TMS320 DSP
Number 10
DESIGNER'S NOTEBOOK
TEXAS INSTRUMENTS

Initializing the Fixed-point EVM's AIC

Contributed by Jason Chyan

Design Problem How do I program the AIC registers for a given sampling rate, f_s, and low-pass filter cutoff frequency, f_c?

Solution There are two pairs of registers, TA, TB and RA, RB. The T and R mean transmit and receive. Both pairs work the same way, so only one pair will be discussed here. The TA and TB registers can be written to via the DSP's serial port. The word sent to the AIC must have the two LSBs of the data word programmed to indicate that a control word is present. Typically, these two bits are 11. After receive a data word with two LSBs programmed as 11, the AIC will send another FSX signal after four shift clocks delayed to request the DSP to send the control word. The two LSBs of the control word will be programmed as 00 to indicate to program the TA and RA registers, and as 10 to program the TB and RB registers.

A second register, TA′ may also be programmed. The two LSBs for the control word to program the TA′ and RA′ registers are 01. The TA′ register will cause a small change in the sampling frequency. The two LSBs of the data word are again used to program the use of the TA′ register. TA+TA′ is programmed as 01 while TA–TA′ is programmed as 10.

There are three equations you can use to determine f_s and f_c.

```
fc = fm/(72*TA)        ; given a masterclock fm = 10.368 MHz
fs = (36/TB) * fc      ; TA' not used, LSBs = 00
fs = (36*fc*fm)/(TB*fm+36*fc*TA')
```

Table 1. TA and TB vs. f_c and f_s

TA	f_c (KHz)	TB	f_s/f_c
31	4.6	63	0.57
29	5.0	36	1.0
24	6.0	18	2.0
21	6.8	12	3.0
18	8.0	9	4.0
16	9.0	6	6.0
14	10.3		
9	16.0		
6	24.0		

The following examples illustrate the use of Table 1.

Suppose f_s = 16 KHz and f_c = 8 KHz are desired.

f_c = 10368/(72*18) = 8 ; TA = 18
f_s = (36/18)*8 = 16 ; TB = 18

If TA′ = 20 is used, the following calculation results:

f_s = (36*8*10368)/(18*10368+36*20) = 15.756

Clearly, TA′ reduced f_s, but not much. It is used in modem applications to advance or retard conversion frequencies.

Other examples:

a. f_c = 16 KHz ; TA = 9
   f_s = 16 KHz ; TB = 36

b. f_c = 6 KHz ; TA = 24
   f_s = 18 KHz ; TB = 12

Some other caveats include:

1. $f_c$min = 4.6 KHz where TA = 31
2. $f_s$min = 2.622 KHz where TB = 63 and TA = 31
3. $f_c$max = 28.8 KHz where TA = 5 (min allowed value)
4. $f_s$max = 25 KHz the maximum conversion rate for AIC

```
           .mmregs
           .global START, AICINIT,AIC_2ND
           .data
TA         .word     18       ; fc = 8 KHz
RA         .word     18       ; fc = 8 KHz
TAp        .word     31
RAp        .word     31
TB         .word     18       ; fs = 2 * fc
RB         .word     18
AIC_CTR    .word     8Dh
ACC_lo     .word     0
ACC_hi     .word     0
TEMP       .word     0
* initialization
           .text
START:     DINT                ; disable interrupts
           LDPK      #0        ; data page pointer == 0
           LARP       0        ; point to AR0
;
           CALL      AICINT    ; initialize AIC and enable ints
* put main program here
           LACK      #010h     ; use RINT as sync for
           SACL      IMR       ; TX and RX
AICINIT:
           SFSM                ; non-continuous mode
           RTXM                ; FSX as input
           FORT      0         ; 16-bit words
           LALK      #0ffefh   ; Pulse AIC reset by setting it low
           SACL      TEMP,0
           OUT       TEMP,PA2  ; Write to AIC
           RPTK      #255      ; and then taking it high after 10k cycles
           NOP                 ; (.5ms at 100nS)
           RPTK      #243
           NOP
```

Figure 1. TMS320C25 code example

```
            LALK    #0FFFFh
            SACL    TEMP,0
            OUT     TEMP,PA2
            LDPK    0
            LACK    020h
            SSXM
            SACL    IMR      ; XINT interrupt
            LAC     TA,9     ; initialize TA register
            ADD     RA,2
            CALL    AIC_2ND
            LAC     TAp,9    ; initialize TA'
            ADD     RAp,2
            ADDK    01h
            CALL    AIC_2ND
            LAC     TB,9     ; initialize TB register
            ADD     RB,2     ;
            ADDK    02h
            CALL    AIC_2ND
            LAC     AIC_CTR,2 ; initialize control register
            ADDK    03h
            CALL    AIC_2ND
            RET
AIC_2ND:
            LDPK    0
            SACH    DXR      ; load transmit data register
            IDLE             ; wait for int
            ADLK    6,16
            SACH    DXR
            IDLE             ; ACC_hi requests 2nd XMIT
            SACL    DXR
            IDLE             ; ACC_lo sets up registers
            ZAC
            SACL    DXR      ; make sure word was sent
            RET
```

Figure 1. TMS320C25 code example (continued)