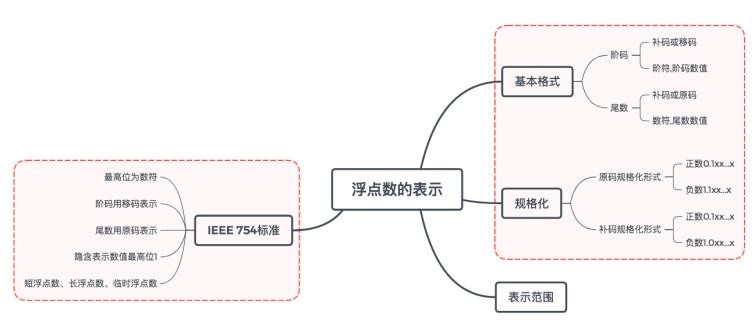
1. E=0且M=0，则真值为0

2. E=0且M≠0，为非规格化数，真值 = (−1)*s*×0.*M*×2−126

3. 1≤E≤254时，真值 = (−1)*s*×1.*M*×2*E*−127

1. E=255且M≠0时，真值为‘NaN’(非数值)
2. E=255且M=0时，真值为正无穷或负无穷(看符号位)

王道考研/CSKAOYAN.COM



本节回顾

王道考研/CSKAOYAN.COM

|  |  |  |
| --- | --- | --- |
| 格 式 | 最 小 值 | 最 大 值 |
| 单精度 | E=1，M=0：1.0×21−127=2−126 | E=254，M=.11…1：1.11…1×2254−127=2127×(2−2−23) |
| 双精度 | E=1，M=0：1.0×21−1023=2−1022 | E=2046，M=.11…1：1.11…1×22046−1023=21023×(2−2−52) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 偏 置 值 | |
| 类 型 | 数 符 | 阶 码 | 尾 数 数 值 | 总 位 数 | 十 六 进  制 |  |
|  |  |  |  |  | 十 进 制 |
| 短浮点数 | 1 | 8 | 23 | 32 | 7FH | 127 |
| 长浮点数 | 1 | 11 | 52 | 64 | 3FFH | 1023 |
| 临时浮点数 | 1 | 15 | 64 | 80 | 3FFFH | 16383 |

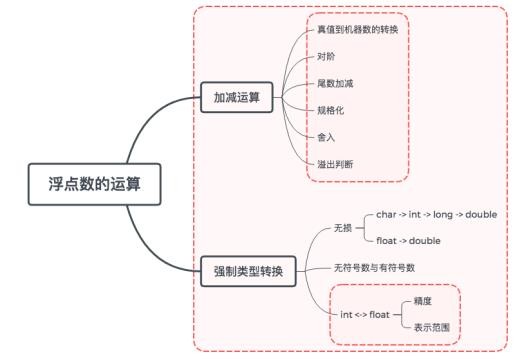


本节内容

浮点数的 表示与运算

加减运算 强制类型转换

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM



浮点数的加减运算

浮点数加减运算步骤：

1. 对阶
2. 尾数加减
3. 规格化
4. 舍入
5. 判溢出

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

59D = 111011B，1/1024 = 2−10  Y = + 111011 × 2−10 = + 0.111011 × 2−4 = + 0.111011 × 2−100

X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：

1. 对阶
2. 尾数加减
3. 规格化
4. 舍入
5. 判溢出

补码：1.011 补码：1011

5D = 101B，1/256 = 2−8  X = - 101 × 2−8 = - 0.101 × 2−5 = - 0.101 × 2−101

浮点数的加减运算

例：已知十进制数*X*=−5/256、*Y*=+59/1024，按机器补码浮点运算规则计算*X*−*Y*，结果

用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

扩展：11.011000000

用补码表示阶码和尾数 双符号位补码：11.011 双符号位补码：11011

0. 转换格式



王道考研/CSKAOYAN.COM

② 对阶：X：11011,11.011000000  11100,11. 101100000 X = - 0.0101 × 2−100

1. 尾数加减
2. 规格化
3. 舍入
4. 判溢出

浮点数的加减运算

例：已知十进制数*X*=−5/256、*Y*=+59/1024，按机器补码浮点运算规则计算*X*−*Y*，结果用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

用补码表示阶码和尾数

1. 转换格式

5D = 101B，1/256 = 2−8  X = - 101 × 2−8 = - 0.101 × 2−5 = - 0.101 × 2−101

59D = 111011B，1/1024 = 2−10  Y = + 111011 × 2−10 = + 0.111011 × 2−4 = + 0.111011 × 2−100

X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：

1. 对阶 使两个数的阶码相等，小阶向大阶看齐，尾数毎右移一位，阶码加1

① 求阶差：[Δ*E*]补=11011+00100=11111，知Δ*E*=−1



王道考研/CSKAOYAN.COM

11.101100000 X-Y

+ 11.000101000 = (-0.0101 × 2−100 )- (+0.111011 × 2−100)

10.110001000 = (-0.0101-0.111011) × 2−100

= -1.001111 × 2−100

2. 尾数加减 -Y：11100,11.000101000

X-Y：11100, 10.110001000

1. 规格化
2. 舍入
3. 判溢出

② 对阶：X：11011,11.011000000  11100,11. 101100000 X = - 0.0101 × 2−100

浮点数的加减运算

例：已知十进制数*X*=−5/256、*Y*=+59/1024，按机器补码浮点运算规则计算*X*−*Y*，结果用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

用补码表示阶码和尾数

1. 转换格式

5D = 101B，1/256 = 2−8  X = - 101 × 2−8 = - 0.101 × 2−5 = - 0.101 × 2−101

59D = 111011B，1/1024 = 2−10  Y = + 111011 × 2−10 = + 0.111011 × 2−4 = + 0.111011 × 2−100

X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：

1. 对阶 使两个数的阶码相等，小阶向大阶看齐，尾数毎右移一位，阶码加1

① 求阶差：[Δ*E*]补=11011+00100=11111，知Δ*E*=−1



强制类型转换

char  int  long  double float  double

范围、精度从小到大，转换过程没有损失

32位

int  float：可能损失精度

float  int：可能溢出及损失精度

int：表示整数，范围 -231 ～ 231-1 ，有效数字32位

float：表示整数及小数，范围 ±[2-126 ～ 2127×(2−2−23)]，有效数字23+1=24位

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

5. 判溢出 常阶码，无溢出，结果真值为2−3×(−0.1001111)2

= -1.001111 × 2−100

= -0.1001111 × 2−011

X-Y：11100, 10.110001000  11101,11.011000100

4. 舍入 无舍入

11.101100000 X-Y

+ 11.000101000 = (-0.0101 × 2−100 )- (+0.111011 × 2−100)

10.110001000 = (-0.0101-0.111011) × 2−100

2. 尾数加减 -Y：11100,11.000101000

X-Y：11100, 10.110001000

3. 规格化

② 对阶：X：11011,11.011000000  11100,11.101100000 X = - 0.0101 × 2−100

浮点数的加减运算

例：已知十进制数*X*=−5/256、*Y*=+59/1024，按机器补码浮点运算规则计算*X*−*Y*，结果

用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

用补码表示阶码和尾数

1. 转换格式

5D = 101B，1/256 = 2−8  X = - 101 × 2−8 = - 0.101 × 2−5 = - 0.101 × 2−101

59D = 111011B，1/1024 = 2−10  Y = + 111011 × 2−10 = + 0.111011 × 2−4 = + 0.111011 × 2−100

X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：

1. 对阶 使两个数的阶码相等，小阶向大阶看齐，尾数毎右移一位，阶码加1

① 求阶差：[Δ*E*]补=11011+00100=11111，知Δ*E*=−1



浮点数的加减运算**-**舍入

“**0**”舍“**1**”入法：类似于十进制数运算中的“四舍五入”法，即在尾数右移时，被移去的最高数值位为0，则舍去；被移去的最高数值位为1， 则在尾数的末位加1。这样做可能会使尾数又溢出，此时需再做一次右规。

恒置“**1**”法：尾数右移时，不论丢掉的最高数值位是“1”还是“0”，都使右移后的尾数末位恒置“1”。这种方法同样有使尾数变大和变小的两 种可能。

浮点数加减运算步骤：

1. 对 阶

2. 尾数加减 如：加减结果为11100,10.110001011

3. 规格化 0舍1入：11100,10.110001011  11101,11.011000101 1

 11101,11.011000110 1

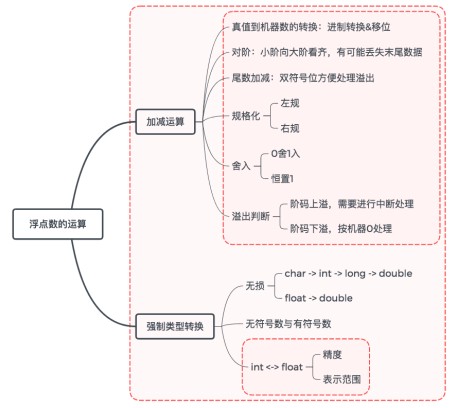
恒置1:11100,10.110001011  11101,11.011000101 1

 11101,11.011000101 1

1. 舍入
2. 判溢出

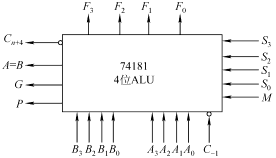
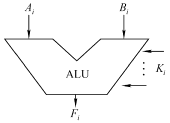
王道考研/CSKAOYAN.COM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 类型 | **16**位机器 | **32**位机器 | **64**位机器 | |
|  | char | 8 | 8 | 8 |  |
|  | short | 16 | 16 | 16 |  |
|  | int | 16 | 32 | 32 |  |
|  | long | 32 | 32 | 64 |  |
| long long | | 64 | 64 | 64 |  |
|  | float | 16 | 32 | 32 |  |
|  | double | 64 | 64 | 64 |  |



本节回顾

王道考研/CSKAOYAN.COM



算术逻辑单元

运算器

算术运算：加、减、乘、除等

Arithmetic and Logic Unit 逻辑运算：与、或、非、异或等

辅助功能：移位、求补等

输入信号(操作数)

实例：

如：M=1、 *S*3～*S*0=1001时， 做逻辑运算AB

控制信号

(指令译码产生)

输出信号(运算结果)

王道考研/CSKAOYAN.COM

PSW



逻辑符号

与

或

非

表达式

Y = A・B

电路逻辑

0 1

A B

Y

Y = A + B

A B

Y = A

Y

A

Y

真值表

**A B Y**

0 0 0

0 1 0

1 0 0

1 1 1

**A B Y**

0 0 0

0 1 1

1 0 1

1 1 1

电路符号

与门的天然逻辑： “屏蔽”

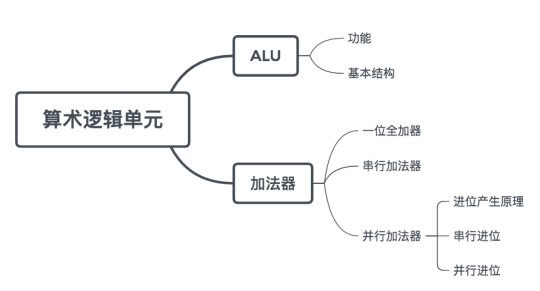
王道考研/CSKAOYAN.COM



本节内容

算术逻辑单元

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM

|  |
| --- |
| MQ |
| ACC |
| ALU |
| X |

|  |  |
| --- | --- |
| **A** | **Y** |
| 0 | 0 |
| 1 | 1 |



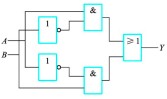
**B** 𝑺i

0 0

1 1

0 1

1 0



复合逻辑

反演律：

A + B = A・B A・B = A + B

表达式

与非 现起来更方便 或非

半导体电路实

异或

Y = A・B

A + B

Y = A + B

A・B

Y = A⨁B

A和B不同

—> A=0且B=1或A=1且B=0

1 0 1

1 1 0

1 0 0

1 1 0

1 0 1

1 1 0

—> A・B + A・B

电路符号

与门的天然逻辑“屏蔽”

异或的天然逻辑“加法”

“奇偶”

王道考研/CSKAOYAN.COM



复合逻辑

反演律：

A + B = A・B

A・B = A + B

表达式

与非 现起来更方便 或非

半导体电路实

异或

Y = A・B

A + B

真值表

0 1

**A**

0

0

1

1

Y = A + B

A・B

**B**

0

1

0

1

Y = A⨁B

**Y**

1

0

0

0

二进制加法

0 + 0 =

0 + 1 =

1 + 0 =

1 + 1 =

电路符号

与门的天然逻辑“屏蔽”

异或的天然逻辑“加法”

“奇偶”

王道考研/CSKAOYAN.COM



复合逻辑

反演律：

A + B = A・B

A・B = A + B

表达式

与非 现起来更方便 或非

半导体电路实

异或

同或

Y = A・B

A + B

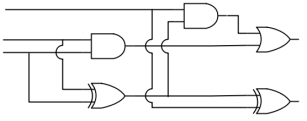
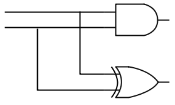
Y = A + B

A・B

Y = A⨁B

Y = A⨀B

王道考研/CSKAOYAN.COM



组合逻辑电路设计**-**一位全加器

100110...0110

**A B** 𝑪i

**A** A

𝑆𝑖

𝐶

𝑖

𝐶𝑖

𝑆𝑖

𝐶𝑖 = 𝐺𝑖 + 𝑃𝑖 𝐶𝑖−1

王道考研/CSKAOYAN.COM



𝑃

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| + 101100…1010 | 0 | 0 | 0 | 0 B | 𝑖 |  | |
| \*\*\*\*\*\*…0000 | 0 | 1 | 0 | 0 | 𝐶𝑖 | FA | 𝐶𝑖−1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **Y** |  | **A B** | **Y** |  | **A** | **B Y** |
| 真值表 | 0 | 0 | 1 |  | 0 0 | 1 |  | 0 | 0 0 |
| 0 1 | 0 | 1 | 1 |  | 0 1 | 0 |  | 0 | 1 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Y** |  |
| 0 | 0 | 1 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Y** |  |
| 0 | 0 | 0 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **Y** |  | **A** | **B** | **Y** |  | **A** | **B** | **Y** |  | **A** | **B** | **Y** |  | 1 | 0 | 0 |  | 1 |  |  | 𝑆 |  |
| 真值表 | 0 | 0 | 1 |  | 0 | 0 | 1 |  | 0 | 0 | 0 |  | 0 | 0 | 1 |  | 1 | 1 | 1 |  | 1 |  |  |  | 𝐴𝑖 𝐵𝑖 |
| 0 1 | 0 | 1 | 1 |  | 0 | 1 | 0 |  | 0 | 1 | 1 |  | 0 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 0 | 1 |  | 1 | 0 | 0 |  | 1 | 0 | 1 |  | 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 1 | 0 |  | 1 | 1 | 0 |  | 1 | 1 | 0 |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |
| 电路符号  与门的天然逻辑 异或的天然逻辑  “屏蔽” “加法”  “奇偶” | | | | | | | | | | | | | | | | 𝐴𝑖 𝑆𝑖 ：输入中有奇数个1时为1(异或) 𝐶𝑖−1  𝐵𝑖 𝐴𝑖  𝐶𝑖−1 𝐶𝑖 ：思路1. 输入中至少2个1 𝐵𝑖  思路2. 进位来源  -产生(来自本级𝐴𝑖和𝐵𝑖)  -传递(来自前一级𝐶𝑖−1) | | | | | |  |  | 𝐺𝑖  𝑖 |  |



王道考研/CSKAOYAN.COM

𝐴𝑖 𝐵𝑖

进位触发器

串行加法器：只有一个全加器，数据逐位串行送入加法器中进行运算。进位触发器用来寄存进位信号，以便参与下一次运算。

如果操作数长*n*位，加法就要分*n*次进行，每次产生一位和，并且串行

逐位地送回寄存器。

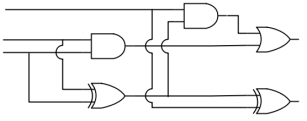
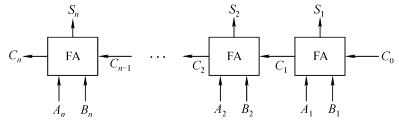
𝐶𝑖−1

𝐶𝑖

𝑆𝑖

串行加法器

FA



𝐶𝑖 = 𝐺𝑖 + 𝑃𝑖 𝐶𝑖−1

𝐺𝑖= 𝐴𝑖𝐵𝑖 𝑃𝑖 = 𝐴𝑖𝐵𝑖

*C*1=*G*1+*P*1*C*0 *C*2=*G*2+*P*2*C*1=*G*2+*P*2*G*1+*P*2*P*1*C*0 *C*3=*G*3+*P*3*C*2=*G*3+*P*3*G*2+*P*3*P*2*G*1+*P*3*P*2*P*1*C*0

王道考研/CSKAOYAN.COM

𝑆𝑖

𝑃𝑖

𝐶𝑖

𝐺𝑖

𝐶𝑖−1

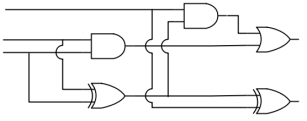
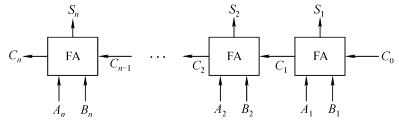
𝐴𝑖

𝐵𝑖

串行进位的并行加法器：把n个全加器串接起来，就可进行两个n位数的相加。

串行进位又称为行波进位，每一级进位直接依赖于前一级的进位，即进位信号是逐级形成的。

并行加法器



𝐶𝑖 = 𝐺𝑖 + 𝑃𝑖 𝐶𝑖−1

𝐺𝑖= 𝐴𝑖𝐵𝑖 𝑃𝑖 = 𝐴𝑖𝐵𝑖

*C*1=*G*1+*P*1*C*0 *C*2=*G*2+*P*2*C*1=*G*2+*P*2*G*1+*P*2*P*1*C*0 *C*3=*G*3+*P*3*C*2=*G*3+*P*3*G*2+*P*3*P*2*G*1+*P*3*P*2*P*1*C*0

王道考研/CSKAOYAN.COM

𝑆𝑖

𝑃𝑖

𝐶𝑖

𝐺𝑖

𝐶𝑖−1

𝐴𝑖

𝐵𝑖

并行进位的并行加法器：各级进位信号同时形成，又称为先行进位、同时进位

𝐴3～ 𝐴0 𝐵3～ 𝐵0

…

…

𝐶0

𝐶4

*G*1 *P*1

*G2 P2*

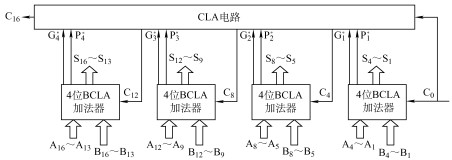
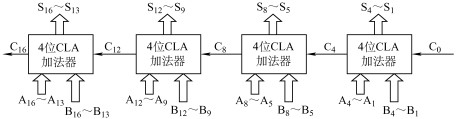
*Gn-1 Pn-1*

𝑆3～ 𝑆0

并行加法器

4位CLA

加法器



王道考研/CSKAOYAN.COM

多级先行进位方式，又称为组内并行、组间并行进位方式

单级先行进位方式，又称为组内并行、组间串行进位方式。

𝐴3～ 𝐴0 𝐵3～ 𝐵0

𝐶0

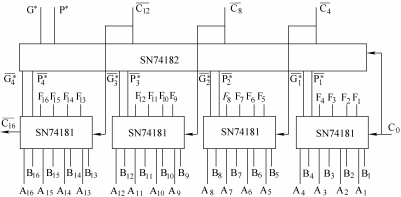
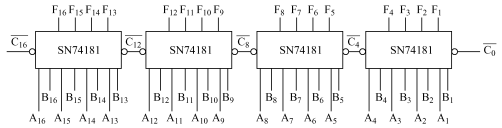
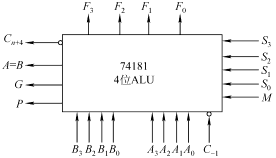
𝐶4

𝑆3～ 𝑆0

并行加法器

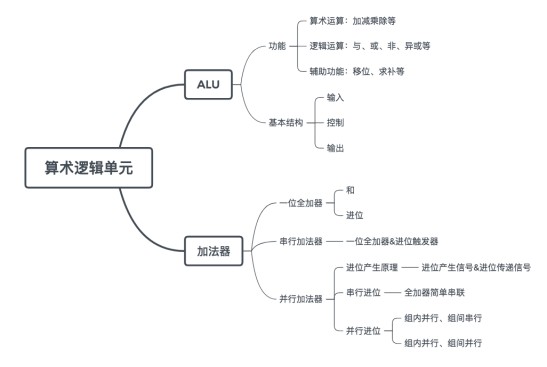
4位CLA

加法器



**ALU**芯片的组织

王道考研/CSKAOYAN.COM



本节总览

王道考研/CSKAOYAN.COM