TIMING BOARD

This chapter describes the timing board for the Generation II systems, Revision 4, dated 2/2/97. The timing board serves two main functions - generating digital timing signals to control other, primarily analog, circuit boards that connect to the detector array, and to serve as a communication hub between the controller and the host computer interface board. The central element of the timing board is a Motorola digital signal processor, part number DSP56002FC66, running at a clock speed of 50 MHz that executes instructions in 40 nanosec. Analog circuit functions are controlled at this minimum time scale of 40 nanosec. Timing is accomplished by having the DSP simply write 24-bit data words from its internal memory to its external bus, wherein external circuitry on the timing and analog boards decodes the word to control such functions as delay counters, analog switches, data latches and analog-to-digital (A/D) converters. Communication with the host computer is done over a duplex fiber optic link that receives data from the host computer at a speed 4 Mbits per second and transmits to it at 50 Mbits per second. A block diagram of the board is shown in Fig. 1.

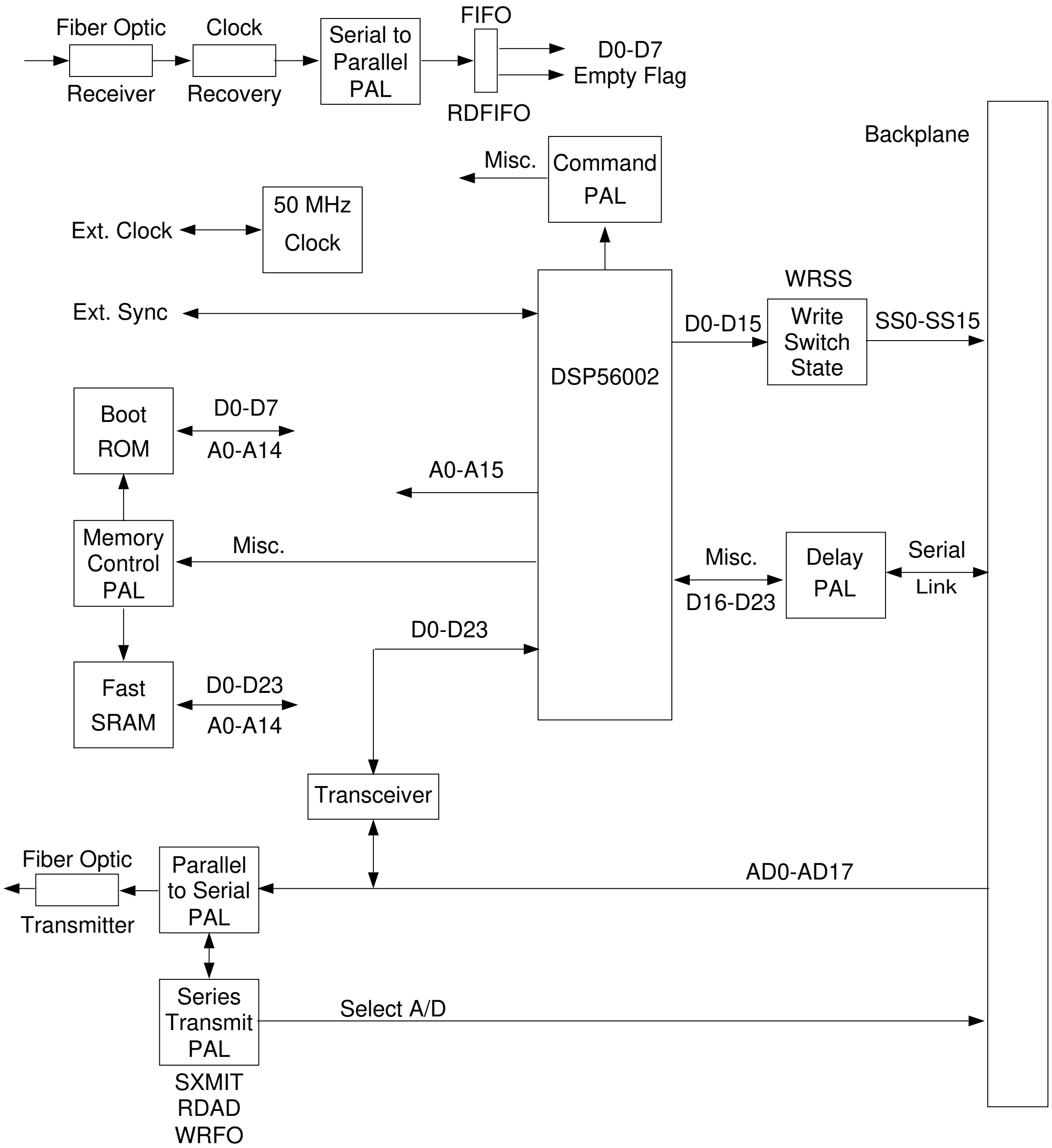

Fig. 3-1: Timing Board Block Diagram

Referring to Fig. 1 and the accompanying timing board circuit schematic, the fiber optic receiver U5 converts optical data from the host computer to electrical signals, the clock recovery chip U7 extracts a clock that is synchronous with the incoming data, the programmable array logic (PAL) device U2 converts the serial data stream into 8-bit wide parallel data, and the byte-wide first-in-first-out (FIFO) memory U28 device holds the data until they are read by the DSP a byte at a time with a RDFIFO command. The jumper block JP4-JP10 is used to select the speed range of the incoming data, and is normally set to 4 MHz for compatability with the host computer interface boards. An LED is installed near the fiber optic components to be lit when the circuit is receiving correct data from the host computer interface board, though a jumper can be removed to disable the LED if its too close to sensitive optical sensors. The DSP polls the Empty Flag line of the FIFO prior to reading its contents, eliminating the need for an interrupt service routine. The FIFO depth of 1k bytes allows many commands to be stored in the FIFO awaiting service, and eliminates the need for the delay between the first and second words of commands that was needed in first generation timing boards that were serviced by interrupts. The default speed is 4 Mbits per second for compatibility with the host computer interface boards, although the jumpers described below allow selection of speeds up to 50 Mbits per second.

The incoming serial data contains 33 bits per word. There is a start bit, an 8-bit preamble, and then 24-bits of data, most significant bits first. If the preamble byte contains an $AC then the 24 data bits are passed on to the FIFO a byte at a time. If the preamble contains a $53 then the DSP is reset by asserting its RESET line and the data bits are thrown away. If neither of these bytes is in the preamble then the word is simply thrown away by the serial receiver. For compatability with Generation I host computer interface boards the insertion of a jumper in JP14 will enable the servicing of its simpler 24-bit command format.

When the DSP exits the reset state it loads the boot program into its internal memory from external program memory starting at address $C000. The memory control PAL asserts the ROM access signal whenever there is a program access with bit A15 is set, and the bit A14 selects the boot program from the upper half of a 32k x 8-bit Read-Only-Memory (ROM) U3. The boot code configures the DSP, manages communication and command processing with the host computer and the utility board, and supports several elementary commands (read memory, write memory, test data link and load application). The lower half of the ROM contains application code that can be loaded after booting, allowing the user to load DSP memory from either ROM (loading an application) or from the host computer (downloading) by writing the program to DSP memory over the fiber optic data link. The ROM part can be either electrically erasable (EEPROM) or UV erasable (UVPROM) memory, of generic part numbers 28C256 or 27C256 respectively, contained in 28-pin, 600-mil DIP packages and socketed on the board. The jumpers JP2 and JP3 select between the two type of ROM, and enable the EEPROMs to be write protected if desired. The ROM read by the DSP with a move program memory instruction over the address range $8000 to $FFFF. Both read and writes are allowed, but writes will only occur if the jumper JP3 is installed and the EEPROM timing rules are followed.Programming the ROM is further discussed in the DSP software chapter. EEPROMs can be written to while they're installed in the timing board with the write memory instruction, and care should be exercised by the user to not fatally overwrite the boot program.

Timing is accomplished by having the DSP write a 24-bit data value to the WRSS "write switch state" address. The most significant 8 bits select a delay function in the delay PAL that delays the execution of the next DSP instruction by a programmable number of clock cycles, according to the scheme -

If D23 = 0 then delay by the number D16-D22 of 20 nanosec clock cycles.
If D23 = 1 then delay by the number D16-D22 of 160 nanosec clock cycles.

The most significant bit D23 is a fine/coarse adjustment. The actual execution time of the instruction will be the number of clock cycles chosen times either 20 or 160 nanosec plus 80 nanosec, the normal 40 nanosec execution time plus a 40 nanosec time for using the delay function. This gives a range of 80 nanosec to 20.40 microsec delay time. Delays of more than that can be implemented by repeating the same instruction. During the delay time the DSP is idle - it cannot process anything. Summarizing these rules, the following times between waveforms are obtained with the indicated delay numbers -

40 nanosec	delay = 0	40 nanosec for instruction
60 nanosec	not possible	
80 nanosec	delay = 0, repeat instruction	40 x 2 nanosec for two instructions
100 nanosec	delay = 1	80 nanosec for instruction + 20 nsec delay
120 nanosec	delay = 2	80 nanosec for instruction + 40 nsec delay
2.620 microsec	delay = $7f	80 nanosec for instruction + 127x20 nsec delay
40 nanosec	delay = $80	40 nanosec for instruction + 0 nsec delay
240 nanosec	delay = $81	80 nanosec for instruction + 160 nsec delay
400 nanosec	delay = $82	80 nanosec for instruction + 2x160 nsec delay
20.40 microsec	delay = $ff	80 nanosec for instruction + 127x160 nsec delay

To generate timing control for the analog boards in the controller, the DSP data lines D0-D15 are driven by the high current buffers U18 and U21 onto the backplane, renamed as the switch state signals SS0-SS15. The strobe signal WRSS is generated if a DSP write instruction is made to the address X:$FF80=WRSS, and is used by the analog boards to latch the signals SS0-SS15. The analog boards generally use the bits SS12-SS15 for board selection and SS0-SS11 to control switches or other timing circuits.

Fast static random access memory (SRAM) is implemented in a 32k x 24-bit format by U1, U4 and U6 to extend the limited internal DSP memory. The SRAM is split between the three address spaces of the DSP (program, X: and Y: data memory). The 32k word memory space is allocated by the PAL U9 to occupy the lower address space of each type, according to the following -

P: memory is 512 words internal (P:$0000-01ff) and 8k words external (P:$0200 to $1FFF)
X: memory is 256 words internal (X:$0000-00ff) and 8k words external (X:$0100 to $1FFF)
Y: memory is 256 words internal (Y:$0000-00ff) and 16k words external (Y:$0100 to $3FFF)

Normally, addresses in the range P:$0000-$01FF, X:$0000-$00FF and Y:$0000-$00FF access internal DSP memory, although external SRAM still exists there. The DSP operating mode register can be programmed to disable internal DSP memory access, in which case the external SRAM memory will be accessed instead.

The external SRAM memory is fast enough to support no wait state access. The only execution time degradation will occur if more than one external bus access is required per instruction, in which case 40 nanosec of instruction execution time are needed for each bus access. The following instruction is used to write timing waveforms to external analog boards over the bus -

```
MOVE    A,X:(R6)    Y:(R0)+,A
```

In one instruction it write the contents of the accumulator A to the memory mapped location in R6 (here its mapped to WRSS) and reads the next waveform entry from the from the waveform table indexed by the address register R0 into the accumulator. If this instruction were located in external program memory, and the waveform table were in external Y: memory, then the instruction would require 120 nanosec to execute since it requires three external bus accesses - 40 nanosec to access the instruction from program memory, 40 nanosec to access the Y: memory contents from external memory and 40 nanosec to write the accumulator contents to WRSS. At the other extreme, the instruction in the timing board code that sets video processor gain -

```
MOVE    X:(R4)+,A
```

will execute with no wait states even if the instruction is located in external program memory because the register R4 points to internal DSP memory space. In the timing board code supplied with the board use of external program memory is made to locate infrequently and non-time critical code for initializing and tweaking the system. The time critical command processing and timing waveform generation is located in internal program space, as are the timing waveforms. However, the external Y: memory space is made quite large to allow for extensive clocking waveforms to be stored there, which will take 80 nanosec per instruction to execute if the clocking program instruction is located in internal DSP memory.

The synchronous serial interface (SSI) of the DSP is used to write numbers to digital-to-analog converters (DACs) and related circuits on the analog boards that don't need to be rapidly updated. The serial clock, data and frame sync signals from the DSP are passed through the PAL U14 where the frame sync is merged with the data to give a start bit following the same protocol as data is passed between the timing board and the host computer, all with the clock continuously running. The serial data is sent as 24-bit words, where PAL circuits decode the most significant 4-bits as a board select and decode the remaining bits depending on the context. Twelve-bit digital-to-analog converters (DACs) take 16-bits as a serial data stream input, and use 12 of them to set the output voltage and the remaining ones to select one of several circuits to be written to. The same SSI circuit is used to communicate between the timing and utility board. Separate pins on the backplane are used for communicating with the analog boards and the utility board, and the DSP with the delay PAL steers the serial data to one or the other type of board with the signal H0. When transmitting to the analog boards the serial clock TIM-A-SCK runs continuously for a short time, whereas bidirectional communication with the utility board is with a gated clock TIM-U-SCK to minimize interference with the analog board functions.

Analog-to-digital numbers from the video processor board representing image pixel counts can be read over the backplane into the DSP by reading from one of 32 memory mapped locations over the range of Y:$FFA0 to Y:$FFBF, known collectively as RDAD. The data is asserted by the video processor on the backplane lines AD0-AD17 and read by the timing board through the transceivers U16, U19 and U22 onto the DSP data lines D0-D17, with D18-D23 indeterminate. The data is then written to the host computer by writing to the WRFO memory location, causing the data to go in the reverse direction through the transceivers and to the parallel-to-serial PAL circuit U17 that generates a serial data stream for the fiber optic transmitter U23. Data can be written in either 24-bit mode which will get expanded into 32-bit mode by the PAL or in 16-bit image mode, determined by the word width bit WW. The capability of generating only 24-bit data exists for compatability with the earlier Gen I systems by installing a jumper on JP14 to clear the word length line WL.

To speed the transfer of images an image data path is implemented that bypasses the DSP so image transfer to the host computer can take place at the same time as timing functions. In this mode the series transmit PAL U12 selects a series of A/D values to transfer, asserting the select A/D lines that cause data to be placed on the AD0-AD17 lines by the video processor for the serial transmitter to send to the host computer. It is enabled by writing to the memory mapped location SXMIT.

A diagnostic port is implemented to help the developer debug DSP code. While the timing board is operating at full speed, the port allows the user to examine memory and register values, install breakpoints, download code and a host of other things. It implements the OnCE (for On Chip Emulation) protocol developed by Motorola and simply consists of a 14-pin connector mounted close to the DSP that the user can connect to a cable and special hardware that runs to a host computer. Low cost support hardware and software is available from Motorola for operating the port from Suns, PCs or MACs via a host computer interface board and a command converter that connects the interface board to the OnCE port ribbon cable.

A master/slave capability exists wherein two or more controllers can be connected to the same mosaic of arrays and operated simultaneously. This may be useful for operating large numbers of readouts from a single mosaic as a way of limiting the number of analog boards that are placed on the same backplane to a manageable number, and for increasing the data transmission rate from the controller to the host computer by having more than one fiber optic data path. The main difficulty in operating a single mosaic from two or more controllers is to synchronize the readout enough to avoid beating effects from unsynchronized readouts. The timing board has provision for operation as a master or as a slave, where one master and one or more slave timing boards are established in a multiple controller system. The master timing board generates the 50 MHz system clock and transmits it to all slaves to assure that all DSP are synchronized at the clock level. To synchronize instruction execution all timing boards are loaded with identical readout code, and then the master timing board generates a SYNC signal that is connected to all slave boards, as well as to itself, to generate an interrupt that causes all the boards to execute a long interrupt service routine containing identical readout code. The buffer chip U26 is used either to transmit the clock and sync signals from the master to the slaves or to receive them on slave boards, with jumpers explained below used to configure a board as either a master or a slave. Two small coax SMB push-on connectors are used to route the clock and sync signals between the boards in the system. Synchronization of the clocks has been demonstrated to within a phase delay of about 10 nanosec, and synchronization of the readout to within an instruction and a half (60 nanosec) has been demonstrated.

A watchdog timer and reset circuit U24 is installed to reset the DSP under several conditions. The DSP will be reset whenever the on-board push button S1 is depressed. The DSP will be reset if the +5volts digital power line crosses below 4.75 volts, with the RESET line de-asserted only if it rises above that threshold. A watchdog timer functions whereby the timing board must write to the memory location RSTWDT every 62.5 milliseconds or the circuit will reset the DSP. This watchdog timer function can be disabled by simply installing the jumper as described below, which generates a hardware level reset watchdog timer signal from the clock. The DSP can also be reset from external sources, by setting the backplane signal EXT-T-RST low.

A latch circuit U25 is installed to assert miscellaneous signals for system support. The latch is written to a byte at a time by writing to the address WRLATCH, and the current contents of the latch are stored in the DSP code in a variable named LATCH. Finally, the address decode PAL U8 decodes DSP addresses and control line to generate the control signals described above for routing throughout the board.

Hardware Preparation

The principal components of the timing board are shown in Fig. 2. There are many jumpers on the board for selecting operating modes which are described below and in Fig. 3. Much of the text in the figure is self-explanatory, so only a few additional comments are included here. Some of this discussion repeats the section above, and is included in abbreviated fashion for easy reference.

A radiating LED between the fiber optic transmitter and receiver indicates that a valid signal is being received from the host computer interface board, and acts in the same way as the LED installed there. A nearby jumper can be removed to disable the LED when the controller is in use in a low light level environment.

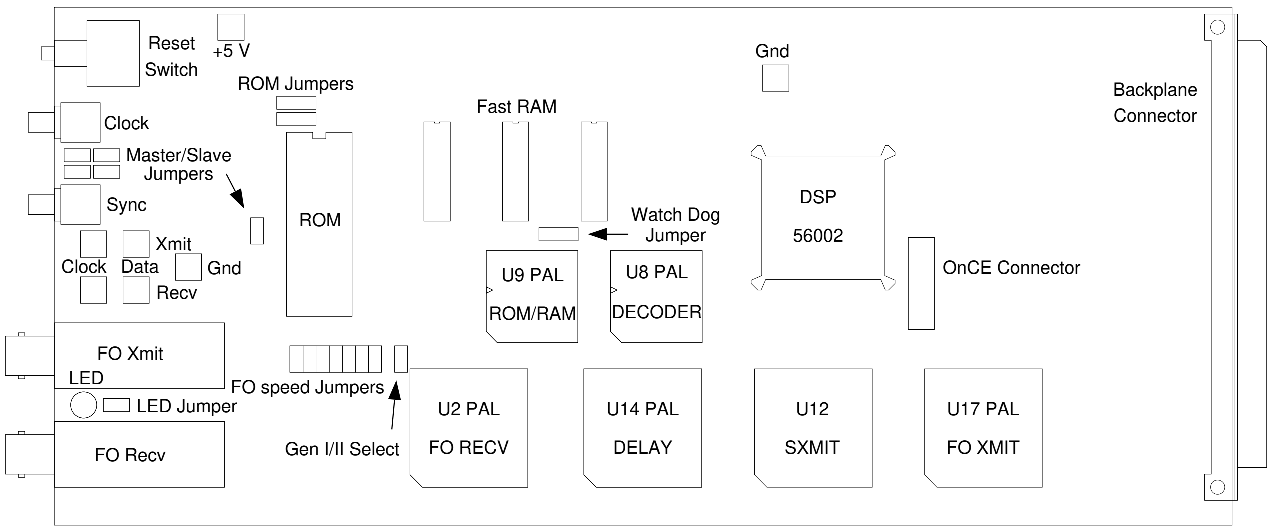

Fig. 3-2: Parts layout of Rev. 4B timing board

The DSP boots its program from on-board ROM, which is socketed for easy external programming. The jumpers JP2 and JP3 just above the ROM socket select EEPROM or UVPROM. If EEPROM is selected JP3 can be used to write enable the part so it can be written to in its socket by the DSP. The

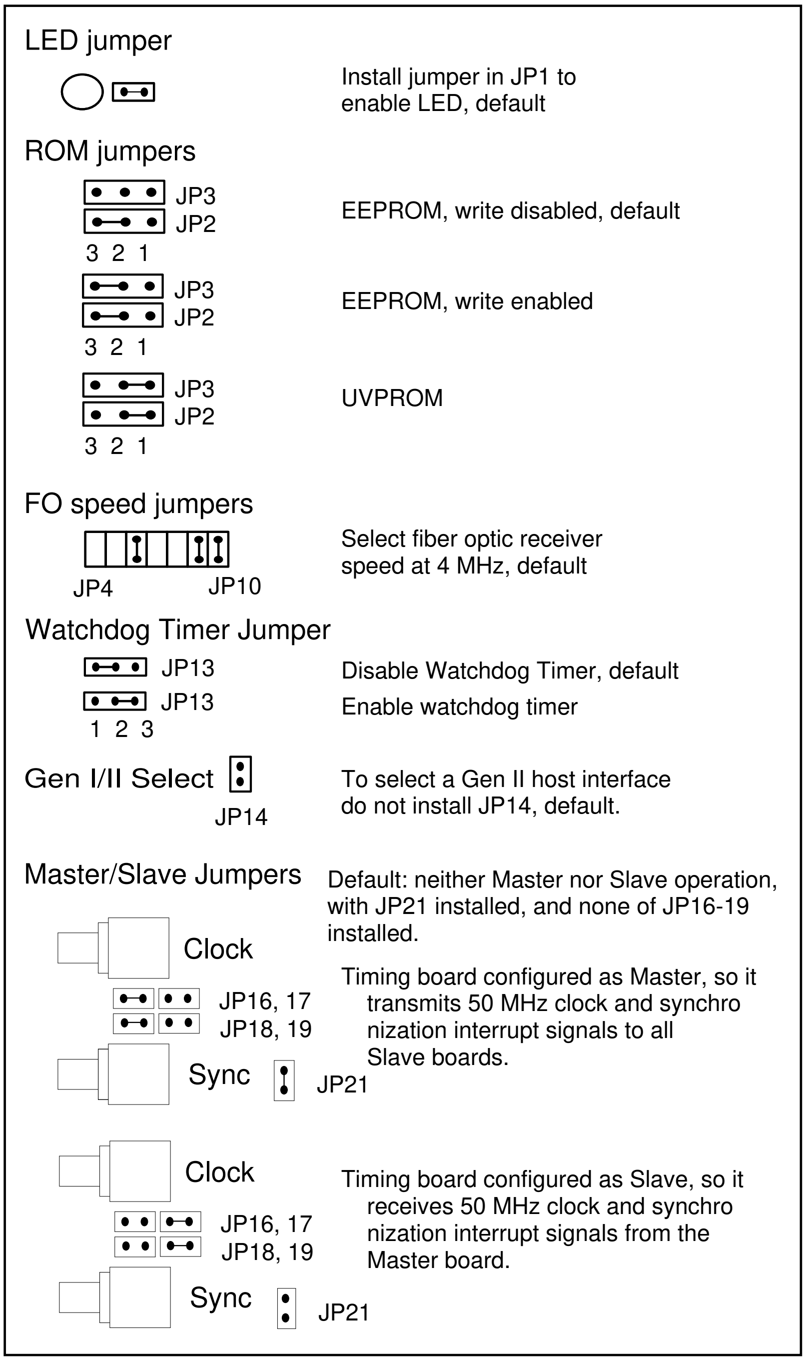

Fig. 3-3: Jumper map for timing board, Rev. 4B

intention of providing both options is to allow the user to install UVPROMs once the instrument is installed in the field since they are less volatile than EEPROMs.

The bank of jumpers JP4-JP10 selects the speed of the fiber optic data being received, and is normally set to 4 MHz as shown. There is a watchdog timer circuit on the board that can be used to reset the DSP if the DSP program does not execute properly. For compatibility with both Generation I host computer interface boards and the newer Gen II boards the jumpers JP14 is included. It configures the serial receiver and transmitter to process 24-bit data words for Gen I systems and 16- and 32-bit words for Gen II systems.

The jumpers JP16-19 and 21 are used to allow operation of two or more timing boards in a system wherein the master timing board generates the 50 MHz system clock and an instruction synchronization signal for itself and for the slave boards so all the DSPs in the system are synchronized. For single timing board operation only JP21 should be installed.

The default jumper settings are shown in the figure, and are summarized here - LED activated, EEPROM write disabled, 4 MHz serial input data speed, watchdog timer disabled, Gen II host computer connection, and neither master nor slave operation. While all operating modes of these jumpers are believed to function properly, the user should exercise a certain degree of caution when selecting non-default jumper settings since the degree of testing and reliability for these modes is not as high as for the default mode.

DSP instruction summary

The following table gives the memory map for the DSP instructions. The mnemonics used in the discussion above are used, which generally follow the ones used in the DSP source code files.

Read a byte from the FIFO	MOVEP	Y:$FFC0,A	RDFIFO
Write 32-bit word to fiber optic transmitter	BCLR	#1,X:PBD	
(commands and replies)	MOVEP	A,Y:$FFC0	WRFO
Write 16-bit word to fiber optic transmitter	BSET	#1,X:PBD	
(image data)	MOVEP	A,Y:$FFC0	WRFO
Write serial data word to clock driver and video processor boards	MOVEP	A,X:$FFEF	
Write serial data word to utility board	BCLR	#0,X:PBD	
	MOVEP	A,X:$FFEF	SSITX
Wrtte serial data word to an analog board	BSET	#0,X:PBD	
	MOVEP	A,X:$FFEF	SSITX
Read serial data word from utility board	MOVEP	X:$FFEF,A	SSIRX
Write word to switch state register	MOVE	A,X:$FF80	WRSS
Read A/D datum from A/D number 0 to $1F	MOVE	Y:$FF##,A	RDAD
		## = $A0 to $BF	
Write to latch on timing board	MOVEP	A,Y:$FFC1	WRLATCH
Reset timing board watchdog timer	MOVE	A,P:$6000	RSTWDT
Initiate series transmission of A/D data	MOVE	A,X:$FF80	SXMIT

The contents of the 24-bit accumulator A for the series transmit A/D data command are defined as follows -

A = %00000000111100eeeeebbbbb,

The five bit field bbbbb specifies the beginning number of the A/D to be read and transmitted, and the five bit field eeeee specifies the ending number. From one to 32 A/D values may be transmitted by a single instruction. Common values for the accumulator are shown -

A = $00F000	transmit A/D #0 only	A = $00F020	transmit A/Ds #0 and #1
A = $00F021	transmit A/D #1 only	A = $00F060	transmit A/Ds #0 to #3

DSP Parallel Port Definition

Port B on the DSP is configure as a parallel port wherein its 15 bits that are wired to a miscellany of functions throughout the board. Many of these functions have been discussed above, and some will be discussed following this summary table.

Bit #	DSP Name	Board Name	Direction	Comment
0	H0	H0	Output	Direct SSI to utility (=1) or analog (=0)
1	H1	WW	Output	Fiber optic word width 32 (=0) or 16(=1)
2	H2	LVEN	Either	Low voltage enable, low true

3	H3	HVEN	Either	High voltage enable, low true
4	H4	STATUS0	Either	General use - backplane A25
5	H5	STATUS1	Either	General use - backplane A26
6	H6	STATUS2	Either	General use - backplane A27
7	H7	STATUS3	Either	General use - backplane A28
8	HA0	AUX1	Either	not used
9	HA1	EF*	Input	FIFO Empty flag, low true
10	HA2	FD15	Output	Force D15 to this value if FMODE = 1
11	HR/W	SYNC	Either	Master/Slave sync bit
12	HEN*	FMODE	Output	Forced D15 mode if =1
13	HREQ*	PWRST	Either	Reset power board, low true
14	HACK*	AUX2	Either	not used

The power control board lines LVEN (low voltage enable, to turn on +/- 6.5 and +/- 16.5 volts supplies to the backplane), HVEN (high voltage enable, to turn on the +36 volts supply) and PWRST (turn off all supplies) are programmed as inputs in controllers that have a utility board installed, since the utility board controls these lines. In controllers without a utility board the timing board has these lines programmed as outputs and directs the power control board. The four lines STATUS0 to 3 are wired as general purpose communication lines between the DSP and the outside world and may be used as the user desires. The lines AUX1 and AUX2 are wired from the DSP to PALs and are not presently used for anything and are wired for future expansion. The signal EF* signals that the fiber optic receiver PAL has a command that the DSP should service. The SYNC bit is either an input or output that is used to synchrnize the readout of two or more controllers, and can be used for general purpose I/O between the controller and the outside world since it has a high current buffer and accessible SMB connector on the front of the board.

The signals FD15 and FMODE were developed to support adaptive optics applications that send many images continously to the host computer, pointing to the need for a frame sync signal to alert the host computer that the beginning of an image transmission is at hand so synchronization would not be lost between the host computer and the timing board. Detail of this implementation are available in the file AOpals.txt.

Implemented Commands

The following commands are implemented in released versions of the DSP software. The commands discussed in the software chapter (RDM, WRM, LDA, and TDL) are also implemented. The boot program is contained in a file named "timboot.asm".

LDA number - "Load Application". This is a boot command also described elsewhere that loads application code from EEPROM on the timing board into the DSP and fast static RAM memory. On the timing board the application number may be between 0 and three, allowing four applications to be strored in EEPROM at the same time. $500 words are allocated for each application in EEPROM to contain P: program memory, X: command table entries and Y: waveform table entries. The maximum memory sizes allowed for the applications is listed, with APL_LEN being the length of the application in DSP P: memory, typically 256 words -

P: $200 + APL_LEN words, typically 768 words.
Y: $300 - APL_LEN - $40 words, typically 448 words.

IDL - "Start Idling". Put clocks in the readout sequence, but don't transfer any data, to keep the CCD from building up charge. Once the idle mode is entered by executing this command the camera will idle after executing any other command, except stop, which exits the idle mode.

STP - "Stop Idling". Exit idle mode and leave the clocks not clocking.

SBV - "Set Bias Voltages". This will write the voltage codes in the appropriate Y: data memory area to the video processor DC bias and clock driver DACs.

RDC - "Read CCD". Immediately read out the CCD. The entire image will be read out according to the constants contained in the following memory locations:

Y:1 Number of pixels per line.
Y:2 Number of lines.
Y:3 Number of pixels to clear from the serial shift register before image readout. This will typically be five times the number of pixels in the serial shift register
Y:4 Number of lines to clear before exposing.

CLR - "Clear". This clears the image from the CCD by executing Y:4 parallel clock shifts. It should be done before an exposure.

SGN gain speed - "Set Gain". The gains of all the video processors will be set to the value of the gain argument. The integrator speed will be set by the speed argument by selecting whether the large capacitor is inserted in the integrator feedback.

gain = 1, 2 5 or 10 to select the gain of the switchable gain stage between the values of 1.0, 2.0, 4.75 and 9.5.

speed = 0 for slow integrator speed
= 1 for fast integrator speed

WRC number - "Write Control Word". The number will simply be written to the analog boards over the serial link, giving the user a low-level command.

SDC mode - "Set DC". The video processor preamp output can be directly coupled to the A/D converter for measuring its value as an aid to setting the input offset voltage.

mode = 1 for DC diagnostic mode
mode = 0 for normal readout mode

SBN #board #DAC type value - "Set Bias Number". A specified DAC on a specified analog board of the specified type can be set to the 12-bit value.

#board = 0 to 14 to select the numbered analog board, as determined by its jumpers.

#DAC = 0 to 15 for the video processor board

= 0 to 47 for the clock driver board

type = 'VID' to set the video processor board

= 'CLK' to set the clock driver board

value = 12-bit value that gets written to the indicated DAC

SMX #clk_board #MUX1 #MUX2 - "Set MUX". There are two analog multiplexers on the clock driver board can select from each of 24 clock driver outputs to be switched onto the diagnostic SMB connectors.

#clk_board = 0 to 15 to select the desired clock driver board

#MUX1 = 0 to 23 to select which clock driver output gets connected to the first SMB connector.

#MUX2 = 0 to 23 to select which clock driver output gets connected to the first SMB connector.

ABR - "Abort". Stop reading out the CCD immediately.

CRD - "Continue Read". Keep reading the CCD from where it was aborted.

RAD - A do-nothing command implemented for compatibility with Gen I systems where it recalibrated the A/D converters.

LGN - This will put the video processors to gain = 1 and fast readout

HGN - This will put the video processors to gain = 2 and fast readout

These commands all generate a DON reply to the board that issued the command, except for the readout command RDC that only generates pixel data that is sent along the fast fiber optic link.

Application Program

The above command list is for generic timing programs. Application programs as they get developed will be made available, and are described below.

"timEEV39.asm"

This does single readout in the simplest possible manner - no binning, no sub-images - of a four readout CCD, the EEV39 chip, manufactured for wavefront sensing in adaptive optics system. The parallel clocks are set up for full MPP (multiple phase pinned) operation wherein all parallel clocks are set to low voltages so they are inverted, both during image integration and readout as well. This reduces dark current by a significant amount. The readout code has been optimized for speed, as the pixel time is 2.56 microsec per pixel with 2.0 microsec of that time being devoted to signal integrations. It is implemented with the series transmit command SXMIT being set to transmit four readouts. The commands ABR and CRD have not been implemented.