ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Abaco Systems / SBS ABI-V4* at our website: ***Click HERE***

Table OVW-5 ABI Error Table

Word Address	Byte Address	Class	Description
12Ch	258h	P	An unexpected gap followed a receive mode command containing a data word.
12Eh	25Ch	P	After a receive mode command with data, a word with a command sync was received.
130h	260h	P	An apparent mode command was received, but it contained a parity error and was not processed.
132h	264h		Undefined
134h	268h	P	A mode command was received containing an illegal MIL-STD-1553B mode code.
136h	26Ch	P	A data word was expected but a command word was received.
138h	270h	I	Interrupt buffer overflow. Too many interrupts were received before host servicing.
13Ah	274h		Undefined
13Ch	278h	I	A number of entries in the pointer table must contain non-zero values for proper operation. This error is incremented for each entry that is "0" when the CMD word is set to "1". The address of the last entry is placed in the parameter word associated with this error.
13Eh	27Ch	I	Bus Controller program block word error.
140h	280h	P	No gap preceded an apparent received status word.
142h	284h	H	The transmitter was unable to send data on the bus.
144h	288h	I	HOB buffer has error (block transfer only).
146h	28Ch	P	RT-RT word count error.
148h	290h	P	Mode code timeout while waiting for data.
14Ah	294h	P	Invalid data on receive status.
14Ch	298h	P	Invalid data on last data word of BC-RT message.
14Eh	29Ch	M	Found DRDY without SARDY
150h	2A0h	P	No gap after last word of RT-BC.
152h	2A4h	P	Intermessage gap error.
154h	2A8h	P	No gap timeout detected for delay.
156h	2ACh	P	BC-RT long message error. (No gap at end of data).
158h	2BOh	H	Timer busy

Class Definitions:
H Internal hardware failure
I Application initialization or response related
P Bus protocol
M Microcode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hardware Reset

If the hardware reset button on the host system is pressed, the CSR is reset and microcode execution is halted. Also, the DMA interface registers, word addresses 01h through 06h, will be cleared. Memory above "003Fh" will remain intact and accessible until the microcode is re-started. See the "Microcode Initialization" subsection for details on restarting the microcode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MICROCODE INITIALIZATION

When an ABI module is powered-up or reset, the host software must perform the proper startup procedure to initialize the microcode program. However, before performing any initialization procedure, the user must first determine which microcode program to execute. Table MC-1 lists the two major types of ABI microcode programs: Built-In Tests (program numbers 1-7) and Application Main Body (program number 13).

Table MC-1 Microcode Program Identification

Microprogram ID #		Microprogram Description	Where Discussed
1	0001 h	Built-In Test 1	Built-In Test Routines
2	0002 h	Built-In Test 2	
3	0003 h	Built-In Test 3	
4	0004 h	Built-In Test 4	
5	0005h	Reserved	
6	0006 h	Built-In Test 6	
7	0007h	Reserved	
8	0008 h	Reserved	--------
9	0009 h	Reserved	--------
10	000A h	Reserved	
11	000B h	Reserved	
12	000C h	Reserved	-------
13	**000D h**	**Full Function 1553B**	**"1553 Microcode Program Startup Procedure" in this manual subsection**
14	000E h	Reserved	-------
15	000F h	Reserved	

The first type of microcode program is for Built-In-Test functions. These functions are described in the "Built-In Test Routines" section of this manual. The Application Main Body program is the entry point for the "Full Function 1553B" microcode program that executes the BC, RT, Monitor, and Interrupt functions of this interface.

The ABI can support two separate and independent microcode programs. This is possible due to a bit in the CSR which selects either the upper or the lower area of PROM for storage (See Table OVW-2). The desired area is selected as part of the "1553 Microcode Program Startup Procedure" which follows in this subsection.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1553 Microcode Program Startup Procedure

To start a desired ABI microcode program, perform the following procedure:

1. Write "0001" to the CSR (offset 0000h) to reset the ABI hardware.

2. Write "000D" to the MAP Register (offset 0010h) to address the "Full Function 1553B" microcode program.

3. Write "0000" to the PSTEP Register (offset 003Fh) to latch the microcode instruction for execution.

4. Write the desired operation mode to the CSR ("0140h" is a typical value). See Table OVW-2 for CSR bit descriptions. Set the **run** bit (bit 1) to "0". Set the **operation enable** bit (bit 6) to "1". Set the **interrupt** bit (bit 7) to "1" to clear any interrupts. Set the **prom select** bit (bit 9) to "1" if executing upper-half microcode loads; set it to "0" for lower-half microcode loads. If using interrupts, set the desired **interrupt level** (bits 13, 14, and 15).

5. If using interrupts, write the interrupt vector to INTV_1553 (offset 000Eh).

6. Write "0000" to control register offset 0040h. Upon proper startup of the application microcode, this register will be loaded with the microcode version number.

7. Logically "OR" the CSR value from step 4 with the value "0002" and write result to the CSR. This will set the ABI to run mode.

8. Go into a software loop, checking the control register (offset 0040h) for a non-zero value. Exit the loop when the register contains a non-zero value.

9. Load data structures and control registers into ABI memory.

10. Set the CMD control register (offset 0080h) to "0001". This enables 1553 bus processing.

11. Verify that the RESP control register (offset 0081h) is incrementing. This ensures that the microcode has successfully performed startup and is currently processing 1553 data.

When this startup procedure is complete, control registers 0040h and 0041h will contain the microcode product and version numbers (see the cover page of this manual for these codes). Please have these values when calling SBS Engineering with product operation questions. As step 11 above describes, the RESP register can be checked any time after startup for an incrementing number. (Bus traffic may cause RESP to increment slowly.) If this register is not incrementing, the microcode has "crashed". This is probably due to an improper read from or write to a reserved control register or from an incorrectly programmed data structure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BUS CONTROLLER SIMULATION

The ABI is capable of full Bus Controller (BC) simulation while performing RT simulation and Sequential and Map Monitoring. BC simulation in the ABI is governed by a single data structure consisting of a linked list of command blocks. Each command block in the linked list contains encoded information for the transmission of different BC command messages. This linked list can be loaded at any location in the data structure memory area (0400h to FFFFh). Each command block of the linked list consists of eighteen words, containing the following information:

- A Type word designating the type of command block: either BC message transmission type or time delay block type.
- Two command words to be transmitted.
- Registers to store the status-response words transmitted by remote terminals at the completion of each operation.
- DMA Control Registers.
- Three Time Words.
- A pointer to a data buffer which stores the data from the bus transaction (if the command block is a time delay function, this word will contain a "time ticks" value).
- A pointer to the next block in the chain.

The host computer constructs a linked-list data structure of command blocks and loads this structure into ABI memory. It then programs register BCIPTR (offset 008Fh) with the address of the first word of the first command block. During the idle routine of the microcode 1553 program, BCIPTR is monitored. When the microcode finds a non-zero value, it uses the value as a pointer to the first command block of the linked-list. It sets BCIPTR to "0" and begins to transmit the BC command message as it is encoded in each command block. (There may be up to a 20 microsecond delay from the time the host software writes the value to BCIPTR to when the first BC transmission begins.) The host software can monitor BCIPTR to determine when BC transmission begins.

The BC chain data structure is one of the few data structures that does not need to be defined in ABI memory at microcode startup.

The ABI-V4 provides VMEbus master block transfer (DMA) capability, greatly enhancing BC simulation for the host system. Every BC block has DMA control registers which allow BC block data to be automatically transferred to/from the host system/processor (or other VMEbus memory). This function can be used to automatically transfer completed BC blocks to/from VMEbus memory at every block execution, or at the end of minor/major frames by initiating a transfer at the last block of the chain. Thus, the host system/processor can build and manage BC execution from local memory, and the ABI can provide the timing of loading/storing BC messages (and RT responses).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Control Block Structure

ABI transmitted BC command messages are specified by the host system in a series of linked-list command blocks. The host software can construct any length command block linked-list that ABI memory will allow (over 40,000 blocks if the ABI is performing only BC simulation with a four megabyte memory model). Figure BC-1 shows the data structure for BC command block linked-list chains.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

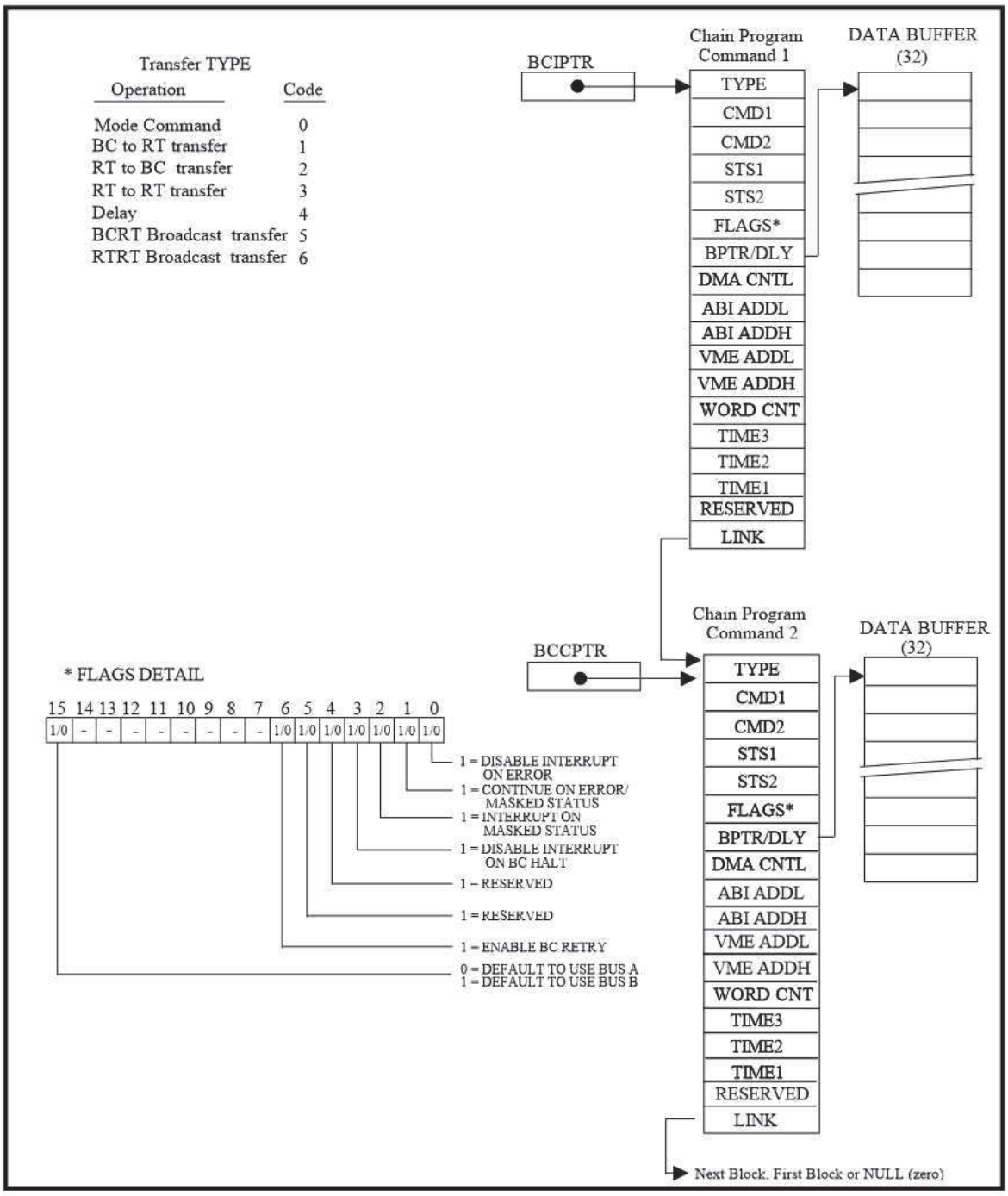


Figure BC-1 BC Control Structures

The definition of each word is presented below.

TYPE

This word holds a code which designates the type of bus-control operation to be performed (BC-RT, RT-BC, RT-RT, Broadcast, etc). The available codes are shown in Table BC-1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table BC-1 BC Type Codes	
Operation	Code
Mode Command	0
BC to RT Transfer	1
RT To BC Transfer	2
RT to RT Transfer	3
Delay	4
BC to RT Broadcast Command	5
RT to RT Broadcast Command	6

CMD 1
This is the command word associated with a particular operation, or the first command word for an RT-RT transfer.

CMD 2
This is the second command word for an RT-RT transfer.

STS 1
This is the status word returned by the remote terminal in response to an operation, or the second status word returned in an RT-RT transfer from the receive command word (CMD1).

STS 2
This is the first status word returned in an RT-RT transfer. This status word is associated with the transmit command word (CMD2).

FLAGS
This word holds a set of flags indicating the response the ABI system makes to certain bus control error conditions. The first three bits in the flag word determine how the ABI microcode treats errors and status word returns encountered during bus controller operation. The last bit controls which bus (A or B) is used for transmitting commands.

Bit 0- Disable Interrupt on Error
Set this bit to a "1" to disable interrupts on any error conditions. Set this bit to "0" (the default condition) to generate an interrupt for any error encountered by the microcode. *Note: The proper error code in the Error Table is updated regardless of the value of this bit.*

Bit 1 - Continue on Error/Masked Status
This bit determines whether the microcode ceases execution of the chain program upon encountering an error or masked status. Set to "1" to cause the microcode to

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

continue executing the chain program after encountering an error or interrupting on masked status. Set to "0" to cause the microcode to terminate the chain program upon encountering any errors or interrupting on masked status.

Bit 2 - Interrupt on Masked Status
This bit determines which status word conditions cause the bus controller portion of the microcode to interrupt the host system. Set this bit to "1" to cause status returns to the bus controller to be compared to the bus controller status word mask (BCSMSK) entry in the pointer table. If any bits set to "1" in the returned status word(s) match bits set to "1" in the bus controller status word mask, the microcode interrupts the host processor. When this occurs, the microcode reads the Continue on Error/Masked Status flag to determine whether to halt the chain program.

Bit 3 - Disable Interrupt on BC Halt
This bit disables interrupt 06h when the BC chain execution halts. Halting of a BC chain most commonly occurs due to a zero link in the last control block of a chain, but may also occur due to an error or an interrupt on masked status.

Bit 6 - Enable BC Retry
This bit enables BC retry for this block.

Bit 15 - Bus Select
This bit tells the microcode to send the command on bus A or B. Set this bit to "0" (the default setting) to send commands on bus A. Set to "1" to use bus B.

BPTR/DLY
This word points to a data buffer which provides transmitted data or accepts received data. For delay blocks, this word contains the delay interval in clock ticks. Determine the clock tick value as follows:

1. Determine the MIPS rating of the ABI: MIPS=14 for ABI-V4

2. Divide this rating into "1".

For example, if the MIPS value is 14, the clock tick value is 71.43 nanoseconds (or 1/14,000,000).

> Note: It takes 15 microseconds to process a delay block. Add this to the delay interval value (clock tick value) when determining the intermessage gap time.

DMA CNTL, ABI ADDL, ABI ADDH, VME ADDL, VME ADDH, WORD CNT
These six words control DMA operations for writing ABI memory to VMEbus memory at the end of a BC control block. Table BC-2 defines the bit values for **DMA CNTL** (VMEbus Direct Memory Access Control Word). DMA operations, if

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

selected by bit 8 of DMA CNTL, will be initiated at the end of the completed 1553 message of the current BC Block - after the RT response has been received or a No Response Time-out has occurred and the status word has been copied to the BC Block.

If an Interrupt on DMA Transfer is desired, set the DMA interrupt vector in INTV_DMA (offset 0007h) and the DMA interrupt level using bits 12, 13, and 14 of HOST DMA CNTL (offset 0001h).

Table BC-2 DMA CONTROL WORD
Bit 0 is right-most bit and least-significant.

BIT(s)/NAME	Bit Values	Comment
Bits 0-5 Reserved	Reserved - Set to "0"	These bits are reserved for ABI microcode and hardware settings - set to "0".
Bit 6 Interrupt Enable	0 = Disable Interrupt after DMA 1 = Enable Interrupt after DMA	This bit enables/disables an interrupt by the ABI to the VMEbus after a block (DMA) transfer has completed. This feature allows the ABI to notify the host processor that a DMA transfer has completed for synchronization.
Bit 7 Reserved	Reserved - Set to "0"	These bits are reserved for ABI microcode and hardware settings - set to "0".
Bit 8 Master Block Transfer (DMA) Enable/Disable	0 = Disable DMA Function 1 = Enable DMA Transfer upon buffer full.	This bit determines if the ABI will perform a DMA transfer of the respective sequential monitor buffer when it is filled.
Bit 9 Direction	0 = Transfer from ABI to VME 1 = Transfer from VME to ABI	This bit controls the direction of master transfers by the ABI. Set to "0" to cause the ABI to transfer data from ABI ADDH/L to VME ADDH/L. Set to "1" to cause the ABI to perform a master transfer from VME ADDH/L memory to ABI ADDH/L.
Bits 10,11 Bus Request Level	00 = Bus Request Level 0 01 = Bus Request Level 1 10 = Bus Request Level 2 11 = Bus Request Level 3	These bits control the bus level for the VME transfer. "11" is typically the highest, but this setting is dependent on the bus control settings/protocol of the slot 0 VME arbiter.
Bit 12 Address Width	0 = A32 1 = A24	This bit selects either a 32 or 24 bit address width for the VME transfer. The VME address corresponds to the 32 or 24 least significant bits of VME ADDH/L, respectively.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table BC-2 DMA CONTROL WORD Bit 0 is right-most bit and least-significant.		
BIT(s)/NAME	**Bit Values**	**Comment**
Bit 13 Data Width	0 = D32 1 = D64	This bit controls the width of data transfers to/from VMEbus memory. The WORD CNT value is in increments of this bit setting - i.e., if bit 5 is set to "0" and WORD CNT is 16, then 16 32-bit transfers will occur.
Bit 14 Supervisor/Non Privileged Access	0 = Non Privileged (NORMAL) 1 = Supervisory	This bit controls memory address modifiers. The ABI-V4 responds to either mode.
Bit 15 Address Increment	0 = Increment VME Address 1 = No Address Increment	This bit controls automatic VME address increments for ABI block transfers. In short, this bit allows for transfers to/from regular memory or FIFOs. The amount the address is incremented depends on the setting of bit 5; a D32 transfer increments the VME address by 4 and a D64 transfer increments the address by 8.

ABI ADDL and **ABI ADDH** (ABI Source Address Low and High) define the ABI source address for DMA operation to the VME bus. These two words combine to make up a 21-bit ABI address (includes all bits of ABI ADDL and bits 0-4 of ABI ADDH); bits 5-31 of ABI ADDH are not used and should be set to zero. The source address can be any ABI local address. (ABI addresses are in words, not bytes.)

DMA transfers to the VMEbus can be in 32-bit or 64-bit words (two and four ABI words, respectively). When performing a D32 DMA transfer to the VMEbus, the ABI hardware will increment the ABI ADDL/H value by two for every transfer. For D64 transfers, the ABI will increment the ABI ADDL/H value by four after every transfer. With D32 DMA transfers, the ABI ADDL/H value must lie on a two word ABI address boundary (bit 0 must be "0"). D64 transfers require that the ABI ADDL/H address value start on a four word ABI address boundary (bits 0 and 1 must be "0"). These address requirements must be considered when programming BC data structures in ABI memory (make sure the starting address, ABI ADDL/H, falls on a two or four boundary as required for D32 or D64 transfers, respectively). ABI memory (like 1553 words) is organized as 16-bit words and DMA transfers are in 32 or 64 bit word lengths.

NOTE: Since a BC block is 50 words in length, the host may want to "pad" the block with either two or six fill words (for D32 and D64 transfers, respectively) in order to preserve address boundaries for long and double long word transfers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VME ADDL and **VME ADDH** (VMEbus Address Low and High) combine to make the VMEbus destination address words (24 or 32 bits as defined by bit 4 of DMA CNTL). VME memory, as with ABI memory, is big-endian oriented. Most VMEbus processor or reflective memory modules are byte oriented; thus, D32 or D64 transfers must start on a four byte or eight byte boundary.

WORD CNT is the number of transfers to be performed over the VME bus, minus one. The count value is for either D64 or D32 words, as designated by bit 5 of the DMA CNTL word. For example, if WORD CNT is 127 and bit 5 is set to "1" (selecting D64), 128 64-bit words will be transferred. Thus, the word count is not a byte count, but a VME transfer count. The minimum transfer value is zero and the maximum is 65535.

D64 transfers may consist of one to 65536 64-bit words (512K bytes), but the ABI must allow for VMEbus arbitration at 2K byte address boundaries (as defined by the VME64 specification). D32 DMA transfers may consist of one to 65536 32-bit words (256K bytes), but the ABI must allow for VMEbus arbitration at 256 byte address boundaries (as defined by the VME specification). These boundaries should be kept in mind when defining ABI data structures for maximum efficiency of VMEbus transfers (although the delay time through an address boundary is typically less than one microsecond). The ABI will only relinquish the VMEbus if a higher priority bus request is pending (which means the VMEbus arbiter has BCLR* {Bus Clear} pending).

DMA Options

One possible automatic DMA transfer scenario may be the storing and downloading of BC minor frames during the dead-bus period usually found at the end of a minor frame. The last BC block of the minor frame may invoke a DMA transfer from the ABI to VMEbus memory to store the entire minor frame to host/processor memory. Then, a dummy BC block could follow that would invoke the automatic transfer of the next BC minor frame into ABI memory. (A dummy block is a delay block whose value is large enough for the storing DMA function over the VMEbus.) The user's application program would need to manage the last BC block's link pointer (the last link pointer must not be overwritten if an endless loop is set up), or the restarting of BC block execution through the resetting of BCIPTR. (See the paragraphs following the "Link" description).

Another DMA option may be the transfer of each BC block to VMEbus memory as it is executed.

TIME3, TIME2, TIME1

These three words make up a 48 bit time stamp for each BC Block. The time value is taken from the ABI hardware clock and is updated within two microseconds of when the first command word of the BC Block has started transmission. **TIME3** is the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

most significant word, followed by **TIME2** and **TIME1**. The ABI clock is programmable for 1 or 32 microsecond resolution, can be driven from an internal or external source, and may be preset as described in the "Introduction: ABI Operations Overview" section (See Table OVW-1). If you ever see bit 45 roll over, you have been doing your job way too long.

LINK
This word is a link pointer to the next command block in the chain program. A zero value indicates the end of the linked-list.

During bus controller operation, the microcode decodes each block of the linked-list program and takes the appropriate action. If a BC-RT, RT-BC, or mode type is indicated, CMD1 is transmitted on the bus and any resulting status is placed in STS1. BPTR/DLY defines the buffer from which data is transmitted or into which data is received. If an RT-RT transfer is specified, then CMD1, CMD2, STS1, and STS2 are used. BPTR/DLY is not used when RT-RT is specified.

Controlling BC Operation - BCIPTR, BCCPTR, BCLPTR

To operate the ABI in bus controller mode, set up the appropriate chain and data buffers in ABI RAM. Initiate chain program execution by loading BCIPTR (offset 008Fh) with the offset of the first command block. As soon as the ABI begins to execute the chain program, it clears the contents of BCIPTR. While the chain program is executing, the ABI updates the BCCPTR entry in the Control Register table (offset 0090h) so that it points to the chain program block currently being executed. Execution of the chain program continues until a "0" is encountered in the LINK word of a chain program command block. If the LINK in the last command block of the chain program points back to the first command block of the linked-list, the ABI operates in a continuous-loop mode. Halt or redirect chain program execution during operation by entering a new value in the LINK word of a particular command block, prior to execution of that block. Enter a "0" to halt execution or a different offset to change the execution path. This would allow special BC messages to be inserted into the minor frame for cases such as the service requested bit being set by the RT Status Word. The control register BCLPTR (offset 0091h) points to the last BC block that was executed.

Status Word Mask - BCSMSK

The control register BCSMSK (offset 0092h) is used as a status word filter to allow the BC to interrupt when a certain status response is received. This feature is based on two values: 1) bit 2 in the FLAG word of the BC control block and 2) the entry in BCSMSK. When bit 2 of the FLAG word is set to "1", the Returned Status word is logically "ANDed" with the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSMSK word. If the resulting value contains any bits set to "1", an interrupt is generated. The microcode then reads the Continue On Error/Masked Status flag to determine whether to halt the chain program.

BC Retries - BRTCNT, BRTBUS, BRTCMD, BRTRTC

A BC program block can be set up to retry the message if there is an error in the RT response. The retry feature is enabled by setting bit 6 of the FLAGS word (in the BC block shown in Figure BC-1) to "1". There are four control registers for BC retries: BRTCNT (offset 00AFh), BRTBUS (offset 00B0h), BRTCMD (offset 00B1h), and BRTRTC (offset 00B2h). BRTCNT specifies the maximum number of retries (a maximum of 16 is allowed).

BRTBUS is a word which defines which bus (A or B) will be used for the retried message. Each bit of this word pertains to a particular retried message: the first or least-significant bit applies to the first retry of the message, the second bit applies to the second retry of the message, and so on, up to the bit which applies to the number of retry messages defined by BRTCNT. If a bit is set to "0", the associated retry of the message is retried on the same bus as the original message from which the error was detected. If a bit is set to "1", the associated retry of the message is sent on the alternate bus. For example, if the first message was sent on the primary bus and the retry bit is a "1", the retry of the message will be on the secondary bus.

BRTCMD contains the command that was last retried. The value is set to "0" at the start of each new BC chain.

BRTRTC contains the actual number of retries. It is set to "0" at the start of each new BC chain.

NOTE: Retries are always attempted before the state of the Continue on Error/Masked Status bit (bit 1) in the BC FLAG word is checked. Therefore, if bit 1 is set to "0", retries will be attempted before execution of the BC chain is halted; if bit 1 is "1", retries will be attempted and normal BC chain execution will continue.

Mode Codes

The ABI processes mode codes in the same way a normal bus controller handles them. For mode code commands without data words, the microcode simply transmits the command word. For mode code commands with data words, the data word is transmitted or stored as the first word of the command block data word buffer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

REMOTE TERMINAL SIMULATION

This subsection of the manual reviews the key data structures required for ABI RT simulation. The ABI is capable of simulating all 31 possible remote terminals (RT) while performing BC and Monitoring functions.

The ABI provides great flexibility for RT simulation and real-time monitoring. Linked-lists of data word buffers are used for both the transmission of RT messages and receiving of BC transmitted messages. The linked-list buffers may be chained indefinitely, limited only by the available ABI memory. (If the ABI is performing only RT simulation, nearly the entire 64K-2M word memory range is available.) The ABI also provides interrupt capabilities for each of the 31 RT addresses and their related 64 subaddresses (32 transmit and 32 receive buffers). This provides the host system with real-time access to RT data buffers for the most complex simulation applications.

DMA (VME block master) operations allow single or entire RT buffer lists to be transferred automatically to/from host memory. This can minimize host VME access, and allow host operations (reading and updating RT buffers) to remain in local memory for efficient processor performance.

ABI Control Registers BCIGP, RSPGPA and RSPGPS may be used to change the **intermessage and status response gap times** and are described at the end of this subsection.

Note: There are areas in this subsection which overlap the "Bus Monitoring" subsection. RT Map Monitoring utilizes the same linked-list buffers as RT Simulation for real-time storage of data words. This subsection discusses these buffers as they relate to RT Map Monitoring. Please also read the "Bus Monitoring" subsection for a complete understanding of Map Monitoring functions.

The key data structures for RT Simulation are:

- **The Status Word Table** which enables desired RTs for simulation and stores status response information.
- **The RT Address Data Structure** which enables the status responses for the RTs enabled in the Status Word Table and provides the leading address for the linked-list buffers (for both RT simulation and RT Map Monitoring).
- **The Filter Table** which provides interrupt enabling, parity error generation, and Sequential Monitoring control bits (see "Bus Monitoring" subsection).

Other important data structures include:

- The **Protocol Table** which selects different RT responses, such as "1553A subaddress 31", Dynamic Bus Control Accept, etc.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The **Mode Code Tables** which contain dedicated data structures for data word mode codes.

This subsection details each of these data structures.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Status Word Table - Enabling RT Simulation

The Status Word Table is a 64-word table containing two entries for each of the possible 32 remote terminals being simulated. It provides the ABI microcode with information about the remote terminals currently being simulated, and stores information used during a Remote Terminal Status Response. The starting offset of the Status Word Table is loaded by the host software into the SWTPTR control register (offset 0088h). Figure RT-1 shows the basic structure of this table and the bit definitions of a status word.

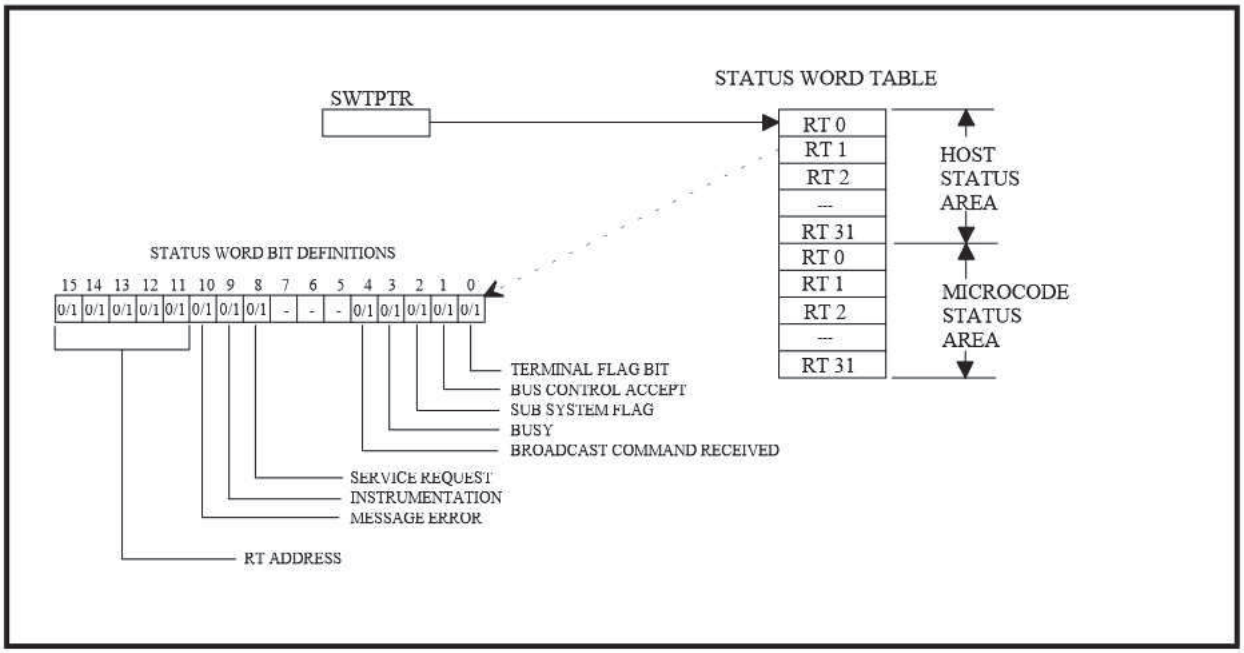


Figure RT-1 RT Status Word Table (SWTPTR)

One RT entry is contained in the **host** status area and the other in the **microcode** status area. The status word in the host status area is initialized and maintained by the host software. **The status word in the microcode status area is maintained by ABI microcode and should not be modified by host software.**

The **host** status word performs two functions: 1) It indicates to the ABI that a particular RT is active if the associated host status word contains a nonzero entry, and 2) It is logically "ORed" with the microcode status word to obtain the transmitted status response word. Any bit set to "1" in the host status word will be set to "1" in the transmitted **status response** word (with the exception of the **message error** bit, described below).

Normally, the host status word only contains the address for a given RT. A nonzero address value must be loaded into the table to enable a particular RT. *To enable RT 0, the host sets the message error bit (in the host status word) to "1".* This action does not affect the status response word because the message error bit actually transmitted is taken from the microcode status area.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The other bits in the status response word are assigned a value by a logical "OR" operation involving the host status word and the microcode status word. Therefore, the host may force any bit in the status response word to "1" by setting it to "1" in the host status word (with the exception of the message error bit). An erroneous RT address could be entered, for example. The function of each bit in the status response word and a description of how the bits are generated is presented below.

Note: Those bits which are normally set by the hardware during 1553 operation are handled by the ABI microcode. Those bits normally set by sub-system software during 1553 operation are handled by the host.

Status Word Response Bits

Bit 0 - Terminal Flag Bit

The host software sets this bit to "1" to indicate a "terminal-failed" condition. The ABI microcode also controls this bit when responding to Inhibit Terminal Flag Bit and Override Inhibit Terminal Flag Bit mode commands. The microcode responds to an Inhibit Terminal Flag Bit mode command by sending a status word containing a terminal flag bit set to "0", regardless of the actual status of this bit. The microcode responds to a subsequent Override Inhibit Terminal Flag Bit by sending a status word containing a terminal flag bit with the actual state of this bit in the microcode status word table. Please see "RT Phase Data Structure", under "RT Mode Code Simulation" in this subsection, for more details.

Bit 1 - Bus Control Accept Bit

This bit is set by the ABI microcode in response to a Dynamic Bus Control mode command. (The state of the bit is determined by the state of bit 2 in the Protocol Table entry for the particular RT. If the bit is set to "1" in the Protocol Table, the **bus control accept** bit will be set to "1" in the status response word.) The host software must also respond to this mode command by enabling an interrupt so that it may initiate a bus controller program. Both of these actions must take place in order for Dynamic Bus Control to be accepted.

Bit 2 - Subsystem Flag Bit

This bit is set to "1" by the host software to indicate a potentially failed subsystem. It is never set by the ABI microcode.

Bit 3 - Busy

This bit is set to "1" by the host software to indicate a "terminal-busy" condition. It is never set by the ABI microcode. When this bit is set to "1", the RT will not transmit any data words, only status words.

If the busy bit is set in the status word of a receiving RT involved in a BC-RT or RT-RT message, data will be stored in the RT data buffer. Upon successful completion

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

of the BC-RT or RT-RT message, the RT will transmit a status word with the busy bit set.

When the busy bit is set in the status word of a transmitting RT, the resulting action depends on the message type. For both an RT-BC and RT-RT message, the RT will transmit its status response with the busy bit set and will suppress the transmission of data words. For an RT-RT message, error code 120h, unexpected gap, will be generated with respect to both the BC and receiving RT. For RTs involved in mode codes, the transmission of an associated data word will be suppressed.

Bit 4 - Broadcast Command Received Bit
On receipt of a broadcast command, the ABI microcode sets this bit to "1" for all active RT simulations. In response to a Transmit Status Word and/or Transmit Last Command mode code immediately following a broadcast command, the microcode sets this bit to "1" in the RT's status response word. Otherwise, the microcode resets this bit to "0".

Bit 8 - Service Request Bit
This bit in the host status word is set to "1" by the host software; thereafter, this bit appears in each status response word until it is cleared by the host software. The host software should respond by enabling an interrupt on a Transmit Vector Word mode command and clearing this bit (in the host status word) as part of the interrupt service routine.

Bit 9 - Instrumentation Bit
This bit is used by the host software for specialized applications. It is never set by the ABI microcode.

Bit 10 - Message Error Bit
This bit indicates that the previous message received by the RT did not pass all validity checks. It remains set to "1" in the status word table until the next message is received, at which time it is updated to reflect the validity of that message. It is set only by the ABI microcode. The proper code in the Error Table is updated when an invalid message is detected.

RT Address Data Structures

Defining Data Word Buffers

The RT Address Data Structures consist of the Address Table, Buffer Pointer Table, and linked data buffers. The control register ATPTR (offset 0089h) contains the address of the first word of the Address Table. The Address Table provides addresses to the Buffer Pointer Tables associated with each RT. The Buffer Pointer Table gives the location of the first linked-list buffer for each RT subaddress (receive and transmit). The size of these linked-lists

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

are limited only by the available ABI memory. Figure RT-2 illustrates the RT Address Data Structures.

These RT Address Data Structures are used for both RT Simulation and real-time Map Monitoring. The Address Data Structures are programmed identically for both simulation and monitoring functions, and the ABI can simulate and Map Monitor different RT messages within these data structures simultaneously.

The same RT message cannot be both simulated and Map Monitored. (This would not be a useful function because the data words would just be duplicated.) Sequential Monitoring provides for self-monitoring of simulated and external messages. The following paragraphs explain programming the ABI for RT message simulation. However, the information applies to setup for RT Map Monitoring, including Broadcast Commands with receive data words.

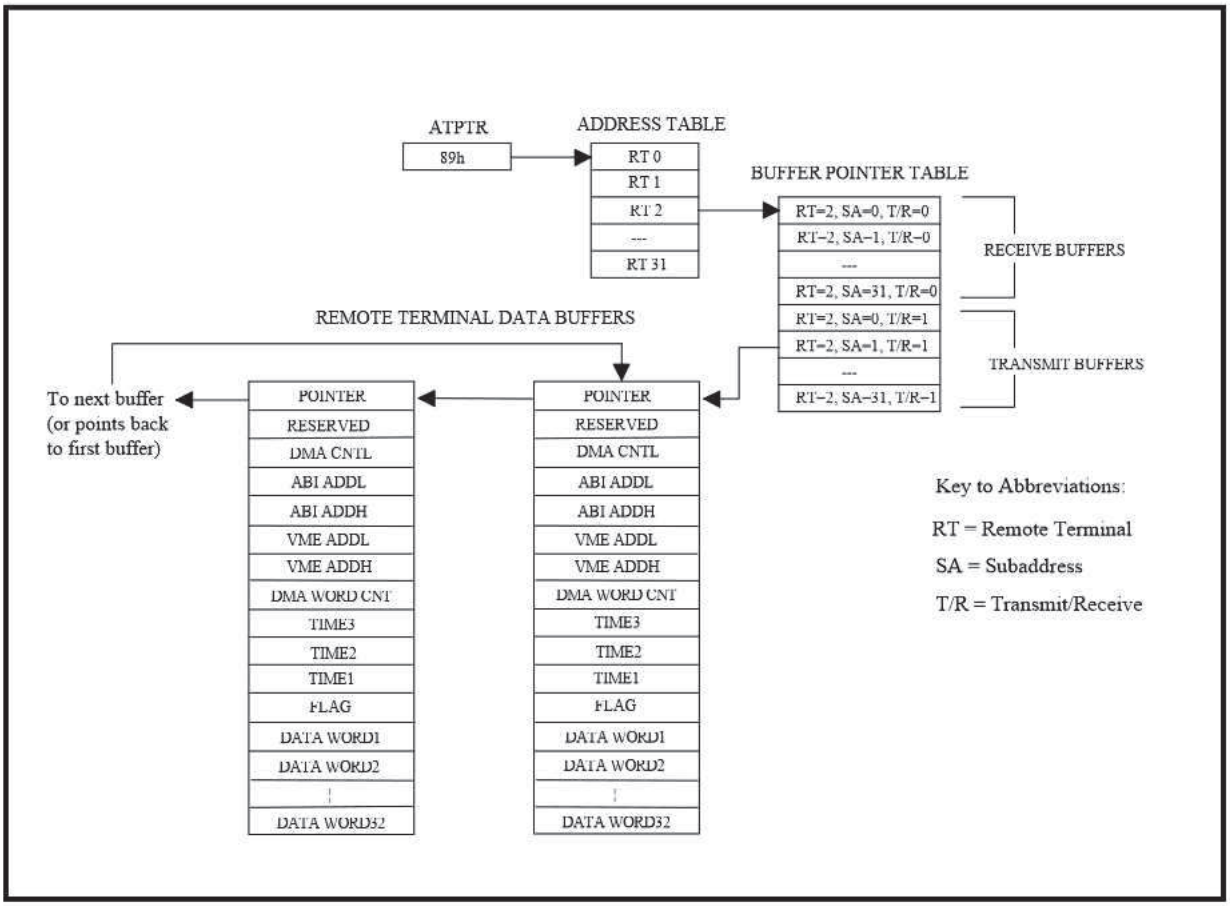


Figure RT-2 RT Address Data Structures

Address Table

The Address Table is 32 words long and holds the Buffer Pointer Table address for each of the 32 Remote Terminals. The Buffer Pointer Table address of an RT must be programmed into the address table prior to RT Simulation or Map Monitoring. An RT not being simulated or

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

monitored must have a zero value programmed into its respective element in the address table. When the ABI microcode detects a 1553 command word on the bus, it checks the buffer pointer table location (respective to the command word terminal address) programmed into the address table for a nonzero value. If the value is "0", the ABI will not respond with status or data words. If there are no data buffers defined and the appropriate element of this structure is "0", the ABI will not respond with a status word.

For example, if RT 6 is to be simulated, element 7 of the Address Table must contain the address to the Buffer Pointer Table for RT 6. If RT 6 is the only RT to be simulated, all other elements must have a value of "0".

If there are no data buffers defined and the appropriate element of this structure is "0", the ABI will not respond with a status word (in Map Monitor mode, the microcode will simply ignore the message and not store the command or data words).

Buffer Pointer Table

The Buffer Pointer Table has one entry for each possible receive subaddress and one for each possible transmit subaddress. The first 32 pointers correspond to receive (T/R bit = 0) subaddresses 0-31, and the second 32 pointers correspond to transmit (T/R bit = 1) subaddresses 0-31. Each of the 64 elements may point to the beginning of a linked-list of data buffers which is used to transmit or receive the RT bus data words. A "null" entry in this table indicates that the corresponding subaddress is not active for this particular RT (no data buffers are defined).

Data Buffer Format

Each data buffer consists of 44 words of memory. The last 32 of these 44 words are used to store RT message data words. The first twelve words provide special features.

POINTER
The first word is a **pointer** that may provide the address to another data buffer. The link of buffers joined together by the pointer words may be as large as ABI memory allows. Usually the last data buffer points back to the first data buffer to create a circular chain. Figure RT-7 shows two buffers that alternate in this manner. If only one buffer is required, which is often the case, the pointer must contain its own address (points back to itself). After transmitting data words to a buffer, or receiving data words that are stored in a buffer, the ABI microcode copies the Data Buffer Pointer address into the respective Buffer Pointer Table location. The ABI microcode always uses the address stored in the Buffer Pointer Table as the target address for storage or transmission. This allows the host computer to monitor the Buffer Pointer Table and determine the current location of the microcode in the data buffer linked-list.

DMA CNTL, ABI ADDL, ABI ADDH, VME ADDL, VME ADDH, WORD CNT
These six words control DMA operations for writing/reading ABI RT data buffers to/from VMEbus memory. Table RT-1 defines the bit values for **DMA CNTL**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

(VMEbus Direct Memory Access Control Word). DMA operations, if selected by bit 8 of DMA CNTL, will be initiated upon the receipt of an error free 1553 message (at the end of a message including an ABI Status/Data Word response if appropriate).

If an Interrupt on DMA Transfer is desired, set the DMA interrupt vector in INTV_DMA (offset 0007h) and the DMA interrupt level using bits 12, 13, and 14 of HOST DMA CNTL (offset 0001h).

Table RT-1 DMA CONTROL WORD
Bit 0 is right-most bit and least-significant.

BIT(s)/NAME	Bit Values	Comment
Bits 0-5 Reserved	Reserved - Set to "0"	These bits are reserved for ABI microcode and hardware settings - set to "0".
Bit 6 Interrupt Enable	0 = Disable Interrupt after DMA 1 = Enable Interrupt after DMA	This bit enables/disables an interrupt by the ABI to the VMEbus after a block (DMA) transfer has completed. This feature allows the ABI to notify the host processor that a DMA transfer has completed for synchronization.
Bit 7 Reserved	Reserved - Set to "0"	These bits are reserved for ABI microcode and hardware settings - set to "0".
Bit 8 Master Block Transfer (DMA) Enable/Disable	0 = Disable DMA Function 1 = Enable DMA Transfer upon buffer full.	This bit determines if the ABI will perform a DMA transfer of the respective sequential monitor buffer when it is filled.
Bit 9 Direction	0 = Transfer from ABI to VME 1 = Transfer from VME to ABI	This bit controls the direction of master transfers by the ABI. Set to "0" to cause the ABI to transfer data from ABI ADDH/L to VME ADDH/L. Set to "1" to cause the ABI to perform a master transfer from VME ADDH/L memory to ABI ADDH/L.
Bits 10,11 Bus Request Level	00 = Bus Request Level 0 01 = Bus Request Level 1 10 = Bus Request Level 2 11 = Bus Request Level 3	These bits control the bus level for the VME transfer. "11" is typically the highest, but this setting is dependent on the bus control settings/protocol of the slot 0 VME arbiter.
Bit 12 Address Width	0 = A32 1 = A24	This bit selects either a 32 or 24 bit address width for the VME transfer. The VME address corresponds to the 32 or 24 least significant bits of VME ADDH/L, respectively.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table RT-1 DMA CONTROL WORD Bit 0 is right-most bit and least-significant.		
BIT(s)/NAME	**Bit Values**	**Comment**
Bit 13 Data Width	0 = D32 1 = D64	This bit controls the width of data transfers to/from VMEbus memory. The WORD CNT value is in increments of this bit setting - i.e., if bit 5 is set to "0" and WORD CNT is 16, then 16 32-bit transfers will occur.
Bit 14 Supervisor/Non Privileged Access	0 = Non Privileged (NORMAL) 1 = Supervisory	This bit controls memory address modifiers. The ABI-V4 responds to either mode.
Bit 15 Address Increment	0 = Increment VME Address 1 = No Address Increment	This bit controls automatic VME address increments for ABI block transfers. In short, this bit allows for transfers to/from regular memory or FIFOs. The amount the address is incremented depends on the setting of bit 5; a D32 transfer increments the VME address by 4 and a D64 transfer increments the address by 8.

ABI ADDL and **ABI ADDH** (ABI Address Low and High) words define the ABI source/destination address for a DMA operation to/from the VME bus. These two words combine to make up a 21-bit ABI address (includes all bits of ABI ADDL and bits 0-4 of ABI ADDH); bits 5-31 of ABI ADDH are not used and should be set to zero. Bit 9 of the DMA CNTL word determines the direction of master transfers by the ABI and thus determines whether this address is a source or destination address. If bit 9 of the DMA CNTL word is "0", the RT data buffer is a receive buffer and the user will probably want the ABI address to represent the *source* address in ABI memory (the beginning memory location from where ABI memory will be transferred to VME memory). If bit 9 is "1", the RT data buffer is a transmit buffer and ABI ADDL/H will probably hold the *destination* address for a DMA transfer from VME memory to ABI memory.

The user may wish to transfer one RT data buffer at a time or may transfer large blocks of RT data buffers in one DMA transfer at the end of a minor frame. ABI memory (like 1553 words) is organized as 16-bit words and DMA transfers are in 32 or 64-bit word lengths. The ABI ADDL/H is a 16-bit word address.

VME ADDL and **VME ADDH** (VMEbus Address Low and High) combine to make the VMEbus destination address words (24 or 32 bits as defined by bit 4 of DMA CNTL). VME memory, as with ABI memory, is big-endian oriented. Most VMEbus processor or reflective memory modules are byte oriented; thus, D32 or D64 transfers must start on a four byte or eight byte boundary. Bit 9 of the DMA CNTL word determines if the ABI master transfer is to or from ABI memory. If bit 9 is set to "0",

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

transfers are to VME ADDL/H from ABI ADDL/H, and if bit 9 is set to "1", transfers are from VME ADDL/H to ABI ADDL/H.

WORD CNT is the number of transfers to be performed over the VME bus, minus one. The count value is for D64 and D32 words as designated by bit 5 of the DMA CNTL word. For example, if this word value is 127 and bit 5 is set to "1" (D64), 128 64-bit words will be transferred. Thus, the word count is not a byte count, but a VME transfer count. The minimum transfer value is zero and the maximum is 65535.

D64 transfers may consist of one to 65536 64-bit words (512K bytes), but the ABI must allow for VMEbus arbitration at 2K byte address boundaries (as defined by the VME64 specification). D32 DMA transfers may consist of one to 65536 32-bit words (256K bytes), but the ABI must allow for VMEbus arbitration at 256 byte address boundaries (as defined by the VME specification). These boundaries should be kept in mind when defining ABI data structures for maximum efficiency of VMEbus transfers (although the delay time through an address boundary is typically less than one microsecond). The ABI will only relinquish the VMEbus if a higher priority bus request is pending (which means the VMEbus arbiter has BCLR* - Bus Clear - pending).

TIME3, TIME2, TIME1

These three words make up a 48-bit time stamp for each BC Block. The time value is taken from the ABI hardware clock and is updated within two microseconds of when the first command word of the BC Block has started transmission. TIME3 is the most significant word, followed by TIME2 and TIME1. The ABI clock is programmable for 1 or 32 microsecond resolution, can be driven from an internal or external source, and may be preset as described in the "Introduction: ABI Operations Overview" section (See Table OVW-1). If you ever see bit 45 roll over, you have been doing your job way too long.

FLAG WORD

After a valid message transfer is complete, the **flag** word is loaded with the command word portion of the message. If the message is not valid, this word will be set to "0" (in most cases the data words will be written to the buffer upon error, but the microcode cannot guarantee proper values or word count). The host program should set the flag word to "0" after reading the buffer. This permits the programmer to determine whether new data has been received at this subaddress by looking for a non-zero value in the flag word. For variable-length messages (less than 32 words), the word count field of the command word (flag word) will tell the host system the number of valid data words in the buffer.

Broadcast Considerations

To store the data from a broadcast command, define a subaddress data buffer for RT 31 and enable the map monitor bit (bit 1) for RT 31 in the Protocol Table. Data buffers defined for

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RT 31 only function as map monitor data buffers for broadcast commands (see the "Bus Monitoring" subsection of the manual for more information on map monitoring). Data from a broadcast command is never placed in the corresponding subaddress data buffers of RTs 0 - 30.

For example, assume that RT2/r/SA4 and RT4/r/SA4 are being simulated in ABI memory and each RT/SA has a receive data buffer defined for subaddress 4. Also, assume that a data buffer has been allocated for RT31/r/SA4 and map monitoring has been enabled for RT 31. If a BC-RT broadcast command is sent on the bus, the data from that bus transfer will be found in RT31/r/SA4's data buffer. The data will not be present in the subaddress data buffers of RT 2 or RT 4.

Minimum Programming Requirements for the RT Address Data Structures

The ATPTR must be programmed to point to an Address Table of 32 elements. The Address Table does not have to point to any Buffer Pointer Tables (if the ABI is not simulating or Map Monitoring Remote Terminals, all the Address Table elements will have NULL values).

Filter Table

The Filter Table Data Structures provide setup flags for Message Interrupt, Sequential Monitoring, and parity error generation for simulated RT messages. Figure RT-3 illustrates the Filter Table Data Structures.

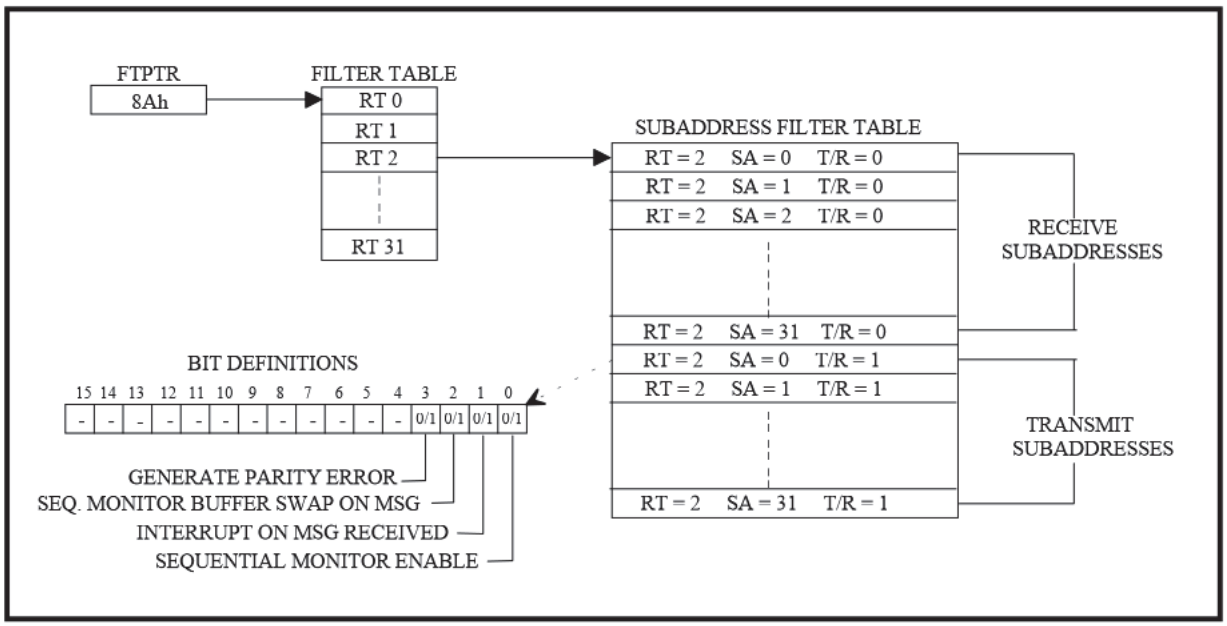


Figure RT-3 Filter Table Data Structures

The control register FTPTR (offset 8Ah) holds the address of the first word of the Filter Table array. The Filter Table provides 32 pointer elements, one for each RT address. Each element

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

points to an array of 64 flag words called the Subaddress Filter Table which provides operation flag bits for all receive and transmit subaddresses. This provides operational control to the subaddress level of each RT.

The following bits may be programmed for each receive and transmit flag word in the Subaddress Filter Table. Bits RT-15 should be set to "0".

Bit 0 - Sequential Monitor Enable
If this bit is set to "1", data to or from this RT and subaddress combination is placed in the Sequential Monitor Buffer. This data includes command, status, data, and time stamp words.

Bit 1 - Interrupt on Message Received
If this bit is set to "1", an interrupt occurs when a command word from a simulated or external Bus Controller is detected for this RT/subaddress combination. The interrupt will occur after the message is completely received.

Bit 2 - Sequential Monitor Buffer Swap on Message
If this bit is set to "1", message activity to or from this RT/subaddress combination will cause the ABI microcode to first swap the Sequential Monitor Buffers and then generate a hardware interrupt (code 0010h). If message activity to or from this RT/subaddress combination is also being sequentially monitored (bit 0 = 1), the following actions will take place: 1) the message for this RT/subaddress combination will be placed in the active buffer as the last message, 2) the monitor buffers pointers will be swapped. Following the swap, a hardware interrupt code 0010h will be generated.

Bit 3 - Generate Parity Error
If this bit is set to "1", this simulated RT/subaddress combination will respond with a parity error. If this RT responds with only a status word, the status word will contain a parity error. If this RT responds with a status word and data word(s), the last word will contain a parity error.

Minimum Programming Requirements for the Filter Table

The FTPTR must be programmed to point to the first word of the Filter Table. Each pointer element of the Filter Table must point to the first word of the 64 word Subaddress Table array. None of the bits of the Subaddress Filter Table have to be set to "1"; if no flag bit operations are desired for the respective RT subaddress, set all bits to "0".

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Protocol Table

The Protocol Table allows the user to program ABI 1553 protocol characteristics and to select which Remote Terminals will be monitored in the RT Address Data Structures for Map Monitoring. Figure RT-4 illustrates the Protocol Table.

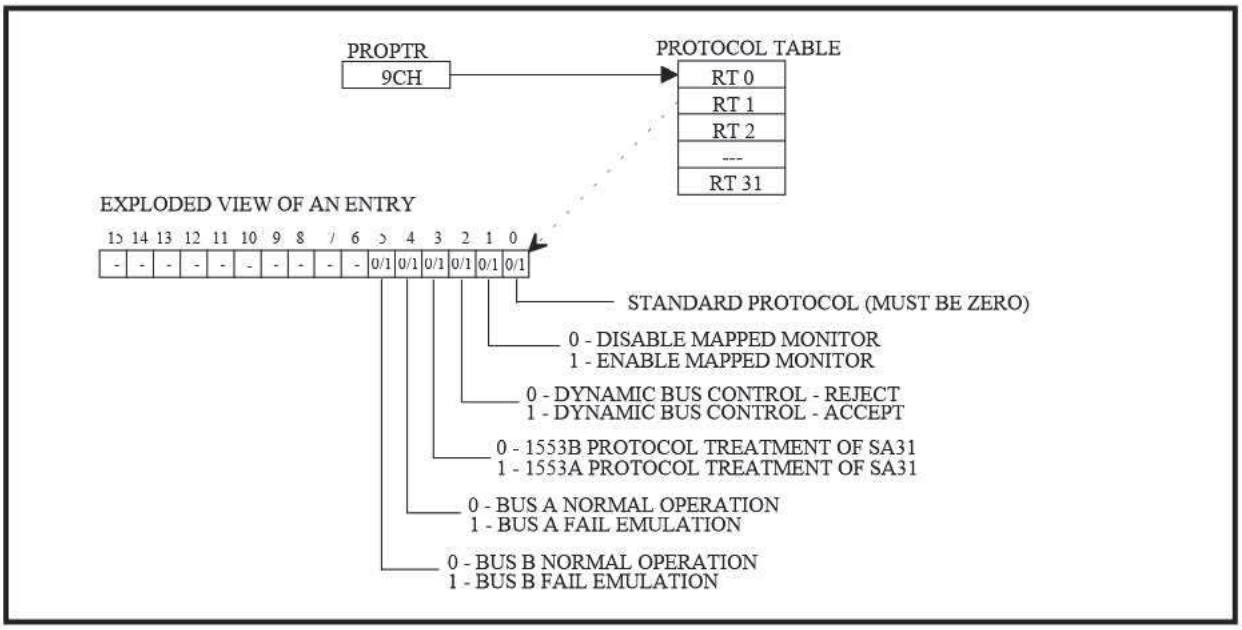


Figure RT-4 Protocol Table

The Control Register PROPTR (offset 009Ch) must be programmed with the address of the first word of the 32-word Protocol Table. Each element corresponds to an RT address and functions as a flag word. Bits can be set to control ABI protocol changes or enable Map Monitoring at the RT level. The following paragraphs describe flag bit options.

Bit 0 - Standard Protocol
Must always be set to "0".

Bit 1 - Mapped Monitor Enable
Set this bit to "1" to enable Map Monitoring for the respective RT. This causes the ABI to store received RT messages in the appropriate RT data buffer of the RT Address Data Structures. Set this bit to "0" to disable Map Monitoring for the respective RT. If the host program attempts to simulate and Map Monitor the same RT (the Status Word Table is programmed for the same RT), the Simulation function will override the Map Monitoring function.

Bit 2 - Dynamic Bus Control
This bit affects the status response to a Dynamic Bus Control Acceptance mode command. This bit is copied to the associated Status Word bit (bit 1) before the Status

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Word is transmitted for simulated RT responses. Please see the Mode Code discussion later in this subsection for other Mode Code program options.

Bit 3 - Treatment of SA31
If this bit is set to "0", the ABI microcode handles any subaddress equal to 31 as a Mode Code command per 1553B specifications. If this bit is set to "1", the microcode treats subaddress value of 31 as a normal subaddress (non-Mode Code) per the 1553A specification.

Bit 4 - Bus A Operation
If this bit is set to "0", the simulated RT will respond properly to messages received on bus A. If this bit is set to "1", the RT will not respond to messages received on bus A.

Bit 5 - Bus B Operation
If this bit is set to "0", the simulated RT will respond properly to messages received on bus B. If this bit is set to "1", the RT will not respond to messages received on bus B.

Minimum Programming Requirements for the Protocol Table

The PROPTR pointer must be programmed with the address of the first element of the 32-word Protocol Table array. The Protocol Table flag words are programmed as required by the 1553 system. Set all flag words to "0000h" for normal 1553B operations.

RT Mode Code Simulation

The ABI responds properly to all Mode Code commands in the RT Simulation mode. For some Mode Codes, a proper response is the simple transmission of a Status Word. Other Mode Codes require actions such as the transmission or reception of data words, the setting of status word bits, etc. For many Mode Code operations, the ABI provides special data structures to store data words or other operation words. Table RT-2 lists 1553B Mode Codes, explains the function of each and indicates which have associated data words.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table RT-2 ABI Mode Code Operations				
Mode Code	Transmit / Receive Bit	Associated Data Word	Broadcast Command Allowed?	Function
00000	1	No	No	Dynamic Bus Control
00001	1	No	Yes	Synchronize
00010	*1	No	No	Transmit Status Word
00011	1	No	Yes	Initiate Self-Test
00100	1	No	Yes	Transmitter Shutdown
00101	*1	No	Yes	Override Transmitter Shutdown
00110	1	No	Yes	Inhibit Terminal Flag Bit
00111	1	No	Yes	Override Inhibit Terminal Flag Bit
01000	*1	No	Yes	Reset RT
10000	1	Yes	No	Transmit Vector Word
10001	0	Yes	Yes	Synchronize
10010	1	Yes	No	Transmit Last Command
10011	1	Yes	No	Transmit Bit Word
10100	0	Yes	Yes	Selected Transmitter Shutdown
10101	0	Yes	Yes	Override Selected Transmitter Shutdown
* Minimum Required Mode Codes				

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Mode Code Data Structures Required for ABI Operations

Last Status Word Data Structure

The status word the ABI microcode sends in response to a valid command is stored in a 32-word Last Status Word table. These status words are stored in a location corresponding to the particular RT involved in the transmission. Each time a response to a valid command is made, this table is updated to reflect the most current response. The LSWPTR control register (offset 0099h) holds the starting address of this table, as shown in Figure RT-5. The host software must allocate memory space for the table but the buffer contents must only be updated by the ABI microcode. **This is a required data structure for ABI operations.**

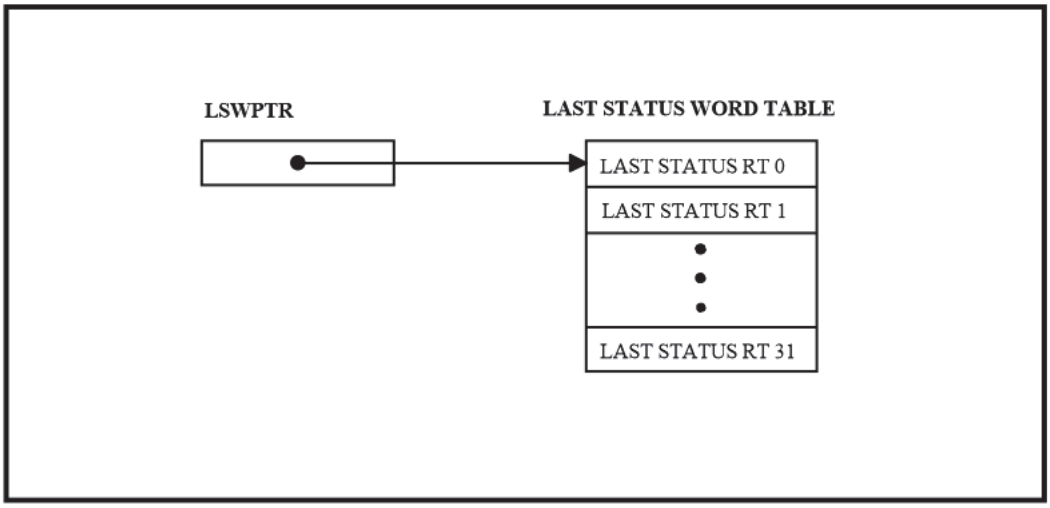


Figure RT-5 Last Status Word Data Structure

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RT Phase Data Structure

Includes:

- Transmitter Shutdown
- Override Transmitter Shutdown
- Inhibit Terminal Flag
- Override Inhibit Terminal Flag

The RT Phase Table stores status information for Transmitter Shutdown and Terminal Flag Inhibit. The table is 32 words long, with one word for each RT. The RTPPTR control register (offset 0095h) holds the starting address of this table. Figure RT-6 illustrates the structure of the table.

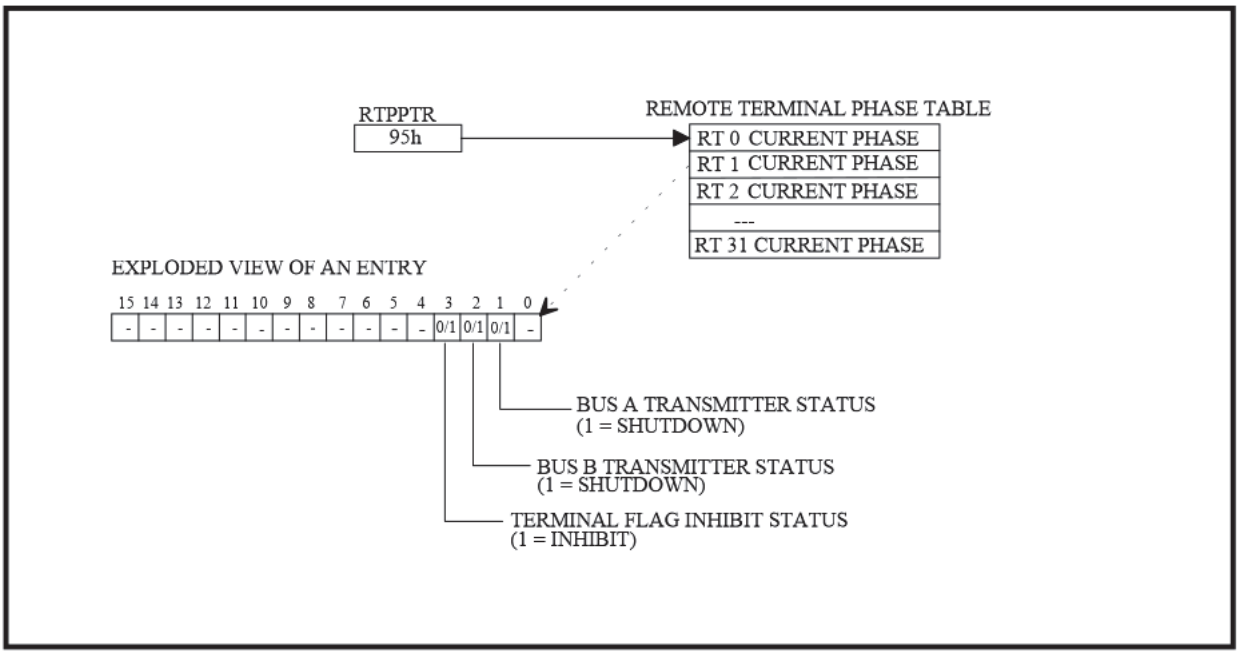


Figure RT-6 RT Phase Table Data Structure

Bits 1 and 2 of each word in the table store the shutdown status of the corresponding RT transmitter. A "1" in either bit indicates that the ABI has responded to a Transmitter Shutdown mode code and has shutdown the transmitter. (If the "1" is contained in bit 1, the transmitter for Bus A is shutdown; if the "1" is in bit 2, the transmitter for Bus B is shutdown) The Override Transmitter Shutdown mode code sets the appropriate bit to "0".

Bit 3 indicates the status of the Terminal Flag Inhibit function for each RT Status Word. When this mode code is received, the ABI microcode sets the appropriate bit in the table to "1" to inhibit the transmission of a Terminal Flag bit set to "1" (from the Status Word Table for the respective RT). The ABI microcode checks this bit before transmitting any status word. If it is set to "1", the microcode sets the Terminal Flag bit to "0", regardless of the actual value of this bit in the host Status Word. The receipt of an Override Inhibit Terminal Flag mode code

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

sets bit 3 to "0", allowing the Terminal Flag bit in the Status Word Table for the respective RT to be programmed. **The host must allocate memory locations for this table.**

Transmit Vector Word Data Structure

The ABI microcode responds to a Transmit Vector Word mode code command for a simulated RT by returning the appropriate vector word (as described in the Service Request bit discussion in the Status Word Table description). The returned Transmit Vector data word is taken from the specific RT location of this 32-word table. The TVWPTR control register (offset 0098h) holds the starting address for the table. The table array provides storage for 32 Transmit Vector words, one for each RT. Figure RT-7 illustrates the table structure. The host software always has access to this structure in order to store the desired Vector Word value for RT responses. **This is a required data structure for ABI operations.**

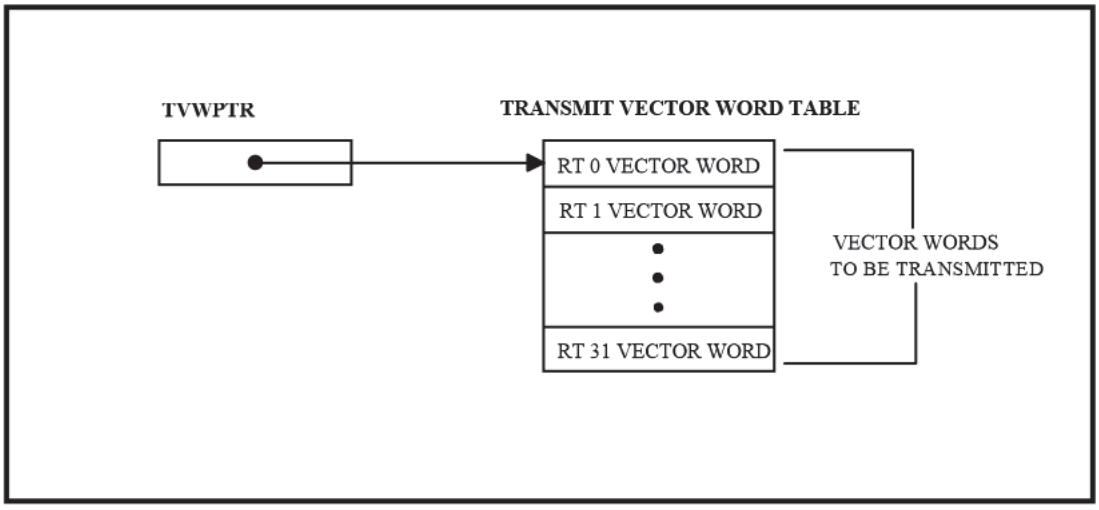


Figure RT-7 Transmit Vector Word Table

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Last Sync Word Data Structure

The ABI microcode responds to a Synchronize with Data Word mode code command for a simulated RT by storing the data word from the received mode code command in the specific RT location of this 32-word table. The ABI updates this table regardless of whether the ABI is simulating or monitoring the RT (this is the only mode code structure which functions in Monitor Mode). The LSYPTR control register (offset 009Ah) holds the starting address for the table. The table provides storage for 32 Synchronize Data words, one for each RT. Figure RT-8 illustrates the table structure. The host software always has access to this structure in order to read the desired data word value. **This is a required data structure for ABI operations.**

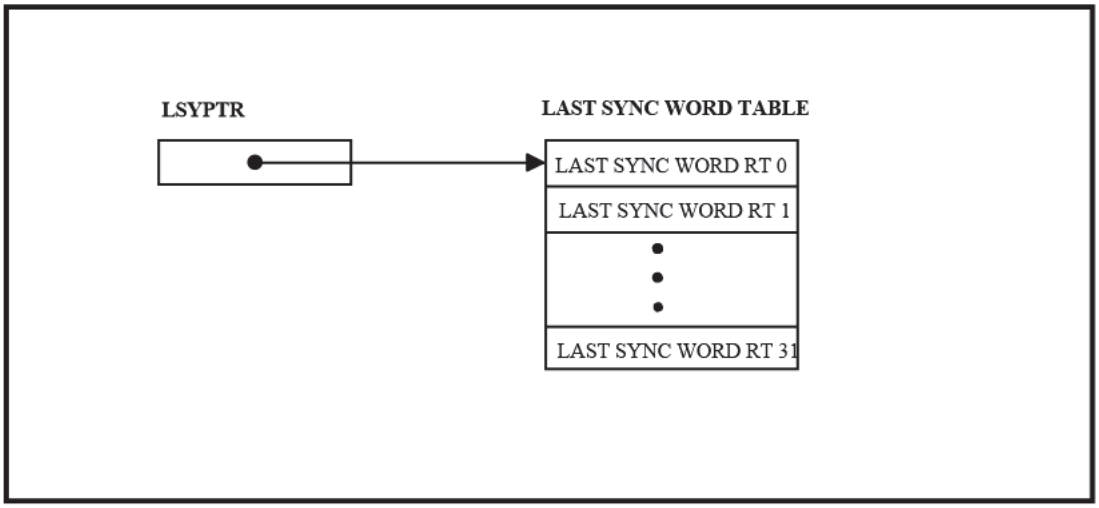


Figure RT-8 Last Sync Word Table

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Last Command Data Structure

The last valid message the ABI received from a simulated RT is stored in this 32-word table. The table provides storage for the Last Command Word for each of the 32 Remote Terminals (The table position corresponds to the RT address). The ABI microcode responds to a Transmit Last Command mode code for a simulated RT with the last command word stored in the table. The LCDPTR control register (offset 0097h) holds the starting address. Figure RT-9 illustrates the table structure. The host should not access this structure. **This is a required data structure for ABI operations.**

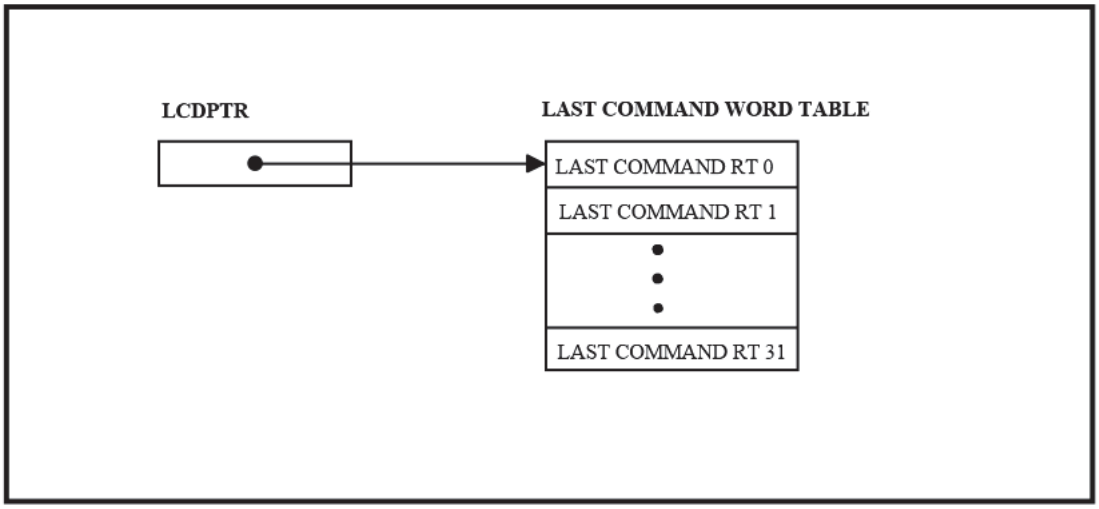


Figure RT-9 Last Command Word Table

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit Word Data Structure

The host software is responsible for writing the Bit Word value to this 32-word table for each RT being emulated. The ABI microcode responds to a Transmit Bit Word mode code command for a simulated RT by returning the appropriate bit word from the table (The table position corresponds to the RT address). Figure RT-10 illustrates the table structure. The BITPTR control register (offset 0096h) holds the table starting address. **This is a required data structure for ABI operations.**

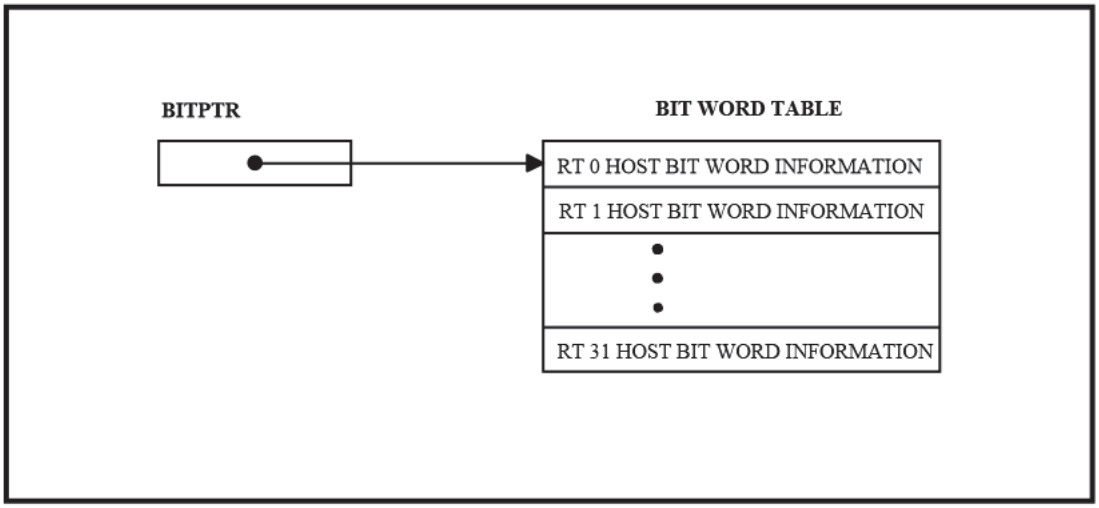


Figure RT-10 Bit Word Table

Optional Mode Code Data Structures

Filter Table

Provides an interrupt upon the detection of a message received. See the Filter Table discussion of Bit 1 earlier in this subsection.

RT Address Data Structures

Provides a special data buffer format for mode codes that can be used to detect the occurrence of a mode code for specific Remote Terminals (See "RT Address Mode Code Data Buffer Format" below.).

Mode Command Interrupt Table

This structure may be used to set up interrupts for specific mode codes (for specific Remote Terminals). It may be used in conjunction with or instead of the RT Address Data Structure for host detection of desired mode codes. This data structure provides a lower-level interrupt capability for mode code detection than that provided by the Filter Table. (See "Mode Command Interrupt Table" description later in this subsection).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RT Address Mode Code Data Buffer Format

The Address Data Structure provides a special mode code format for Receive Subaddresses 0 and 31. This format allows the ABI microcode to store the number of times each mode code has occurred. It is available for all mode code commands detected by the ABI (including mode codes generated by ABI Bus Controller simulation). To utilize this structure, define a 34-word array data buffer (using the host software) for Receive Subaddresses 0 and 31, as illustrated in Figure RT-11. Set the Buffer Pointer Table Transmit Subaddresses 0 and 31 to NULL. (Only the data buffers for Receive Subaddresses 0 and 31 are used to store mode code occurrence counter values, regardless of the state of the T/R bit in the mode code command). Set the Pointer in each of these special data buffers so it points back to itself, creating a single-link data buffer list.

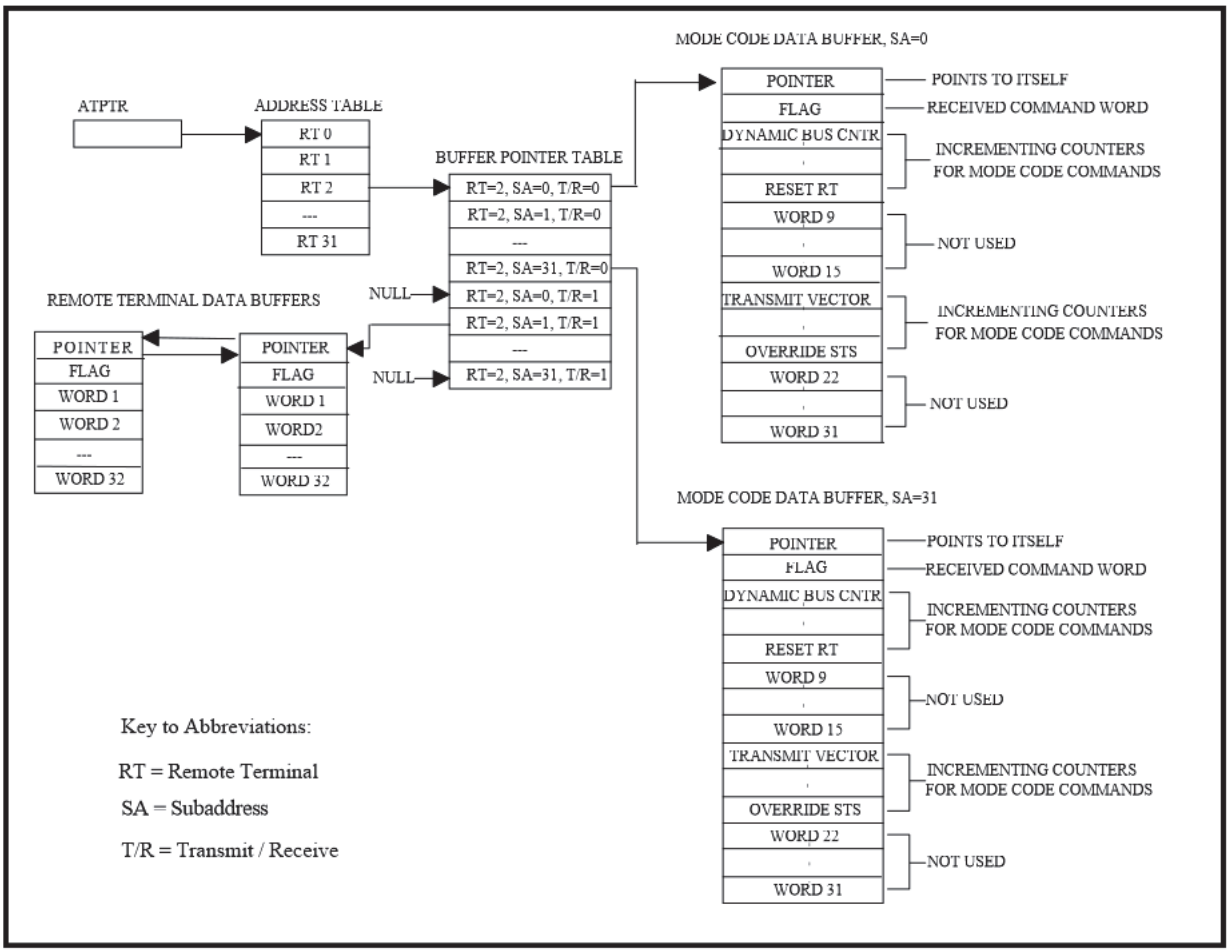


Figure RT-11 RT Address Mode Code Data Buffer

The ABI microcode increments the word value in this buffer by using the Word Count/Mode Code value as an offset to the data word area. For example, if a Transmit Vector Word mode code (value of 10000h or 16) is received for RT 9, Subaddress 31, the 19th word of the data buffer for RT 9 is incremented by "1". The received Command Word is stored in the flag word of the buffer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This same structure may be used for Map Monitoring functions. See the Protocol Table (Figure RT-4) for information on enabling Map Monitor functions for specific Remote Terminals, and the "Bus Monitoring" subsection for an overview of monitoring mode code messages.

Mode Interrupt Mask Table

The MIMPTR control register (offset 009Eh) holds the starting address of the Mode Interrupt Mask Table. This table allows the host software to select mode codes (for each RT) that will cause a hardware interrupt. *The ABI microcode will only generate interrupts from this table if the corresponding RT is being simulated.* The ABI microcode checks for the appropriate mode code mask bit to be set in this table when a particular mode code command is received. If the bit is set to "1", the ABI microcode generates a hardware interrupt to the host software.

The table structure, illustrated in Figure RT-12, consists of a 64-word array, 2-word mask field for each RT, with the mask words arranged by RT address number. The first mask word for each RT provides bit selections for mode codes 00h to 08h (Dynamic Bus Control to Reset RT). The second mask word provides bit selection for mode codes 10h to 15h (Transmit Vector Word to Override Selected Transmitter Shutdown). See the "Interrupts" subsection for more details on ABI interrupt handling.

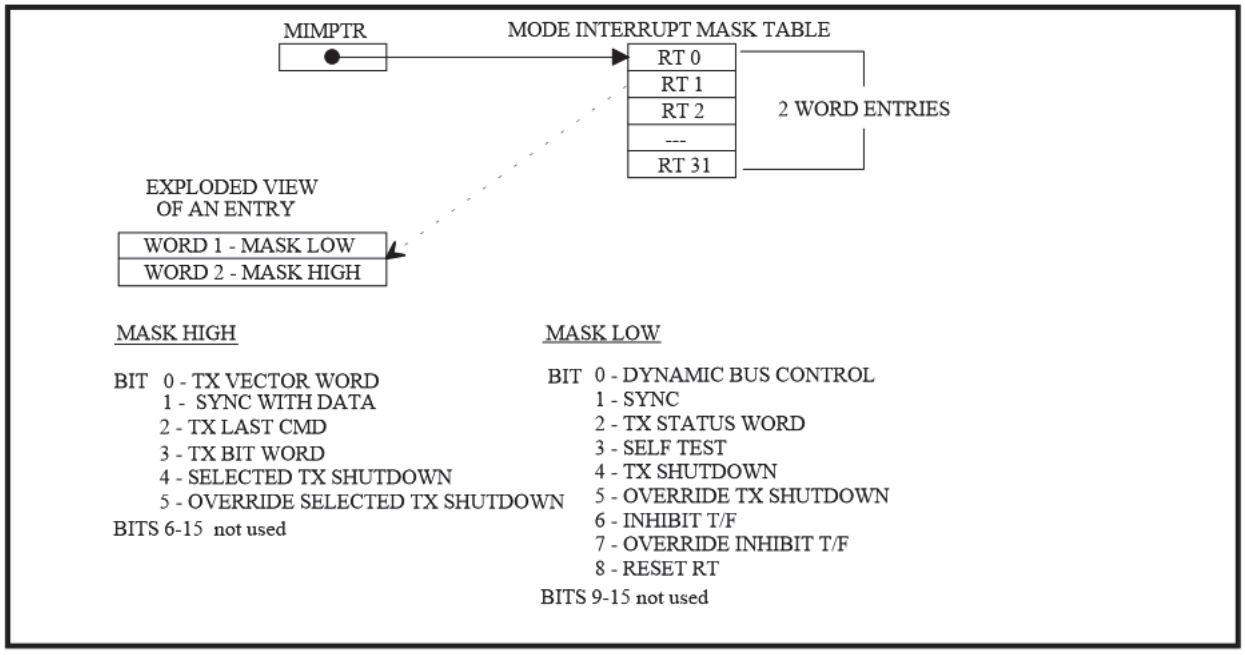


Figure RT-12 Mode Interrupt Mask Table

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Variable Intermessage and Status Response Gap Times

BC Intermessage and RT Status Response Gap Times are variable through the use of ABI control registers **BCIGP** (offset AEh), **RSPGPA** (offset 00ACh) and **RSPGPS** (offset 00ADh). BCIGP determines the intermessage gap time for BC messages. RSPGPA determines the maximum gap time allowed by a Bus Controller or Remote Terminal waiting for an RT status response. RSPGPS determines the actual gap time a Remote Terminal waits before transmitting a status response.

To determine the value to be entered in each of these control registers, use the following formulas:

RSPGPS, BCIGP = (time_in_microseconds - 4.0) * gap_count

RSPGPA = (time_in_microseconds - 2.0) * gap_count

where time_in_microseconds is the desired gap time (for RSPGPS and BCIGP, must be greater than four; for RSPGPA, must be greater than two) and gap_count is the MIP rating of the ABI (see "BUFPTR" description in the "BC Simulation" subsection of this manual). All values are in decimal; convert the Gap Time to hexadecimal before entering it into the control registers. This value will be accurate to within five microseconds of the desired gap time.

The actual bus gap time is dependent on the type of message being sent. Mode codes may have longer response times than do normal status messages and errors will cause "wait timeouts" because the Remote Terminal will not respond with a status word.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1553 BUS MONITORING

The ABI can perform two independent modes of bus monitoring (Sequential Monitoring and Map Monitoring) while performing simulation tasks for BC and 31 RTs. This subsection of the manual details these bus monitoring modes.

Sequential Monitoring stores raw 1553 traffic in double ("ping-pong") buffers for real-time data logging, recording and analysis applications. The ABI is capable of storing every 1553 message to double buffers and notifying the host system of buffer swaps through interrupt and polling functions. The messages are stored with 48-bit, one or 32-microsecond time stamps. This monitoring method is useful when the host system has tight processing constraints (relative to 1553 traffic) but must still receive every 1553 message word. The key data structures for Sequential Monitoring are **Sequential Monitor Buffers**, the **Filter Table**, and the **Interrupt Queue**.

Map Monitoring provides user defined linked-list data buffers for capturing RT commands and data words. This monitoring method allows the host system to set up fixed buffer locations for 1553 messages and then poll these buffers as host processing time permits. These data buffers may be chained together indefinitely, limited only by the available ABI memory. Map Monitoring is often used in conjunction with RT simulation applications where real-time polling of RT data buffers is preferred. The key data structures for Map Monitoring are **RT Address Table**, **Filter Table**, and **Protocol Table**. The details of these structures may be found in the "RT Simulation" subsection.

Both of the these monitoring methods can use the VME bus master capabilities of the ABI-V4. For map monitor buffers, the ability and method for performing DMA transfers of RT data buffers to VMEbus memory (usually VME/host processor) is identical to simulated receive RT data buffers of the ABI. Please see RT Simulation section for details to setup Map Monitor buffers for DMA transfers. Sequential monitor buffers may also be automatically transferred to VMEbus memory, and the control registers for each sequential monitor buffer are detailed in the following paragraphs. The use of automatic transfer of monitored 1553 data to VMEbus memory can greatly enhance the system architecture, as well as increase overall VME processor/host performance.

Sequential Monitoring

Sequential Monitoring provides the host system a method of receiving every 1553 message *in the same data format as is presented on the bus*. The host system must define two equally sized buffers of any length (up to the maximum available ABI memory). It must also specify which RT and subaddress (SA) combinations will have their messages stored in these buffers. Messages will fill one buffer at a time. Switching to the second buffer will automatically occur only when the first is filled (when the second buffer is filled, the ABI will switch back to the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

first buffer to continue storage - the buffers will "ping-pong"). The ABI microcode switches buffers when there are less than 50 words remaining in the current buffer (the size of the largest 1553 RT-RT message plus overhead words). This action continues indefinitely, or until the host system disables this function.

Three structures must be defined before Sequential Monitoring can take place: 1) The Filter Table must be defined with the desired RT terminals programmed for traffic, 2) The Sequential Monitor Buffers must be defined and 3) The interrupt queue buffers must be defined. The starting addresses for all these data structures must be programmed in the Pointer Table. The following paragraphs as well as the "RT Simulation" subsection provide programming details for each of the above data structures.

Filter Table

The Filter Table provides programming options for RT Simulation and Sequential Monitor designation. Each of the possible 32 RT registers holds a pointer to its subaddress (SA) filter table. Each of the possible 64 SAs (32 receive and 32 transmit) has an integer flag word, used for different programming options. Each flag word contains a Sequential Monitor bit. **This bit indicates whether the messages for the respective RT/SA are to be placed in a Sequential Monitor buffer.** This function allows message filtering at the RT/SA address level. There are no limits on the number of RT/SA messages that can be filtered or stored to buffers. Please see the "RT Simulation" subsection for details about this structure (use the Table of Contents to locate the "Filter Table" discussion). Figure MON-1 is provided for reference.

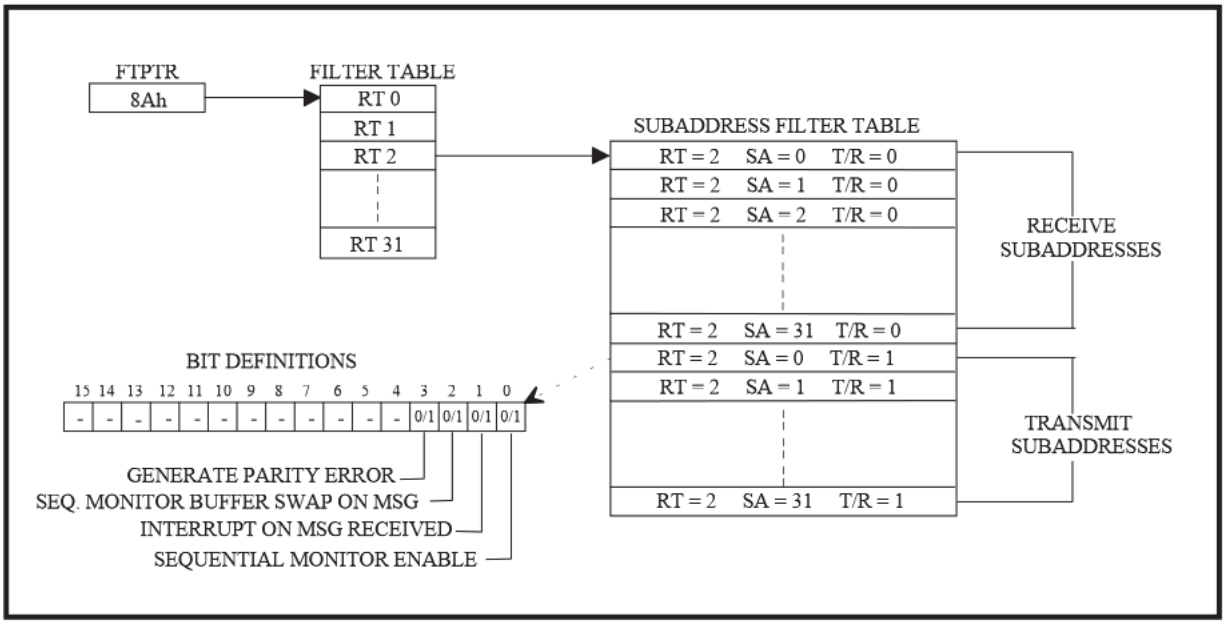


Figure MON-1 Filter Table Data Structures

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Sequential Monitor Buffers

The Sequential Monitor Buffers provide a double buffered storage area for 1553 messages selected using the Filter Table. The two control registers, M1PTR (offset 8Ch) and M2PTR (offset 8Dh), provide the starting addresses of two message blocks in ABI memory (in the 0400h to FFFFh area). Both buffers must be the same word length and this length must be programmed into the MBLEN register (offset 008Eh).

The size of sequential buffers is determined by considering the host system's processing capabilities and the amount of 1553 bus loading. For example, at the maximum transfer rate of the 1553 bus (about 49K words/second), two 25K buffers would be required to produce two buffer swaps per second (and probably two interrupts to the host system per second). Most 1553 buses have much lower transfer rates, and most host systems can process much more than two interrupts per second; therefore, the actual buffer size required for most applications is 4K-16K words per buffer.

If the sequential monitor bit is set to "1" for a particular RT/SA in the Filter Table, the respective 1553 messages will be logged to sequential monitor buffers. When the ABI microcode detects a 1553 message for a respective RT/SA message, it time stamps the message and stores it in the next available message block of the current buffer pointed to by M1PTR. The microcode always uses the value in M1PTR as the starting address for the current buffer. When the current buffer is filled, the microcode swaps the values between M1PTR and M2PTR; therefore, M1PTR always points to the beginning of the "current" buffer. Figure MON-2 provides a description of the Sequential Monitor Buffer data structure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

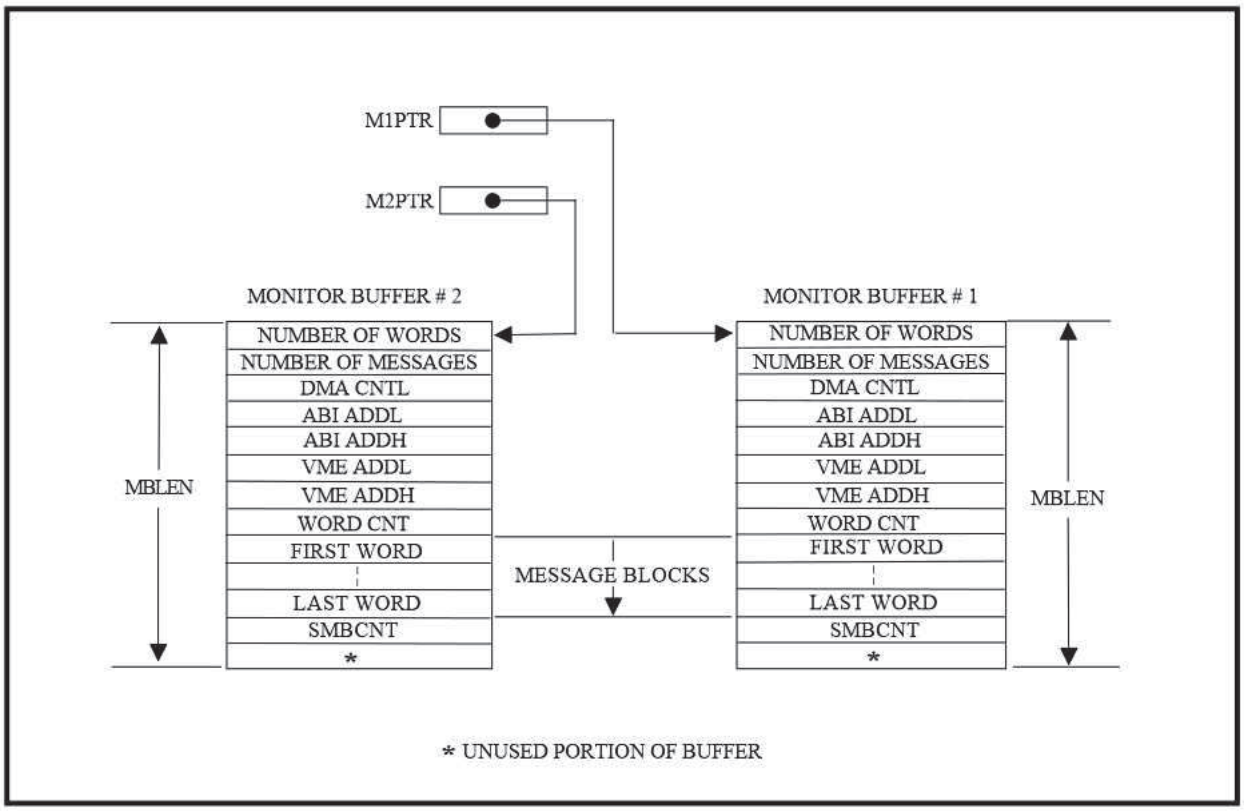


Figure MON-2 Sequential Monitor Buffer Data Structures

A message buffer counter is located at position "number of words + 1". It is initialized at "0" and, at the time of a buffer swap, is incremented once for each monitor buffer swap. It can be used by the host system to track message buffers and insure no buffer losses occurred. The word value stored in the Sequential Monitor Buffer Counter, SMBCNT (offset ABh), is used as the message counter in the subsequent buffer swap. The user may write a value to this register to preset this counter.

When a buffer has been filled with selected 1553 messages, the microcode updates the first and second word of the buffer with the **total buffer word count**, and the **1553 message count**, respectively. These first two words are included in the word count.

The remainder of the buffer contains message blocks which consist of one **type** word and three time stamp words followed by the complete 1553 message as it appeared on the bus. An error word is stored after the bus data if an error occurred for the message. The message block structures are illustrated in Figure MON-3.

DMA CNTL, ABI ADDL, ABI ADDH, VME ADDL, VME ADDH, WORD CNT
These six words (one set for each sequential monitor buffer) control DMA operations for writing sequential monitor buffers to VMEbus memory. Table MON-1 defines the bit values for **DMA CNTL** (DMA Control Word).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set bit 9 of DMA CNTL to "0" to indicate the transfer of ABI sequential monitor buffers to VMEbus memory.

If an Interrupt on DMA Transfer is desired, set the DMA interrupt vector in INTV_DMA (offset 0007h) and the DMA interrupt level using bits 12, 13, and 14 of HOST DMA CNTL (offset 0001h).

Table MON-1 DMA CONTROL WORD
Bit 0 is right-most bit and least-significant.

BIT(s)/NAME	Bit Values	Comment
Bits 0-5 Reserved	Reserved - Set to "0"	These bits are reserved for ABI microcode and hardware settings - set to "0".
Bit 6 Interrupt Enable	0 = Disable Interrupt after DMA 1 = Enable Interrupt after DMA	This bit enables/disables an interrupt by the ABI to the VMEbus after a block (DMA) transfer has completed. This feature allows the ABI to notify the host processor that a DMA transfer has completed for synchronization.
Bit 7 Reserved	Reserved - Set to "0"	These bits are reserved for ABI microcode and hardware settings - set to "0".
Bit 8 Master Block Transfer (DMA) Enable/Disable	0 = Disable DMA Function 1 = Enable DMA Transfer upon buffer full.	This bit determines if the ABI will perform a DMA transfer of the respective sequential monitor buffer when it is filled.
Bit 9 Direction	0 = Transfer from ABI to VME 1 = Transfer from VME to ABI	This bit controls the direction of master transfers by the ABI. Set to "0" to cause the ABI to transfer data from ABI ADDH/L to VME ADDH/L. Set to "1" to cause the ABI to perform a master transfer from VME ADDH/L memory to ABI ADDH/L.
Bits 10,11 Bus Request Level	00 = Bus Request Level 0 01 = Bus Request Level 1 10 = Bus Request Level 2 11 = Bus Request Level 3	These bits control the bus level for the VME transfer. "11" is typically the highest, but this setting is dependent on the bus control settings/protocol of the slot 0 VME arbiter.
Bit 12 Address Width	0 = A32 1 = A24	This bit selects either a 32 or 24 bit address width for the VME transfer. The VME address corresponds to the 32 or 24 least significant bits of VME ADDH/L, respectively.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table MON-1 DMA CONTROL WORD Bit 0 is right-most bit and least-significant.		
BIT(s)/NAME	**Bit Values**	**Comment**
Bit 13 Data Width	0 = D32 1 = D64	This bit controls the width of data transfers to/from VMEbus memory. The WORD CNT value is in increments of this bit setting - i.e., if bit 5 is set to "0" and WORD CNT is 16, then 16 32-bit transfers will occur.
Bit 14 Supervisor/Non Privileged Access	0 = Non Privileged (NORMAL) 1 = Supervisory	This bit controls memory address modifiers. The ABI-V4 responds to either mode.
Bit 15 Address Increment	0 = Increment VME Address 1 = No Address Increment	This bit controls automatic VME address increments for ABI block transfers. In short, this bit allows for transfers to/from regular memory or FIFOs. The amount the address is incremented depends on the setting of bit 5; a D32 transfer increments the VME address by 4 and a D64 transfer increments the address by 8.

ABI ADDL and **ABI ADDH** (ABI Address Low and High) define the ABI source address for DMA operation to the VME bus. These two words combine to make up a 21-bit ABI address (includes all bits of ABI ADDL and bits 0-4 of ABI ADDH); bits 5-31 of ABI ADDH are not used and should be set to zero. ABI memory (like 1553 words) is organized as 16 bit words and DMA transfers are in 32 or 64 bit word lengths. The ABI ADDL/H is a 16-bit word address. ABI ADDH is used only for extended memory models with more than 128K bytes on-board. For default memory models (128K of memory), ABI ADDH is set to "0" and ABI ADDL values are set to the same value as the user's application placed in the M1PTR and M2PTR pointers, respectively.

VME ADDL and **VME ADDH** (VMEbus Address Low and High) combine to make the VMEbus destination address words (24 or 32 bits as defined by bit 4 of DMA CNTL). VME memory, as with ABI memory, is big-endian oriented. Most VMEbus processor or reflective memory modules are byte oriented; thus, D32 or D64 transfers must start on a four byte or eight byte boundary.

WORD CNT is the number of transfers to be performed over the VME bus, minus one. The count value is for either D64 or D32 words, as designated by bit 5 of the DMA CNTL word. For example, if WORD CNT is 127 and bit 5 is set to "1" (selecting D64), 128 64-bit words will be transferred. Thus, the word count is not a byte count, but a VME transfer count. The minimum transfer value is zero and the maximum is 65535.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D64 transfers may consist of one to 65536 64-bit words (512K bytes), but the ABI must allow for VMEbus arbitration at 2K byte address boundaries (as defined by the VME64 specification). D32 DMA transfers may consist of one to 65536 32-bit words (256K bytes), but the ABI must allow for VMEbus arbitration at 256 byte address boundaries (as defined by the VME specification). These boundaries should be kept in mind when defining ABI data structures for maximum efficiency of VMEbus transfers (although the delay time through an address boundary is typically less than one microsecond). The ABI will only relinquish the VMEbus if a higher priority bus request is pending (which means the VMEbus arbiter has BCLR* {Bus Clear} pending).

Message Block Format

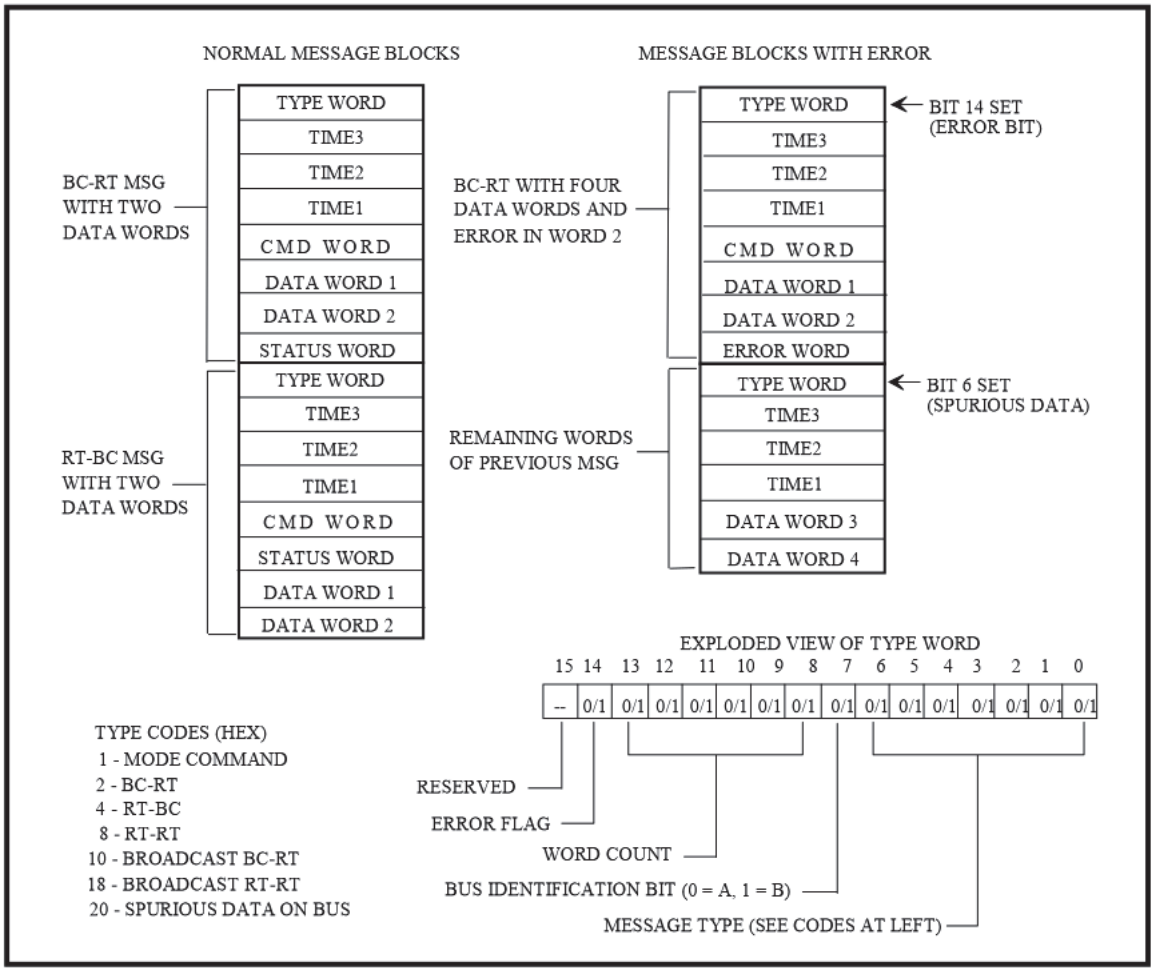


Figure MON-3 Sequential Monitor Buffer Message Blocks

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TYPE WORD

The **type** word indicates the type of 1553 message that has been stored, the bus on which the message was detected, the word count of the message block (including the type word) and whether or not an error occurred. The host software can use this information to read the command word and determine the configuration of the status and data words (the word count field specifies the number of words in non-Mode Code messages, including the type, command, data and status words).

Bits 0-6 - Message Type
These bits define the type of 1553 transfer for the message.

Bit 7 - Bus Identification Bit
If this bit is "0", the message was received/transmitted on Bus A; if "1", the message was received/transmitted on Bus B.

Bits 8-13 - Word Count
These bits indicate the total word count of the message, including the Type, Time Stamp, Command, Data, Status, and Error words.

Bit 14 - Error Flag
If this bit is "1", the message contained an error; if "0", no error.

TIME3, TIME2, TIME1

These three words make up a 48-bit time stamp for each BC Block. The time value is taken from the ABI hardware clock and is updated within two microseconds of when the first command word of the BC Block has started transmission. **TIME3** is the most significant word, followed by **TIME2** and **TIME1**. The ABI clock is programmable for 1 or 32 microsecond resolution, can be driven from an internal or external source, and may be preset as described in the "Introduction: ABI Operations Overview" section (See Table OVW-1). If you ever see bit 45 roll over, you have been doing your job way too long.

Data Words

After the TIME words, the 1553 data words are stored exactly as they appear on the 1553 bus. The format for BC-RT, RT-BC, RT-RT, Mode Code, and Broadcast Commands can be found in the 1553 standard.

Message With Error

If the ABI microcode detects an error during storage, it will add an error word to the end of the message block and set bit 14 of the type word to "1". The error word represents the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

address of an error in the Error Table (see Table OVW-4 in the "ABI Operations Overview" subsection of this *Full Function Programmer's Reference)*. The error condition applies to the 1553 word immediately prior to the error word. For example, if a "parity error" is detected on the sixth word of a ten word BC-RT transfer, two message blocks will be generated. The first block will contain the command word and the first six data words. The error word will be 011Eh, the address for the "parity" error. The next message block will contain the remaining words of the message and bit 6 of the type word will be set (bit 6 designates spurious data).

The ABI microcode will place errors or spurious data into the buffers regardless of Filter Table bit settings. (Spurious data refers to unexpected data which appears on the bus.) This can cause unexpected buffer swaps if 1553 words appear that are of poor transmission quality. This condition can also be caused by error conditions such as RT "no response".

Minimum Programming Requirements for Sequential Monitor Buffers

Sequential Monitor buffers are a required data structure. MBPTR1 and MBPTR2 must point to buffers which are a minimum of 50 words in length (MBLEN must be at least 50 for each buffer).

Buffer Swap Detection

The user can detect when a buffer swap has occurred through hardware interrupts or by monitoring M1PTR and M2PTR for address changes. The ABI will force a hardware interrupt to the host system when a buffer swap occurs only if interrupts are enabled in the CSR control register (offset 0000h). Please see the "ABI Operations Overview" subsection discussion on Control I/O registers and the "Interrupts" subsection for more details.

The host system may detect a buffer swap by polling M1PTR for a change of address. Upon startup, M1PTR contains the address to the first sequential monitor buffer and M2PTR contains the address to the second buffer. When a buffer swap occurs, the address for M2PTR is copied to M1PTR and the M1PTR address value is copied to M2PTR. Thus, M1PTR always contains the address of the current buffer. When a swap occurs, the respective SM DMAX DMA CNTRL WORD is checked to see if a DMA transfer of the buffer is to occur.

Forcing Buffer Swaps

The host software can force two types of buffer swaps: **simple buffer swap** and **buffer swap with interrupt**. Control register MBFLG (offset 009Fh) has two flags used to force these swaps. Write 0001h to the register to cause a simple buffer swap (without an interrupt) and write 0002h to the register to cause a buffer swap with a hardware interrupt (interrupt code 0002h). The ABI microcode will set the register to "0" when the swap has occurred, and will check SM DMA registers to see if a DMA transfer is to be performed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Map Monitoring

Map Monitoring is typically used by data analysis systems or screen update applications. In these situations, the host computer must poll fixed locations in ABI memory for 1553 data at specific time intervals or on event interrupts. Special linked-list data buffers are set up by the host system to be used for this function. Data word buffers for any non-simulated RT can be indefinitely linked together, limited only by the available ABI memory. This allows the host system to configure the length of buffer lists as required by system resources. For example, if the host can only read data buffers 10 times a second, and the particular RT subaddress is updated 20 times a second, the linked-list should be two or three buffers.

Some ABI applications combine Map Monitoring with message interrupts. The host system first programs the Filter Table to produce a hardware interrupt when a critical message has occurred. The host then reads the Map Monitor data word buffer(s) for the respective RT during the interrupt service routine.

All of the Map Monitoring data structures are also used for RT simulation functions. A review of these data structures is included in this subsection with Figures MON-4 and MON-5 provided for reference. Detailed discussions are provided in the "RT Simulation" subsection.

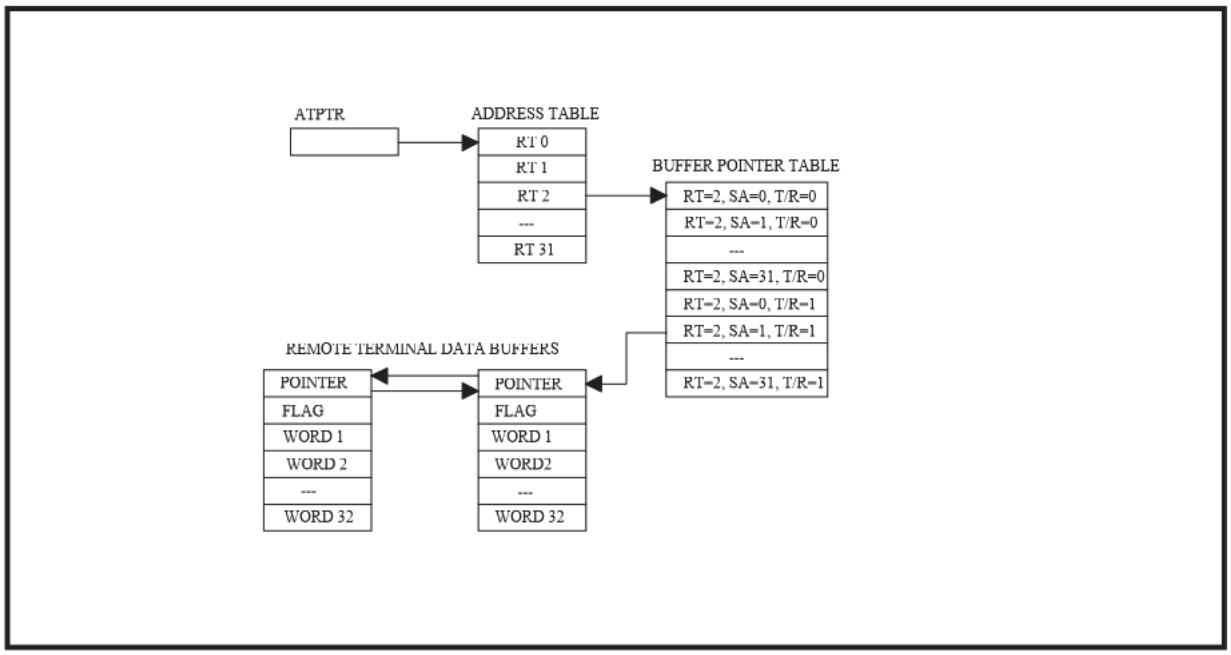


Figure MON-4 RT Address Data Structures

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

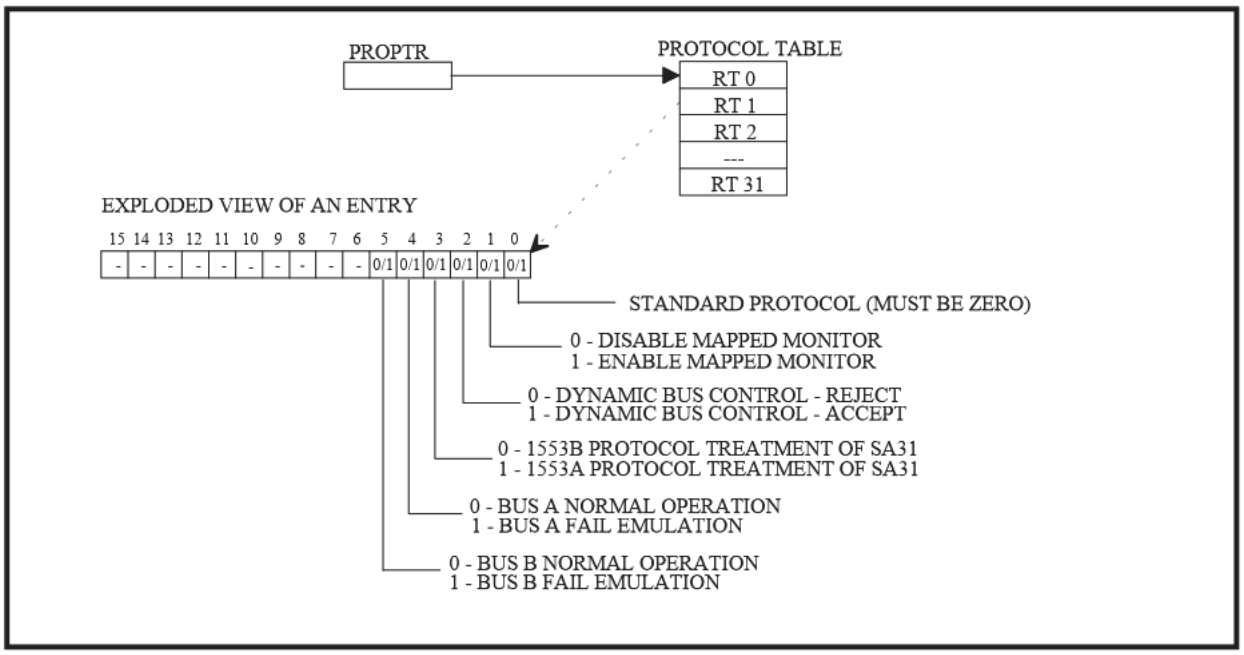


Figure MON-5 Protocol Table

Map Monitoring Data Structures

The key data structures for Map Monitoring are the RT Address Table and the RT Protocol Table. The RT Address Table is the same structure used for RT Simulation data buffers. The ABI cannot Map Monitor the RT if it is using the Address Table for simulation of the same RT.

The Protocol Table provides operation flags for ABI simulated RT responses, and provides a flag to enable and disable individual RT Map Monitoring functions. If Bit 1 is set to "0", the data words are not saved to RT Address Table Buffers. If Bit 1 is set to "one", the proper pointers and linked-list buffers must be established for the respective RT address, and subaddress-T/R bit combination. The ABI microcode first saves data to the buffer addressed by the Buffer Pointer Table, and then uses the pointer value in the data buffer link to determine the storage address for the next message.

Receive Broadcast Commands may be Mapped Monitored just as any other message. The RT Address Data Structures must be set up for RT address 31 with the appropriate subaddress data buffers defined.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INTERRUPTS

The ABI offers extensive interrupt capabilities for notifying the host processor of BC, RT, Monitoring, and system events that have occurred on the 1553 bus. Interrupts may be enabled via the following structures: 1) Filter Table, 2) flag word of a BC block, 3) MBFLG, 4) BCSMSK, and 5) mode code interrupt data structures in MIMPTR.

The ABI utilizes a doubled buffered queue structure to store detected or generated events for the host system interrupt service routines. The queue addresses and event counters for this data structure are managed by control registers (See "Control Registers" in the "Introduction: ABI Operations Overview" subsection). The host system reads these control registers during an interrupt service procedure to determine which interrupt queue is currently active and how many interrupt events are stored in the active queue.

The host system may handle interrupt events in one of two ways: 1) by polling the interrupt control registers with host software with the hardware interrupts disabled, 2) by using an interrupt service routine (ISR) written for the specific device with the hardware interrupts enabled. (SBS has developed many ISR routines for PC and UNIX systems which are available upon request to ABI customers.) The ABI fills the interrupt queues and updates interrupt control registers regardless which method is used. If the host system does not require hardware interrupts, the **interrupt enable** bit in the CSR control register (offset 0000h) should be set to "0" at startup.

The interrupt queue data structures and control registers allow the host system to easily determine what type of BC, RT, Monitor or system event occurred. This subsection details the types of ABI interrupts and the data structures and control registers used by interrupt service procedures to manage interrupts.

Interrupt Types

There are interrupts associated with each major ABI function (BC and RT simulation and Monitoring) and interrupts for general system functions. Table INT-1 details the various ABI interrupt conditions. The "Code" column lists the numbers assigned to each interrupt condition. Each entry in an interrupt queue consists of four words, the first word representing the **code** type. The host system uses this code to determine how to handle each interrupt.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table INT-1 ABI Interrupt Types

Type	Code (hex)	Description
BC Simulation		
RT Response Error	0005	By default, this interrupt is generated when an error is detected in an associated RT response. The most common error is RT "no response". Set bit 0 of a BC command block flag word to "1" to disable this interrupt.
Masked Status	0007	If bit 2 of a BC command block flag word is set to "1", the ABI will logically "AND" the associated RT status word with the control register BCSMSK (offset 0092h). If the result is greater than "0", the ABI will generate an interrupt. This interrupt is useful in determining when status word bits, other than address bits, are set to "1".
BC Halt	0006	By default, this interrupt is generated whenever BC execution is halted. This most commonly occurs due to a zero link in the last control block of a BC chain but an RT response error or an interrupt on masked status may also cause BC execution to halt. Set bit 3 of a BC command block flag word to "1" to disable this interrupt.
RT Simulation		
Command Word Detected	0003	This interrupt is generated if bit 1 of the Filter Table for a specific RT address and subaddress is set to "1", and the command word to the respective RT is received. This function is extremely useful for RT simulation and monitoring in determining when important command words have been received on the 1553 bus. The interrupt occurs after the completion of the entire BC message.
Mode Code Command Word Detected	0009	This interrupt is generated when a Mode Code command word is detected on the 1553 bus and the respective bit has been set to "1" in the Mode Command Interrupt Table.
Monitoring		
Sequential Monitor Buffer Swap	0002	This interrupt is generated when a monitor buffer swap occurs. A monitor buffer swap can occur during normal bus monitoring or via a forced buffer swap (write a "2" to the MBFLG control register 9Fh to force a buffer swap). The microcode stores 1553 messages, including messages with errors, according to Filter Table programming. (Spurious data on the bus is always stored in the Sequential Monitor buffer regardless of Filter Table programming.) This interrupt is frequently used by the host system to read one monitor buffer while the other buffer is filling with data.
Sequential Monitor Buffer Swap on Message	0010	The ABI generates this interrupt when bit 2 of the Filter Table for a specific RT/SA is set to "1" and the command word is received. This interrupt forces the sequential monitor buffers to swap. The interrupt is generated after the message has completed and the monitor buffers have swapped.
Message Received	0003	This interrupt is generated if bit 1 of the Filter Table for a specific RT/SA is set to "1", and a command word for the RT/SA combination is detected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table INT-1 ABI Interrupt Types		
Type	Code (hex)	Description
General System		
Timer Data Ready	0001	The ABI generates this interrupt when control registers SCHIGH and SCLOW (offsets 0093h and 0094h) contain valid time words. This condition exists when the host software has set bit 0 of the control register CCW (offset 009D) to "1". Bit 0 of CCW is cleared by the ABI microcode when the interrupt occurs.
Interrupt Overflow	0008	This interrupt indicates that the interrupt queue has overflowed. The host system neglected to service the first interrupt of the queue before the entire queue filled with data.

Interrupt Queue Data Structures and Control Registers

The ABI manages interrupts using double buffered queues and counter control registers. When an interrupt event occurs, the ABI stores four words in the current interrupt queue. These four words consist of an interrupt code and three information words. The host reads these words to determine what type of interrupt occurred and to obtain related information such as the 1553 command words, pointer values and time values associated with the interrupt event.

The interrupt queues and control registers are illustrated in Figure INT-1.

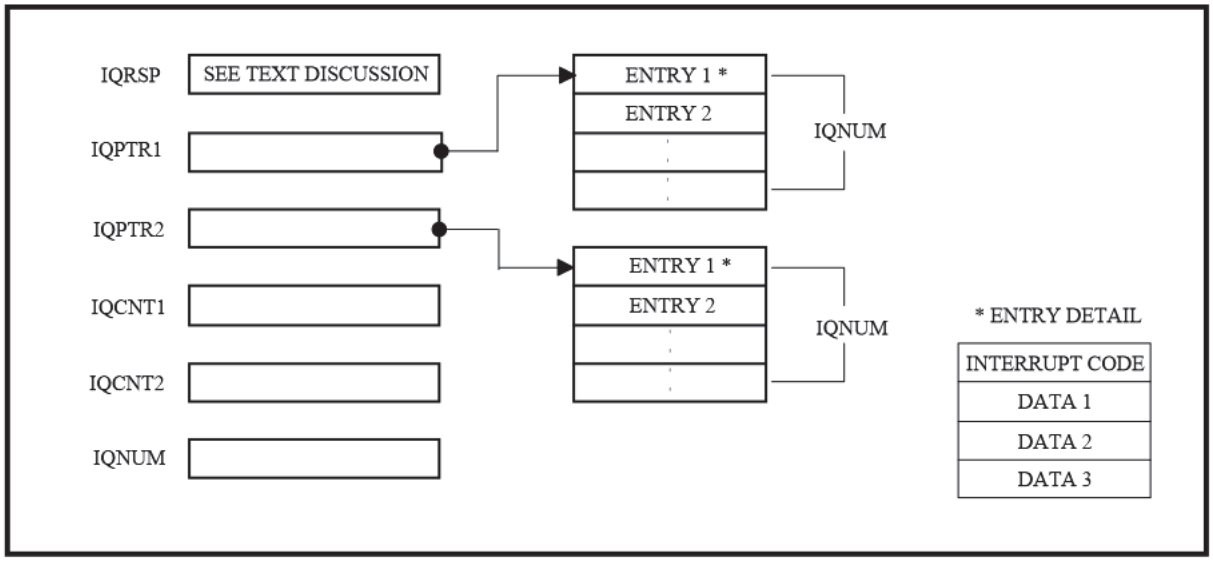


Figure INT-1 ABI Interrupt Queues and Control Registers

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Minimum Programming Requirements for Interrupt Queues

The host software must program two interrupt queues. These queues must be identical in length and must hold a minimum of four interrupt entries each. A pointer (IQPTR1 and IQPTR2) must be programmed for each queue to indicate the starting address. Each queue must also have a counter (IQCNT1 and IQCNT2) to count interrupt entries as they occur. Each entry will consist of one to three words of valid data.

The host programs the starting address of the first queue into IQPTR1 (offset 0083h) and the address of the second queue into IQPTR2 (offset 0084h). The host programs the length of the queues (these lengths must be identical) into control register IQNUM (offset 0087h). The value in IQNUM determines the number of entries in each queue; each entry contains one to four words. For example, enter a value of "10" ("000Ah") into IQNUM for a 40-word queue. Ten interrupt entries may be stored in a queue this size. The ABI microcode defaults IQNUM to "4" at startup.

When an interrupt event occurs, the microcode sets the interrupt pending bit of the CSR control register (bit 7 of offset 0000h). If bit 3 of the CSR is set to "1", the ABI generates a hardware interrupt. The host ISR writes the value "FFFF" to IQRSP when starting the service routine and writes "0001h" to IQRSP when the routine is complete. The control registers IQCNT1 and IQCNT2 (offset 0085h and 0086h) provide an event count for each buffer (there are four words per event).

After microcode startup, the microcode writes interrupt events to the queue indicated by IQPTR1. When an interrupt occurs, the host ISR sets IQRSP to FFFFh. **The host must then wait two microseconds** for the microcode to finish processing any partially completed asynchronous interrupt events (see NOTE in box). If there is a partial interrupt pending, the microcode finishes processing, places the event information in the queue indicated by IQPTR1, and increments IQCNT1. If a new interrupt occurs during the host service (after FFFFh is written to IQRSP), the new interrupt event information is placed in the queue indicated by IQPTR2. After servicing the interrupts, the host ISR writes a "1" to IQRSP and the microcode swaps the values from IQPTR2 to IQPTR1 and from IQCNT2 to IQCNT1. If an interrupt event was placed in IQPTR2, the ABI will generate a new hardware interrupt.

> NOTE: If the host has programmed the ABI for interrupt events that are known to be synchronous, and these events occur more than 10 microseconds apart, the host ISR does not need to wait two microseconds before servicing the interrupts.

When the host services an interrupt, it reads the **code** word from the interrupt entry to determine the interrupt type. Some interrupt types consist of only this code word and others also contain data words. Table INT-2 defines each of the code words and their associated data words. Some words are undefined (their values are unknown) and should be ignored by the host software.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table INT-2 Interrupt Word Descriptions

Type	Word 1 - Code (hex)	Word 2	Word 3	Word 4
Timer Data Ready	1	Low 16 bits of system timer.	High 16 bits of system timer.	Undefined.
Sequential Monitor Buffer Swap	2	Undefined	Undefined	Undefined
Message Received	3	This command word for the respective RT Filter Bit for BC-RT or RT-BC transfers. For RT-RT transfers, this word contains the "receive command" word.	Set to "0" for BC-RT and RT-BC transfers. For RT-RT transfers, this word contains the "transmit command" word.	Undefined
RESERVED	4	Undefined	Undefined	Undefined
RT Transmit Error	5	The offset address for the BC command block from which the error was detected. Copied from BCCPTR pointer value at time of BC transmission. This usually indicates an RT transmission error or an illegal BC command block program setting.	The error code as listed in the error table. - See the "Error Table" in the "ABI Overview" subsection.	Undefined
BC Halt	6	Undefined	Undefined	Undefined
Masked Status	7	The offset address for the BC command block from which the RT status word "ANDed" with BCSMSK had a result greater than "0".	Undefined	Undefined
Interrupt Overflow	8	Undefined	Undefined	Undefined
Mode Code Command Word Detected	9	The mode code command word for the selected bit in the MIM table.	Undefined	Undefined
Sequential Monitor Buffer Swap on Message	10	The command word detected for the selected bit in the Filter Table.	Undefined	Undefined

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hardware Interrupt Service Procedure

The procedure shown in Table INT-3 is recommended for servicing ABI hardware interrupts.

Table INT-3 Hardware Interrupt Service Procedure

Step Number	Description
1	Host receives ABI hardware interrupt and starts ISR.
2	Host ISR reads IQRSP (offset 0082h) in a loop until IQRSP is equal to "0".
2	Host ISR writes "FFFF" to IQRSP.
3	Host ISR *waits 2 microseconds* for the microcode to swap values between IQPTRs and IQCNTs.
4	Host ISR reads IQCNT1 (offset 0085h) to determine the number of interrupt events to process.
5	Host ISR reads IQPTR1 to get the address of the interrupt queue to process.
6	The host reads each interrupt event, beginning at the address indicated by IQPTR1. The host reads each interrupt event in the queue, deciphering the code word to determine if data words are present. The host ISR may either process the interrupt data at this point or after execution has returned to the host software.
7	Clear the pending interrupt bit. This step is performed by the hardware for VME systems.
8	Host ISR writes 0001h to IQRSP.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Software Polling Interrupt Service Procedure

The procedure shown in Table INT-4 is recommended for software polling of ABI interrupt events.

Table INT-4 Software Polling Interrupt Service Procedure

Step Number	Description
1	Host polls (reads) bit 7 of the CSR (offset 0000h) until it detects a value of "1". A "1" indicates an interrupt event has occurred on the ABI.
2	Host ISR reads IQRSP (offset 0082h) in a loop until IQRSP is equal to "0".
2	Host ISR writes FFFFh IQRSP.
3	Host ISR *waits 2 microseconds* for the microcode to swap values between IQPTRs and IQCNTs.
4	Host ISR reads IQCNT1 (offset 0085h) to determine the number of interrupt events to process.
5	Host ISR reads IQPTR1 to get the address of the interrupt queue buffer to process.
6	The host reads each interrupt event, beginning at the address indicated by IQPTR1. The host reads each interrupt event in the queue, deciphering the code word to determine if data words are present. The host ISR may either process the interrupt data at this point or after execution has returned to the host software.
7	Host ISR writes a "1" to bit 7 of the CSR control register to clear the interrupt request.
8	Host ISR writes 0001h to IQRSP.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Addendum to:

1553 Advanced Bus Interface for VME (ABI-V4)
Bus Master
Reference Manual

The following information should be added to the last paragraph on page 2-3 (under the topic *Setting the ABI RAM Base Address in VMEbus Memory*):

"Recommended memory address offset when using multiple cards of both Rev A and Rev B printed circuit board is 200000h."

SBS Technologies is dedicated to providing technically superior products and the best customer support possible. SBS provides full support for the ABI V4, including any reasonable assistance with the entire integration effort. For assistance, please contact SBS Technologies via phone, fax, regular mail, or e-mail.

SBS Technologies, Inc.
2400 Louisiana Boulevard NE, Building 5, Suite 600
Albuquerque, New Mexico 87110

1-800-SBS-1553 or 505-875-0600
FAX (505) 875-0400

e-mail: sbshelp@sbse.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP