1030-M-002, Version 2.07
December 1996

Standard Firmware Manual for the DCR-1030 VME64 Ampex DCRsi Interface Board

Table of Contents

List of Figures

List of Tables

1 Overview

The DCR-1030 Standard Firmware (1030 SF) implements a simple, efficient control interface for the DCR-1030, allowing a VME-based host processor to issue high-level commands for recording and playing back tape cartridges on a Ampex Digital Cartridge Recorder/Reproducer models DCRsi 240, DCRsi 107, and DCRsi Classic. The interface to a VME-based host is based on a set of data structures in VME accessible memory. This memory-based interface is host operating system independent, thereby avoiding (in most cases) unnecessary operating system overhead and eliminating the need for special device drivers.

The DCRsi reference in this manual will apply to the DCRsi 240, DCRsi 107, and DCRsi Classic machines unless otherwise noted.

The 1030 SF includes the following features:

1. Command Processing, including:
   a. Pass-through commands and acknowledges between a VME-Host and the Ampex DCRsi control interface.
   b. Macro commands including Initialize, Record, Playback, and Stop.
2. Wide-Band Data Processing, including:
   a. DMA transfer of data between host memory (VME, VME64, VSB, VME RACEway singles accessible) and the Ampex DCRsi data ports.
   b. High throughput processing (50 MBytes/second from VME64 memory to on-board FIFO buffers, 35 and 26 MBytes/second for VSB and VME respectively)
   c. Low-overhead buffer management (once running, a VME-based host performs a single VME write for each new data buffer).
   d. User programmable notification after the completion of each data buffer (VME interrupt, mailbox access, and/or memory flag update).

The 1030 SF is EPROM based software running in a VxWorks environment on the DCR-1030 (figure 1). A VME-based host issues commands to the 1030 SF by writing to VME accessible data structures in DCR-1030 local RAM. Record and playback operations are managed via a set of Buffer Access Blocks (BABs) in VME accessible memory. Each BAB is a buffer description which provides a pointer to and description of one buffer in VME, VSB, or RACEway accessible memory. Each buffer is individually defined in the BAB and can be up to 2 GBytes in size. BABs are organized as a circular queue (each BAB contains a pointer to the next, with the last BAB pointing to the first). BAB head and tail pointers indicate the next available BAB and the next BAB to be processed. As buffers are available for record or playback operation, the Host (or other VME processor) fills the BAB and informs the 1030 SF by updating the BAB head pointer. The 1030 SF determines how many buffers are available by subtracting the tail pointer from the

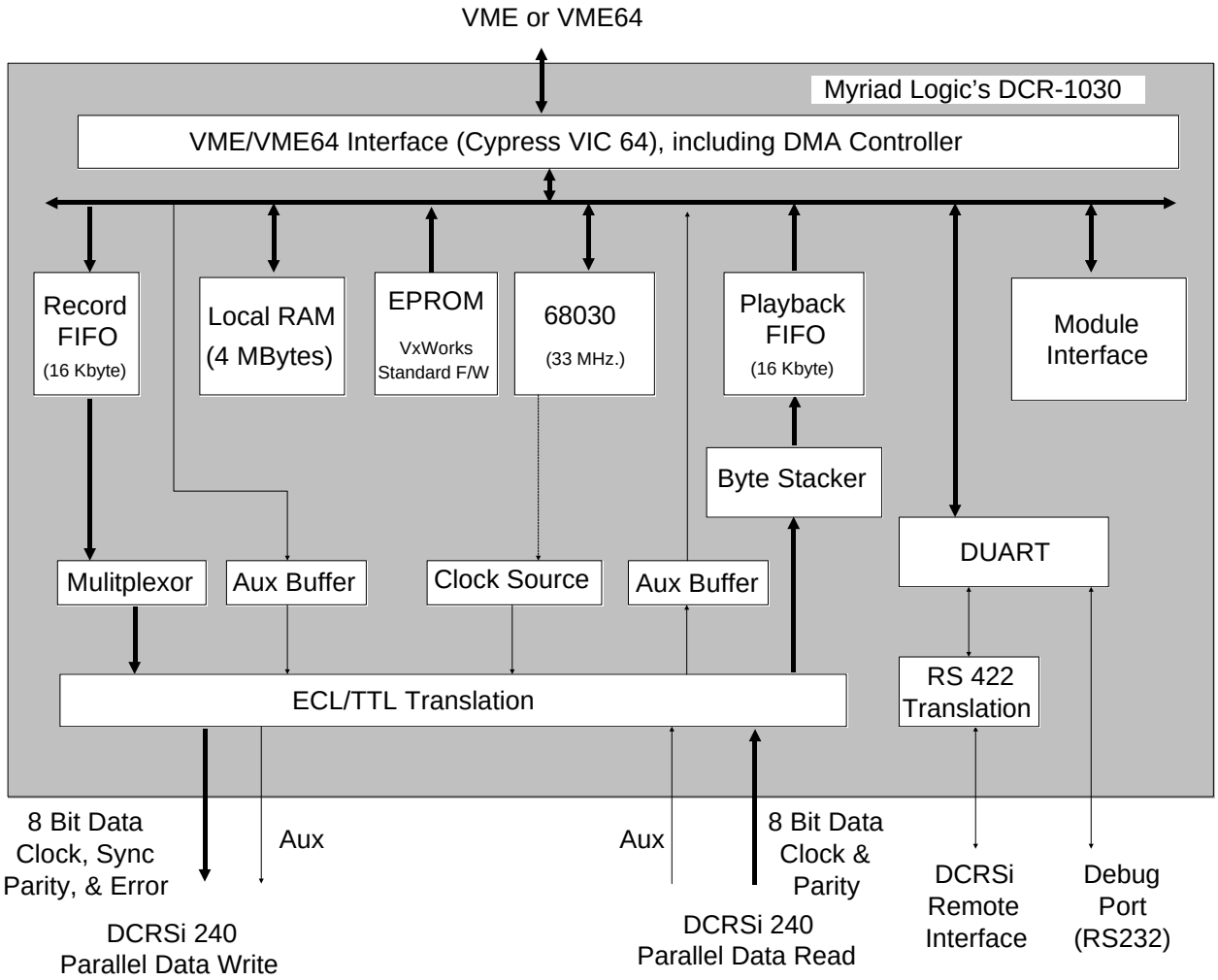

Figure 1 - DCR-1030 Hardware Block Diagram

head pointer (modulo number of BABs). When output to/input from the Ampex DCRsi is required (as defined by the Record or Playback macro command), the 1030 SF reads the appropriate BAB and initiates a DMA between the VME, VSB, or RACEway buffer and the Ampex DCRsi. Upon completion of each DMA, a flag is written in the BAB, the BAB tail pointer is updated, and an optional user signal is provided (the signal can be a Host mailbox access, a VME interrupt, both or neither).

Since all command, record, and playback operations are implemented via the VME accessible structures, specific device drivers for the Myriad DCR-1030 (with the 1030 SF) are not required. Any Host operating system which supports a VME memory driver can perform all control, record and playback functions. Additionally, the BAB structures are designed for flexibility and minimal Host overhead. The BABs can be initialized with pointers and buffer sizes pre-pass so that the only Host overhead required for each buffer is a single DCR-1030 head pointer update.

The following paragraphs describe operation of the 1030 SF and define the data structures required to interface to the 1030 SF. References for additional information on the Ampex DCRsi and Myriad Logic DCR-1030 are also provided.

2 References

The following documents contain important information regarding use and operation of the Ampex DCRsi 240, Ampex DCRsi 107, and the Myriad Logic DCR-1030:

1. "Ampex Digital Instrumentation Recorder DCRsi 240 Users Manual" available from Ampex Corporation
2. "Ampex Digital Instrumentation Recorder DCRsi 107 Users Manual" available from Ampex Corporation
3. "Myriad Logic DCR-1030 User's Manual" available from Myriad Logic, Inc., 1109 Spring Street, Silver Spring, MD 20910, (301) 588-0604.

Reference 2(a) provides detailed formats for all Ampex DCRsi 240 commands and responses. General operation of the recorder is also discussed.

Reference 2(b) provides detailed formats for all Ampex DCRsi 107 commands and responses. General operation of the recorder is also discussed.

Reference 2(c) provides installation and maintenance information regarding the DCR-1030, including options for setting the VME base address of the DCR-1030.

3 Operation

Operation of the 1030 SF is directed by three concurrent tasks operating in a VxWorks environment on the DCR-1030 68030 processor (Figure 2). The Command Task processes commands from a VME-based Host CPU, the Transfer Task manages either a record or playback session and the DCRsi Response Task handles control input from the Ampex DCRsi.

The Command, Transfer and Response tasks operate independently. Therefore, during record/playback operations a VME-Host can issue commands. Normally, commanding during record or playback operations consists of either issuing a stop macro or requesting recorder status.

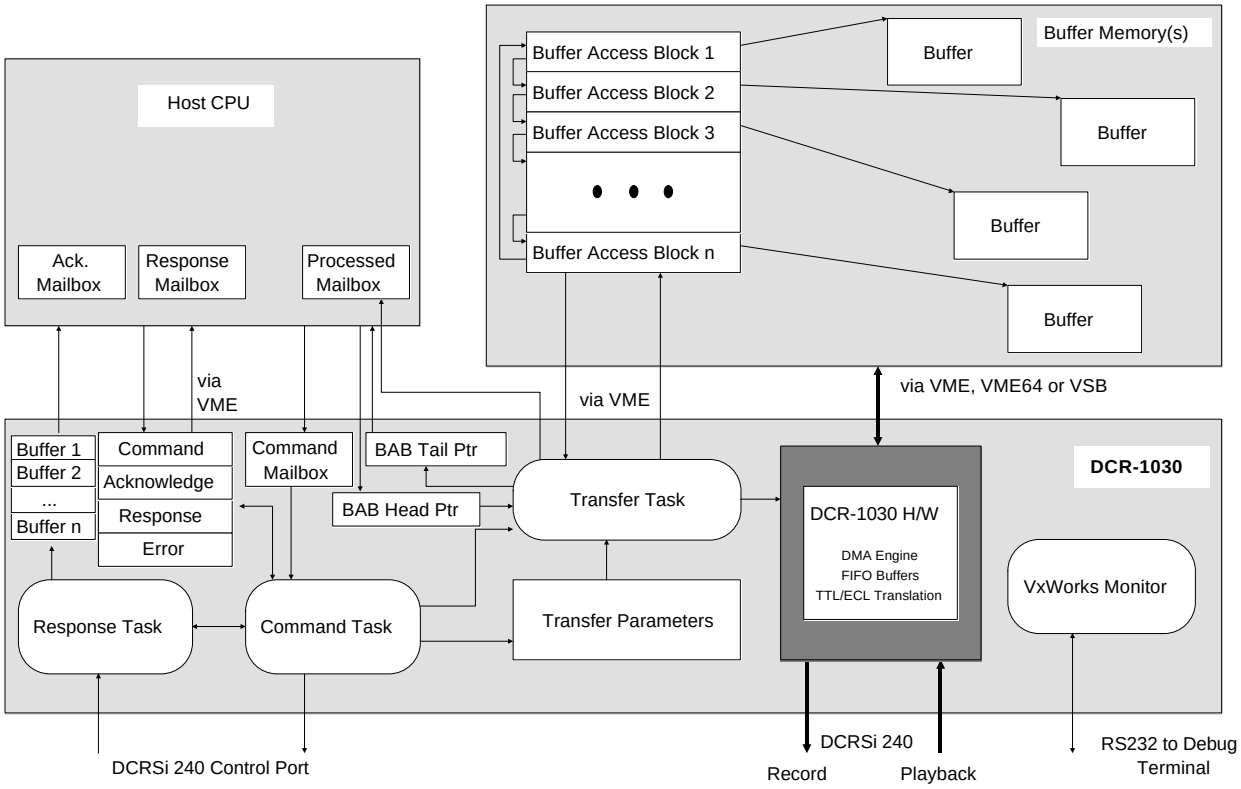

Figure 2 - Myriad DCR-1030 Standard Firmware Overview

3.1 Command Processing

The Command Task processes two types of commands, pass-through Ampex DCRsi commands and Macro commands. All commands will generate a signal to the VME Host. The three types of response signals generated in response to a command mailbox access are: the Acknowledge, the Response and the Error. Each signal is separately defined in the command structure and may generate a VME interrupt, a VME mailbox access, or a write of a specific data value to a user defined VME location. The response signals can be disabled if desired. Table 3-1 shows the operation of the signalling mechanisms for each type of command, or action initiated by the host.

Pass-through commands are all valid Ampex DCRsi control interface commands (as defined in reference 2.a.). Any VME-based processor can request a pass-through command by:

1. Testing the Command busy flag in the 1030 SF Command data structure
2. When the Command data structure is not busy, placing the desired Ampex DCRsi command in the data structure
3. Accessing the 1030 SF Command Mailbox

Accessing the 1030 SF Command Mailbox will activate the Command Task, which will determine that a pass-through type command is in the command buffer. The Command Task will forward the command to the Ampex DCRsi via the Ampex control interface, use the Acknowledge signalling mechanism, and can be instructed to wait for a response from the Ampex DCRsi. If the Response signalling mechanism is enabled, the 1030 SF will wait either 10 seconds for a response from the DCRsi or the "*" prompt. If a response is received, it is placed in the response section of the command structure. The Command Task then generates the signal to the Host as defined in the original command structure. When the signal is given, the Command Task sets the command busy flag to "not busy". If the response to the command is needed and it is not received within 10 seconds then the Error signalling mechanism is used.

Macro commands cause the Command Task to interpret the command in the command buffer and perform a sequence of operations. A complete description of the macro commands is provided in Section 4 of this document. The macro commands are:

1. Initialize transfer parameters (INIT_MACRO)
2. Start the Record process (RECORD_MACRO and move head pointer)
3. Start the Playback process (PLAYBACK_MACRO and move head pointer)
4. Stop the Record or Playback process (STOP_MACRO)

Table 3-1 Command Processing Signalling Mechanisms

ACTION	SIGNALLING MECHANISM :				
	ACK	RESPONSE	ERROR	PROC	DCR
Pass-Thru Command	Yes	Yes	Yes if timeout	No	Yes if response from DCRsi
Init Macro	No	Yes	Yes if bad parameter	No	No
Record Macro	Yes (when DCRsi is positioned)	Yes if end scan address	Yes if DMA errors or DMA timeout	No	Yes
Playback Macro	Yes (when DCRsi is positioned)	Yes if end scan address	Yes	No	Yes
Increment Head Pointer	No	No	Yes if bad BAB data, DMA error, or timeout	Yes	Yes
Responses from DCRsi	No	No	No	No	Yes

When the Command Task receives a macro command, it may respond to the Host by using either the Host defined Acknowledge signalling mechanism, the Host defined Response signalling mechanism, or both. The Initialize and Stop macros use only the Response signalling mechanism. The Record macro and Playback macro use the Acknowledge signalling mechanism when the command is processed and may use the Response signal if the command completes. The command busy flag is set "not busy" after the entire response and Host signal has been generated.The Passthrough commands use both the Acknowledge and Response signalling mechanisms.

Commands cannot be stacked or queued to the Command Task. The VME-Host processor must not enter a new command into the command buffer until processing of the previous command is complete.

3.2 Transfer Processing

Transfer processing is controlled by a set of user-defined Transfer parameters and the Buffer Access Blocks (BABs). Prior to a Record or Playback session, the VME-Host must issue the Initialize macro command, which causes the Command task to load the Transfer parameters, zero the BAB head and tail pointers, and zero the DCR Response Buffer head pointer.

After initialization, the VME-Host issues the Record or Playback macro commands. Transfer processing is the same for record and playback operations. For record operations, the term "available buffer" refers to a VME buffer that has been filled and has not yet been output to the Ampex DCRsi. For playback operations the term "available buffer" refers to an empty VME buffer.

The VME-host manages 1030 SF transfer processing using the BABs and associated head and tail pointers. Each time the host determines there is an "available buffer", the host updates the associated BAB and BAB Head pointer. The Transfer Task monitors the BAB Head pointer which tracks the number of available buffers in VME memory. When there is at least one available buffer, the Transfer Task outputs/inputs the next buffer to/from the Ampex DCRsi. The buffer is defined by the BAB, which includes a pointer to the buffer, the buffer size, access mode and a usage flag. Using the information in the BAB, the Transfer Task programs the DCR-1030 hardware to DMA the buffer to/from VME memory from/to the Ampex DCRsi. When the DMA is complete the Transfer Task resets the usage flag, updates the BAB Tail pointer and signals the host (via the processed signal mechanism). The processed signal mechanism can be programmed to write the updated tail pointer to the mailbox to reduce VME overhead when to checking the tail pointer. This process continues until the Record/Playback macro command completes or a Stop macro is issued by the Host. It is the Hosts responsibility to insure that the available buffers do not exceed the maximum number defined in the Initialize macro. Note that for best performance for VME transfers, data buffers should be aligned on 256 byte boundaries.

3.2.1 Initialization

The Initialization process sets up parameters that remain fixed for the duration of a record or playback session. There are 12 initialization parameters loaded from the Initialize Macro (INIT_MACRO). These parameters are described below:

1. Address of the first BAB: This address is a VMEbus A32 address located in system memory or in VME accessible memory on the Host processor.
2. Buffer Processed signalling mechanism: The Processed signal is used when a buffer (BAB) has been fully processed. The signalling mechanism is defined by the VME Interrupt (Level and Vector) and the VME Mailbox (Address value, Address space, Data width, and Data value).
3. DCR Response Buffer Mailbox signalling mechanism: The DCR Response signal is used each time a DCRsi control message is received from the DCRsi. The

signalling mechanism is defined by the VME Interrupt (Level and Vector) and the VME Mailbox (Address value, Address space, Data width, and Data value).

4. Recorder Type parameter. There are six selections currently supported for the 3 Ampex DCRsi machine types.

   a. DCR240 - assumes connection to a DCRsi 240 machine. This selection uses the clock oscillator located in U45. It does not use the BufferLimit signal from the DCRsi interface and therefore requires that U45 contain a value of 30MHz or less (default factory oscillator is 30.00 MHz).
   b. DCR107 - assumes connection to a DCRsi 107 machine. This selection uses the clock oscillator located in U35. It does not use the BufferLimit signal from the DCRsi interface and therefore requires that U35 contain a value of 13.375MHz or less (default factory oscillator is 13.375 MHz).
   c. DCRCLASSIC - assumes connection to a DCRsi Classic machine. This selection uses the clock oscillator located in U35. It does not use the BufferLimit signal or DataAvailable signal from the DCRsi interface and therefore requires that U35 contain a value of 13.375MHz or less.
   d. DCR240NOFLOW - assumes connection to a DCRsi 240 emulator that does not contain any flow control hardware. This selection uses the clock oscillator located in U45. It does not use the BufferLimit or DataAvailable signals from the DCRsi interface.
   e. DCR240BUFLIM - assumes connection to a DCRsi 240 machine that has updated firmware and buffer boards that supports BufferLimit. This selection uses the clock oscillator located in U45. It does use the BufferLimit signal from the DCRsi interface and therefore can support the full burst rate of 37.5MHz to the DCRsi. U45 can contain a value up to 37.5MHz.
   f. DCR107BUFLIM - assumes connection to a DCRsi 107 machine that has updated firmware and buffer boards that supports BufferLimit. This selection uses the clock oscillator located in U35. It does use the BufferLimit signal from the DCRsi interface and therefore can support the full burst rate of MHz to the DCRsi. U35 can contain a value up to MHz.

5. Transfer Mode: This is used to inform the firmware that it may be using an installed module for the data transfers. If MODE_VSB (2) is selected, then the VSB module will be initialized with the parameters contained in the Module Control Word. If MODE_RACEWAY is selected themodel MLR_1470 RACEway Module will be initialized with the parameters in the Module Contro Word (see the following for more information: item II, table 4-3, and table 4-5) If the subsequent transfers will use both VSB and VME then MODE_VSB must be selected.

6. Number of BABs/Maximum Available Buffers: This value is used for the Head and Tail pointer modulus.

7. Byte Ordering Control: Controls byte stacking/unstacking from 32 bit words as follows:

   a. BYTE_LL (D0-D7 First value = 0) - stacks/unstacks D0-D7, D8-D15, D16-D23, D24-D31.

b. BYTE_UU (D24-D31 First value = 1) - stacks/unstacks D24-D31, D16-D23, D8-D15, D0-D7.
c. BYTE_UM (D16-D23 First value = 2) - stacks/unstacks D16-D23, D24-D31, D0-D7, D8-D15.
d. BYTE_LM (D8-D15 First value = 3) - stacks/unstacks D8-D15, D0-D7, D24-D31, D16-D23.

8. DMA timeout: Used as a watchdog timer for the transfer of one buffer. This value is in 20ms ticks (i.e. 50 = 1 second).

9. Response Echo: Used to enable/disable the echo of the control port responses from the DCRsi to the console port. (1 = Enable)

10. Error Echo: Used to enable/disable the echo of standard firmware error messages to the console port. (1 = Enable)

11. Module Control Word: This parameter contains setup information specific to an installed module such as VSB, Ethernet, or RACEway. Only the VSB and RACEway module are currently supported by the firmware for data transfers.

12. VSB Module (XferMode = 2):

a. VSB Size: Defines the data width of the VSB transfer. Comprised of two bits and are located in data bit 0 and 1. These bits should always be set to zero (VDI_VSB_SIZE_QUAD) for 32 bit data transfers.
b. VSB Space. Defines the VSB address space. Comprised of two bits and are located in data bits 2 and 3. Three address spaces are defined by the VSB specification: Alternate Space (VDI_VSB_SPACE_ALT = 0x4), I/O Space (VDI_VSB_SPACE_IO = 0x8), and System Space (VDI_VSB_SPACE_SYS = 0xC). Normally programmed to VDI_VSB_SPACE_SYS.
c. VSB Lock: Defines the VSB locking control. This is a single bit located in bit position 4. If data 4 is set high the VSB lock signal will not be used for the data transfer (VDI_VSB_NOT_LOCK = 0x10). Normally programmed to VDI_VSB_NOT_LOCK.
d. VSB Arbitration: Controls the VSB Module Bus request logic. If data bit 5 is set high (VDI_VSB_ONLY_MASTER = 0x20) then the VSB module will not participate in bus arbitration. If the bit is cleared then it will arbitrate.
e. VSB Release Mode: Defines the bus mastership release mode. If data bit 6 is set high then Release On Request (VDI_VSB_ROR = 0x40) is chosen. If data bit 6 is low then Release When Done (VDI_VSB_RWD = 0) is selected.
f. VSB Block Limit: Maximum number of bytes permitted in a consecutive burst. Ranges from 0-1024. Data bits D8..D23. Default is 256 bytes (VDI_VSB_BLOCK_LIMIT = 0x00010000).

RACEway Module (xferMode = 4)

g. Block Size: Defines the local bus burst size for the Module Block Transfer Valid values -- 0x300 (2 Kbytes), 0x200 (1 Kbytes), 0x100(512 bytes), and 0 (256 bytes). A value of 0x300 for 2 Kbytes would provide the highest

performance tranfer but also hold off VMEbus slave reads and writes to the DCR-1030 for approximately 30 uS (2048/70Mytes/sec).

13. VME Access Control: This parameter select the privilege level for VMEbus accesses. (0 = Privileged, 1 = User Mode).

14. Stop on Errors: The parameter selects whether to stop or ignore DE errors reported by the DCRsi. A value of 1 causes the board to write the Error mailbox and then stop on DE Error messages from DCRsi. A value of 0 will tell the 1030 SF to ignore DE errors while transferring data.

3.2.2 Record Operation

The DCR-1030 does not impose a format on the tape. VME buffers are written to tape without any added data. At the end of a Record session, the 1030 SF does not add fill data to complete a 4356 byte scan.

Prior to Record operations the VME-Host must send the Initialize Macro command to the 1030 SF and may pre-initialize all BABs with BAB link addresses, buffer addresses, and the usage flag set to "Empty". When ready to start the record process, the VME-Host sends the Record macro command. The 1030 SF will immediately acknowledge the reception of the Record Macro by signalling the Host defined Acknowledge Mailbox and/or VME Interrupt and may send the appropriate control commands to the DCRsi to begin recording. The VME-Host then fills the memory buffers and marks the associated BAB host usage flags as "Available". The host may update the BAB Head pointer as each buffer is filled, or may update the BAB Head pointer once for several buffers (this is sometimes useful in startup conditions or when buffers are available in pairs). The 1030 SF monitors the difference between the BAB Head and Tail pointers to determine the number of buffers available. When this difference is greater than zero the DCR-1030 initiates a DMA from the first available buffer to the record FIFO buffers on the DCR-1030. The DCR-1030 hardware then handles output of the data to the Ampex DCRsi.

When the DMA is complete (all data has been extracted from the VME buffer) the 1030 SF clears the DCR-1030 usage flag in the BAB, signals the Host (either accessing a "Processed Mailbox" or generating a VME interrupt as requested) and then updates the BAB Tail pointer. The 1030 SF then checks the number of buffers available, if a buffer is available the 1030 SF will initiate the transfer of the next buffer. If not, the 1030 SF Transfer Task will monitor the Head and Tail difference and wait until a buffer becomes available.

There are two options for controlling the recording of data to the DCRsi - the Host Controlled, and Record using Scan Address.

Host Controlled requires the host to fully control the DCRsi via pass-through commands (or other means), then issue the Record Macro with the flag set appropriately. In the Host Controlled option, the Host must position the recorder, ensure that there are no errors and wait for the Data Transfer Ready State before issuing the Record Macro.

In the Record using Scan Address option, the 1030 SF controls the DCRsi using a start scan address and the number of scans. The start scan address is the scan address for the record session or can be set to "-1" if recording is to begin at the current tape position. The number of scans can also be set to "-1" if the length record session is not known. If the number of scans is "-1" then the output of data to the recorder will continue until the Stop Macro is received. When the Stop Macro is received by the 1030 SF, it will immediately issue the "SL" control command to the DCRsi and abort any data transfers. The Stop Macro responds to the Host via the Response signal mechanism and returns the actual start scan address and the last scan address recorded by the DCRsi.

If the number of scans parameter is set, the 1030 SF will calculate the number of bytes (number of scans * 4356) to be output to the recorder. If the data in the BABs given for the record session is greater than number of calculated bytes, then only the amount of data specified by the number of scans will be output to the recorder. If there is extra data in the BAB buffer, it will be ignored and the BAB will be marked as 'Empty' and normal BAB completion will be performed. Upon completion of the last BAB of the Record Macro, the 1030 SF will respond to the Host via the Response signal mechanism and will return the actual start scan address and the last scan address recorded by the DCRsi in the Response section of command buffer.

3.2.3 Playback Operation

The playback operation is similar to the record operation. The Host first issues the Initialize Macro command to the 1030 SF and waits for the response via the Response signalling mechanism . The VME-Host may then issue the Playback Macro command. After receiving the response from the Acknowledge signalling mechanism, the Host can load "n" BABs structures and set the Head pointer to indicate that "n" buffers are available (empty). "N" can be any number up to the number of BABs - 1. Since all VME buffers are initially available, the Transfer Task will initiate DMAs to the buffers specified in the first BAB as the data comes from the recorder. As in the record operation, when each of the buffers are filled the Transfer Task notifies the host (either VME interrupt or mailbox access), updates the BAB Tail pointer, sets the Usage flag, and proceeds to the next BAB. If the BAB Tail pointer (after update) equals the BAB head pointer, transfer from the DCRsi is suspended until at least one buffer is available.

There are two options for controlling the playback of data from the DCRsi - Host Controlled and Play using Scan Address.

The Host Controlled mode requires the host to fully control the DCRsi via PASS-Thru commands (or other means). If the VME-Host chooses to use the Host Controlled option, it must position the recorder, ensure that there are no errors and wait for the Data Transfer Ready State before issuing the Playback Macro.

In the Playback using Scan Address option, the 1030 SF controls the DCRsi using the start scan address and number of scans parameters. The start scan address is the initial scan address for

the playback session or can be set to "-1" if playback is to begin at the current tape position. The number of scans can also be set to "-1" if the length of the playback session is not known. If the number of scans is "-1", the input of data from the recorder will continue until the Stop Macro is received. When the Stop Macro is received by the 1030 SF, it will immediately issue the "SL" control command to the DCRsi and abort any data transfers. The Stop Macro responds to the Host via the Response signal mechanism and returns the actual start scan address and the last address output by the DCRsi.

If the number of scans parameter is set, then the 1030 SF will calculate the number of bytes (number of scans * 4356) to be input from the recorder. If the data in the BABs given for the playback session is greater than number of calculated bytes, then only the amount of data specified by the number of scans will be input from the recorder. If there is extra space in the BAB buffer, it will be ignored and the BAB will be marked as 'Full' and normal BAB completion will be performed. Upon completion of the last BAB of the Playback Macro, the 1030 SF will respond to the Host via the Response signal mechanism and will return the actual start scan address and the last scan address played by the DCRsi in the Response section of command buffer.

3.3 DCRsi Response Processing

The DCRsi Response Task handles the control data from the DCRsi. The ASCII responses from the DCRsi are placed in a circular array in VME accessible memory called the DCRsi Response buffer. The DCRsi Response buffer can be used to process the asynchronous pass-through command responses from the DCRsi.

The DCRsi Response Task can be programmed to signal the Host for each response received from the DCRsi. The Initialize Macro is used to define the signalling mechanism for the DCRsi Responses placed in the DCRsi Response Buffer. The DCRsi Response Head pointer, in VME accessible memory, is an index into the array of Responses and points to the first empty location in the buffer.

4 Data Structures

The data structures used by the 1030 SF are:

1. Mailboxes,
2. Command/Acknowledge/Response Structures in DCR-1030 Local Memory,
3. DCR Response Buffer in DCR-1030 Local Memory,
4. DCR Response Buffer Head Pointer in DCR-1030 Local Memory
5. Busy Flag in DCR-1030 Local Memory,
6. Buffer Access Block (BAB) Head and Tail Pointers in DCR-1030 Local Memory,
7. BABs in VME Accessible Memory, and
8. Data Buffers in VME, VME64, or VSB accessible memory.

Table 4-1 defines the location of items 1 through 5 above as an offset from the DCR-1030 base VME A16 and A32 slave address. The VME base address is determined by switch S3 on the DCR-1030. The DCR-1030 User's Manual (reference 2-2) describes configuring the DCR-1030 in detail, including recommended settings for all switches and jumpers. Appendix A of this document is an excerpt from the User's Manual relating to setting the VME base address.

Table 4-1 Location of Command Mailbox and Local Data Structures

Item	Offset (Hexidecimal)	Base	Access Mode
Command Mailbox	0x0023	Register Space	A16/D8
DCR Response Buffer	0x007E,0000	Memory Space	A32/D32
Command Section	0x007F,0000	Memory Space	A32/D32
Acknowledge Section	0x007F,0080	Memory Space	A32/D32
Response Section	0x007F,0100	Memory Space	A32/D32
Error Section	0x007F,0180	Memory Space	A32/D32
BAB Head Pointer	0x007F,0400	Memory Space	A32/D32
BAB Tail Pointer	0x007F,0404	Memory Space	A32/D32
Command Buffer Busy Flag	0x007F,0408	Memory Space	A32/D32
DCRsi Response Head Pointer	0x007F,040C	Memory Space	A32/D32
Self-Test Mailbox	0x007F,0410	Memory Space	A32/D32

The following paragraphs define these structures.

4.1 Mailboxes

Mailboxes are used to signal events between the 1030 SF and VME-host application software. VME hosts use Command mailbox to indicate new commands are in the Command/Acknowledge/Response data structure, and activate the Command mailbox by accessing the VME addresses specified in table 4-1. Note that the Command Mailbox is a VME A16, D8 access (i.e., a byte write in the Hosts VME A16 (Short) address space).

The 1030 SF has 5 different signals to the VME-host (as shown in Table 3-1). The Acknowledge Mailbox provides immediate feedback to the Host for commands that may take a long time to execute such as: pass-through commands, Record macro commands and Playback macro commands. The Response Mailbox provides an indication to the Host that the command has fully completed. The Error Mailbox provides an indication that an error has occurred with a command or an error occurred during Transfer processing. The Processed Mailbox indicates that a BAB buffer has been processed. The DCR Response Buffer Mailbox indicates that a message was received from the DCRsi control port. From the 1030 SF point of view, the mailbox accesses are simply writes to defined VME locations. The Mailboxes can be used with a single VME Interrupt to distinguish the type of response from the 1030 SF.

4.2 DCRsi Response Buffer in DCR-1030 Local Memory

The DCRsi Response Buffer data structure exists in DCR-1030 local memory. The DCRsi Response Buffer is a circular array of 512 fixed length (128 bytes) text strings received from the DCRsi control port. The DCRsi Response Head pointer is an index to the last response from the DCRsi. The Head pointer is reset to zero and the Buffer cleared (0x20) when the Initialize Macro is executed. The Initialize Macro is used to define the signalling mechanism for the DCRsi Responses placed in the DCR Response Buffer. If the response mechanism is enabled, the 1030 SF will use that mechanism each time a response is placed in the DCRsi Response Buffer. The 1030 SF will always update the Head pointer for each response.

4.3 Command/Acknowledge/Response Structures in DCR-1030 Local Memory

The Command/Acknowledge/Response data structure exists in DCR-1030 local memory. Tables 4-2 through 4-6 define the Command/Acknowledge/Response data structure.

4.4 Self-Test Mailbox

The Self-Test Mailbox exists in DCR-1030 local memory. The Self-Test Mailbox is a single longword containing the results of the power-up self-test. A value of 0x11 indicates that the self-test is in progress. A value of 0x12 indicates that the self-test has passed, and a value of 0x13 indicates that an error has occurred.

Table 4-2 Command/Acknowledge/Response Structure Format

Byte Offset	Description
Command Section	
0.	Command type 0 = Ampex DCRsi pass-through 1 = Initialize Macro 2 = Record Macro 3 = Playback Macro 4 = Stop Macro
4 - 127	Pass-through command (terminated by 0 byte) For macro commands see tables 4-3 through 4-7
Acknowledge Section	
128 - 131	Acknowledge VME interrupt number (0 or -1 for no interrupt)
132 - 135	Acknowledge VME vector
136 - 139	Acknowledge mailbox address (0 or -1 for no mailbox access on acknowledge)
140 - 143	Acknowledge mailbox access mode: 0 = A16 1 = A24 2 = A32
144 - 147	Acknowledge mailbox data width: 0 = D8 1 = D16 2 = D32
148 - 151	Acknowledge mailbox write value
152 - 191	Reserved
192 - 255	Acknowledge Status: 0 = OK -1 = Error
Response Section	
256 - 259	Response VME interrupt number (0 or -1 for no interrupt)
260 - 263	Response VME vector
264 - 267	Response mailbox address (0 or -1 for no mailbox access on response)

268 - 271	Response mailbox access mode: 0 = A16 1 = A24 2 = A32
272 - 275	Response mailbox data width: 0 = D8 1 = D16 2 = D32
276- 279	Response mailbox write value
280 - 319	Reserved
320 - 447	Response area: Text string from DCRsi on Pass-thru commands For macro commands see tables 4-3 through 4-7
Error Section	
448 - 451	Error VME interrupt number (0 or -1 for no interrupt)
452 - 455	Error VME vector
456 - 459	Error mailbox address (0 or -1 for no mailbox access on error)
460 - 463	Error mailbox access mode: 0 = A16 1 = A24 2 = A32
464 - 467	Error mailbox data width: 0 = D8 1 = D16 2 = D32
468 - 471	Error mailbox write value
472 - 511	Reserved
512 - 515	Error Status Code (see Table 4-7 for list)
516 - 519	General Status Register (see User's Manual)
520 - 523	Recorder Status Register (see User's Manual)
524 - 527	VIC64 DMA Status Register (see Cypress VIC64 Manual)
528 - 531	VIC64 Bus Error Status Register (see Cypress VIC64 Manual)
532 - 575	Reserved

Table 4-3 Initialize Macro Command Structure

Byte Offset	Description
4 - 7	Address of first BAB in VME A32 memory
8 - 11	Processed Signal VME interrupt number (0 or -1 for no interrupt)
12 - 15	Processed Signal VME vector
16 - 19	Processed mailbox address (0 or -1 for no mailbox access on response)
20 - 23	Processed mailbox access mode: 0 = A16 1 = A24 2 = A32
24 - 27	Processed mailbox data width: 0 = D8 1 = D16 2 = D32
28 - 31	Processed mailbox write value (if 0 Tail pointer is written to mailbox)
32 - 35	DCR Response Signal VME interrupt number (0 or -1 for no interrupt)
36 - 39	DCR Response Signal VME vector
40 - 43	DCR Response mailbox address (0 or -1 for no mailbox access on response)
44 - 47	DCR Response mailbox access mode: 0 = A16 1 = A24 2 = A32
48 - 51	DCR Response mailbox data width: 0 = D8 1 = D16 2 = D32
52 - 55	DCR Response mailbox write value
56 - 59	Recorder Type (0 = DCRsi 240, 1 = DCRsi 107, 2 = DCRsi Classic, 3= DCRsi 240 NoFlow Control, 4 = DCRsi 240 w/ BUFLIM, 5 = DCRsi 107 w/BUFLIM
60 - 63	Transfer Mode (2 if using VSB, 4 if using RACEway)

64 - 67	Number of BABs
68 - 71	Byte Order (0=D0-D7 first, 1= D24..D31 first, 2=D16..D23 first, 3= D8..D15 first)
72 - 75	DMA Timeout (timeout for each BAB in 20 msec ticks)
76 - 79	Response Echo (1=echo DCRsi responses to console)
80 - 83	Error Echo (1=echo 1030 Error messages to console)
84 - 89	Module Control Word: - - if Transfer Mode = 2 (VSB) then VSB Control Word - see Table 4-4 - if Transfer Mode = 4(RACEway) then RACEway Control Word - see Table 4-5)
90 - 93	VME Access: 0 = Supervisor, 1 = User
94 - 97	Stop Error: 0 = Do not stop on DCRsi DE errors, 1 = stop on DCRsi DE errors
Response Section	
320 - 323	Response Status: 0 = OK -1 = Error

Table 4-4 VSB Control Register Bit Definitions

Definition	Value
VDI_VSB_SIZE_QUAD	0x00000000
VDI_VSB_SPACE_ALT	0x00000004
VDI_VSB_SPACE_IO	0x00000008
VDI_VSB_SPACE_SYS	0x0000000C
VDI_VSB_NOT_LOCK	0x00000010
VDI_VSB_LOCK	0x00000000
VDI_VSB_ONLY_MASTER	0x00000020
VDI_VSB_RWD	0x00000000
VDI_VSB_BLOCK_LIMIT_256	0x00010000

Table 4-5 RACEway Control Register Bit Definitions

Definition	Value
RACE_BLOCK_LIMIT_256	0
RACE_BLOCK_LIMIT_512	0x100
RACE_BLOCK_LIMIT_1K	0x200
RACE_BLOCK_LIMIT_2K	0x300

Table 4-6 Record Macro Command Structure

Byte Offset	Description
4 - 7	Record Flag: 1 = Host Controlled 2 = Use Scan Address
8 - 11	Start Scan Address (-1 if current position)
12 - 15	Number of Scans (-1 if use Stop Macro to terminate)
16 - 127	Reserved
Response Section	
320 - 323	Actual Start Scan Address
324 - 327	Actual End Scan Address

Table 4-7 Playback Macro Command Structure

Byte Offset	Description
4 - 7	Record Flag: 1 = Host Controlled 2 = Use Scan Address
8 - 11	Start Scan Address (-1 if current position)
12 - 15	Number of Scans (-1 if use Stop Macro to terminate)
16 - 127	Reserved
Response Section	
320 - 323	Actual Start Scan Address
324 - 327	Actual End Scan Address

Table 4-8 Stop Macro Command Structure

Byte Offset	Description
4 - 127	Reserved
Response Section	
320 - 323	Actual Start Scan Address
324 - 327	Actual End Scan Address

4.5 Buffer Access Block Head and Tail Pointers

The Buffer Access Block head and tail pointers are each 32 bit integers in DCR-1030 local DRAM (see table 4-1). The head pointer is always written by the host and read by the DCR-1030 transfer task. The pointer is interpreted as an index into the set of BABs. A zero value in the head pointer indicates that the first BAB (as defined in the initialize macro) will describe the next buffer to be made available by the host. The integer 1 in the head pointer indicates the next BAB and so on. The host should always ensure that the buffer is filled (for record) or empty (for playback) before updating the BAB head pointer.

The tail pointer is always written by the DCR-1030 transfer task. The host may read the tail pointer to determine where the DCR-1030 is in the process. The processed mailbox can be instructed to write the tail pointer value to the mailbox location. Enabling this tail pointer write back feature can improve system performance by reducing VMEbus traffic.

Neither the head nor tail pointer will ever equal or exceed the number of BABs defined in the initialize macro. The host must ensure that head pointer is updated modulo the number of BABs. When the head and tail pointers are equal the DCR-1030 assumes that there are no available buffers. In processing the initialization macro, the command task zeros both the head and tail pointers. Therefore, the host should wait until initialization is complete before updating the head pointer. The host must also ensure that the head pointer is not incremented such that it equals the tail pointer. This would indicate a buffer overrun condition that cannot be detected by the DCR-1030 SF.

4.6 DCRsi Response Buffer Head Pointer

The DCRsi Response head pointer is a 32 bit integer in DCR-1030 local DRAM (see table 4-1). The head pointer is always written by the DCR-1030 and read by the host. The pointer is interpreted as an index into the array of Responses. An integer 2 value in the head pointer indicates that the first location will contain a valid DCRsi response. The integer 3 in the head pointer indicates the next response and so on. The Initialize macro, clears the head pointer to a 1. Therefore the host should wait until initialization is complete before using the head pointer.

4.7 Buffer Access Blocks (BABs) in VME Accessible Memory

BABs are a circular data structure in VME memory. The first word of each BAB is the address of the next BAB. The last BAB points to the first BAB.

Each BAB describes a buffer in VME/VSB memory. This description includes the address (physical VME or VSB address), access mode, usage flags, and other data. Table 4-7 defines the structure of each BAB.

Note that bit 24 of the access mode flag can be set to disable the assertion of Write Data Ready to the DCRsi.

Table 4-9 DCR-1030 Error Messages in ErrorStatus Field

ERROR MESSAGE	OFFSET (HEXIDECIMAL)	DESCRIPTION
INVALID_COMMAND	0x80001001	Invalid Command Type
INVALID_PARAM	0x80001002	Invalid Parameter
COMMAND_SEQ_ERROR	0x80001003	Bad command sequence
INTERNAL_STATE_ERROR	0x80002001	Internal DCR-1030 error
DMA_TIMEOUT	0x80003001	Timeout transferring data
DMA_ERROR	0x80003002	Bus error transferring data
DMA_PLAY_FIFO_OVER	0x80003003	Playback FIFO overrun
DMA_REC_FIFO_UNDER	0x80003004	Record FIFO underrun
PASSTHRU_RSP_TIMEOUT	0x80004001	10 second response timeout
DCR_DE_RESPONSE	0x80005001	DE responses from DCRsi
DCR_DLB_TIMEOUT	0x80005002	Timeout waiting for DL-B DCR response
DCR_DLE_TIMEOUT	0x80005003	Timeout waiting for DL-E DCR response
DCR_DTR_TIMEOUT	0x80005004	Timeout waiting for DS DTR DCR response
DCR_DLB_ERROR	0x80005005	Error with DL-B DCR response
DCR_COMM_TIMEOUT	0x80005006	DCRsi not responding
INVALID_BAB_ADDRESS	0x80006001	Invalid VME address
INVALID_BAB_SIZE	0x80006002	Invalid transfer size
INVALID_BAB_MODE	0x80006003	Invalid access mode

Table 4-10 Buffer Access Block Structure

Word Offset	Description
0	VME address of next BAB
1	VME Buffer address
2	Buffer access mode 0 = VME32 1 = VME64 2 = VSB 3 = VME Singles 4 = RACEway Note: Data Bit 24 is Data Ready Control if set then Write Data Ready will not be asserted by the hardware for the entire buffer.
3	Buffer size in bytes
4	Usage Flag 0 = Empty 1 = Full
5	RACEway Route Word (transfer mode 4 only), or RACEway

4.8 Data Buffers in VME, VME64, or VSB Accessible Memory

The data buffers are located in VME/VSB memory. The user is responsible for ensuring that the access mode defined in the BAB is appropriate for the VME buffer type.

While each buffer must be in contiguous memory, successive buffers need not be in contiguous VME memory. Each VME buffer may be a different size.

Best performance is obtained by aligning VME buffers on a 256 byte address boundary. The VME block transfer hardware requires that block transfer addresses are 256 byte aligned if the block transfer size is large enough to cross a 256 byte address boundary. Therefore, if a VME buffer address is note 256 byte aligned the 1030 SF will perform a short transfer to align the address, then a second DMA which performs multiple, FIFO sized transfers, and possible a third to transfer the remaining data. Furthermore, VME64 block transfers have a 2048 byte address alignment restriction and may require up to 4 DMA setups when the buffer address is not 256 byte aligned. Address alignement is only critical for buffers less than 128 Kbytes.

Appendix A
DCR-1030 Base Address Settings

The DCR-1030 supports separate VME base addresses for Register space and Memory space. Register space is always addressed as A16 (short address space) and is used to access the DCR-1030 Command Mailbox. Memory space may be addressed as A24 or A32. All access to the DCR-1030 DRAM is performed as a memory space access (this includes the command/response/acknowledge data structure and the BAB head and tail pointers).

The switch S3 on the DCR-1030 selects one of 16 sets of register and memory space. Table A-1 defines the DCR-1030 base address in both register and memory space for each of the 16 settings.

Table A-1 DCR-1030 Base Address Settings

S3(4321) 0=ON 1=OFF	Memory Space (A32 or A24) Base and Bound	Register Space (A16) Base and Bound
0 0 0 0	0x10000000-0x10FFFFFF (A32)	0x0000-0x00FF
0 0 0 1	0x18000000-0x18FFFFFF (A32)	0x0800-0x08FF
0 0 1 0	0x30000000-0x30FFFFFF (A32)	0x1000-0x10FF
0 0 1 1	0x40000000-0x40FFFFFF (A32)	0x3000-0x30FF
0 1 0 0	0x50000000-0x50FFFFFF (A32)	0x4000-0x40FF
0 1 0 1	0x60000000-0x60FFFFFF (A32)	0x5000-0x50FF
0 1 1 0	0x70000000-0x70FFFFFF (A32)	0x6000-0x60FF
0 1 1 1	0x80000000-0x80FFFFFF (A32)	0x7000-0x70FF
1 0 0 0	0x90000000-0x90FFFFFF (A32)	0x8000-0x80FF
1 0 0 1	0xA0000000-0xA0FFFFFF (A32)	0x9000-0x90FF
1 0 1 0	0xB0000000-0xB0FFFFFF (A32)	0xA000-0xA0FF
1 0 1 1	0xC0000000-0xC0FFFFFF (A32)	0xB000-0xB0FF
1 1 0 0	0xD0000000-0xD0FFFFFF (A32)	0x0000-0x00FF
1 1 0 1	0xE0000000-0xE0FFFFFF (A32)	0x1000-0x10FF
1 1 1 0	0x00000000-0x00FFFFFF (A24)	0x0000-0x00FF
1 1 1 1	0x00000000-0x00FFFFFF (A24)	0x8000-0x80FF

Appendix B
DCR-1030 Confidence Tests

The DCR-1030 EPROM contains two programs that can be executed from the console terminal to validate the operation of the board with the DCRsi recorder. The first test will check the remote control interface to ensure that the DCRsi can communicate with the DCR-1030. The remote control interface on the DCRsi must be setup for RS422, 9600 Baud, 8 data bits, and Even Parity. The EPROM resident function 'DRC' can be executed from the console prompt to send ASCII DCRsi commands to the remote control interface. The 'DRC' function is passed a ASCII text string terminated by a semicolon (;). An example of sending a DS (Display Command Mode Status) command to the DCRsi is shown below:

```
-> DRC "DS;"
```

The 'DRC' function will send the string to the DCRsi then return a value (i.e., value=0=0x0). If the remote control interface is functioning properly, then the response from the DCSsi will be displayed on the console. The console output should look like below:

```
-> value = 0 = 0x0
       -> - DS 4000;
```

If the 'DS 4000' response is not displayed on the console check the following items:

1. The remote control cable connection between DCR-1030 and DCRsi
2. Verify that the DCRsi unit is powered ON
3. Verify that the DCRsi serial port setting is RS422, 9600 Baud, 8 data bits, and Even Parity
4. Type 'Echo' at the VxWorks console prompt to verify that DCRsi responses are being echoed to the console. The value of Echo should be a 1. If Echo is 0 then console output is disabled. Console output can be enable by typing 'Echo = 1'.

If the 'DRC' function hangs and does not return a value then a Control-C will abort the function and restore the console. If the 'DRC' function hangs, it may be due to the DCR-1030 not receiving the asterisk (*) prompt. The asterisk prompt from the DCRsi indicates that it is ready to receive a command. If this lockup happens, use Control-C, then type 'xmitOK = 1' and retry the command.

The second test will control the DCR-1030 Standard Firmware tasks from a function called 'sf1030ctl'. The 'sf1030ctl' function will require VMEbus memory for the BAB structures and the

Data buffers. This test can be used to record a pre-initialized data pattern to the Ampex DCRsi recorder then play and verify that the data matches the recorded pattern. The text below is a capture of the console output used to record and play 1 minute of an incrementing longword data pattern at scan address 1000000. Note that the actual record scan address was at 1010000 instead of at 1000000.

```
-> sf1030ctl
Enter DCR-1030 VME slave switch setting (S3): 0x0
Enter BAB structure VMEbus (A32) address: 0x21f00000
Enter VMEbus (A32) data buffer address: 0x20000000
Enter VMEbus buffer size in bytes: 0x1f00000



             DCR-1030 Standard Firmware Test Control
             Myriad Logic, Inc
                 1994
    ----------------------------------------
1)  Enter/View Command/Ack/Response Buffer
2)  Record Tests
3)  Playback Tests
4)  Loop Test
5)  Init/View/Compare DMA Buffer
X)  Exit Tests
Enter Selection:5

Select function:
1)  Init Buffer
2)  Display Buffer
3)  Compare Buffer
Otherwise) - return to main menu 1

Select type of fill pattern:
1)  Zero
2)  Incrementing Bytes
3)  Alternating longwords
4)  Incrementing longwords
5)  Ampex Random Bytes
Otherwise) - return to main menu 4
```

```
    DCR-1030 Standard Firmware Test Control
            Myriad Logic, Inc
                 1994
 ----------------------------------------
1)  Enter/View Command/Ack/Response Buffer
2)  Record Tests
3)  Playback Tests
4)  Loop Test
5)  Init/View/Compare DMA Buffer
X)  Exit Tests
Enter Selection:2

Enter Record Flag (1=Host Control,2=Use Scan Address) : 2
Enter Start Scan Address(Decimal): 1000000
Enter Number of Scans(Decimal): -1
Enter BAB Size: 0x10000
Enter AccessMode[0=MODE_VME32,1=MODE_VME64,2=MODE_VSB, 3=MODE_PIO] :0
Enter byte Order [0=BYTE_LL, 1=BYTE_UU, 2=BYTE_UM, 3=BYTE_LM] :0
Enter RcdrType [0=DCR240, 1=DCR107, 2=DCRCLASSIC, 3=DCR240NOFLOW] :0
Enter Length of Record Session (minutes): 1
ctlNumBABs = 496

init_dcr1030() - OK
open_dcr1030_write() - OK
beginAddr=1010000 endAddr=1171131
close_dcr1030() - OK

Hit any key to EXIT

   DCR-1030 Standard Firmware Test Control
             Myriad Logic, Inc
                  1994
----------------------------------------
1)  Enter/View Command/Ack/Response Buffer
2)  Record Tests
3)  Playback Tests
4)  Loop Test
5)  Init/View/Compare DMA Buffer
X)  Exit Tests
Enter Selection:3
```

```
Compare Input Data? [1=yes]:1
Enter Play Flag (1=Host Control,2=Use Scan Address) : 2
Enter Start Scan Address(Decimal): 1010000
Enter Number of Scans(Decimal): -1
Enter BAB Size: 0x10000
Enter AccessMode [0=MODE_VME32,1=MODE_VME64,2=MODE_VSB,3=MODE_PIO] :0
Enter byteOrder [0=BYTE_LL, 1=BYTE_UU, 2=BYTE_UM, 3=BYTE_LM] :0
Enter RcdrType [0=DCR240, 1=DCR107, 2=DCRCLASSIC, 3=DCR240NOFLOW] :0
Enter Length of Playback Session (minutes): 1

ctlNumBABs = 496
init_dcr1030() - OK
open_dcr1030_read() - OK

Comparing data ... Passed

beginAddr=1010000 endAddr=1017465
close_dcr1030() - OK

Hit any key to EXIT

   DCR-1030 Standard Firmware Test Control
         Myriad Logic, Inc
               1994
-----------------------------------------
1)  Enter/View Command/Ack/Response Buffer
2)  Record Tests
3)  Playback Tests
4)  Loop Test
5)  Init/View/Compare DMA Buffer
X)  Exit Tests

Enter Selection:x
value = 0 = 0x0
->
```

Myriad Logic provides the "C" language source for the sf1030ctl function in a file call d1030demo.c. This "C" language program provides an example of how to control the 1030 SF from itself, another 1030 board and a Motorola MVME162 CPU board and Mercury MCU6. The 1030 SF EPROM alos contains a diagnostic program to check the DCR-1030 hardware without the use of a DCRsi recorder. This program can be run from the console prompt by typing 'dcr1030atp'. The program prompts for VME record and playback buffer addresses, a VME

buffer (4000 should be entered), and a menu will be displayed. Menu option 7 (Functionality Test) will perform an automatic test of most functions of the board but requires an ECL loopback cable (PN DTI-C-100xxx) to successfully complete all tests. Menu option 3 allows the selection of either TTL or ECL loopback and can be used to perform diagnostics when an ECL loopback cable is not available.

Appendix C
DCR-1030 Console Port Debug Utilities

The following functions can be called from the console part of the DCR-1030:

1. BABdiff - return difference between head and tail pointers
2. showCommand - shows value of last command in DCR-1030 command buffer
3. show SF control - show values of internal Standard Firmware control structure
4. *(0x3F0400) - prints value of Head pointer
5. *(0x3F0404) - prints value of Tail pointer
6. BABptrTail - prints current value of 1030 SF's pointer to the BAB structures
7. DCR "ampex serial command:" - send serial control command to the DCRsi control port - response will be echoed to the console