INFRARED COADDER BOARD

This chapter describes the infrared coadder board. It contains independent analog video processing circuits and 16-bit A/D converters for four channels, and a common digital coaddition circuit consisting of a DSP, fast static RAM memory buffer and interface logic to the system backplane. It is designed for fast readout of midband IR arrays with many output channels, such as Aladdins and BIBs. Fig. 1 is a block diagram of one of the four analog front ends.

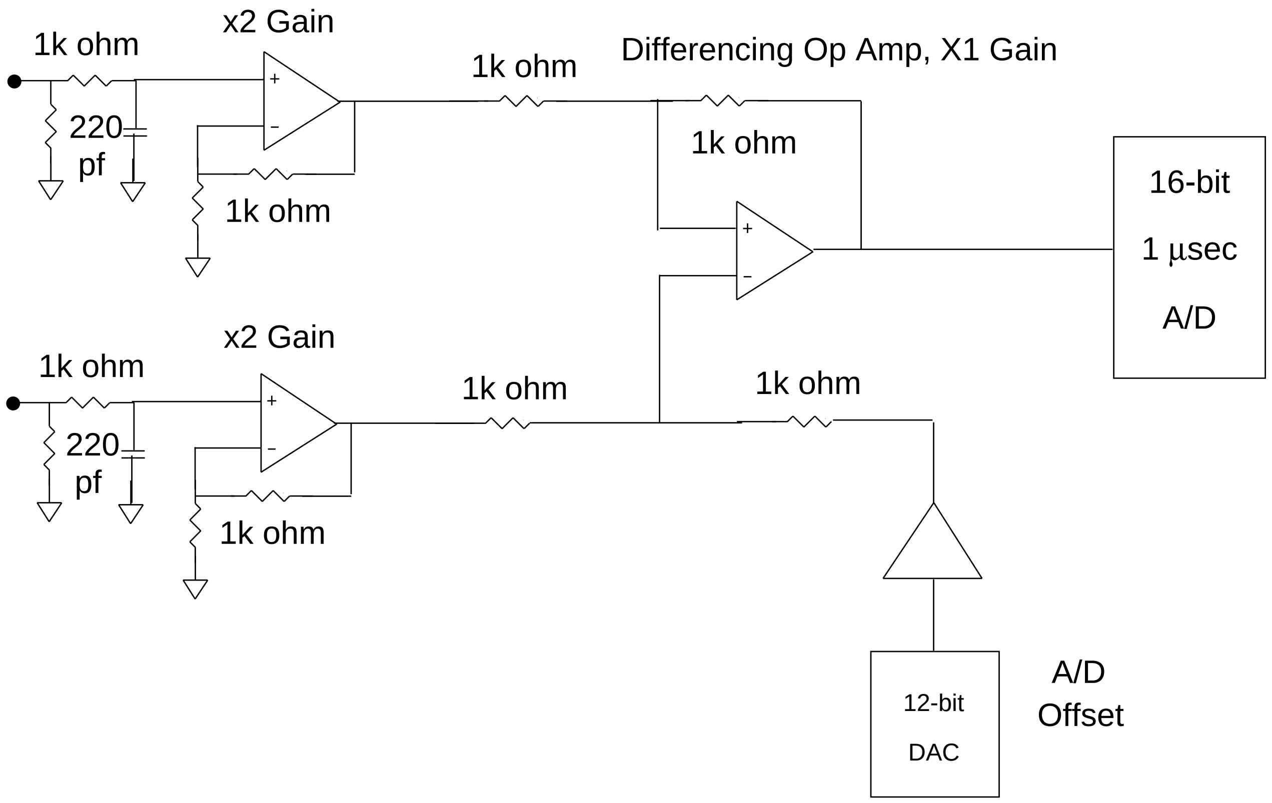

Fig. 1: One of four analog input circuits

THEORY OF OPERATION

Each of the four channels contains a three op amp, fully differential stage with an overall gain of x2. The input to the board is through a 9-pin DB male connector, with the pinouts listed below.

Pin #	Function	Pin #	Function
1	Inverting input channel #0	2	Non-inverting input, channel #0
3	Inverting input channel #3	4	Non-inverting input, channel #3
6	Inverting input channel #1	7	Non-inverting input, channel #1
8	Inverting input channel #2	9	Non-inverting input, channel #2
5	Ground		

Each of the eight input amplifiers has a user-configurable area where the user can solder in through-the-hole components to match the operational requirements of the array. These include provisions for load resistors (input to ground, R1, R3, R5, R7, R10, R12, R14 and R16), resistor/capacitor pairs that determines the pixel processing time constant (R2/C1, R4/C2, R6/C3, R8/C4, R9/C5, R11/C6, R13/C7 and R15/C8), and overvoltage zener protection diodes (D1 to D8). Default boards are shipped with an RC time constant of 220 nanosecoc (R2 = 1k ohms and C1 = 220 pf), and no load resistors or zener diodes.

Each input passes through a non-inverting amplifier stage consisting of the low noise part AD829 with a gain of 2x. The outputs from these two amplifiers connects to a unity gain AD829 op amp differencing amplifier. The differencing amplifier contains pads for installing a limiting diode and a bandwidth limiting capacitor, but these are not normally installed because the input stage RC circuit is a more effective bandwidth limiter. A fourth op amp buffers the output of a 12-bit DAC that provides an individual offset voltage to each of the four channels so the four A/D offset levels can be closely matched. Each of the four analog channels is connected to a 16-bit sampling A/D converter (Datel part number ADS937, U18 to U21). The A/D converters have a conversion time of 1.0 microseconds that is initiated on the rising edge of the CONV signal.

Fig. 2 illustrates the digital back end of the board. Digital data are written from each A/D converter to a set of latches on the rising edge of the XFER signal. The XFER line also signals the PAL U45 to generate four sequential pulses of WRFIFO* to write 16-bit image data from each of the four latches to a single FIFO, and to pulse each of the lines WRFIFO0* to WRFIFO3* that are connected to the latches in turn to cause each of the four latches to assert its data onto the FIFO input data lines. The A/D convert and XFER signals are both generated by the encoded timing board signals WRSS and SS# that are also used by the clock driver board. In this way A/D conversion and data transfer is fully synchronized to the array clocking and the coadder board DSP is only used to process the resultant image data.Once the image data are written into the FIFO they are usually allowed to accumulate until the FIFO is half full and they can be written out in one block transfer. The half full flag HF* on the FIFO is connected to bit #10 of Port B of the DSP on the board.

There is a large static RAM memory into which image data are coadded. It consists of eight 512k x 8 bits memory chips configured to provide a one Meg x 32-bit deep memory. Image data are read from or written to this memory by a Motorola DSP56002 operating at an internal clock speed of 75 MHz. The SRAM memory is mapped to the DSP address space Y:$100 to $FFFF. Addresses over the range of Y:0 to $FF are mapped to the internal Y: memory of the DSP, which is reserved for coadded image data. All parameter storage is in the internal X: memory space. To overcome the limited DSP address space of 64k words, access to the full megaword of image memory is made by having the DSP control the four higher order address lines SRAMADR19 to SRAMADR16 by writing to the four bits 7 to 4 of the parallel I/O Port B. Crossing across a page boundary requires the DSP to explicitly change one or more of these four higher order address linees, while addresses within each 64k page can be directly controlled with any of the eight 16-bit address registers R0 to R7. The high order paging address lines can be controlled with instructions such as -

```
        BCLR        #SRAMADDR16,X:PBD      ;  A17, A16 = (1,0)
        BSET        #SRAMADDR17,X:PBD
```

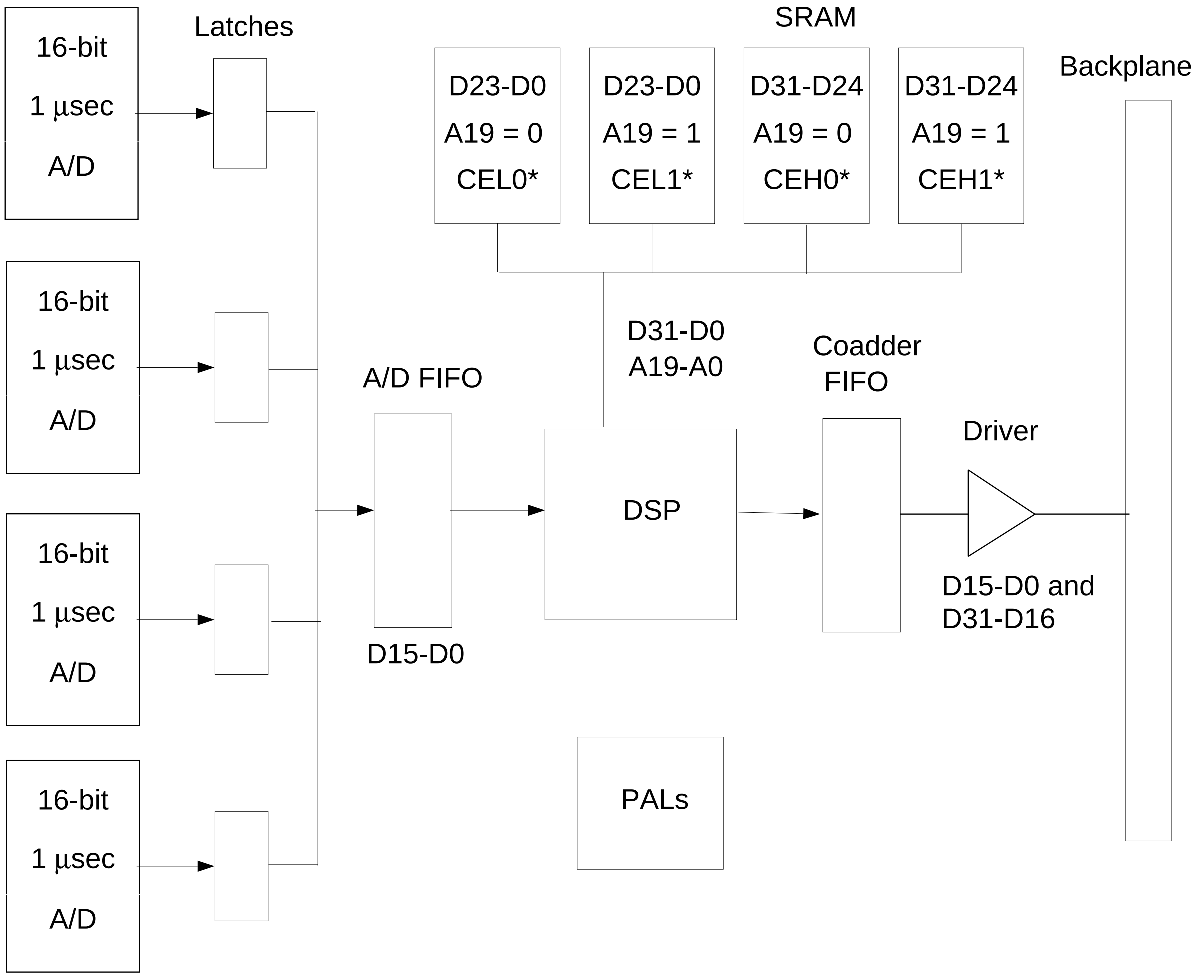

Fig. 2: Block diagram of the digital back end of the coadder board

Because the individual memory chips are 512k bytes deep, access to the 1 Mword space is accomplished at the hardware level by having the most significant address line SRAMADR19 control whether the lower or upper bank of memory is enabled, which is effected with the two pairs of lines CEL0*/CEH0* (lower bank, SRAMADR19 = 0) and CEL1*/CEH1* upper bank, SRAMADR19 =1). The numbers 0 and 1 follow the address line SRAMADR19, and the letters L and H follow the data lines, described below.

In addition to segmenting the SRAM memory into 64k pages it has its data lines segmented as well. Because the DSP only reads or writes external data 24 bits at a time access to the full 32 bits of the coadder memory requires two memory instructions. Some instruments require coaddition to 32 bits whereas some only require coaddition to 24 bits, so a software switch is provided to allow the user to select memory access of either 24 or 32 bits. If bit #2 (named M24_32 in the software) of Port C is cleared then 24-bit mode is selected, and accesses to the SRAM memory space will always assert the SRAM enable lines CEL0* or CEL1*, which are each mapped to three 512k x 8-bit memory chips. If M24_32 = 1 then 32-bit coaddition is enabled and successive coadder memory accesses are toggled back and forth between the D23-D0 memories and the D31-D24 memories. In summary -

	M24_32 = 0, 24-bit coadd	M24_32 = 1, 32-bit coadd
SRAMADR19 = 0	CEL0* asserted for D23-D0	CEL0* asserted for D23-D0

		CEH0* asserted for D31-D24
SRAMADR19 = 1	CEL1* asserted for D23-D0	CEL1* asserted for D23-D0
		CEH1* asserted for D31-D24

The coadder memory partitioning has been made somewhat complex by the finite address space of the DSP (64k words) and its limited data word width (24 bits). However, the hardware was designed to allow the DSP software to execute efficiently since most of the bandwidth of the DSP is required to keep up with four readouts operating at 1 MHz. Discussion of a DSP code segment may clarify much of the boards' operation, as follows, where the numbers in parentheses at the beginning of each line are line numbers for this discussion only.

```
(1)     BCLR      #M24_32,X:PCD          ; Select 24-bit coaddition
(2)     MOVE      #$FFFF,X1              ; Mask to zero D23-D16
(3)     JSET      #HF,X:PBD,*            ; Wait for the FIFO to be half full
(4)     DO        #512,RD_PXLS           ; Read 512 pixels = half of FIFO
(5)     MOVE      X:(R5),A1              ; Read from the A/D FIFO
(6)     AND       X1,A  Y:(R0),B1        ; D23-D16 = zero; read D23-D0 from SRAM
(7)     ADD       A,B                    ; Co-add current pixel to on-going summation
(8)     MOVE      B1,Y:(R0)+             ; Write D23-0 to SRAM
(9) RD_PXLS
```

Line (1) sets the hardware coadd mode to 24-bits, so only CEL0* or CEL1* will be asserted. A mask is put in line (2) so the higher order data lines D23-D16 will be zeroed out when the DSP reads the input image data lines D15-D0. The lines D24-D16 are not driven by any circuitry, are undefined and could corrupt the coaddition memory. Line (3) checks that the FIFO connected to the A/D latches is half full with 512 words or more so a whole block of data can be moved without checking flags for each image word. Image data is written to the FIFO independently of the coadder DSP by having the timing board assert the A/D conversion line and XFER which transfer data from the A/D converters to the FIFO.

Line (4) starts up a loop through the 512 FIFO locations that are now known to be not empty. Line (5) simply reads the current FIFO contents and places it in the accumulator A. Line (6) zeroes out the D24-D16 bits of the accumulator and reads the coadder memory contents of the current location pointed to by the address register R0 into the accumulator B1. Line (7) simply adds the current image counts to the coadded counts, and line (8) writes the coadded counts back to the same coadder memory location. Line (8) also increments the coadder memory address register by one to point to the next coadder pixel location.

A code segment for doing 32-bit coaddition is somewhat longer, as follows -

```
(1)     BSET      #M24_32,X:PCD          ; Select 32-bit coaddition
(2)     MOVE      #$FFFF,X1              ; Mask to zero D23-D16
(3)     MOVE      A,X:RSTSRAM            ; Initialize the Ping-Pong data pointer
(4)     JSET      #HF,X:PBD,*            ; Wait for the FIFO to be half full
```

```
(5)     DO          #512,RD_PXLS           ; Read 512 pixels = half of FIFO
(6)     MOVE        X:(R5),A1              ; Read from the A/D FIFO
(7)     AND         X1,A  Y:(R0),B1        ; D23-D16 = zero; read D23-D0 from SRAM
(8)     MOVE        Y:(R0),B2              ; Get coadded D31-D24 from SRAM
(9)     ADD         A,B                    ; Co-add current pixel to on-going summation
(a)     MOVE        B1,Y:(R0)              ; Write D23-D0 to SRAM
(b)     MOVE        B2,Y:(R0)+             ; Write D31-D24 to SRAM
RD_PXLS
```

Here bit M24_32 is set to one to select 32-bit coaddition, and the hardware pointer that selects either the low memory bank D23-D0 or the high bank D31-D24 is cleared to zero to select the low memory bank. Line (7) reads the low order data bits D23-D0 into accumulator B1 because the hardware asserted the enable line CEL0* or CEL1*. Line (8) appends the high order 8 bits to the same accumulator B by writing to B2, but now with CEH0* or CEH1* asserted because the hardware toggled these line since M24_32 was set. Accumulator B now has 32-bits of coadded image data properly placed for the add instruction in line (9), after which all 32 data bits D31-D0 are written back to the coadder memory in two instructions to the same coadder memory location.

Finally, to round out the digital end of the board a 16-bit with FIFO named CAFIFO in the software and a set of bus driver chips are used to write coadded image data to the backplane for transmission through the timing board to the host computer. The timing board reads image data the same way it does for non-coadding IR and CCD video boards by asserting the address of the desired A/D on the lines TIM-ADADR4 to TIM-ADADR2 along with the strobe signal TIM-ADSTR to signal that the address lines are valid. The PAL U45 on the coadder board decodes these lines, reads the next image data value from the FIFOs U28 and U29 and asserts them on the backplane whenever the address lines TIM-ADADR4 to TIM-ADADR2 match the jumper values JP7 to JP5. The two address lines TIM-ADADR1 and 0 are ignored to reflect the fact that the individual A/Ds cannot be accessed on the coadder board, but only the single FIFO into which all image data is written. Even so, in the timing board DSP software supplied for reading from coadder boards the command SXMIT = $00F060 for reading from four successive addresses is employed as a programmer’s aid.

Usually coadded data is not written concurrently with coadding frames, but it is possible to overlap them. The following code segment illustrates the writing process for 32-bit coadded data -

```
        JCLR        #CAHF,X:PCD,*
        DO          #256,L_XMIT
        MOVE        Y:(R3),A1       ; Get bits D23-D0 of coadded image
        MOVE        A1,X:(R6)       ; Write bits D15-D0 to CAFIFO
        MOVE        Y:(R3)+,A2      ; Get bits D31-D24 of coadded image
        REP         #16             ; Shift bits D31-D16 into position
        ASR         A
        MOVE        A1,X:(R6)       ; Write bits D31-D16 of coadded image to CAFIFO
L_XMIT
```

Here the coadder FIFO is kept half full or more so that the timing board that is always reading image data by monitoring the coadder FIFO half full flag CAHF. The image data path over the backplane is 16 bits wide, while the image data in the DSP is split into a 24-bit and 8-bit segments, so the most significant 16-bits are left shifted into position by 16 bits before writing them to the coadder FIFO. The timing board is executing code such as follows to read the coadder FIFO data and transmit them to the host computer over the fiber optics data link.

```
(1)     BSET        #WW,X:PBD            ; Set WW to 1 for 16-bit image data
(2)     MOVE        #$4000,X0            ;   = (256 x 256) / 4
(3)     MOVE        #$00F060,A           ; Read the image from the coadder board
(4)     DO          X0,L_TXMIT           ;     four pixels at a time
(5)     MOVE        A,X:(R6)             ; Write SXMIT out
(6)     MOVE        #$4A0000,X0          ; Delay for pixel transmission
(7)     MOVE        X0,X:(R6)
(8) L_TXMIT
```

Line (1) sets the hardware on the timing board serial transmitter for 16-bit long image data. Lines (2) and (3) set up the loop counter to transmit 64k image words, four every time line (5) is executed since the SXMIT = $00F060 parameter is set to transmit four pixels. Line (5) causes the timing board to automatically read four image values from the coadder board jumpered to zero, and transmit them sequentially over the fiber optic link. Lines (6) and (7) impose a delay on the loop execution to ensure that the image data has enough time to be transmitted over the link by calling the hardware delay circuit with the delay parameter $4A. If more than one coadder board is installed in the system then the SXMIT parameter would be increased to $00F0E0 for two boards, and so on.

The lines STATUS3 to 0 are connected to the DSP parallel I/O port PORT B pins and routed to the backplane where they connect to pins of the same name on the timing board DSP. These are bi-directional, programmable pins that can be used for communication between the timing and coadder boards, or between multiple coadder boards in a system. However, they are presently unused by released software.

JUMPERS

There are two sets of jumpers on each coadder board. One set is located next to the LM2990 regulator and simply connects the analog signal ground plane to the digital ground plane. The default configuration has that jumper installed. The second set of jumpers is located near the corner of the board, towards the top, next to the DIN-96 backplane connector. It is not to be confused with the OnCE connector located nearer to the DSP. It contains eight addressing jumpers, labelled with tiny silk screen letters JP5 to JP12, with JP5 nearest the edge of the board.

Jumpers JP7 to JP5 determine the address of the board when the timing board is reading image data, with JP5 being the least significant bit. When the address lines TIM-ADADR4 to 2 match the jumpers JP7 to JP5 then the next contents of the coadder output FIFOs U28 and U29 are driven onto the image data pins AD0 to AD5. Notice that the two least significant address lines TIM-ADADR0 and 1 are not used by this board. Up to eight coadder boards may be installed and addressed in a system.

The jumpers JP8 to JP5 (relabelled JP3 to JP0 when used internally on the board) are connected to the

DSP parallel I/O port C. They are currently not used, but are provided in case each DSP in a multiple coadder board system needs to be addressed separately.

Jumpers JP12 to JP9 determine the value of the switch select bits SS15-SS12 that result in the switch state bit SS0 being written to the start A/D conversion line and bit SS1 being written to transfer A/D data to the latch signal (XFER). In the DSP software these address lines are set by the VIDEO parameter which is set to zero in all supplied DSP software. All four jumper JP11-JP8 must be installed, and all coadder boards in the system will have simultaneous A/D and XFER execution.

These is single 12-bit DAC U24 that is used to generate four video offsets. It is a serial input DAC of the same type used on the non-coadding video boards and clock driver boards, with the same serial protocol for writing values to it. The board selection for writing to it is determined by the same jumpers JP7-JP5 that are used for reading coadded image data.

DSP COMMANDS

There are four implemented timing board commands that may be executed by the user that involve the coadder board. These are timing board commands, so they must be issued to the timing board, not the coadder board. commands. These are all generated by the timing board, as commands are not passed directly through the timing board to the coadder board. The four commands below are all executed by the timing board and may pass secondary commands on to the coadder board.

DCA - ’Download CoAdder’ writes the coadder DSP program from the extended SRAM memory on the timing board to the coadder’s internal DSP memory. The timing board DSP source code file ’tim.asm’ contains both the timing board application code and the complete coadder DSP code, the latter being written into the timing board SRAM memory starting at location P:$500 when a download is made to the timing board from the host computer.

LDA - ’Load CoAdder’ writes the coadder DSP program from the EEPROM memory of the timing board to the coadder board’s internal memory. The program must first have been written to the EEPROM either by externally programming the part or executing a program such as TimRomBurn.

SNC number of coadds - ’Set Number of Coadds’ takes the argument of the command and writes it to the parameter NCOADDS in the timing board code that is then used by both the timing and coadder boards to set the number of times the array is read out and its image data coadded before being transmitted to the host computer.

TCM - ’Test Coadder Memory’ tests the large SRAM coadder memory by writing and then reading patterns of numbers. It takes a few seconds to execute and returns a ’DON’ to the timing board if it executed correctly. The address of the error is returned if it failed.

MULTIPLE COADDER BOARDS

The coadder board and the supporting controller housing, power supply and timing board are all designed to accommodate up to eight coadder boards. Each of the eight coadder boards must have its address jumpers JP7 to JP5 set to different values, and it is best to have the switch state jumpers JP11 to JP8 set to the same value of zero (all jumpers installed). In this configuration all the coadder boards will execute the start A/D and XFER command simultaneously, and have their image values read sequentially.

Serial communication between the timing and coadder boards occurs over both the SCI (serial communication interface) and SSI (synchronous serial interface). The SCI is used only for downloading DSP code from the timing board memory to the internal coadder DSP memory. All coadder boards installed in the system are written to simultaneously by the timing board, and no addressing or board selection takes place. The SSI is used for sending commands from the timing board to the coadder boards, and for the timing board receiving replies in turn. The same protocol used for other serial communication in the controller is used here, with the header, three character ASCII command and optional arguments being implemented. However, a peculiarity of the current implementation is that all coadder boards execute the same DSP code at the same time, and this applies to serial command processing as well. The timing board sends the same serial command to all coadder boards, all coadder boards process them at the same time, and all coadder boards drive the same serial transmit pin with the same replies to the timing board at the same time.

REVISION HISTORY

The first revision of the coadder board is Rev. 3B dated June 16, 1998. It uses a custom Datel A/D (part number 30372), adjustable low overhead regulators, and a DSP reset line derived from the utility board reset line TIM-U-RST. Before downloading DSP code to its internal memory the coadder DSP is be reset by the timing board to invoke the coadder downloading program. Using the utility board reset line for this has the unfortunate side effect of also turning off the analog power by resetting the power control board. Users of the Rev. 3B board must explicitly turn the power back on after downloading software to the coadder DSP with either the LDA or DCA commands.

The newer board is Rev. 4B, dated March 7, 2000, and uses fixed low dropout regulators, the standard A/D part ADS937, and a DSP reset line derived from an auxiliary control line named TIM-V-AUX1. The more benign reset line does not turn off analog power to the system when coadder DSP software is downloaded.

In the timing board code for systems using a coadder there is a coadder reset routine that is called before code is to written to the coadder DSP. The parameter that points to the line that needs to be reset is different for the two revisions, as follows -

```
; Reset the coadder board to force it to re-boot
RST_CA      EQU      3                        ; Rev. 3B value
RST_CA      EQU      5                        ; Rev. 4B value
CA_RST      BCLR     #RST_CA,X:<LATCH
            MOVEP    X:LATCH,Y:WRLATCH        ; Clear reset coadder bit
            DO       #600,*+3                 ; Delay by 100 microseconds
            NOP
            BSET     #RST_CA,X:<LATCH
            MOVEP    X:LATCH,Y:WRLATCH        ; Set reset coadder bit
            RTS
```