定点 DSP 处理浮点数

BG6RDF

TMS320C5509A 是 16 位定点数处理器，其累加器是 32 位/40 位的。在定点处理器中处理浮点数需要对定点数进行定标。下面所说的定点数都是指有符号数。

通常采用的定标有 Q15 和 Q31，分别表示小数点后有二进制 15 位和二进制 16 位。因此 16 位 Q15 最大能表示的数是$1-2^{-15}$，32 位 Q31 最大能表示的数是$1-2^{-31}$。定标不同的数可以直接相乘，例如 Q13*Q15=Q28。两个定标不同的数不能直接相加，比如 Q13+Q15，通常要将 Q15 右移两位，转换为 Q13 后再相加，当然这样损失了两位的精度。

DSP 进行的乘累加操作常常 Q15 的数，这样结果为 Q30，存储在累加器中。为了将累加器的结果转换为更为常用的 Q31 定标，55x 系列 DSP 在状态寄存器 ST1_55 中设置了 FRCT 控制位，FRCT=1 时，乘积自动左移一位，这样乘积变成了 Q31。对于累加器中 Q31 定标的数，直接取累加器中高 16 位，就能获得结果的 Q15 定标数。

很多时候 Q15 不能解决问题，比如 IIR 滤波器 num, den 系数中通常有大于等于 1 的系数，超过了 Q15 的范围，只能用 Q14，Q13 等定标。这时乘累加操作就需要进行修正了，比如 IIR 滤波器中系数用 Qx 定标，输入数据和输出数据用 Qy 定标，$Q_x * Q_y = Q_{x+y}$，为获得 Qy 的输出数据必须将累加器中的乘积右移 x 位，另外在乘累加操作开始前必须将输入数据在累加器中左移 y 位，进行对齐后才能进行乘累加。当然，这种修正都是在没有考虑 FRCT 的情况下。

在 C5500 DSPLIB 中 iircas51 函数中，FRCT 设置为 1，输入输出数据采用 Q15 定标，如果系数也是 Q15 定标，则程序运行无误，如果系数采用 Q14 及以下定标则会产生严重的问题。以下是其代码片段：

```
MOV     *AR0+ << #16, AC1 ; HI(AC1) = x(n)
||RPTBLOCAL  loop2-1        ;inner loop: process a bi-quad

MPYM    *AR1+, AC1, AC0         ; AC0 = b0*x(n)

MACM    *AR1+, *(AR3+T0), AC0  ; AC0 += b1*x(n-1)

MACM    *AR1+, *AR3, AC0   ; AC0 += b2*x(n-2)

MOV     HI(AC1), *AR3       ; x(n) replaces x(n-2)
||AADD  T1, AR3             ; point to next x(n-1)

MASM    *AR1+, *(AR4+T0), AC0  ; AC0 -= a0*y(n-1)

MASM    *AR1+, *AR4, AC0   ; AC0 -= a1*y(n-2)

MOV     rnd(HI(AC0)), *AR4 ; y(n) replaces y(n-2)
||AADD  T1, AR4             ;point to next y(n-1)

MOV     AC0, AC1        ;input to next biquad
```

从代码片段可以看出，累加器 AC0 为 Q31 定标，输出数据是累加器高 16 位。如果系数是 Q13，则累加器中是$Q_{15+13+1} = Q_{29}$定标，直接取累加器高 16 位显然是不正确的。如果

要获得 Q15 的输入结果，必须将累加器左移 2 位，即转换为 Q31 定标后，再取累加器高 16 位作为结果才是正确的。下面是修改后的代码片段，适用于 Q13 定标的 IIR 系数。

```
MOV       *AR0+ << #16, AC1 ; HI(AC1) = x(n)
     ||RPTBLOCAL  loop2-1          ;inner loop: process a bi-quad

     MPYM     *AR1+, AC1, AC0           ; AC0 = b0*x(n)

     MACM     *AR1+, *(AR3+T0), AC0  ; AC0 += b1*x(n-1)

     MACM     *AR1+, *AR3, AC0   ; AC0 += b2*x(n-2)

     MOV      HI(AC1), *AR3         ; x(n) replaces x(n-2)
     ||AADD   T1, AR3               ; point to next x(n-1)

     MASM     *AR1+, *(AR4+T0), AC0  ; AC0 -= a0*y(n-1)

     MASM     *AR1+, *AR4, AC0   ; AC0 -= a1*y(n-2)

     SFTS AC0, #2

     MOV      rnd(HI(AC0)), *AR4 ; y(n) replaces y(n-2)
     ||AADD   T1, AR4               ;point to next y(n-1)

     MOV      AC0, AC1         ;input to next biquad
```

另外 DSPLIB Programmer's Guide 说明 iircas51 的 nx 参数可以设置为 1。事实上，nx 必须为偶数。iircas51 使用了一个深度为 2 的循环缓冲区，作为 delay line buffer，也就是下次调用 iircas51 函数时，必须使用上次返回的缓冲区中的数据。如果 nx 为奇数，则 delay line buffer 在下次调用时会发生错位。