TMS320 DSP DESIGNER'S NOTEBOOK

TEXAS INSTRUMENTS

Extending Fixed-Point Dynamic Ranges

Contributed by Alex Tessarolo

Design Problem

How can you extend the fixed-point math dynamic range beyond the range of a Q15 number with a minimum of instructions?

Solution

In many advanced control problems such as state estimators, Kalman filters and some high Q filters, the dynamic range/accuracy of the coefficient can sometimes be beyond the range of a Q15 number while the data value can be typically represented as a Q15 number.

Aside from trying to dynamically scale the coefficients to extract as much accuracy as possible or trying to use floating point math, there is a technique that can perform 32-bit × 16-bit math at an effective 4 cycles per Tap and potentially 2 cycles per Tap for larger then 6th order systems (+ some fixed overhead of about 8-13 cycles).

The trick is to re-scale the numbers and represent the problem as an integer value + a fractional value. For example: 16q16

$$Y = 2.391456*X0 - 0.0235045*X1 + 0.000329758*X2 - 34.3392345*X3$$ 2q14

34

In the above equation, the filter Coefficients have a dynamic range exceeding a 16-bit Q15 number. If we re-scale the problem as follows:

7q9

$$Y = [1224.425472*X0 - 12.034304*X1 + 0.168836096*X2 - 17581.68806*X3]/512$$

And then allocate the following coefficient values:

$$Y = [(A0i+A0f)*X0 + (A1i+A1f)*X1 + (A2i+A2f)*X2 + (A3i+A3f)*X3]/512$$

7q 27

where:

A0i = 1224 = 04C8h
A0f = 0.425472 = 3676h (= 0.425476074)
A1i = –12 = FFF4h
A1f = –0.034304 = FB9Ch (= –0.034301758) 16q16
A2i = 0 = 0000h
A2f = 0.168836096 = 159Ch (= 0.168823242)
A3i = –17581 = BB53h
A3f = –0.68806 = A7EEh (= –0.688049316)

2

1q15

The problem then reduces to calculating the following:

Y = (A0i*X0 + A1i*X1 + A2i*X2 + A3i*X3) + (A0f*X0 + A1f*X1 + A2f*X2 + A3f*X3)

This is like calculating two filter banks. The above problem is coded in the example below:

```
; Assume:         X0,X1,X2,X3 = Q15 (-1    range    0.999053955)
;                 Y = Q10 (-32  range  +31.99902344)
;  Ymin-max = 2.391456 + 0.0235045 + 0.000329758 + 34.3392345
;                 = +/- 36.75452476
;                 Sat     = 06000h
;                 Round   = 08000h

        SETC    OVM       ; Enable saturation.
        SETC    SXM       ; Enable sign extension.
        SPM     3         ; Set shift mode = -6
        LT      A0f
        MPY     X0        ; P = A0f*X0
        LTP     A1f       ; ACC = A0f*X0
        MPY     X1        ; P = A1f*X1
        LTA     A2f       ; ACC = ACC + A1F*X1
        MPY     X2        ; P = A2f*X2
        LTA     A3f       ; ACC = ACC + A2f*X2
        MPY     X3        ; P = A3f*X3
        LTA     A0i       ; ACC = ACC + A3f*X3
        SPM     0
        SACH    Temp,6    ; On C5X replace by BSAR 9
        LAC     Temp,1    ; ACC = ACC/512
        ; instruction.
        MPY     X0        ; P = A0i*X0
        LTA     A1i       ; ACC = ACC + A0i*X0
        MPY     X1        ; P = A1i*X1
        LTA     A2i       ; ACC = ACC + A1i*X1
        MPY     X2        ; P = A2i*X2
        LTA     A3i       ; ACC = ACC + A2i*X2
        MPY     X3        ; P = A3i*X3
        APAC              ; ACC = ACC + A3i*X3
        ADDS    Round     ; Round result.
        ADDH    Sat       ; Saturate Y to Q10 value
        SUBH    Sat
        SUBH    Sat
        ADDH    Sat
        SACH    Y,1       ; Y = Q10 number.

; Cycles = 13 + 4n cycles (n = number of taps).

; Note: If saturation is not required, Cycles = 8 + 4n cycles
```

Figure 1.

Q14

sY = 1k0*sX0 + . . . + 1k3*sX3

movdn Y, ROUND
madd.q Y, Y, k, XL, #0
madd.q Y, Y, k, XU, #0
madd.q Y, Y, k, XL, #0
madd.q Y, Y, k, XU, #0

to do

this is a neat trick;

If the number of taps is greater then 6, then a RPT loop can be used for each bank and the effective cycles/tap can be approximately 2.

The above technique is almost equivalent to a floating-point notation with a 4-bit exponent and a 16-bit mantissa.