TMS320 DSP
DESIGNER'S NOTEBOOK

Initializing the Fixed-point EVM's AIC

APPLICATION BRIEF: SPRA206

Jason Chyan
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
December 1992

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Examples

Tables

Initializing the Fixed-point EVM's AIC

Abstract

There are two pairs of AIC registers, TA, TB and RA, RB, which control the sampling rate, f_s, and low-pass filter cutoff frequency, f_c. (The T and R mean transmit and receive.) This document discusses how to program these registers via the DSP's serial port. A data table and a lengthy code example provide the information necessary to determine these frequencies.

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 receiving a data word with two LSBs programmed as 11, the AIC will send another FSX signal, delayed four shift clocks, to request the DSP to send the control word. The two LSBs of the control word will be programmed as 00 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:
$f_c = f_m / (72 * TA)$; given a master clock f_m = 10.368 MHz
$f_s = (36 / TB) * f_c$; TA' not used, LSBs = 00
$f_s = (36 * f_c * f_m) / (TB * f_m + 36 * f_c * TA')$

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:

```
fc = 10368/(72*18) = 8     ; TA = 18
fs = (36/18)*8 = 16        ; TB = 18
```

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

```
fs = (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.    fc = 16 KHz     ; TA =  9
      fs = 16 KHz     ; TB = 36
b.    fc =  6 KHz     ; TA = 24
      fs = 18 KHz     ; TB = 12
```

Some other caveats include:

```
1. fcmin = 4.6 KHz   where TA = 31
2. fsmin = 2.622 KHz where TB = 63 and TA = 31
3. fcmax = 28.8 KHz  where TA = 5 (min allowed value)
4. fsmax = 25 KHz the maximum conversion rate for AIC
```

Example 1. TMS320C25 Code Example

```
        .mmregs
        .global START, AICINIT,AIC_2ND
        .data
TA      .word   18    ; f c = 8 KHz
RA      .word   18    ; f c = 8 KHz
Tap     .word   31
Rap     .word   31
TB      .word   18    ; f s = 2 * f c
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
                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
```