**饱和**

在多媒体中，对于颜色或音量达到最大或最小值时，继续增大或减小应使结果保持为最大或最小值。

**溢出的检测**

 可能产生溢出的情况

 两正数加，变负数，上溢（大于机器所能表示的最大数）

 两负数加，变正数，下溢（小于机器所能表示的最小数）

**保护位(Guard Digits):** 在浮点数中间计算中，在尾数中右边多保留两位，第一位为保护位，第二位为舍入位。

**舍入位(Round Digits)：**规格化后有效尾数大小的右边还有一些非零数字，这个数据就需要舍入。

一个舍入数字必须被进位到保护位的右边，因而在规格化左移之后，根据舍入位的数值，可以对该结果进行舍入处理。

**粘位（Sticky Bit）：**

为进一步改进舍入处理的结果，在舍入数字右边的附加位。

为了更高的准确性，当舍入数字为 B/2 时，需要一个粘位，即如果有在舍入数字的尾部有任何1位丢失，则置为1。

3.6 假定185和122是无符号8位十进制整数。计算185-122.是否有上溢或者下溢？

***∵ 185 – 122 = 185 + (-122) = 63***

***∴ 无上下溢***

3.7 假定185和122是带符号8位十进制整数，以有符号数格式存储。计算185+122.是否有上溢或者下溢？

***∵ 有符号数格式：sign-magnitude format，首位是符号位，后面是数值，表示+0 ~ 127, -0 ~ -127***

***∴ 185： - 57***

***122：122***

***∴ 185 + 122：-57 + 122 = 65***

***∴ 无上下溢***

3.8. 假定185和122是带符号8位十进制整数。计算185-122.是否有上溢或者下溢？

***∵ 185： - 57***

***-122：-122***

***∴ 185 - 122：-57 + (-122) = -179 < -127***

***∴ 有下溢***

3.9. 假定151和214是带符号8位十进制整数并以补码形式存放。使用饱和算术计算151+214.结果以十进制表示。

***∵ 补码：two’s complement format，表示0 ~ 127，-128 ~ -1***

***∴ 151：-105***

***214：-42***

***∴ 151 + 214：-105 + (-42) = -147 < -128***

***∴ -128***

3.10. 假定151和214是带符号8位十进制整数并以补码形式存放。使用饱和算术计算151-214，结果以十进制表示。

***∴ 151：-105***

***214：-42***

***-214：42***

***∴ 151 - 214：-105 + 42 = -63***

3.11. 假定151和214是无符号8位十进制整数并以补码形式存放。使用饱和算术计算151+214.结果以十进制表示。

***∵ 151 + 214：151 + 214 = 365 > 255***

***∴ 255***

**3.13 使用图3-5的类似表格及硬件算法计算无符号数16进制表示的0x**62和0x12的积。

|  |  |  |  |
| --- | --- | --- | --- |
| **StepS** | **Action** | **Multiplicand** | **Product/Multiplier** |
| **0** | Initial Vals | 0110 0010 | 0000 0000 0001 0010 |
| ***1*** | ***lsb=0, no op*** | ***110 010*** | ***000 000 001 010*** |
| ***Rshift Product*** | ***110 010*** | ***000 000 000 101*** |
| ***2*** | ***Prod=Prod+Mcand*** | ***110 010*** | ***110 010 000 101*** |
| ***Rshift Mplier*** | ***110 010*** | ***011 001 000 010*** |
| ***3*** | ***lsb=0, no op*** | ***110 010*** | ***011 001 000 010*** |
| ***Rshift Mplier*** | ***110 010*** | ***001 100 100 001*** |
| ***4*** | ***Prod=Prod+Mcand*** | ***110 010*** | ***111 110 100 001*** |
| ***Rshift Mplier*** | ***110 010*** | ***011 111 010 000*** |
| ***5*** | ***lsb=0, no op*** | ***110 010*** | ***011 111 010 000*** |
| ***Rshift Mplier*** | ***110 010*** | ***001 111 101 000*** |
| ***6*** | ***lsb=0, no op*** | ***110 010*** | ***001 111 101 000*** |
| ***Rshift Mplier*** | ***110 010*** | ***000 111 110 100*** |

3.19 按照课件除法算法3，假设A（74）和B（21）是6位无符号整数计算(74)8除以(21)8。相当于60/17=3余9

|  |  |  |  |
| --- | --- | --- | --- |
| iteration | step | Divisor | Remainder |
| 0 | Initial Values | 010 001 | 000 000 111 100 |
| 1 | ***Rem=Rem–Div*** | ***000 000*** | ***010 001 000 000*** |
| ***Rem<0,R+D,Q<<*** | ***000 000*** | ***010 001 000 000*** |
| ***Rshift Div*** | ***000 000*** | ***001 000 100 000*** |
| 2 | ***Rem=Rem–Div*** | ***000 000*** | ***001 000 100 000*** |
| ***Rem<0,R+D,Q<<*** | ***000 000*** | ***001 000 100 000*** |
| ***Rshift Div*** | ***000 000*** | ***000 100 010 000*** |
| 3 | ***Rem=Rem–Div*** | ***000 000*** | ***000 100 010 000*** |
| ***Rem<0,R+D,Q<<*** | ***000 000*** | ***000 100 010 000*** |
| ***Rshift Div*** | ***000 000*** | ***000 010 001 000*** |
| 4 | ***Rem=Rem–Div*** | ***000 000*** | ***000 010 001 000*** |
| ***Rem<0,R+D,Q<<*** | ***000 000*** | ***000 010 001 000*** |
| ***Rshift Div*** | ***000 000*** | ***000 001 000 100*** |
| 5 | ***Rem=Rem–Div*** | ***000 000*** | ***000 001 000 100*** |
| ***Rem<0,R+D,Q<<*** | ***000 000*** | ***000 001 000 100*** |
| ***Rshift Div*** | ***000 000*** | ***000 000 100 010*** |
| 6 | ***Rem=Rem–Div*** | ***000 000*** | ***000 000 100 010*** |
| ***Rem>0,Q<<1*** | ***000 001*** | ***000 000 100 010*** |
| ***Rshift Div*** | ***000 001*** | ***000 000 010 001*** |

3.29. 手算的和，假设有1位保护位，1位舍入位和1位黏贴位，并采用向最靠近的偶数舍入的模式，给出所有步骤（16位精度表示，1位符号位，指数5位，尾数10位）

***∵ 2.6125 \* 10^1 = 1.10100010 00 0 \* 2^4（G0，R0，S0***

***4.150390625 \* 10^(-1) = 1.1010100111 \* 2^(-2)***

***= 0.0000011010 10 0111 \* 2^4（G1，R0，S1***

***∴ sum = 1.1010100010 10 1 \* 2^4***

***= 1.1010100011 \* 2^4***

***= 2.6546875 \* 10^1***

**3.30** 手算 and的积，假设有1位保护位，1位舍入位和1位黏贴位，并采用向最靠近的偶数舍入的模式，给出所有步骤。（16位精度表示，1位符号位，指数5位，尾数10位）

***∵ -8.0546875 \* 10^0***

***= 1.0000000111 \* 2^3***

***-1.79931640625 \* 10^(-1)***

***= -1.0111000010 \* 2^(-3)***

***∴ sign：-1 \* (-1) = +1***

***exponent：3 + -3 = 0***

***significand：1.0000000111 \* 1.0111000010***

***= 1.0111001100 00 01001110（G0，R0，S1***

***= 1.0111001100***

***∴ pro = 1.0111001100 \* 2^0 = 1.44921875 \* 10^0***

3.33. 手算 ，假设有1位保护位，1位舍入位和1位黏贴位，并采用向最靠近的偶数舍入的模式，给出所有步骤。（16位精度表示，1位符号位，指数5位，尾数10位）

***∵ 3.4375×10^(-1) = 1.0110000000 \* 2^(-2)***

***= 0.0000000000 01 0110000000 \* 2^(10)***

***（G0，R1，S1***

***1.771×10^3 = 1.1011101011 00 0 \* 2^10（G0，R0，S0***

***∴ sum1 = 1.1011101011 01 0 \* 2^(10)***

***= 1.1011101011 \* 2^(10)***

***∵ 3.984375×10^(-1) = 1.1001100000 \* 2^(-2)***

***= 0.0000000000 01 1001100000 \* 2^(10)***

***（G0，R1，S1***

***∴ sum2 = 1.1011101011 01 1 \* 2^(10)***

***= 1.1011101011 \* 2^(10)***

***= 1771***