DDC
Data Device Corporation

APPLICATION NOTES

ACE RT MEMORY MANAGEMENT OPTIONS

ABSTRACT

The software design of an RT for a microprocessor-based system is a common MIL-STD-1553 interface requirement. System requirements and Interface Control Drawings (ICDs) often provide a multitude of demands on the software engineer. Among these requirements are the need to ensure data integrity and data consistency, the need to perform bulk (multi-message) data transfers, the demand for interrupt-driven software, and the need to offload the operation of the host CPU to the greatest degree possible This latter requirement is reflected in such specifications as spare CPU bandwidth.

This application note provides a discussion of the application of the ACE's various memory management modes: single message, circular buffered, and double buffered. Other issues discussed are broadcast separation, considerations regarding data consistency and bulk data transfers, the use of polling techniques and interrupts, software initialization, and the servicing of synchronous and asynchronous messages.

INTRODUCTION

The ACE RT architecture offers a high degree of programmable flexibility. As a result, it is able to provide a solution to a wide variety of applications. One of the salient attributes, which is the main subject of this application note, is the ACE's memory management architecture. This includes a variety of options, most of which are programmable on a subaddress basis.

The ACE allows the memory management configuration for each T/R (/broadcast) subaddress to be selected between a single message buffer or a circular buffer The size of each circular buffer is programmable with choices between 128 and 8192 words, in integral powers of two.

For receive messages, there is a third option, providing a form of double buffering on a subaddress basis. In addition, for each transmit/receive/broadcast subaddress, there is a choice of interrupts: for any message to the T/R (/brdcst)-subaddress, or following a circular buffer rollover for the T/R (/brdcst)-subaddress.

MEMORY MAP, DATA STRUCTURES

Table 1 illustrates a typical memory map for the ACE RT with the enhanced mode features enabled. The two Stack Pointers reside in fixed locations in the shared RAM address space: address 0100 (hex) for the Area A Stack Pointer and address 0104 for the Area B Stack Pointer. In addition to the Stack Pointer, for the ACE enhanced RT, there are several other areas of the ACE address space that are designated as fixed locations. These are for the Area A and Area B Lookup Tables, the Mode Code Selective Interrupt Table, the Mode Code Data Table, the Busy Lookup Table, and the Command Illegalizing Table.

The ACE provides a global area A/B select mechanism, programmable by means of bit 13 of Configuration Register #1. This allows the host CPU to

TABLE 1. TYPICAL RT MEMORY MAP
(SHOWN FOR 4K RAM WITH ENHANCED MODE FEATURES ENABLED)

ADDRESS (HEX)	DESCRIPTION
0000-00FF	Stack A
0100	RT Command Stack Pointer A (fixed location)
0101-0103	RESERVED
0104	RT Command Stack Pointer B (fixed location)
0105-0107	RESERVED
0108-010F	Mode Code Selective Interrupt (fixed area)
0110-013F	Mode Code Data Table (fixed area)
0140-01BF	Lookup Table A (fixed area)
01C0-023F	Lookup Table B (fixed area)
0240-0247	Busy Bit Location Table (fixed area)
0248-025F	(not used)
0260-027F	Data Block 0
0280-029F • • •	Data Block 1 • • •
02E0-02FF	Data Block 4
0300-03FF	Command Illegalizing Table (fixed area)
0400-041F	Data Block 5
0420-043F	Data Block 6
• • •	• • •
0FEO-0FFF	Data Block 100

be able to toggle between two alternative sets of corresponding data structures. These data structures include the Stack Pointer and Lookup Table. Each Stack Pointer references an individual stack. Each lookup table references a set of data tables for the individual RT subaddresses. A provision of MIL-STD-1553B Notice 2 (30.6) requires that "...RT shall be capable of distinguishing between a broadcast and a non-broadcast message to the same subaddress for non-mode command messages." The ACE exceeds the letter of this requirement by including a programmable option for separate lookup tables for non-broadcast and broadcast receive messages. Use of separate lookup tables allows data words received from broadcast messages to be stored in separate blocks from non-broadcast data in the ACE address space. BROADCAST SEPARATION may be enabled by programming bit 0 of Configuration Register #2 to logic "1".

The RT lookup tables provide a mechanism for mapping data blocks for individual Tx/Rx/Bcst-subaddresses to areas in the shared RAM. The two lookup tables are located in address range 0140 to 01BF for Area A and address range 01C0 to 023F for Area B. The RT lookup tables include Subaddress Control Words as well as the individual data block pointers. The actual Stack RAM area as well as the individual data blocks may be located in any of the non-fixed areas in the shared RAM address space.

Table 2 illustrates the organization of the RT Lookup Tables. It should be noted that if SEPARATE BROADCAST DATA, bit 2 of Configuration Register #2, is logic "0", the data block pointers for both broadcast and nonbroadcast receive messages will be stored in the first 32 locations of the lookup tables. If SEPARATE BROADCAST DATA is logic "l", the pointers to data words for broadcast messages are stored in the third block of 32 locations within the respective lookup table.

It should be noted that the Subaddress Control Words, located in the fourth block of 32 locations within each lookup table, are only used if ENHANCED RT MEMORY MANAGEMENT, bit 1 of Configuration Register #2, is logic "1".

SUBADDRESS MEMORY MANAGEMENT

The most salient attribute of the ACE's RT architecture is the flexibility offered by its options for subaddress memory management. The ACE supplies a host of programmable features to support ease of operation, guarantee data consistency, and facilitate bulk data transfers.

The ability to program the memory management mechanism on a subaddress basis is essential to most complex system designs since the data content that will be transferred to and from an RT generally varies in size (bulk data versus single messages).

TABLE 2. RT LOOKUP TABLE

AREA A	AREA B	DESCRIPTION	COMMENT
0140 • • • 015F	01C0 • • • 01DF	RX (Bcst) SA0 • • • RX (Bcst) SA31	Receive (/Broadcast) Lookup Table
0160 • • • 017F	0150 • • • 01FF	TX SA0 • • • TX SA31	Transmit Lookup Table
0180 • • • 019F	0200 • • • 021F	Bcst SA0 • • • Bcst SA31	Broadcast Lookup Table (Optional)
01A0 • • • 01BF	0220 • • • 023F	SACWSA0 • • • SACW SA31	Subaddress Control Word Lookup Table (Optional)

and CPU servicing requirements (received data is retrieved after immediately after every message versus asynchronous use of data). The key to the ACE's RT memory management architecture is that it allows the selection of single message. double buffering, and circular buffer (including circular buffer size) to be programmed on a transmit/ receive/broadcast-subaddress basis.

In its default (power turn-on) configuration, the ACE initializes to its non-enhanced mode. In this configuration, the data tables for broadcast and non-broadcast receive messages are referenced by the same set of lookup table pointers and the subaddress memory management scheme defaults to the single message mode for all receive and transmit subaddresses. In order to use the various enhanced features. various configuration register bits need to be set. Reference Table 3. It should be noted that for most of the advanced features (broadcast separation is the exception), combinations of bits in the Subaddress Control Word must also be used.

SUBADDRESS CONTROL WORD

If ENHANCED RT MEMORY management has been enabled by means of bit I of Configuration Register #2, each of the 32 Subaddress Control Words specifies the memory management and interrupt schemes for the respective subaddress. Refer to Tables 4 and 5.

For each Subaddress Control Word, five bits specify the memory management scheme and interrupts for each of transmit, receive, and broadcast messages. In addition, bit 15 (MSB) of each Subaddress Control Word may be used to enable double buffering for individual receive (and broadcast receive) messages to the particular subaddress. Bit 15 of Configuration Register #3, followed by bit 12 of Configuration Register #2 **(in that order)** must be set to logic "1" in order to enable use of the subaddress double buffering mode.

For each transmit, receive, or broadcast subaddress, three bits are used to specify the memory management scheme. For each Tx/Rx/Bcst subaddress, the memory management scheme may be selected for either the "single message" mode (as in the nonenhanced mode), or the "circular buffer" mode. In addition, each receive (and broadcast, if separated) subaddress may be configured for the subaddress double buffering mode if the RX:(BCST:) MM2, RX:(BCST:) MMI, and RX:(BCST:) MM0 bits (7, 6, and 5 for receive; bits 2, 1, and 0 for broadcast if separated) are programmed to logic "0" **and** Subaddress Double Buffering (bit 15) is logic "1".

In the single buffer mode, a single data block is repeatedly overread (for transmit data) or overwritten (for receive or broadcast data). Alternatively, in the circular buffer mode, Data Words for successive messages to/from any particular Tx/Rx/Bcst subaddresses are read from or written to the next contiguous block of locations in the respective circular buffer.

TABLE 3. CONFIGURATION REGISTER BITS USED TO ENABLE ADVANCED RT MEMORY MANAGEMENT FEATURES

MEMORY MANAGEMENT FEATURE	C.R. 2, BIT 12 SUBADDRESS DOUBLE BUFFER ENABLE	C.R. 2, BIT 11 OVERWRITE INVALID DATA	C.R. 2, BIT 1 ENHANCED RT MEMORY MANAGEMENT	C.R. 2, BIT 0 SEPARATE BROADCAST	C.R. 2, BIT15 ENHANCED MODE ENABLED
Default (power turn-on) Mode: "Single Message" for all tx/rx (/bcst) subaddresses.	0	0	0	0	0
"Single Message" for all transmit subaddresses: "Double Buffering" for all receive (and bdcast) subaddresses.	1 (See Note 2)	0	0	X	1 (See Note 2)
Broadcast separation	X	X	X	X	X
Enable Circular Buffer for individual subaddresses; NOT overwriting invalid data	X	0	1	X	X
Enable Circular Buffer for individual subaddresses; With overwriting invalid data	X	1	1	X	X
Enable subaddress double buffering for individual subaddresses	1 (See Note 2)	X	1	X	1 (See Note 2)

NOTES: (1) "C.R." denotes Configuration Register.
(2) In order to enable the subaddress double buffering modem bit 15 of Configuration Register 3 must be set to logic "1" before bit 12 of Configuration Register 2 is written as logic "1."

TABLE 4. SUBADDRESS CONTROL WORD BIT MAP	
BIT	**DESCRIPTION**
15 (MSB)	RCV DBL BUFF'NG ENA
14	TX: INT ON EOM
13	TX: INT BUF ROLLOVER
12	TX; MM2
11	TX: MM1
10	TX: MM0
9	RX: INT ON EOM
8	RX: INT BUF ROLLOVER
7	RX: MM2
6	RX: MM1
5	RX: MM0
4	B'CAST: INT ON EOM
3	B'CAST: INT BUF ROLLOVER
2	B'CAST: MM2
1	B'CAST: MM1
0 (LSB)	B'CAST: MM0

For the circular buffer mode, the size of the circular buffer for each transmit, receive, or broadcast subaddress may be programmed for 128, 256, 512, 1024, 2048, 4096, or 8192 words.

For each Tx/Rx/Bcst subaddress, two bits of the subaddress control word are used to enable interrupts. One of these bits will result in an interrupt following every message directed to the specific Tx/Rx/Bcst subaddress. The second interrupt control bit will result in an interrupt at the end of a message if the message resulted in the lookup table pointer for the respective Tx/Rx/Bcst-subaddress crossing the lower boundary of the circular buffer, rolling over to the top of the buffer.

SINGLE MESSAGE MODE

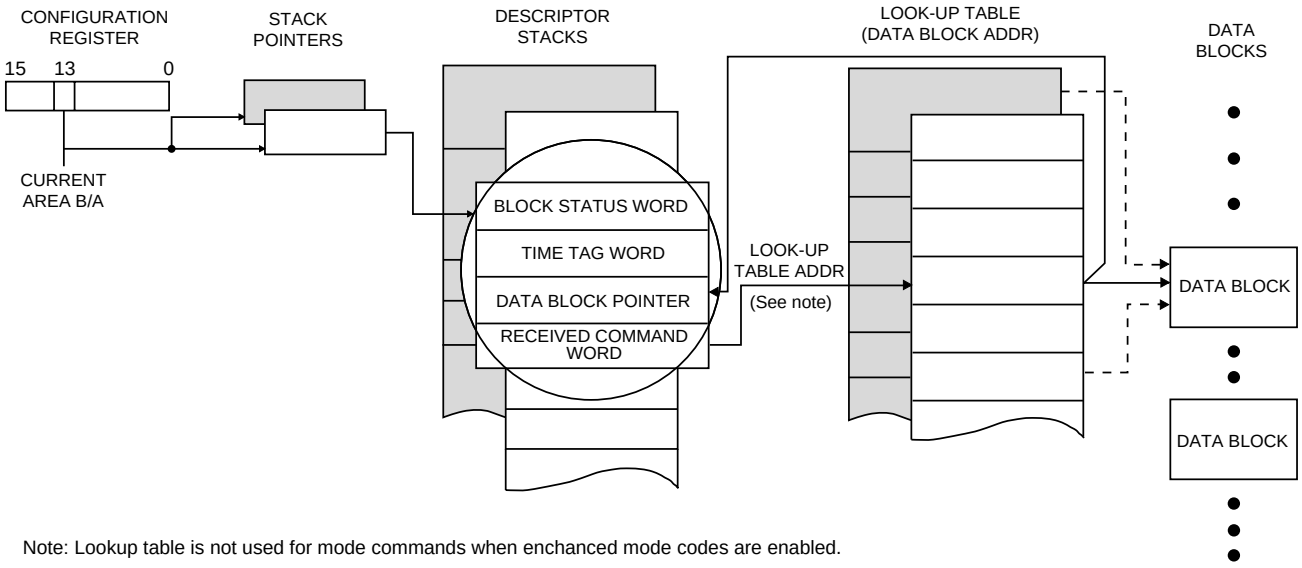
As discussed, the default (power turn-on) configuration for the ACE results in the single message mode for all receive and transmit subaddresses. In addition, individual transmit, receive, and broadcast subaddresses may be designated for the single message mode by means of the Subaddress Control Word.

The operation of the single message RT mode is illustrated in Figure 1. In the single message mode, the respective lookup table entry is loaded by the host processor. At the start of each message, the lookup table entry is stored in the third address location of the respective message block descriptor in the stack area of RAM. Received Data Words are written to or transmitted Data Words are read from the Data Word block referenced by the lookup table pointer. In the single message mode, the current lookup table pointer is not written by the ACE memory management logic at the end of a message. Therefore, if a subsequent message is received for the same subaddress, the same block of Data Words will be overwritten or overread.

The chief advantage of the single message mode is its simplicity. For a given receive subaddress, the CPU does not need to reference the lookup table pointer prior to accessing received Data Words. In comparison to other techniques, the single message buffer uses an absolute minimum amount of memory space. This reduces cost and allows relatively more RAM area to be used for subaddresses where bulk data transfers are being performed (see section on circular buffers).

For applications where the RT synchronously receives a periodic message to a particular subaddress, either the End-of-Message (EOM) or Subaddress Control Word (RX: [BCST:] INT ON EOM) interrupt should be

TABLE 5. SUBADDRESS CONTROL WORD Management Subaddress Buffer Scheme					
MM2	**MM1**	**MM0**	**RCV PBL BUF ENL**	**DESCRIPTION**	
0	0	0	0	Single Message	
0	0	0	1	Double Buffered: for receive (or broadcast) subaddress; if single message for transmit subaddress.	
0	0	1	X	128 Word	Circular Buffer of Specified Size
0	1	0	X	256 Word	
0	1	1	X	512 Word	
1	0	0	X	1024 Word	
1	0	1	X	2048 Word	
1	1	0	X	4096 Word	
1	1	1	X	8192 Word	



Note: Lookup table is not used for mode commands when enchanced mode codes are enabled.

FIGURE 1. SINGLE MESSAGE MODE

used if feasible. This allows the CPU to road the new data block with minimal software overhead. However, for asynchronous applications, it may be necessary to utilize the subaddress double buffering technique for received messages.

For transmitting of synchronous single-message data blocks, a software double buffering technique may be employed. That is, the host may alternate ("ping-pong") between two buffers of Data Words to be transmitted. The critical step is for the CPU to re-assign the lookup table pointer **after it** has written a new block of Data Words to be transmitted. This ensures data consistency by eliminating the possibility of transmitting a mixture of old and new Data Words. It should be noted that the ACE's protocol logic samples (reads) the Lookup Table pointer **only at the start of a message sequence.** That is, after the Command Word is received. As a result, it will not switch blocks during the processing of a message.

Another application of the single message technique involves the data wraparound subaddress. Paragraph 30.7 of MIL-STD-1553B Notice 2 requires that an RT provide such a mechanism to allow for an "end-to-end" self-test by a bus controller. It is suggested that subaddress 30 (11110) be used for the wraparound function. The wraparound subaddress involves receiving a message of "one to N words of any bit pattern" to the wraparound subaddress. In order to comply to Notice 2, the RT must be able to transmit, from the wraparound subaddress, the re received word pattern back to the bus controller. In order to ensure that the same block of words are transmitted as well as received, the single message mode must be used for the data wraparound subaddress.

DESCRIPTOR BLOCK

As shown in Figure 1, the ACE RT stores a 4-word block descriptor in the active area stack for each message processed. The Block Status Word contains information of whether a message is ongoing or has been completed, what bus channel it was processed on, an indication of circular buffer rollover, and a multitude of information identifying message error conditions. These include loop test (self-test) failure, illegal Command Words, errors in received Data Words, RT-to-RT transfer errors and other fault conditions. The Block Status Word may be used as a diagnostic tool in troubleshooting hardware and software system problems.

Following completion of a message, the Time Tag word reflects the contents of the ACE's 16-bit free-running Time Tag Register at the end of the message. One use of the Time Tag word is for the host to be able to determine how long a received message has been stored in a receive buffer; i.e., to determine data staleness or interrupt latency.

In the single message mode, the Data Block Lookup Pointer provides a convenience for the CPU to locate Data Word blocks. In the circular buffer and double buffered modes, its use is necessitated by the fact that the subaddress lookup table pointers are generally updated after every message. The received Command Word is stored in the fourth location of the descriptor for each message. In addition to providing an identifier

for each message, this allows the CPU to be able to easily scan through the stack to locate particular received Command Words.

CIRCULAR BUFFER MODE

The operation of the ACE circular buffer RT memory management mode is illustrated in Figure 2. As in the single message mode, the individual lookup table entries are initially loaded by the host processor. At the start of each message, the lookup table entry is stored in the third position of the respective message block descriptor in the stack area of RAM. Receive or transmit Data Words are transferred to (from) the circular buffer, starting at the location referenced by the Lookup Table pointer.

The size of the circular buffer for each transmit/receive/broadcast subaddress is specified by means of the respective Subaddress Control Word. It is selectable from among 128, 256, 512, 1024, 2048, 4096, or 8192 words.

If the OVERWRITE INVALID DATA bit (bit 11) of Configuration Register #2 is logic "0", the location after the last word accessed for the message is stored into the respective Lookup Table location, **regardless of whether or not there were any errors in the just completed message.** By so doing, data for the next message for the respective Tx/Rx/Bcst subaddress will be accessed to/from the next lower contiguous block of locations in the circular buffer.

If the OVERWRITE INVALID DATA bit (bit 11) of Configuration Register #2 is logic "l ", the location after the last word accessed for the message is written to the respective Lookup Table location **only following a valid received (or transmitted) message.** Assuming that the value of the Lookup Table pointer is updated, data for the next message for the respective Tx/Rx/Bcst subaddress will be accessed to/from the next lower contiguous block of locations in the circular buffer. Assuming that the OVERWRITE INVALID DATA bit is set, the Lookup Table pointer will not be updated at the end of the message, if there was an error in the message. This allows failed messages in a bulk data transfer to be retried without corrupting the circular buffer data structure, and without intervention by the RT's host processor.

It is strongly recommended that OVERWRITE INVALID DATA be set to logic "1" when performing bulk data transfers.

When the pointer reaches the lower boundary of the circular buffer, the pointer moves to the top boundary of the circular buffer, as shown in Figure 3. It is important to note that the boundaries of the circular buffers are determined by 128-word, 256-word, etc. boundaries in the ACE address space. The boundaries are not dependent on the starting address of the buffer.

For example, if a circular buffer size is programmed for 256 and the lookup table pointer is initialized to 400, the bottom boundary of the circular buffer is established as 511, not 655 (400 + 255). That is, the buffer address rolls over from 511 to 256.

As in the single message mode, the pointer to the start of the data block is stored in the third location of

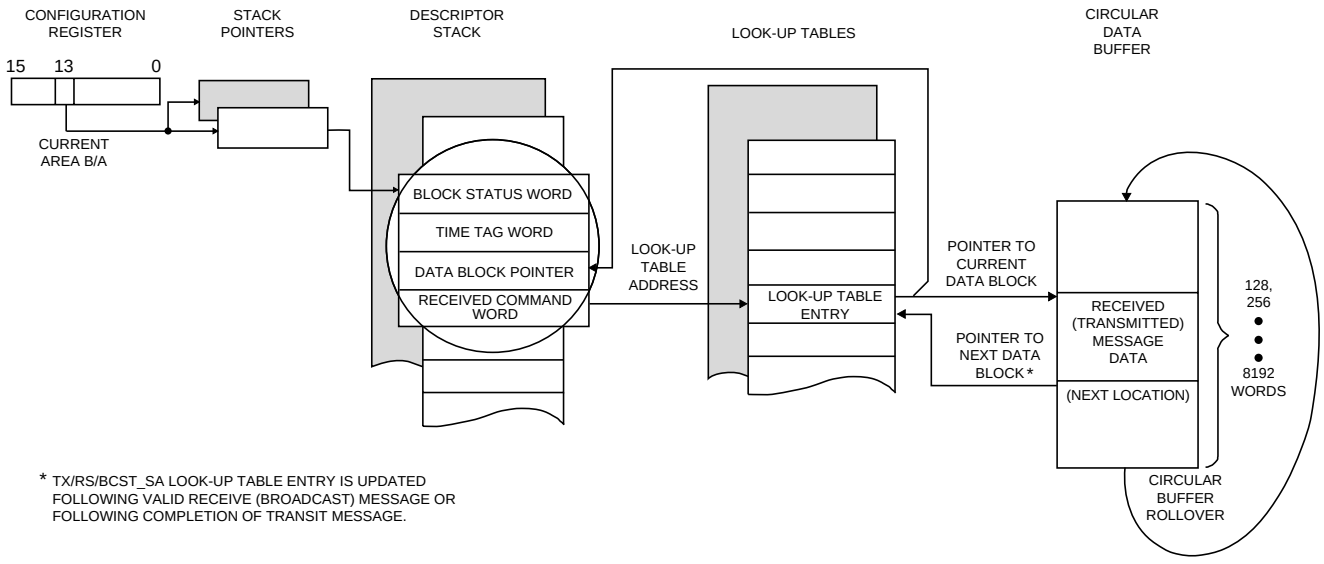


FIGURE 2. RT MEMORY MANAGEMENT – CIRCULAR BUFFER MODE

the message block descriptor (in the stack). This provides a convenience for the CPU to locate Data Words received from individual messages.

Circular Buffer Interrupt.

An important feature of the ACE's circular buffer capability is the circular buffer interrupt. This interrupt may be enabled by means of bits 13,8, and 3 of the Subaddress Control Word for transmit, receive, and broadcast messages, respectively, The use of this interrupt serves to offload the host processor when performing bulk data transfers. The interrupt will be issued after the circular buffer pointer has rolled over from the bottom to the top of the circular buffer address space. It should be noted that the operation of this interrupt request is effected by the OVERWRITE INVALID DATA bit. If OVERWRITE INVALID DATA is logic "0", the interrupt request will occur **immediately** after the word at the bottom boundary of the circular buffer has been accessed. If OVERWRITE INVALID DATA is logic "1", the interrupt request will occur **following the end of a valid (but not Invalid) message** in which the word at the bottom boundary of the circular buffer has been accessed.

Circular Buffer-Bulk Data Transfers.

The primary use of the circular buffer feature is for performing bulk data transfers, such as downloads of program data. Consider the example where the RT anticipates receiving 1000 Data Words. The CPU initializes for such a download by performing the following sequence:

(1) Write to Configuration Register #2, setting the ENHANCED RT MEMORY MANAGEMENT (bit 2) and OVERWRITE INVALID DATA (bit 11) bits to logic "1".

(2) The host should initialize the receive (or possibly, broadcast) bits of the respective Subaddress Control Word for a circular buffer of size 1024 words and enable the RX: (or BCST:) CIRCULAR BUFFER ROLLOVER interrupt request.

(3) The CPU should initialize the lookup table pointer exactly 24 words down from the top of the buffer.

(4) The host should then wait for a CIRCULAR BUFFER interrupt request. This may be verified by means of the Interrupt Status Register. By so doing, the host is able to determined that it has received exactly 1000 valid Data Words that may now be accessed.

The use of the circular buffer feature offloads the host CPU in a number of ways. These include:

(1) Aside from configuration registers, the only initialization required by the CPU is to write the Subaddress Control and Lookup Table pointer words.

(2) All "real time" pointer management is performed by the ACE RT, rather than by the host.

(3) There is no need to service interrupts, or poll for, individual received messages.

(4) There is no need for the host processor to be concerned about errors or message retries. If there is an error in a Received message (eg., a parity error), the ACE RT will not respond and the BC will retry the message. Since the ACE RT's OVERWRITE INVALID DATA bit has been programmed to logic "1", the value of the lookup table pointer will not be updated as a result of the failed message. When the BC retries the message, the valid received Data Words will overwrite the invalid received message.

(5) After initialization, the host need only wait for the Circular Buffer Rollover interrupt request. Following occurrence of the interrupt request, the host need only read the received multi-message data buffer. The operation of the circular buffer scheme enabled the 1000 valid words to be received with no host in intervention.

SUBADDRESS DOUBLE BUFFERING

The purpose of this mode is to provide the host processor with the highest possible degree of data consistency for received Data Words. As illustrated in Figure 3, this is accomplished by allocating two 32-bit Data Word blocks for each individual receive (and/or broadcast receive) subaddress. At any point in time, one of the blocks is designated as the active 1553 block while the other block is considered inactive.

The Data Words for the next receive message to that subaddress will be stored in the active block. Upon completion of the message, provided that the message was valid and Receive Double Buffering is enabled, the ACE will switch ("ping-pong") the active and inactive blocks automatically for that subaddress by writing the starting address of the inactive block to the respective lookup table address. This means that the latest, valid, complete received data block is always readily available to the host processor.

As illustrated in Table 3, subaddress double buffering

may be invoked globally, for all subaddresses, or may be utilized on an individual subaddress basis. In the latter case, double buffering for any particular receive (and/or broadcast) subaddress is programmable via the Subaddress Control Word. To enable double buffering for receive messages, RX: MM2-MM0 (bits 7-5) must set to logic "0") and the DOUBLE BUFFER ENABLE bit (bit 15) must be set to logic "1". If SEPARATE BROADCAST is enabled (bit 0 in Configuration Register #2 is set to logic "1"), then BCST: MM2-MM0 (bits 2-0) should be set to logic "0" and DOUBLE BUFFER ENABLE bit (bit 15) must be set to logic "1" to enable double buffering for a broadcast subaddress.

To make best use of the subaddress double buffering mode, it is best to invoke the ENHANCED RT MEMORY MANAGEMENT mode by setting bit 1 of Configuration Register #2 set to logic "1". This allows the double buffering for individual subaddresses to be disabled while reading the data.

The basic software algorithm for reading the latest, most consistent block of valid data words received to a given subaddress is outlined as follows:

(1) Disable double buffering for the desired subaddress by clearing RECEIVE DOUBLE BUFFER ENABLE, bit 15 of the Subaddress Control Word to logic "0".

(2) The host processor should then read the current value of the data block pointer in the lookup table. This pointer references the active block. The latest consistent data block can be referenced in the inactive block by inverting bit 5 of the current data block pointer. The current lookup table entry should not be modified.

(3) Read the Data Words from the inactive block. By definition, these represent the latest, most consistent sample of valid Data Words received to the particular subaddress (at the time of step 1).

(4) Re-enable double buffering for the accessed subaddress by seeing the RX DOUBLE BUFFER ENABLE bit (bit 15) in the appropriate Subaddress Control Word to logic "1".

Many system Interface Control Drawings (ICDs) require that a method be provided to ensure sample data consistency. The principal advantage of the subaddress double buffering mode is that it ensures data consistency. That is, it guarantees that the host processor will never read a mixture of data words from the previous sample and the most recent sample.

The use of the double buffering capability, particularly the capability to disable double buffering on a subaddress basis, is essential for the case in which the processor is asynchronously reading data from a particular RT subaddress. There are many applications in which the RT is receiving data from the BC to a particular subaddress, but the RT's host processor does not immediately need to make use of the information. The ACE's RT subaddress double buffering allows the host processor to ignore received messages until the information is actually needed and still guarantee data consistency with minimum processor overhead. This allows the host may to read the latest, complete, valid data block from a subaddress at any time.

RT SOFTWARE INITIALIZATION PROCEDURE

The following software sequence is typical of the

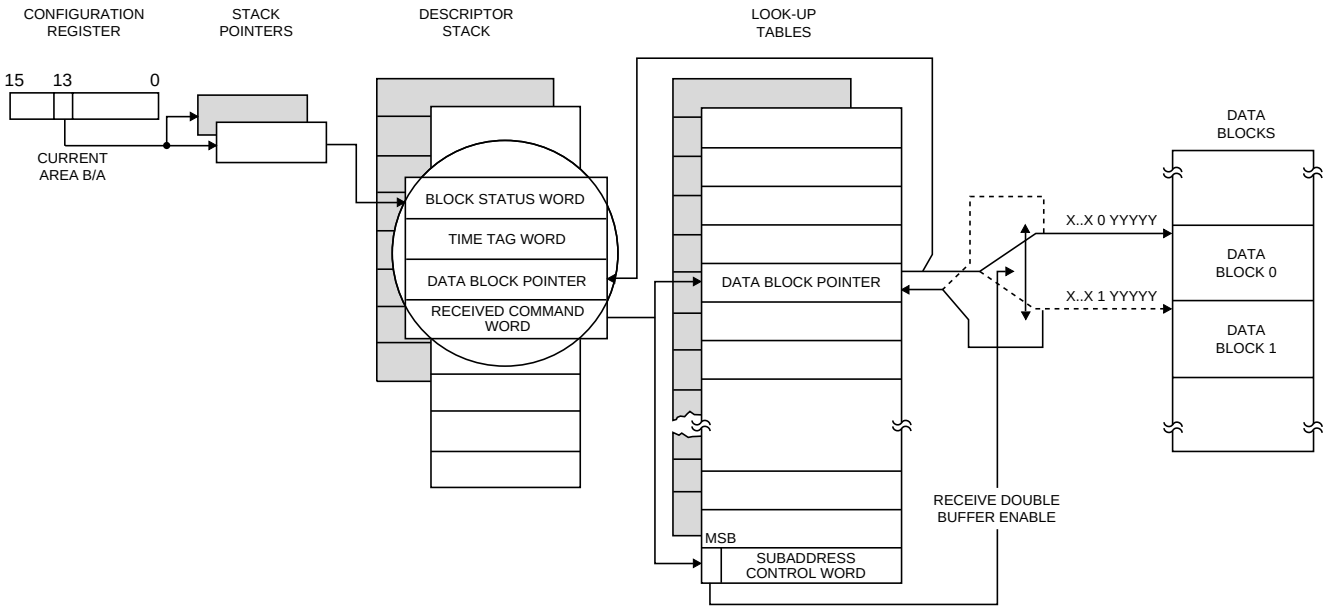


FIGURE 3. SUBADDRESS DOUBLE BUFFERING MODE

steps the host CPU should perform following power turn-on to configure the ACE for RT operation. For most applications, it is possible to skip many of the steps indicated.

(1) Perform a software reset, by writing 0001 (hex) to the Start/Reset Register.

(2) If any of the Enhanced mode functions (eg., subaddress double buffering) are to be used, invoke the ACE RT's enhanced mode by writing 8000 (hex) to Configuration Register #3.

(3) Initialize the Interrupt Mask register. For many RT applications, the EOM interrupt will generally be enabled. In other instances, the RT Subaddress Control Word, RT Circular Buffer Rollover, and/or Format Error Interrupt Requests may also be enabled. The RT Subaddress Control Word Interrupt enables interrupt requests to be issued following messages to specified Transmit, Receive, or Broadcast subaddresses. The RT Circular Buffer Rollover Interrupt may be used to provide an interrupt request following a multi-message reception or transmission of a specified number of Data Words to/from a given subaddress.

(4) Load the starting location of the Stack into the Active Area Stack Pointer location in RAM.

(5) As an option, initialize the Active Area Stack. If there is a desire to poll the Stack RAM while 1553 messages are being processed, the Block Status Word locations for the respective message block descriptors (relative address locations 0, 4, 8...[stack size 4] in the stack) should be cleared to 0000. A Block Status Word of 0000 (SOM = EOM = 0) indicates that a message has not been processed yet.

(6) Initialize the Active Area Lookup Table. The Lookup Table address for each transmit, receive. and (optionally) broadcast subaddress should be initialized as the pointer value for each respective lookup table. If the RT is going to be used in the enhanced RT Memory Management mode, it will also be necessary to select the memory management and interrupt options for each subaddress by initializing the Subaddress Control Words for the Active Area.

If there are several unused subaddresses for an RT, it is recommended that the Lookup Table pointers for these be initialized to the same value in order to conserve memory space.

(7) If ENHANCED MODE CODE HANDLING (bit 0 of Configuration Register #3) is not used, the pointers for receive subaddresses 0 and 31 (for Synchronize with Data messages) generally get loaded with the same pointer value. Similarly, the Lookup Table addresses for transmit subaddresses 0 and 31 (for Transmit Vector Word messages) generally get loaded with the same pointer value.

If ENHANCED MODE CODE HANDLING is enabled. Data Words for these mode codes are stored in locations 0111 (for Synchronize with data) and 0130 (for Transmit Vector Word).

(8) Initialize Configuration Register #2. This involves selecting use of the following functions: ENHANCED RT MEMORY MANAGEMENT should be selected if it is desirable to select the circular buffer feature and/or subaddress double buffering features on an individual subaddress basis. SEPARATE BROADCAST should be set if it is necessary to comply to Notice 2 for broadcast messages.

ENHANCED INTERRUPTS should be enabled if the CPU needs to poll using the Interrupt Status Register and/or it is desired that one or more of the following conditions cause an interrupt: Transmitter Timeout, RT Command Stack Rollover, or RT mode code interrupt. OVERWRITE INVALID DATA should be set to logic "1" if the circular buffer mode is used with one or more subaddresses. SUBADDRESS DOUBLE BUFFERING should be enabled, if desired. Finally, BUSY LOOKUP TABLE ENABLE should be logic "1" if there is a need for the Busy bit in the RT Status Word to be set for particular T/R / Bcst subaddresses.

(9) Initialize Configuration Register #3. If one or more of the Enhanced Mode features are to be used, **bit 15 must be maintained at logic "1".** The RT Stack size is programmable with choices of 256 words (default, 64 messages), 512, 1024, or 2048 words (512 messages) by bits 14 and 13. Other RT features that may be selected by this register include ILLEGALIZATION (default = "0" = enabled), ALTERNATE STATUS (allowing software programming of **all 11** Status Word bits), the choice of storing

or not storing words for illegal or "BUSY" messages, and ENHANCED MODE CODE HANDLING.

If ENHANCED MODE CODE HANDLING is selected, Data Words for mode codes are stored in address locations 0110-013F, and interrupt requests for individual mode codes may be enabled by means of a table in address locations 0108-010F. Other RT options selectable by Configuration Register #3 include 1553A MODE CODES ENABLED, and RTFAIL / RTFLAG WRAP ENABLED. 1553A MODE CODES ENABLED causes only subaddress 00000 to be treated as a mode code subaddress. RTFAIL / RTFLAG WRAP ENABLE causes the RT FLAG Status bit to be automatically set following a failure of the loop test.

(10) Initialize Configuration Registers #4 and #5. If EXTERNAL BIT WORD ENABLE is logic "1", the Data Word for a Transmit BIT Word mode command is accessed from a RAM location, rather than from an internal register. INHIBIT BIT WORD IF BUSY prevents the BIT Word from being transmitted if the RT is Busy. If MODE CODE

OVERRIDE BUSY is logic '1", this enables the ACE RT to transmit a Data Word in response to a Transmit Vector Word or Reserved transmit mode command, even if the RT is busy.

For the BU-65171 and BU-61581 versions of the ACE, LATCH RT ADDRESS WITH C.R. #5 allows the ACE's RT Address to be software programmable by means of bits 5 through 0 of Configuration Register #5. If LATCH RT ADDRESS WITH C.R. #5 is logic "1", writing to Configuration #5 causes the RT Address to be read from pins RTAD4-0 and RTADP and latched internally. After the RT Address has been programmed, it is suggested that LATCH RT ADDRESS WITH C.R. #5 be cleared to logic "0" to prevent an erroneous overwrite.

(11) If RT Illegalization is used, the CPU should initialize the Illegalization Table, address locations 0300-03FF.

(12) If the BUSY LOOKUP TABLE is enabled, select the desired subaddresses to be Busy by programming the Busy table, address locations 0240 through 0247.

(13) If ENHANCED INTERRUPTS are enabled and ENHANCED MODE CODE HANDLING is enabled, interrupts for selective mode code messages may be enabled by programming locations 0108 through 010F.

(14) Data to be transmitted on the 1553 bus (in response to transmit commands) should be written into the appropriate data blocks. As an option, the locations for Data Words for anticipated receive Data Words may be initialized to zero.

(15) To configure the ACE as an on-line RT, write to Configuration Register # I, setting bit 15 (MSB) to logic "l" and bit 14 to logic "0". The current active area is selected by the seeing of bit 13 (0 for A, 1 for B). If ALTERNATE RT STATUS is not enabled, bits 11 through 8 should be initialized to select the values for the RT Status Word bits Dynamic Bus Control Acceptance, Busy, Service Request, and Subsystem Flag. Also, in the enhanced mode, the RT FLAG Status Word bit is programmable by bit 7. These bits must be programmed for the logical inverse of their desired values in the RT Status Word.

If ALTERNATE RT STATUS is enabled, bits 10 through 0 of the ACE's RT Status Word are programmable via bits 11 through I of Configuration Register #1. In this case, the logical values (not the inverse) of the intended Status Word bit values must be programmed.

SERVICING COMPLETED RT MESSAGES

The ACE RT provides a number of techniques for determining when a message has been processed. These methods support both polling-driven and interrupt-driven software. There are several polling methods that may be used. These include:

(1) In the ENHANCED mode, the host may continuously poll RT MESSAGE IN PROGRESS, bit 0 of Configuration Register #l. This bit will return logic "0" while the ACE RT is not processing a message. During the time that the ACE RT is servicing a message (after the receipt of a Command Word), RT MESSAGE IN PROGRESS will return logic "1". When the message completes, RT MESSAGE IN PROGRESS will once again return logic "0".

(2) The CPU can poll the contents of the Stack Pointer RAM location. The active area Stack Pointer increments by four **at the beginning** of each message being processed (after receipt of a Command Word).

(3) If the host needs to determine the occurrence of a particular Command Word, it may do so by

polling the RT Last Command Register. It should be noted that the contents of this register are updated **at the beginning** of a message being processed. The CPU may then poll the EOM (End-of-Message) bit of the Interrupt Status Register to determine when the message has been completed. See explanation below.

(4) If the ACE is programmed for the ENHANCED mode and ENHANCED INTERRUPTS (bit 15 of Configuration Register #2) are enabled, the CPU may poll the Interrupt Status Register. In this mode, the various bits in the Interrupt Status Register will become set **regardless** of the programming of the corresponding bits in the Interrupt Mask register.

In this mode, the ACE may determine that a message has been **completed** by polling the Interrupt Status Register until the EOM (End-of-Message) bit returns logic " 1".

In this mode, the host processor may also poll to determine when a message has been processed for a particular transmit, receive, or broadcast subaddress. To enable do this, ENHANCED RT MEMORY MANAGEMENT (bit 1 of Configuration Register #2) must be invoked. To cause the bit to be set for a particular Tx/Rx /Bcst-subaddress, it is then necessary to set the appropriate bit ([TX:, RX:. or BCST:1] INT on EOM) bit in the desired Subaddress Control Word to logic "1". All other Subaddress Control Word ([TX:, RX:, or BCST:] INT on EOM) bits should be programmed to logic "0". This will cause the RT SUBADDRESS CONTROL WORD EOM bit in the Interrupt Status Register to be set to logic "1" after completion of the desired message.

Similarly, the host may poll for receipt of a particular mode code message. This feature is enabled by invoking ENHANCED MODE CODE HANDLING (bit 0 of Configuration Register #3). The desired mode code may be selected by setting the appropriate bit in the Mode Code Selective Interrupt Table (address range 0108-010F).). When the specific mode code message has been completed, the RT MODE CODE bit of the Interrupt Status Register will return logic "1".

Similarly, the Interrupt Status Register may be polled to determine the occurrence of FORMAT ERROR, CIRCULAR BUFFER ROLLOVER, and/or COMMAND STACK ROLLOVER conditions. FORMAT ERROR indicates any error in a received message, other than an invalid Command Word: sync or Manchester encoding, parity, bit count, word count, or RT-to-RT transfer errors. CIRCULAR BUFFER ROLLOVER may be used to signal completion of a multi-message bulk data transfer, as described above. COMMAND STACK ROLLOVER occurs when the Stack rolls over at an address boundary of 256, 512, 1024, or 2048 words, as programmed in Configuration Register #3.

If interrupts are used, the normal procedure would be to not invoke ENHANCED INTERRUPTS. This allows the Interrupt Mask Register to be used to enable bits in the Interrupt Status Register as well as the corresponding interrupt requests for only selected condition(s), as discussed above.

RT ERROR HANDLING

As discussed above, the preferred method for handling erroneous messages is to make use of the circular buffer and/or double buffering techniques. In the case of the circular buffer mode, the OVERWRITE INVALID DATA bit should be set to logic "1". With these techniques, Data Words received and stored from invalid messages will be automatically overwritten by the ACE RT. In most systems, the bus controller will retry failed messages. By so doing, the occurrence of errors and message retries is transparent to the RT's host processor.

If necessary, the RT's host processor may ascertain the occurrence of failed messages by several methods:

(1) Determine, by polling or interrupt techniques, when a FORMAT ERROR condition occurs.

(2) Read the Block Status Words for all messages processed. Bits 12-9 and 6-0 all indicate error conditions in received messages.

SUMMARY OF SUBADDRESS MEMORY MANAGEMENT TECHNIQUES

Table 5 provides a summary of the subaddress memory management techniques that may be used for various types of receive and transmit data transfers. This encompasses applications involving both synchronous and asynchronous systems. In this context, "synchronous" implies that the action of the RT's host processor is tightly coupled to activity from the 1553 bus. In such applications, which are typically interrupt-driven with low interrupt latency, the host CPU will

TABLE 5. SUMMARY OF SUBADDRESS MEMORY MANAGEMENT METHODS		
TRANSMIT OR RECEIVE	**DESCRIPTION OF DATA TRANSFER**	**RECOMMENDED SUBADDRESS MEMORY MANAGEMENT TECHNIQUE**
Receive	Synchronous. That is, the host processor is reliably able to access received messages prior to the start of the next receive message to the subaddress.	Single message.
Receive, Transmit	Wraparound subaddress.	Single message.
Receive	Bulk (multi-message) data transfer.	Circular buffer.
Receive	Asynchronous, such that the 1553 message rate (messages per second) is much greater than the rate that the RT's host CPU samples received data blocks.	Subaddress double buffering as described above.
Receive	Asynchronous, such that the 1553 message rate (messages per second) is approximately equal to or less than the rate with which the RT's host CPU samples received data blocks.	A modified technique using the subaddress double buffering feature. This is required in order to preclude the possibility that the RT loses a received message from the 1553 bus. Refer to "Asynchronous Received Messages" section below.
Transmit	Synchronous single-message transfer, such that the RT's host CPU is reliably able to rewrite the transmit data table before the beginning of the next transmit message to the subaddress.	Single message.
Transmit	Asynchronous single-message transfer. In this case a transmit message to the subaddress may begin during the time that the RT's host CPU is updating the transmit data table.	Single message. However, a form of software double should be implemented to estimate the possibility of transmitting a combination of old and new Data Words. The CPU should set up two transmit data blocks. The lookup table pointer should be alternated between the starting locations of the two blocks. The pointer should only be updated after the RT's host CPU has finished writing Data Words to the new transmit data table.
Transmit	Bulk (multi-message) data transfer.	Circular buffer. For explanation of error handling, refer to "Bulk Data Transfers" in the section below.

always read a received data table or update a transmitted data table a short period of time after a 1553 message has been processed.

In an "asynchronous" environment, the actions of the host CPU are loosely coupled from the timing of 1553 bus messages. In these situations, data consistency is more of a concern, since it is possible for a 1553 message to be received to a given subaddress during the time that the RT's host processor is accessing the data table for that subaddress.

Referring to Table 5, there are two situations that require special attention. The two situations are:

(1) For an RT receiving asynchronous messages such that the 1553 message rate is less than, or approximately equal to the rate that the host CPU is sampling data from the receive Data Word tables.

(2) For an RT transmitting a bulk (multi-message) data transfer.

ASYNCHRONOUS RECEIVED MESSAGES

First, consider the case of an RT receiving asynchronous messages (see Figure 4). If a message transfer to the respective receive subaddress is completed during the time that the host is accessing the previous message, the new message data will be stored in the current "active" data block. However, in order to ensure data consistency, double buffering was temporarily disabled for the subaddress. As a result, the lookup table pointer **will not toggle** to the starting location of the "inactive" block, being accessed by the CPU. When the CPU finishes accessing the "inactive" block, it re-enables the double buffering for the subaddress. Note, at this time, the ACE **will not update the value of the lookup table pointer.** Consequently, if the host CPU goes to sample the subaddress a second time before a new 1553 message comes in, it will read the same "inactive" data block again. **The data from the most recent received message stored in the "active" data block, will be lost.** If the 1553 message rate is much greater than the CPU's sampling rate, the scenario described above should not present a problem. That is, by the time the CPU goes to read its next sample, the "active" data block will have been overwritten by the next (i.e., frequent) receive message to the subaddress, resulting in a toggling of the pointer. In this case, therefore, data consistency can be ensured by means of

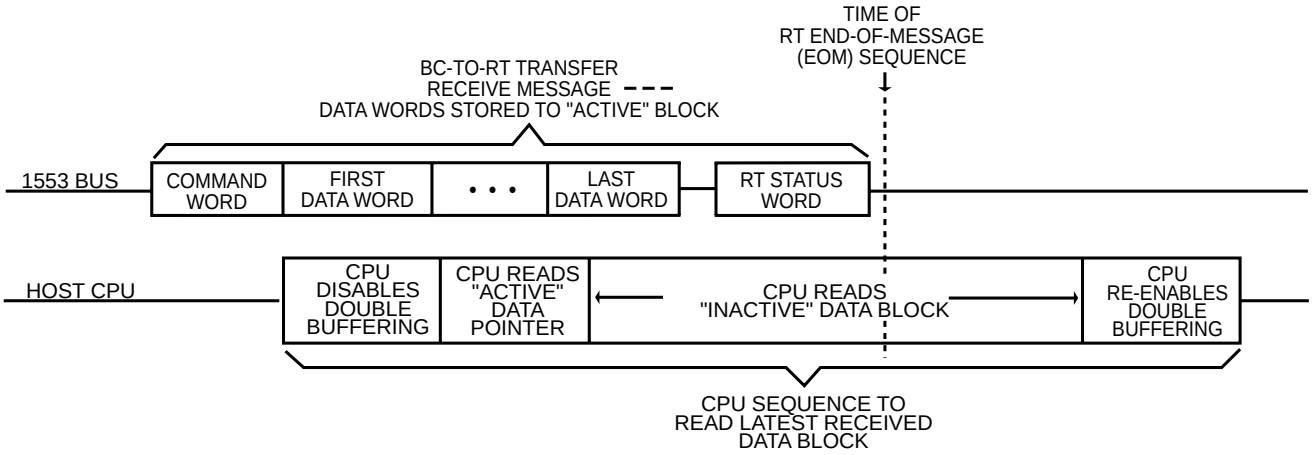


FIGURE 4. ASYNCHRONOUS RECEIVED MESSAGES

the subaddress double buffering mode. That is, the simple procedure outlined above—disabling the double buffering during the CPU "read" time—may be used to eliminate the possibility of reading a mixture of old and new Data Words. Losing the current "active" data block is not a concern.

However, the situation is a bit more complicated for the asynchronous case where the 1553 message rate is less than, or approximately equal to the CPU sampling rate. Reference Figure 5. In this scenario, there is a strong likelihood that a message which completed during the CPU's access time will not be overwritten prior to the next sampling interval. In this case, the same data block will be read again. The more recent data, from the new received message, will be lost.

In this instance, a slight modification is needed in order to guarantee that the CPU always reads the most recent received data block. There are several ways of accomplishing this. A fairly simple technique is outlined in the steps below.

Initialization: For the case of a 1553 interface accessed asynchronously by the host, it is assumed that interrupt will not be used. Instead, to enable polling using the Interrupt Status Register ENHANCED INTERRUPTS should be enabled and the Interrupt Mask Register should be initialized to zero. In addition, the RX:, TX:, and BCST: INT ON EOM and INT ON CIRC BUFFER ROLLOVER bits of the 32 Subaddress Control Words should be initialized to logic "0". OVERWRITE INVALID DATA should be set to logic "1".

The sequence for accessing the most recent received data block is summarized as follows:

(1) Write to the Subaddress Control Word for the subaddress to be accessed. Disable double buffering for the desired subaddress by clearing RECEIVE DOUBLE BUFFER ENABLE, bit 15, to logic "0". During the same write cycle, set the RX: (or BCST:) INT ON EOM bit to logic "1".

(2) The host processor should then read the current value of the data block pointer in the lookup table. This pointer references the active block. The CPU should store the value of this pointer for future reference. The most recent data sample can be referenced in the inactive block by inverting bit 5 of the active data block pointer (ACTIVE PTR x or 32). The value of the lookup table entry should not be modified at this time.

(3) Read the Data Words from the inactive block. By definition, these represent the latest, most consistent sample of valid Data Words received to the particular subaddress (at the time of step [1]).

(4) Re-enable double buffering for the accessed subaddress by setting the RX DOUBLE BUFFER ENABLE bit (bit 15) in the appropriate Subaddress Control Word to logic "1". As part of the same write cycle, clear the RX: (or BCST:) INT ON EOM bit to logic "0".

(5) **IMMEDIATELY** following step (4), read the value of the Interrupt Status Register. If bit 4, RT SUBADDRESS CONTROL WORD EOM, is logic "1", this indicates that the end of a mes-

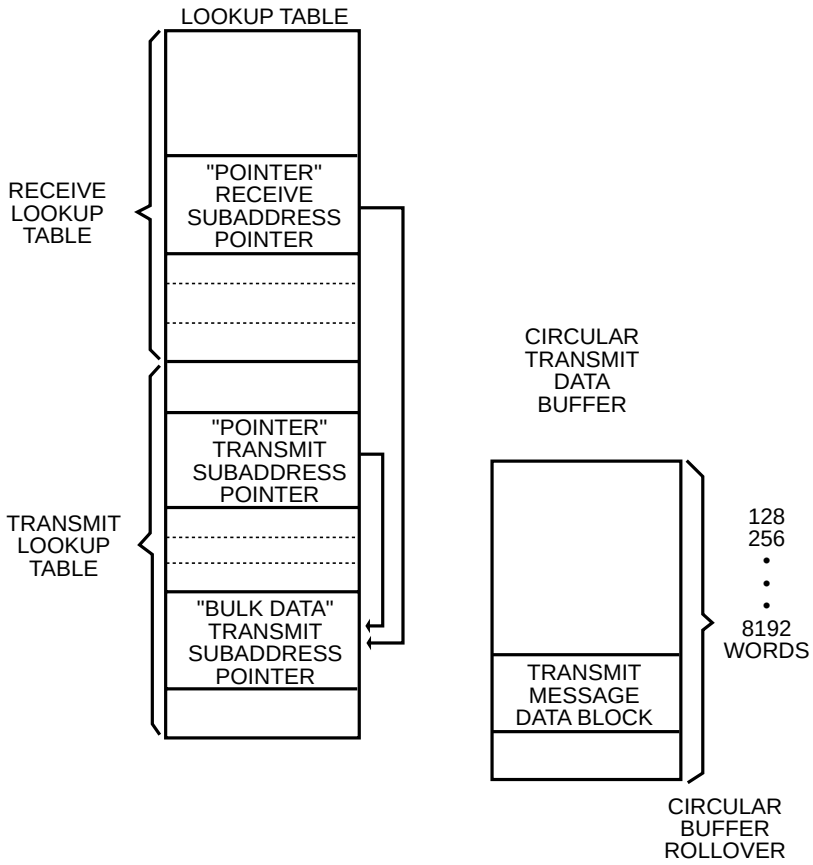


FIGURE 5. BC ACCESS TO AC RT LOOKUP TABLE FOR BULK TRANSMIT TRANSFERS

sage to the accessed subaddress occurred **during the time that the CPU was reading the data block.** In this case, the CPU should **IMMEDIATELY** write the value of the inactive block as the updated value of the lookup table pointer for the receive (or broadcast) subaddress. This ensures that if another message is not received in the interim, the CPU will next read the latest block of received Data Words, rather than reread the previous sample.

BULK DATA TRANSMIT TRANSFERS

The OVERWRITE INVALID DATA feature provides an airtight mechanism for ensuring data validity for received multi-message data transfers. This is a result of MIL-STD-1553's command/response protocol. That is, a bus controller will be informed if an RT does not receive a completely valid message (the RT will no respond). As a reset, the BC can immediately retry the failed message, resulting in the RT overwriting the invalid data.

The situation is a bit more complicated for bulk data transfers in which the RT is transmitting, rather than receiving. This is due to the fact that MIL-STD-1553, for an RT-to-BC transfer, does not provide a direct mechanism for the bus controller to acknowledge receipt of a valid transmission by an RT. It is the responsibility of the BC to "tell" the RT that the message needs to be retried.

Similar to the problem of asynchronous received messages, there are a number of ways of dealing with the problem of bulk data transfers transmitted by an RT. By necessity, the problem must be managed by the bus controller. Ideally, the method used should be transparent to the RT's host processor.

One relatively simple method makes use of the ACE's circular buffer feature and the fact that the ACE's RT lookup tables are memory mapped. This provides a mechanism for the BC to access the lookup table pointer for the subaddress involved in the bulk data transfer.

Very simply, the RT software should allocate one subaddress for the BC to access the lookup table pointer. As shown in Figure 5, the receive and transmit lookup table pointers for one subaddress (NOT the subaddress involved in the bulk data transfer) should both point to the location of the pointer for the bulk transfer subaddress. The

Subaddress Control Word for this "pointer" subaddress should be initialized for "single message mode", for both transmitting and receiving.

By so doing, the BC is able to retry failed RT-to-BC transfer messages in a way that is transparent to the RT's host CPU.

It should be kept in mind that the BC must distinguish a "no response" from a "format error". A "no response" indicates the RT's non-acknowledgement of the transmit Command Word. In this case, the BC should simply retry the message. A "format error" indicates acknowledgement by the RT. However, it also indicates an error such as sync or Manchester encoding, parity, bit count, or word count in the RT's transmission or the BC's reception of the RT response.

For the case of a "Format Error," the BC's retry sequence is outlined as follows:

(1) After the BC has determined that the RT's response was invalid, it should send a transmit command to the RT's "pointer" subaddress. The RT should respond with a Data Word indicating the current value of the lookup table pointer for the "bulk data" subaddress.

(2) Assuming a message size of 32 Data Words for the "bulk data" subaddress, the BC's host CPU should subtract 32 from the received pointer value. That is,

$$PTR \leftarrow PTR - 32.$$

(3) The BC should send a receive command to the "pointer" subaddress, loading the new value (actually, the old value) of the bulk data pointer. This will cause the RT to retransmit the failed data block.

(4) The BC should retry the failed transmit message for the "bulk data" subaddress.

ILLEGALIZATION AND BUSY

Aside from memory management, another important feature for RT operation is programmable command illegalization. Commands to the ACE RT may be illegalized based on broadcast, T/R bit, subaddress, word count, and mode code. The illegalization scheme is programmable by the host processor in the ACE's RAM, providing a degree of self-testability.

A stipulation of MIL-STD-1553B Notice 2 (30.5.3) states that "...the setting of the busy bit, shall occur only as a result of particular commands/messages sent to the RT." This is subject to interpretation. The ACE RT provides compliance by including an option such that the Busy bit may be programmed to be set as a function of the command broadcast, T/R bit, and subaddress.

For both illegalization and busy, the ACE RT provides two different options for received messages. In the case of illegalization, MIL-STD-1553B (4.4.3.4) states that the RT "...not use the information received." For the case of setting of the Busy bit, the standard (4.3.3.5.3.8) states "...the RT or subsystem is unable to move data to or from the subsystem… ." In either case, there is the question of whether a shared RAM is part of the "RT" or part of the "subsystem". The ACE RT supports both interpretations by allowing for either storage or non-storage of received Data Words for messages in which the RT responded with its Busy and/or Message Error (indicating illegal) bit set. In the ACE's enhanced RT mode, these options are programmable by means of bits 4 and 3 of Configuration Register #3.

HEADQUARTERS AND MAIN PLANT: Data Device Corporation, 105 Wilbur Place, Bohemia, NY 11716-2482
TEL: (631) 567-5600, FAX: (631) 567-7358, Toll Free 1-800-DDC-5757
WEST COAST: Data Device Corporation, 5912 Bolsa Ave., Suite 201, Huntington Beach, CA 92649, (714) 895-9777, FAX: (714) 895-4988
SOUTHEAST: Data Device Corporation, 21400 Ridgetop Circle, Suite 110, Sterling, VA 20166, (703) 450-7900, FAX: (703) 450-6610
UNITED KINGDOM: DDC U.K., Ltd., Mill Reef House, 9-14 Cheap Street, Newbury, Berkshire, RG14 5DD, England, TEL: +44-(0)1635-811140, FAX: +44-(0)1635-32264
IRELAND: DDC Ireland Ltd., Cork Business and Technology Park, Model Farm Road, Cork, Ireland, TEL: +353-21-341065, FAX: +353-21-341568
FRANCE: DDC Electronique, 10 Rue Carle-Hebert, 92400 Courbevoie, France, TEL: +33-(0)1-41-16-3424, FAX: +33-(0)1-41-16-3425
GERMANY: DDC Elektronik, GmbH, Ahornstrasse 2, 82296 Schoengeising, Germany, TEL: +49-(0)8141-349-087, FAX: +49-(0)8141-349-089
SWEDEN: DDC Elektronik AB, Danderyd Campus, Plan 6, Morby Centrum, 182 31 Danderyd, Sweden, TEL: +46-(0)8-54490044, FAX +46-(0)8-7550570
JAPAN: DDC Electronics K.K., Koraku 1-chome Bldg. 7F, 2-8, Koraku 1-chome, Bunkyo-ku, Tokyo 112, Japan, TEL: +81-(0)3-3814-7688, FAX: +81-(0)3-3814-7689
WORLD WIDE WEB: http://www.ddc-web.com

ILC DATA DEVICE CORPORATION
REGISTERED TO ISO 9001
FILE NO. A5976