```
    page    132,60
;****************************************************************************
;   ADA_INIT.ASM
;   Initialization program for the Crystal Semi. CS4215 A/D &
;          D/A Converters
;
;   NOTES:
;          1-  The DSP5002EVM analog inputs are connected to "MIC"
;                input pins only.  The "LINE" input pins are connected
;                to analog GROUND.  Line level inputs are supported
;                by disabling the 20 dB Microphone Pre-Amplifier.
;          2-  As shipped, the DSP56002EVM has a 24.576 MHz crystal
;                connected to the XTAL2 inputs of the CD4215.  Do
;                not select any other clock source unless the board
;                has been modified to support the alternate clock
;                source(s).
;
;
;   Copyright (c) MOTOROLA 1995
;             Semiconductor Products Sector
;             Digital Signal Processing Division
;
;****************************************************************************
;
;      portc usage:
;       bit8: SSI TX (from DSP to Codec)
;   bit7: SSI RX (from Codec to DSP)
;   bit6: SSI Clock
;   bit5: SSI Frame Sync
;   bit4: codec reset (from DSP to Codec)
;   bit3:
;       bit2: data/control bar
;             0=control
;             1=data
;
;****************************************************************************

;****************************************************************************
;  Defined Constants used to construct data for the CONTROL TIME SLOTS
;****************************************************************************
NO_PREAMP       equ     $100000 ;0 == enable 20 dB pre-amp
LO_OUT_DRV      equ     $080000 ;0 == 2.8 Vp-p line (1V rms)
                                ;0 == 4.0 Vp-p headphones
                                ;1 == Line and Headphone 2.0Vp-p
HI_PASS_FILT    equ     $008000 ;0 == HPF disabled
SAMP_RATE_9     equ     $003800 ; 9.6 kHz sample rate
SAMP_RATE_48    equ     $003000 ;48   kHz sample rate
SAMP_RATE_32    equ     $001800 ;32   kHz sample rate
SAMP_RATE_27    equ     $001000 ;27.4 kHz sample rate
SAMP_RATE_16    equ     $000800 ;16   kHz sample rate
SAMP_RATE_8     equ     $000000 ; 8   kHz sample rate
STEREO          equ     $000400 ;1 == stereo, 0 == mono
DATA_8LIN       equ     $200300 ; 8-bit unsigned linear
DATA_8A         equ     $200200 ; 8-bit A-law
DATA_8U         equ     $200100 ; 8-bit u-law
DATA_16         equ     $200000 ;16-bit 2s complement linear
IMMED_3STATE    equ     $800000 ;1=SCLK & FS 3-state immediately
XTAL2_SELECT    equ     $200000 ;REQUIRED as this is the only clock
                                ;  source on the board
```

```
BITS_64         equu    $000000 ; 64 bits per frame
BITS_128        equ     $040000 ;128 bits per frame
BITS_256        equ     $080000 ;256 bits per frame
CODEC_MASTER    equ     $020000 ;1 == codec generates SCLK & FS
                                ;0 == codec receives  SCLK & FS
CODEC_TX_OFF    equ     $010000 ;0 == enable codec TX to DSP
                                ;1 == disable (Hi-Z) codec output

;****************************************************************************
;  Defined Constants used to construct data for the DATA TIME SLOTS  (5-8)
;****************************************************************************
HEADPHONE_EN    equ     $800000 ;1 == headphone output enabled, 0 == muted

LINEOUT_EN      equ     $400000 ;1 == line      output enabled, 0 == muted
LEFT_ATTN       equ     $010000 ;63 steps * 1.5 dB = -94.5 dB
SPEAKER_EN      equ     $004000 ;1 == speaker   output enabled, 0 == muted
RIGHT_ATTN      equ     $000100 ;63 steps * 1.5 dB = -94.5 dB
MIC_IN_SELECT   equ     $100000 ;1 == A/D inputs from MIC pins  NOTE: the
                                ;   DSP56002EVM uses these pins.  The line
                                ;   input pins are not used.
LEFT_GAIN       equ     $010000 ;15 steps * 1.5 dB =  22.5 dB
MONITOR_ATTN    equ     $001000 ;15 steps * 6.0 dB =  90.0 dB (mute)
RIGHT_GAIN      equ     $000100 ;15 steps * 1.5 dB =  22.5 dB

;----------------------------------------------------------------------------
;  constructed constants for codec set up/initialize
;----------------------------------------------------------------------------
CTRL_WD_12      equ     NO_PREAMP+HI_PASS_FILT+SAMP_RATE_48+STEREO+DATA_16   ;CL
B=0
CTRL_WD_34      equ     IMMED_3STATE+XTAL2_SELECT+BITS_64+CODEC_MASTER
CTRL_WD_56      equ     $000000
CTRL_WD_78      equ     $000000

;----------------------------------------------------------------------------
;  constructed constants for codec data mode
;----------------------------------------------------------------------------
OUTPUT_SET      equ     HEADPHONE_EN+LINEOUT_EN+(LEFT_ATTN*4)
INPUT_SET       equ     MIC_IN_SELECT+(15*MONITOR_ATTN)+(RIGHT_ATTN*4)

;---DSP56002 on-chip peripheral addresses
IPR             equ     $FFFF   ;Interrupt Priority Register
BCR             equ     $FFFE   ;Bus Control Register
PLL             equ     $FFFD   ;PLL Control Register
SSIDR           equ     $FFEF   ;SSI Data Register
SSISR           equ     $FFEE   ;SSI Status Register
CRB             equ     $FFED   ;SSI Control Register B
CRA             equ     $FFEC   ;SSI Control Register A
PCD             equ     $FFE5   ;Port C Data Register
PCDDR           equ     $FFE3   ;Port C Data Direction Register
PCC             equ     $FFE1   ;Port C Control Register
PBC             equ     $FFE0   ;Port B Control Register



;----------------------------------------------------------------------------
; The two buffers which are defined below are the source and
; destination storage for the codec Input/Output ISRs
;----------------------------------------------------------------------------
       org    x:0
RX_BUFF_BASE    equ     *
```

```
RX_data_1_2     ds      1       ;data time slot 1/2 for RX ISR
RX_data_3_4     ds      1       ;data time slot 3/4 for RX ISR
RX_data_5_6     ds      1       ;data time slot 5/6 for RX ISR
RX_data_7_8     ds      1       ;data time slot 7/8 for RX ISR

TX_BUFF_BASE    equ     *
TX_data_1_2     ds      1       ;data time slot 1/2 for TX ISR
TX_data_3_4     ds      1       ;data time slot 3/4 for TX ISR
TX_data_5_6     ds      1       ;data time slot 5/6 for TX ISR
TX_data_7_8     ds      1       ;data time slot 7/8 for TX ISR

RX_PTR          ds      1       ; Pointer for rx buffer
TX_PTR          ds      1       ; Pointer for tx buffer



;****************************************************************************
;*****          initialize the CS4215 codec                         *****
;****************************************************************************
;    1.  Configure the SSI port and the GPIO lines used (D/C~ & Reset)
;    2.  Select Control Mode and reset the codec (50ms @ 40MHz)
;    3.  Initialize the Transmit Data Buffer with Control Words (CLB=0)
;    4.  Enable SSI
;    5.  Wait until CLB in receive buffer = 0
;    6.  Set CLB = 1 in TX buffer
;    7.  Send 4 frames of data
;    8.  Disable SSI port
;    9.  Program SSI to receive Frame Sync and Clock
;   10.  Select Data Mode (D/C~ = 1)
;   11.  Enable SSI port
;****************************************************************************
;
;
        org     p:
codec_init
    move    #RX_BUFF_BASE,x0
    move         x0,x:RX_PTR             ; Initialize the rx pointer
    move    #TX_BUFF_BASE,x0
    move         x0,x:TX_PTR             ; Initialize the tx pointer


;----- initialize SSI -----
; Assuming the DSP56002 is running at 40 MHz,  this section selects a
;    2.5 MHz SSI clock to be sent from the 56002's SSI clock circuit
;    to the codec.  It also sets up 64-bit frame length (4 words per
;    frame x 16-bits per word), enables SSI TX & RX, enables TX & RX
;    interrupts and synchronous operation with bit-length frame sync.
;-----------------------------------------------------------------------------
    movep   #$0000,x:PCC    ;  turn off ssi port
    movep   #$4303,x:CRA    ;  40MHz/16 = 2.5MHz SCLK, WL=16 bits, 4W/F
    movep   #$FB30,x:CRB    ; RIE,TIE,RE,TE, NTWK, SYN, FSR/RSR->bit
    movep   #$14,x:PCDDR    ; setup pc2 and pc4 as outputs
    movep   #$0,x:PCD       ; D/C~ and RESET~ = 0 ==> control mode
                ;----reset delay for codec ----
    do      #500,_delay_loop
    rep     #2000           ; 100 us delay
    nop
_delay_loop
    bset    #4,x:PCD        ; RESET~ = 1
```

```
    movep   #$3000,x:IPR    ; set interrupt priority level
    movep   #$01E8,x:PCC    ; Turn on ssi port


;--- set up the TX buffer with control mode data
    move    #CTRL_WD_12,x0
    move                     x0,x:TX_BUFF_BASE
    move    #CTRL_WD_34,x0
    move                     x0,x:TX_BUFF_BASE+1
    move    #CTRL_WD_56,x0
    move                     x0,x:TX_BUFF_BASE+2
    move    #CTRL_WD_78,x0
    move                     x0,x:TX_BUFF_BASE+3

    andi    #$FC,mr         ; enable interrupts

;
; CLB == 0 in TX Buffer,  wait for CLB == 1 in RX Buffer
;
    jclr    #3,x:SSISR,*            ; wait until rx frame bit==1
    jset    #3,x:SSISR,*            ; wait until rx frame bit==0
    jclr    #3,x:SSISR,*            ; wait until rx frame bit==1
    jset    #18,x:RX_BUFF_BASE,*    ; loop until CLB set

;
; CLB == 1 in RX Buffer, send 4 frames and then disable SSI
;
    bset    #18,x:TX_BUFF_BASE      ;set CLB
    do      #4,_init_loopB          ; Delay as 4 full frames to pass
    jclr    #2,x:SSISR,*            ; wait until tx frame bit==1
    jset    #2,x:SSISR,*            ; wait until tx frame bit==0
_init_loopB
    movep   #0,x:PCC                ;reset SSI port (disable SSI...)

;
;    now CLB should be 1 -- re-program fsync and sclk direction to input
;
;   movep   #$4303,x:CRA    ; 16bits,4 word/frame, /2/4/2=2.5 MHz
    movep   #$FB00,x:CRB    ; rcv,xmt & int ena,netwk,syn,sclk==inp,msb 1st
    movep   #$14,x:PCD      ; D/C~ pin = 1  ==> data mode
    movep   #$01E8,x:PCC    ;  turn on ssi port (enable SSI now...)

^Z
```