

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

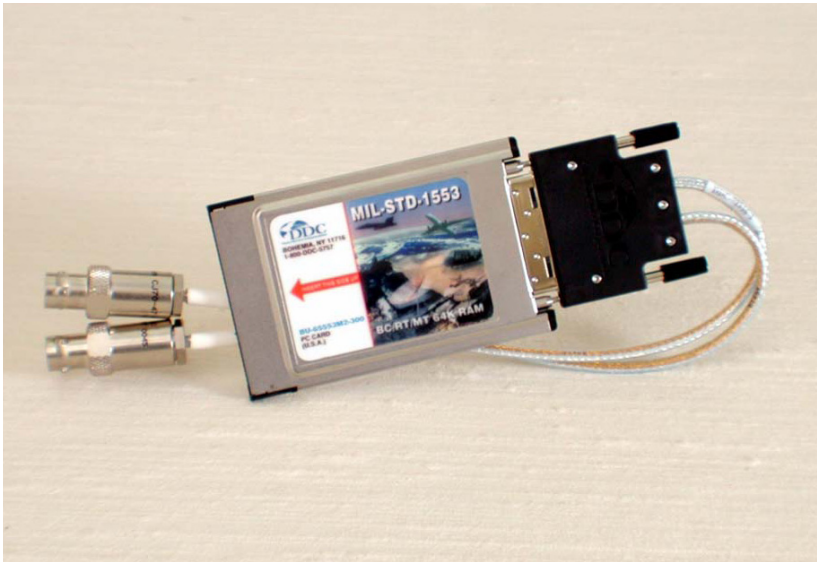
Expert team | Trust guarantee | 100% satisfaction

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

Visit our website - *Click HERE*

BU-65553

MIL-STD-1553 A/B BC/RT/MT PCMCIA INTERFACE CARD



DESCRIPTION

DDC's BU-65553 is a versatile PCMCIA card designed for the test and simulation of MIL-STD-1553 systems. Based on DDC's new Enhanced Mini-ACE®, it provides full, intelligent interfacing between a dual redundant MIL-STD-1553 data bus in a PCMCIA form factor.

Advanced architectural features of the Enhanced Mini-ACE include a highly autonomous bus controller, an RT providing a wide variety of buffering options, and a selective message monitor. The card incorporates 64K by 17 words of RAM.

The on-board Enhanced Mini-ACE is backward compatible with the older generations of ACE, and the older BU-65550/1/2 PCMCIA cards. This allows existing software applications 100% compatibility with the new hardware.

When operating in Enhanced Mini-ACE mode, the Bus Controller provides a new and powerful level of autonomous operation. This is made possible through the use of an on-board message sequence control processor and a programmable instruction list. From this list, the Enhanced Mini-ACE will be able to make operational decisions on its own instead of burdening the host.

The Enhanced Mini-ACE Remote Terminal functions remain the same as previous versions of the ACE, with the addition of a Global Circular buffer which can be set up to contain all messages received by the Remote Terminal, a circular buffer half-full interrupt and an interrupt status queue. With the use of the 50% and 100% rollover interrupts, mass data transfer becomes much easier to implement.

"C," Visual Basic, LabWindows® CVI, and LabVIEW® programming libraries are supplied with the card. These libraries provide complete control of the boards capabilities.

Data Device Corporation
105 Wilbur Place
Bohemia, New York 11716
631-567-5600 Fax: 631-567-7358
www.ddc-web.com

Make sure the next Card you purchase has...

DDC on Board®

FEATURES

- Type II PCMCIA 2.10 Compatible PC Card
- Enhanced Mini-ACE BC/RT/MT Architecture
- 64K Word RAM per Channel
- Highly Autonomous BC Architecture
  - Asynchronous Messages
  - Message Timing Control
  - Bulk Data Transfers
  - Data Block Double Buffering
  - Retries and Bus Switching
- RT Buffering Options
  - Single
  - Double
  - Subaddress Circular
  - Global Circular
- Selective Message Monitor
- PCI Interrupt Support
- Runtime Libraries for Microsoft Linux Operating Systems, Menu Software for Windows® 3.1, Windows 95/98, Windows NT®, and Windows 2000/XP.

FOR MORE INFORMATION CONTACT:

Technical Support:
1-800-DDC-5757 ext. 7771

All Trademarks used herein are the property of their respective owners.
© 2001 Data Device Corporation

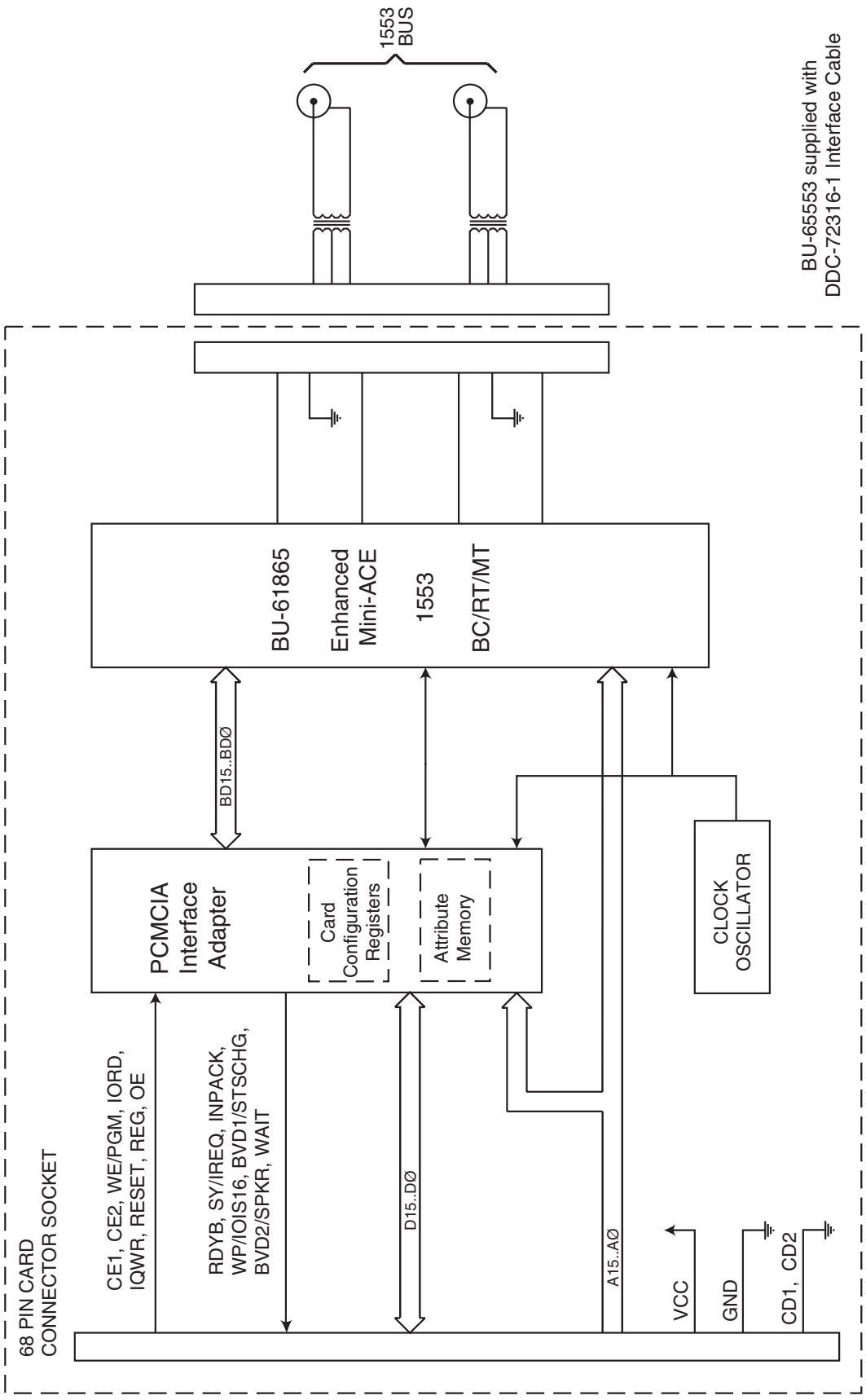


FIGURE 1. BU-65553 BLOCK DIAGRAM

TABLE 1. BU-65553 HARDWARE SPECIFICATIONS

PARAMETER	MIN	TYP	MAX	UNITS
ABSOLUTE MAXIMUM RATINGS				
+5 V Supply Voltage	-0.3		7.0	V
RECEIVER				
Threshold Voltage, Transformer-Coupled, Measured on Stub	0.200		0.860	VP-P
TRANSMITTER				
Differential Output Voltage				
Transformer-Coupled, Measured on Stub	18	20	27	VP-P
POWER SUPPLY REQUIREMENTS				
VOLTAGES/TOLERANCES				
• + 5 V	4.5		5.5	V
Current Drain @ + 5.0 V				
• Idle			180	mA
• 25% Transmitter Duty Cycle			296	mA
• 50% Transmitter Duty Cycle			412	mA
• 100% Transmitter Duty Cycle (See Note 3)			645	mA
1553 MESSAGE TIMING				
RT Response Time	4		7	µsec
BC Intermessage Gap (See Note 1)		9.5		µsec
BC/RT/MT Response Timeout (18.5 µsec nominal, See Note 2)	17.5	18.5	19.5	µsec
Transmitter Watchdog Timeout		668		µsec
THERMAL				
BU-65553				
Operating Temperature	0		55	°C
Storage Temperature	-40		85	°C
MECHANICAL DESIGN				
Shock: per MIL-STD-202F Test Condition J	30g, 11 mS			
Vibration: per MIL-STD-2004D Test Condition A	10g peak, 10 to 500Hz			
Humidity: Non-Condensing	5% - 95%			
PHYSICAL CHARACTERISTICS				
Size — Standard Type II PCMCIA	3.370 x 2.126 x 0.197 (85.6 x 54.0 x 5.0)			in (mm)
Weight	2.8 (80)			oz (gm)

Notes for TABLE 1:

(1) Typical value for minimum intermessage gap time. Under software control, may be lengthened (to 65,535 ms minus message time), in increments of 1 ms.

(2) Software programmable (4 options). Includes RT-to-RT Timeout (Mid-Parity of Transmit Command to Mid-Sync of Transmitting RT Status).

(3) 100% Duty Cycle at MAX Transmit Amplitude [27 V PP].

INTRODUCTION

The BU-65553 is the latest generation DDC PCMCIA interface card based on the new Enhanced Mini-ACE MIL-STD-1553 BC/RT/MT.

The on-board Enhanced Mini-ACE is backward compatible with the previous versions of ACE. This allows existing software applications written for the older BU-65550/1/2 PCMCIA cards 100% compatibility with the new hardware.

The BU-65553 supports PCMCIA “Plug and Play” installation and configuration that greatly simplifies the installation of the card into a PCMCIA compatible computer.

The operating modes of this card are controlled through the use of the 32 on-board registers. 1553 message traffic is stored and retrieved using the dedicated, memory mapped, on-board 64K words of RAM. The registers include the five Configuration Registers, a Start/Reset Register, a Status Register, an Interrupt Mask Register, and an Interrupt Status Register. The Configuration Registers define the operating mode and memory management features. The Start/Reset Register provides various reset and BC/MT start functions. The Interrupt Mask Registers enable desired interrupts for both enhanced and legacy modes.

The design of the Enhanced Mini-ACE provides the capabilities of a highly programmable Bus Controller. This supports message scheduling, asynchronous message insertion, message retry strategies, and bulk data transfers.

Operation of the BU-65553 in RT mode is enhanced by the new global circular buffer feature, the addition of the circular buffer 50% rollover interrupt, and an interrupt status queue.

PCMCIA INTERFACE

The BU-65553 provide a Card Information Structure (CIS) within the attribute memory space of the PCMCIA interface. The CIS contains device configuration information structures called basic compatibility tuples. The format of these configuration tuples is defined within the PCMCIA interface standard. In addition to the CIS, there are the four standard PCMCIA configuration registers (Configuration Option Register, Card Configuration and Status Register, Pin Replacement Register, and Socket and Copy Register).

The CIS, also referred to as the Metaformat, provides a level of device information which allows a card resource manager or an application program to identify and fully configure the card.

ENHANCED MINI-ACE

The Enhanced Mini-ACE comprises a complete, independent interface between the PCMCIA interface and a MIL-STD-1553 bus. The Enhanced Mini-ACE hybrids provide software compatibility with DDC's older generation ACE and Mini-ACE (Plus) terminals.

The BU-61564 Enhanced Mini-ACE provides complete multiprotocol support of MIL-STD-1553A/B/McAir and STANAG 3838. These hybrids include dual transceivers; along with protocol, host interface, memory management logic; and 64K X 17 of RAM. There is built-in parity checking for this RAM.

One of the salient features of the Enhanced Mini-ACE is its enhanced bus controller architecture. The Enhanced BC's highly autonomous message sequence control engine provides a means for offloading the host processor for implementing multiframe message scheduling, message retry and bus switching schemes, data double buffering, and asynchronous message insertion. In addition, the Enhanced BC mode includes 8 general purpose flag bits, a general purpose queue, and user-defined interrupts, for the purpose of performing messaging to the host processor.

Another important feature of the Enhanced Mini-ACE is the incorporation of a fully autonomous built-in self-test. This test provides comprehensive testing of the internal protocol logic. A separate test verifies the operation of the Enhanced Mini-ACE's internal 64K RAM. Since the self-tests are fully autonomous, they eliminate the need for the host to write and read stimulus and response vectors.

The Enhanced Mini-ACE RT offers the choice of single, double, and circular buffering for individual subaddresses along with a global circular buffering option for multiple (or all) receive subaddresses, a 50% rollover interrupt for circular buffers, an interrupt status queue for logging up to 32 interrupt events, and an option to automatically initialize to RT mode with the Busy bit set following power-up.

The transceivers in the Enhanced Mini-ACE series terminals are fully monolithic, requiring only a +5 volt power input. The transmitters are voltage sources, which provide improved line driving capability over current sources. This serves to improve performance on long buses with many taps. The BU-65553's transmitters may be trimmed to meet the MIL-STD-1760 requirement of a minimum of 20 volts peak-to-peak, transformer coupled (consult factory).

If required, the BU-65553 is also available with an option for McAir compatible transmitters (consult factory).

INTERRUPTS

The BU-65553 provides many programmable options for interrupt generation and handling. Individual interrupts are enabled by the Interrupt Mask Register. The host processor may easily determine the cause of the interrupt by using the Interrupt Status Register. The Interrupt Status Register provides the current state of the interrupt conditions. The Interrupt Status Register may be updated in two ways. In the standard interrupt handling mode, a particular bit in the Interrupt Status Register will be updated only if the condition exists and the corresponding bit in the Interrupt Mask Register is enabled. In the enhanced interrupt handling mode, a particular bit in the Interrupt Status Register will be updated if the condition exists regardless of the contents of the corresponding Interrupt Mask Register bit. In any case, the respective Interrupt Mask Register bit enables an interrupt for a particular condition.

The BU-65553 provides maskable interrupts and two 15-bit Interrupt Status Registers for end of message, end of BC message list, erroneous messages, Status Set (BC mode), Time Tag Register Rollover, RT Address Parity Error conditions, BC retry, data stack rollover, command stack rollover, transmitter watchdog timeout, or RAM parity error. The Interrupt Status Register allow the host processor to determine the cause of all interrupts by means of two READ operations.

ADDRESSING

All internal pointers used by these cards are assumed to be word addresses, however byte addressing is used by the PC to access memory and registers in the card. For example, to access the Area A Stack Pointer at word offset address 0100 (hex), a byte address offset of 0200 (hex) must be used (i.e. the byte address is determined by multiplying the word address by two).

The shared RAM space starts at byte location 8000 if the memory allocation were contiguous. The registers are mapped into memory locations 0X0000 through 0X002F. The memory address space between 0X0030 and 0X7FFF is reserved. The 64K x 17 RAM uses page swapping, which is done in the device driver.

COMMON MEMORY ADDRESS MAP

The software interface of the BU-65553 to the host processor consists of 25 internal operational registers for normal operation, an additional 8 test registers, plus 64K x 17 shared memory. Both the registers and the shared memory reside in the PCMCIA common memory space. See TABLE 2.

ENHANCED MINI-ACE REGISTERS

The address mapping for the Enhanced Mini-ACE registers is illustrated in TABLE 2:

TABLE 2. ENHANCED MINI-ACE REGISTERS

ADDRESS							REGISTER
A6	A5	A4	A3	A2	A1	A0	DESCRIPTION / ACCESSIBILITY
0	0	0	0	0	0	0/1	Interrupt Mask Register #1 (RD/WR)
0	0	0	0	0	1	0/1	Configuration Register #1 (RD/WR)
0	0	0	0	1	0	0/1	Configuration Register #2 (RD/WR)
0	0	0	0	1	1	0/1	Start/Reset Register (W/R)
0	0	0	0	1	1	0/1	Non-Enhanced BC or RT Command Stack Pointer/Enhanced BC Instruction List Pointer Register (RD)
0	0	0	1	0	0	0/1	BC Control Word/RT Subaddress Control Word Register (RD/WR)
0	0	0	1	0	1	0/1	Time Tag Register (RD/WR)
0	0	0	1	1	0	0/1	Interrupt Status Register #1 (RD)
0	0	0	1	1	1	0/1	Configuration Register #3 (RD/WR)
0	0	1	0	0	0	0/1	Configuration Register #4 (RD/WR)
0	0	1	0	0	1	0/1	Configuration Register #5 (RD/WR)
0	0	1	0	1	0	0/1	RT/Monitor Data Stack Address Register (RD/WR)
0	0	1	0	1	1	0/1	BC Frame Time Remaining Register (RD)
0	0	1	1	0	0	0/1	BC Time Remaining to Next Message Register (RD)
0	0	1	1	0	1	0/1	BC Frame Time/Enhanced BC Initial Instruction Pointer/RT Last Command/MT Trigger Word Register (RD/WR)
0	0	1	1	1	0	0/1	RT Status Word Register (RD)
0	0	1	1	1	1	0/1	RT BIT Word Register (RD)
0	1	0	0	0	0	0/1	Test Mode Register 0
0	1	0	0	0	1	0/1	Test Mode Register 1
0	1	0	0	1	0	0/1	Test Mode Register 2
0	1	0	0	1	1	0/1	Test Mode Register 3
0	1	0	1	0	0	0/1	Test Mode Register 4
0	1	0	1	0	1	0/1	Test Mode Register 5
0	1	0	1	1	0	0/1	Test Mode Register 6
0	1	0	1	1	1	0/1	Test Mode Register 7
0	1	1	0	0	0	0/1	Configuration Register #6 (RD/WR)
0	1	1	0	0	1	0/1	Configuration Register #7 (RD/WR)
0	1	1	0	1	0	0/1	Reserved
0	1	1	0	1	1	0/1	BC Condition Code Register (RD)
0	1	1	0	1	1	0/1	BC General Purpose Flag Register (WR)
0	1	1	1	0	0	0/1	BIT Test Status Register (RD)
0	1	1	1	0	1	0/1	Interrupt Mask Register #2 (RD/WR)
0	1	1	1	1	0	0/1	Interrupt Status Register #2 (RD)
0	1	1	1	1	1	0/1	BC General Purpose Queue Pointer/RT-MT Interrupt Status Queue Pointer Register (RD/WR)
1	0	0	0	0	0	0/1	Additional Test Mode Registers
• • •							• • •
1	1	1	1	1	1	0/1	Additional Test Mode Registers

BUS CONTROLLER (BC) ARCHITECTURE

The BC functionality for the BU-65553 includes two separate architectures: (1) the older, legacy mode, which provides complete compatibility with the previous ACE and Mini-ACE (Plus) generation products; and (2) the newer, Enhanced BC mode. The Enhanced BC mode offers several new powerful architectural features. These includes the incorporation of a highly autonomous BC message sequence control engine, which greatly serves to offload the operation of the host CPU.

The Enhanced BC's message sequence control engine provides a high degree of flexibility for implementing major and minor frame scheduling; capabilities for inserting asynchronous messages in the middle of a frame; to separate 1553 message data from control/status data for the purpose of implementing double buffering and performing bulk data transfers; for implementing message retry schemes, including the capability for automatic bus channel switchover for failed messages; and for reporting various conditions to the host processor by means of 4 user-defined interrupts and a general purpose queue.

In both the legacy and Enhanced BC modes, the Enhanced Mini-ACE BC implements all MIL-STD-1553B message formats. Message format is programmable on a message-by-message basis by means of the BC Control Word and the T/R bit of the Command Word for the respective message. The BC Control Word allows 1553 message format, 1553A/B type RT, bus channel, self-test, and Status Word masking to be specified on an individual message basis. In addition, automatic retries and/or interrupt requests may be enabled or disabled for individual messages. The BC performs all error checking required by MIL-STD-1553B. This includes validation of response time, sync type and sync encoding, Manchester II encoding, parity, bit count, word count, Status Word RT Address field, and various RT-to-RT transfer errors. The Enhanced Mini-ACE BC response timeout value is programmable with choices of 18, 22, 50, and 130 ms. The longer response timeout values allow for operation over long buses and/or use of the repeaters.

In its legacy mode, the Enhanced Mini-ACE may be programmed to process BC frames of up to 512 messages with no processor intervention. The legacy BC mode provides backward compatibility for the BU-65550/1/2 applications. In the Enhanced BC mode, there is no explicit limit to the number of messages that may be processed in a frame. In both modes, it is possible to program for either single frame or frame auto-repeat operation. In the auto-repeat mode, the frame repetition rate may be controlled either internally, using a programmable BC frame timer, or from an external trigger input.

Enhanced BC Mode: Message sequence control. One of the major new architectural features of the Enhanced Mini-ACE series is its advanced capability for BC message sequence control. The Enhanced Mini-ACE supports highly autonomous BC operation, which greatly offloads the operation of the host processor.

The operation of the Enhanced Mini-ACE's message sequence control engine is illustrated in FIGURE 2. The BC message sequence control involves an instruction list pointer register; an instruction list which contains multiple 2-word entries; a message control/status stack, which contains multiple 8-word or 10-word descriptors; and data blocks for individual messages.

The initial value of the instruction list pointer register is initialized by the host processor (via Register 0D), and is incremented by the BC message sequence processor (host readable via Register 03). During operation, the message sequence control processor fetches the operation referenced by the instruction list pointer register from the instruction list.

Note that the pointer parameter referencing the first word of a message's control/status block (the BC Control Word) must contain an address value that is **modulo 8**. Also, note that if the message is an RT-to-RT transfer, the pointer parameter must contain an address value that is **modulo 16**.

Op Codes. The instruction list pointer register references a pair of words in the BC instruction list: an op code word, followed by a parameter word. The format of the op code word, which is illustrated in FIGURE 3, includes a 5-bit op code field and a 5-bit condition code field. The op code identifies the instruction to be executed by the BC message sequence controller.

Most of the operations are conditional, with execution dependent on the contents of the condition code field. Bits 3-0 of the condition code field identifies the particular condition. Bit 4 of the condition code field identifies the logic sense ("1" or "0") of the selected condition code on which the conditional execution is dependent. TABLE 3 lists all the op codes, along with their respective mnemonic, code value, parameter, and description. TABLE 4 defines all the condition codes.

Eight of the condition codes (8 through F) are set or cleared as the result of the most recent message. The other eight are defined as "General Purpose" condition codes GP0 through GP7. There are three mechanisms for programming the values of the General Purpose Condition Code bits: (1) They may be set, cleared, or toggled by the host processor, by means of the BC GENERAL PURPOSE FLAG REGISTER; (2) they may be set, cleared, or toggled by the BC message sequence control processor, by means of the GP Flag Bits (FLG) instruction; and (3) GP0 and GP1 only (but none of the others) may be set or cleared by means of the BC message sequence control processor's Compare Frame Timer (CFT) or Compare Message Timer (CMT) instructions.

The host processor also has read-only access to the BC condition codes by means of the BC CONDITION CODE REGISTER.

Note that four instructions are **unconditional**. These are Compare to Frame Timer (CFT), Compare to Message Timer (CMT), GP Flag Bits (FLG), and Execute and Flip (XQF). For these instructions, the Condition Code Field is "don't care". That is, these instructions are **always** executed, regardless of the result of the condition code test.

All other instructions are conditional. That is, they will only be executed if the condition code specified by the condition code field in the op code word tests true. If the condition code field tests false, the instruction list pointer will skip down to the next instruction.

As shown in TABLE 3, many of the operations include a single-word parameter. For an XEQ (execute message) operation, the parameter is a pointer to the start of the message's control/status block. For other operations, the parameter may be an address, a time value, an interrupt pattern, a mechanism to set or clear general purpose flag bits, or an immediate value. For several op codes, the parameter is "don't care" (not used).

As described above, some of the op codes will cause the message sequence control processor to execute messages. In this case, the parameter references the first word of a message control/status block. With the exception of RT-to-RT transfer messages, all message status/control blocks are eight words long: a block control word, time-to-next-message parameter, data block pointer, command word, status word, loopback word, block status word, and time tag word.

In the case of an RT-to-RT transfer message, the size of the message control/status block increases to 16 words. However, in this case, the last six words are not used; the ninth and tenth words are for the second command word and second status word.

The third word in the message control/status block is a pointer that references the first word of the message's data word block. Note that the data word block stores only data words, which are to be either transmitted or received by the BC. By segregating data words from command words, status words, and other control and "housekeeping" functions, this architecture enables the use of convenient, usable data structures, such as circular buffers and double buffers.

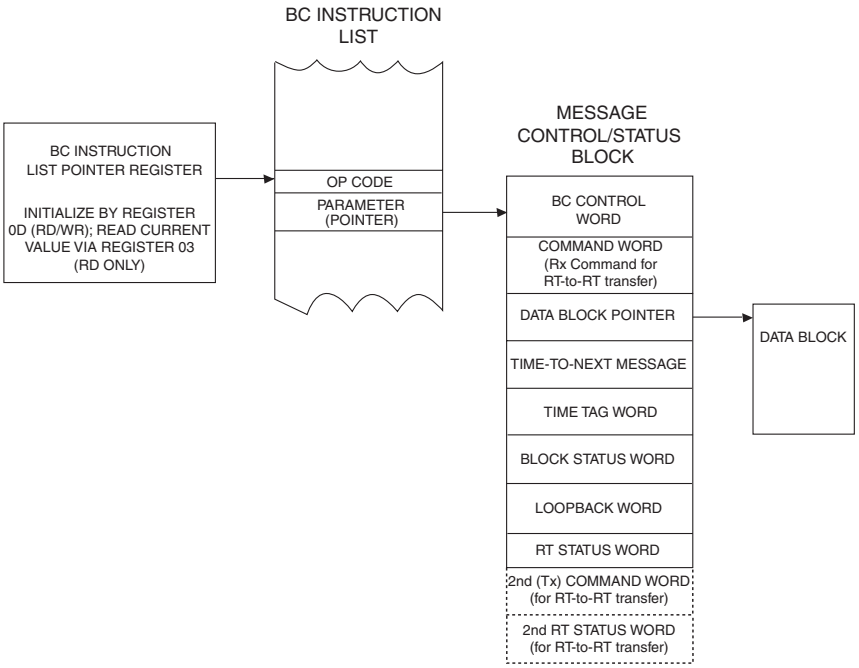


FIGURE 2. BC MESSAGE SEQUENCE CONTROL

Other operations support program flow control; i.e., jump and call capability. The call capability includes maintenance of a call stack which supports a maximum of four (4) entries; there is also a return instruction. In the case of a call stack overrun or underrun, the BC will issue an CALL STACK POINTER REGISTER ERROR interrupt, if enabled.

Other op codes may be used to delay for a specified time; start a new BC frame; wait for an external trigger to start a new frame; do comparisons based on frame time and time-to-next message; load the time tag or frame time registers; halt; and issue host interrupts. In the case of host interrupts, the message control processor passes a 4-bit user-defined interrupt vector to the host, by means of the Enhanced Mini-ACE's Interrupt Status Register.

The purpose of the FLG instruction is to enable the message sequence controller to set, clear, or toggle the value(s) of any or all of the eight general purpose condition flags.

The op code parity bit encompasses all sixteen bits of the op code word. This bit must be programmed for odd parity. If the message sequence control processor fetches an undefined op code word, an op code word with even parity, or bits 9-5 of an op code word do not have a binary pattern of 01010, the message sequence control processor will immediately halt the BC's operation. In addition, if enabled, a BC TRAP OP CODE interrupt will be issued. Also, if enabled, a parity error will result in an OP CODE PARITY ERROR interrupt.

The Enhanced Mini-ACE BC message sequence control capability enables a high degree of offloading of the host processor. This includes using the various timing functions to enable autonomous structuring of major and minor frames. In addition, by implementing conditional jumps and subroutine calls, the message sequence control processor greatly simplifies the insertion of asynchronous, or "out-of-band" messages.

Execute and Flip Operation. The Enhanced Mini-ACE BC's XQF, or "Execute and Flip" operation, provides some unique capabilities. Following execution of this unconditional instruction, if the condition code tests TRUE, the BC will modify the value of the current XQF instruction's pointer parameter by toggling bit 4 in the pointer. That is, if the selected condition flag tests true, the value of the parameter will be updated to the value = old address XOR 0010h. As a result, the **next** time that this line in the instruction list is executed, the Message Control/Status Block at the **updated** address (**old address XOR 0010h**), rather than the one at the old address, will be processed. The operation of the XQF instruction is illustrated in FIGURE 4.

There are multiple ways of utilizing the "execute and flip" functionality. One is to facilitate the implementation of a double buffering data scheme for individual messages. This allows the message sequence control processor to "ping-pong" between a pair of data buffers for a particular message. By so doing, the host processor can access one of the two Data Word blocks, while the BC reads or writes the alternate Data Word block.

A second application of the "execute and flip" capability is in association with message retries. This allows the BC to not only switch buses when retrying a failed message, but to automatically switch buses **permanently** for all future times that the same message is to be processed. This not only provides a high degree of autonomy from the host CPU, but saves BC bandwidth, by eliminating future attempts to process messages on an RT's failed channel.

General Purpose Queue. The Enhanced Mini-ACE BC allows for the creation of a general purpose queue. This data structure provides a means for the message sequence processor to convey information to the BC host. The BC op code repertoire provides mechanisms to push various items on this queue. These include the contents of the Time Tag Register, the Block Status Word for the most recent message, an immediate data value, or the contents of a specified memory address.

FIGURE 5 illustrates the operation of the BC General Purpose Queue. Note that the BC General Purpose Queue Pointer Register will always point to the next address location (modulo 64); that is, the location following the last location written by the BC message sequence control engine.

If enabled, a BC GENERAL PURPOSE QUEUE ROLLOVER interrupt will be issued when the value of the queue pointer address rolls over at a 64-word boundary.

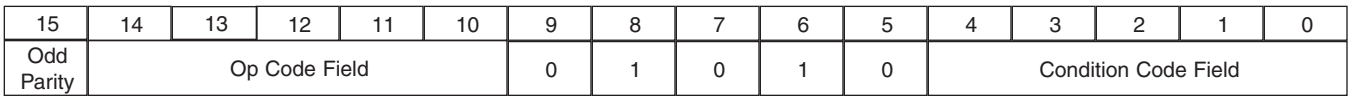


FIGURE 3. BC OP CODE FORMAT

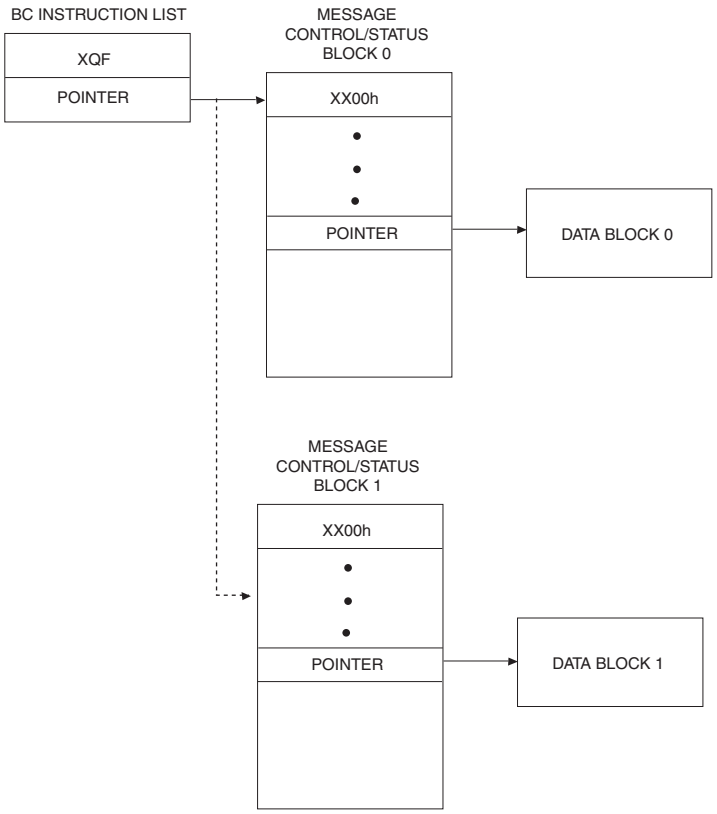


FIGURE 4. EXECUTE AND FLIP (XQP) OPERATION

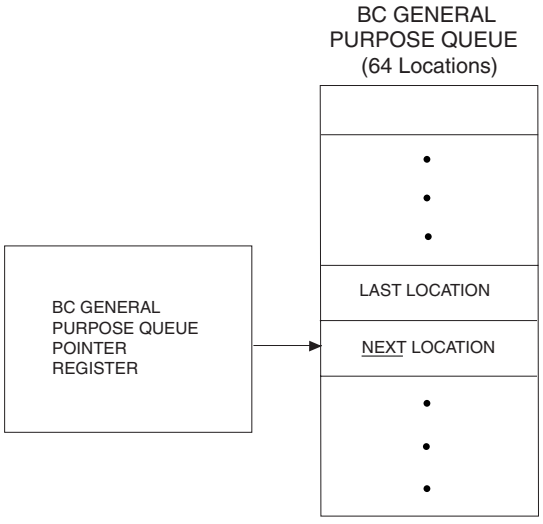


FIGURE 5. BC GENERAL PURPOSE QUEUE

TABLE 3. BC OPERATIONS FOR MESSAGE SEQUENCE CONTROL					
INSTRUCTION	**MNEMONIC**	**OP CODE (HEX)**	**PARAMETER**	**CONDITIONAL OR UNCONDITIONAL**	**DESCRIPTION**
Execute Message	XEQ	0001	Message Control / Status Block Address	Conditional **(See NOTE)**	Execute the message at the specified Message Control/Status Block Address if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Jump	JMP	0002	Instruction List Address	Conditional	Jump to the Op Code specified in the Instruction List if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Subroutine Call	CAL	0003	Instruction List Address	Conditional	Jump to the Op Code specified by the Instruction List Address and push the Address of the Next Op Code on the Call Stack if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list. Note that the maximum depth of the subroutine call stack is four.
Subroutine Return	RTN	0004	Not Used (don't care)	Conditional	Return to the Op Code popped off the Cal Stack if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Interrupt Request	IRQ	0006	Interrupt Bit Pattern in 4 LS bits	Conditional	Generate an interrupt if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list. The passed parameter (IRQ Bit Pattern) specifies which of the ENHANCED BC IRQ bit(s) (bits 5-2) will be set in Interrupt Status Register #2. Only the four LSBs of the passed parameter are used. A parameter where the four LSBs are logic "0" will not generate an interrupt.
Halt	HLT	0007	Not Used (don't care)	Conditional	Stop execution of the Message Sequence Control Program until a new BC Start is issued by the host if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Delay	DLY	0008	Delay Time Value (resolution = 1 ms/LSB)	Conditional	Delay the time specified by the Time parameter before executing the next Op Code if the condition flag tests TRUE, otherwise continue execution at the next Op Code without delay. The delay generated will use the Time to Next Message Timer.
Wait Until Frame Timer = 0	WFT	0009	Not Used (don't care)	Conditional	Wait until Frame Time counter is equal to Zero before continuing execution of Message Sequence Control Program if the condition flag tests TRUE, otherwise continue execution at the next Op Code without delay.
Compare to Frame Timer	CFT	000A	Delay Time Value (resolution = 100 µs/LSB)	Unconditional	Compare Time Value to Frame Time Counter. The LT/GP0 and EQ/GP1 flag bits are set or cleared based on the results of the compare. If the value of the CFT's parameter is less than the value of the frame time counter, then the LT/GP0 and NE/GP1 flags will be set, while the GT-EQ/GP0 and EQ/GP1 flags will be cleared. If the value of the CFT's parameter is equal to the value of the frame time counter, then the GT-EQ/GP0 and EQ/GP1 flags will be set, while the LT/GP0 and NE/GP1 flags will be cleared. If the value of the CFT's parameter is greater than the current value of the frame time counter, then the GT-EQ/GP0 and NE/GP1 flags will be set, while the LT/GP0 and EQ/GP1 flags will be cleared.
Compare to Message Timer	CMT	000B	Delay Time Value (resolution = 1 µs/LSB)	Unconditional	Compare Time Value to Message Time Counter. The LT/GP0 and EQ/GP1 flag bits are set or cleared based on the results of the compare. If the value of the CMT's parameter is less than the value of the message time counter, then the LT/GP0 and NE/GP1 flags will be set, while the GT-EQ/GP0 and EQ/GP1 flags will be cleared. If the value of the CMT's parameter is equal to the value of the message time counter, then the GT-EQ/GP0 and EQ/GP1 flags will be set, while the LT/GP0 and NE/GP1 flags will be cleared. If the value of the CMT's parameter is greater than the current value of the message time counter, then the GT-EQ/GP0 and NE/GP1 flags will be set, while the LT/GP0 and EQ/GP1 flags will be cleared.
GP Flag Bits	FLG	000C	Used to set, clear, or Toggle GP (General Purpose) flag bits (see description)	Unconditional	Used to set, toggle, or clear any or all of the eight general purpose flags. The table below illustrates the use of the GP Flag Bits instruction for the case of GP0 (General Purpose Flag 0). Bits 1 and 9 of the parameter byte affect flag GP1, bits 2 and 10 affect GP2, etc., according to the following rules: **Bit 8 / Bit 0 / Effect on GPO** 0 / 0 / No Change 0 / 1 / Set Flag 1 / 0 / Clear Flag 1 / 1 / Toggle Flag

TABLE 3. BC OPERATIONS FOR MESSAGE SEQUENCE CONTROL (CONT)					
INSTRUCTION	**MNEMONIC**	**OP CODE (HEX)**	**PARAMETER**	**CONDITIONAL OR UNCONDITIONAL**	**DESCRIPTION**
Load Time Tag Counter	LTT	000D	Time Value. Resolution (ms/LSB) is defined by bits 9, 8, and 7 of Configuration Register #2	Conditional	Load Time Tag Counter with Time Value if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Load Frame Timer	LFT	000E	Time Value (resolution = 100 µs/LSB)	Conditional	Load Frame Timer Register with the Time Value parameter if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Start Frame Timer	SFT	000F	Not Used (don't care)	Conditional	Start Frame Time Counter with Time Value in Time Frame register if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Push Time Tag Register	PTT	0010	Not Used (don't care)	Conditional	Push the value of the Time Tag Register on the General Purpose Queue if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Push Block Status Word	PBS	0011	Not Used (don't care)	Conditional	Push the Block Status Word for the most recent message on the General Purpose Queue if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Push Immediate Value	PSI	0012	Immediate Value	Conditional	Push Immediate data on the General Purpose Queue if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Push Indirect	PSM	0013	Memory Address	Conditional	Push the data stored at the specified memory location on the General Purpose Queue if the condition flag tests TRUE, otherwise continue execution at the next Op Code in the instruction list.
Wait for External Trigger	WTG	0014	Not Used (don't care)	Conditional	Wait until a logic "0"-to-logic "1" transition on the EXT_TRIG input signal before proceeding to the next Op Code in the instruction list if the condition flag tests TRUE, otherwise continue execution at the next Op Code without delay.
Execute and Flip	XQF	0015	Message Control/Status Block Address	Unconditional	Execute (unconditionally) the message for the Message Control/Status Block Address. Following the processing of this message, if the condition flag tests TRUE, then flip bit 4 in the Message Control/Status Block Address, and store the new Message Block Address as the updated value of the parameter following the XQF instruction code. As a result, the next time that this line in the instruction list is executed, the Message Control/Status Block at the updated address (old address XOR 0010h), rather than the old address, will be processed.

NOTE: While the XEQ (Execute Message) instruction is conditional, not all condition codes may be used to enable its use. The ALWAYS and NEVER condition codes may be used. The eight general purpose flag bits, GP0 through GP7, may also be used. However, if GP0 through GP7 are used, it is imperative that the host processor not modify the value of the specific general purpose flag bit that enabled a particular message while that message is being processed. Similarly, the LT, GT-EQ, EQ, and NE flags, which the BC only updates by means of the CFT and CMT instructions, may also be used. However, these two flags are dual use. Therefore, if these are used, it is imperative that the host processor not modify the value of the specific flag (GP0 or GP1) that enabled a particular message while that message is being processed. The NORESP, FMT ERR, GD BLK XFER, MASKED STATUS SET, BAD MESSAGE, RETRY0, and RETRY1 condition codes are not available for use with the XEQ instruction and should not be used to enable its execution.

TABLE 4. BC CONDITION CODES

BIT CODE	NAME (BIT 4=0)	INVERSE (BIT 4=1)	FUNCTIONAL DESCRIPTION
0000	LT/GP0	GT-EQ/ GP0	Less than or GP0 flag. This bit is set or cleared based on the results of the compare. If the value of the CMT's parameter is less than the value of the message time counter, then the LT/GP0 and NE/GP1 flags will be set, while the GT-EQ/GP0 and EQ/GP1 flags will be cleared. If the value of the CMT's parameter is equal to the value of the message time counter, then the GT-EQ/GP0 and EQ/GP1 flags will be set, while the LT/GP0 and NE/GP1 flags will be cleared. If the value of the CMT's parameter is greater than the current value of the message time counter, then the GT-EQ/GP0 and NE/GP1 flags will be set , while the LT/GP0 and EQ/GP1 flags will be cleared. Also, General Purpose Flag 1 may be also be set or cleared by a FLG operation.
0001	EQ/GP1	NE/GP1	Equal Flag. This bit is set or cleared after CFT or CMT operation. If the value of the CMT's parameter is equal to the value of the message time counter, then the EQ/GP1 flag will be set and the NE/GP1 bit will be cleared. If the value of the CMT's parameter is not equal to the value of the message time counter, then the NE/GP1 flag will be set and the EQ/GP1 bit will be cleared. Also, General Purpose Flag 1 may be also be set or cleared by a FLG operation.
0002	GP2	GP2	General Purpose Flags set or cleared by FLG operation or by host processor. The host processor can set, clear, or toggle these flags in the same way as the FLG instruction by means of the BC GENERAL PURPOSE FLAG REGISTER.
0003	GP3	GP3	
0004	GP4	GP4	
0005	GP5	GP5	
0006	GP6	GP6	
0007	GP7	GP7	
0008	NORESP	RESP	NORESP indicates that an RT has either not responded or has responded later than the BC No Response Timeout time. The Enhanced Mini-ACE's No Response Timeout Time is defined per MIL-STD-1553B as the time from the mid-bit crossing of the parity bit to the mid-sync crossing of the RT Status Word. The value of the No Response Timeout value is programmable from among the nominal values 18.5, 22.5, 50.5, and 130 ms (±1 ms) by means of bits 10 and 9 of Configuration Register #5.
0009	FMT ERR	FMT ERR	FMT ERR indicates that the received portion of the most recent message contained one or more violations of the 1553 message validation criteria (sync, encoding, parity, bit count, word count, etc.), or the RT's status word received from a responding RT contained an incorrect RT address field.
000A	GD BLK XFER	BAD BLK XFER	For the most recent message, GD BLK XFER will be set to logic "1" following completion of a valid (error-free) RT-to-BC transfer, RT-to-RT transfer, or transmit mode code with data message. This bit is set to logic "0" following an invalid message. GOOD DATA BLOCK TRANSFER is always logic "0" following a BC-to-RT transfer, a mode code with data, or a mode code without data. The Loop Test has no effect on GOOD DATA BLOCK TRANSFER. GOOD DATA BLOCK TRANSFER may be used to determine if the transmitting portion of an RT-to-RT transfer was error free.
000B	MASKED STATUS SET	MASKED STATUS CLR	Indicates that one or both of the following conditions have occurred for the most recent message: (1) If one (or more) of the Status Mask bits (14 through 9) in the BC Control Word is logic "0" and the corresponding bit(s) is (are) set (logic "1") in the received RT Status Word. In the case of the RESERVED BITS MASK (bit 9) set to logic "0," any or all of the 3 Reserved Status bits being set will result in a MASKED STATUS SET condition; and/or (2) If BROADCAST MASK ENABLED/XOR (bit 11 of Configuration Register #4) is logic "1" and the MASK BROADCAST bit of the message's BC Control Word is logic "0" and the BROADCAST COMMAND RECEIVED bit in the received RT Status Word is logic "1."
000C	BAD MESSAGE	GOOD MESSAGE	Indicates either a format error, loop test fail, or no response error for the most recent message. Note that a "Status Set" condition has no effect on the "BAD MESSAGE/GOOD MESSAGE" condition code.
000D	RETRY0	RETRY0	These two bits reflect the retry status of the most recent message. The number of times that the message was retried is delineated by these two bits as shown below:
000E	RETRY1	RETRY1	**Retry Count 1 (bit 14) / Retry Count 0 (bit 13) / Number of Message Retries** 0 / 0 / 0 0 / 1 / 1 1 / 0 / N/A 1 / 1 / 2
000F	ALWAYS	NEVER	The ALWAYS flag should be set (bit 4 = 0) to designate an instruction as unconditional. The NEVER bit (bit 4 = 1) can be used to implement a NOP or “skip” instruction.

REMOTE TERMINAL (RT) ARCHITECTURE

The BU-65553 architecture provides multiprotocol support, with full compliance to all of the commonly used data bus standards, including MIL-STD-1553A, MIL-STD-1553B, Notice 2, STANAG 3838, General Dynamics 16PP303, and McAir A3818, A5232, and A5690. For the Enhanced Mini-ACE RT mode, there is programmable flexibility enabling the RT to be configured to fulfill any set of system requirements. This includes the capability to meet the MIL-STD-1553A response time requirement of 2 to 5 µs, and multiple options for mode code subaddresses, mode codes, RT status word, and RT BIT word.

The Enhanced Mini-ACE RT protocol design implements all of the MIL-STD-1553B message formats and dual redundant mode codes. The design has passed validation testing for MIL-STD-1553B compliance. The Enhanced Mini-ACE RT performs comprehensive error checking, word and format validation, and checks for various RT-to-RT transfer errors. One of the main features of the Enhanced Mini-ACE RT is its choice of memory management options. These include single buffering by subaddress, double buffering for individual receive subaddresses, circular buffering by individual subaddresses, and global circular buffering for multiple (or all) subaddresses.

Other features of the Enhanced Mini-ACE RT include a set of interrupt conditions, an interrupt status queue with filtering based on valid and/or invalid messages, internal command illegalization, programmable busy by subaddress, multiple options on time tagging, and an "auto-boot" feature which allows the RT to initialize as an online RT with the busy bit set following power turn-on.

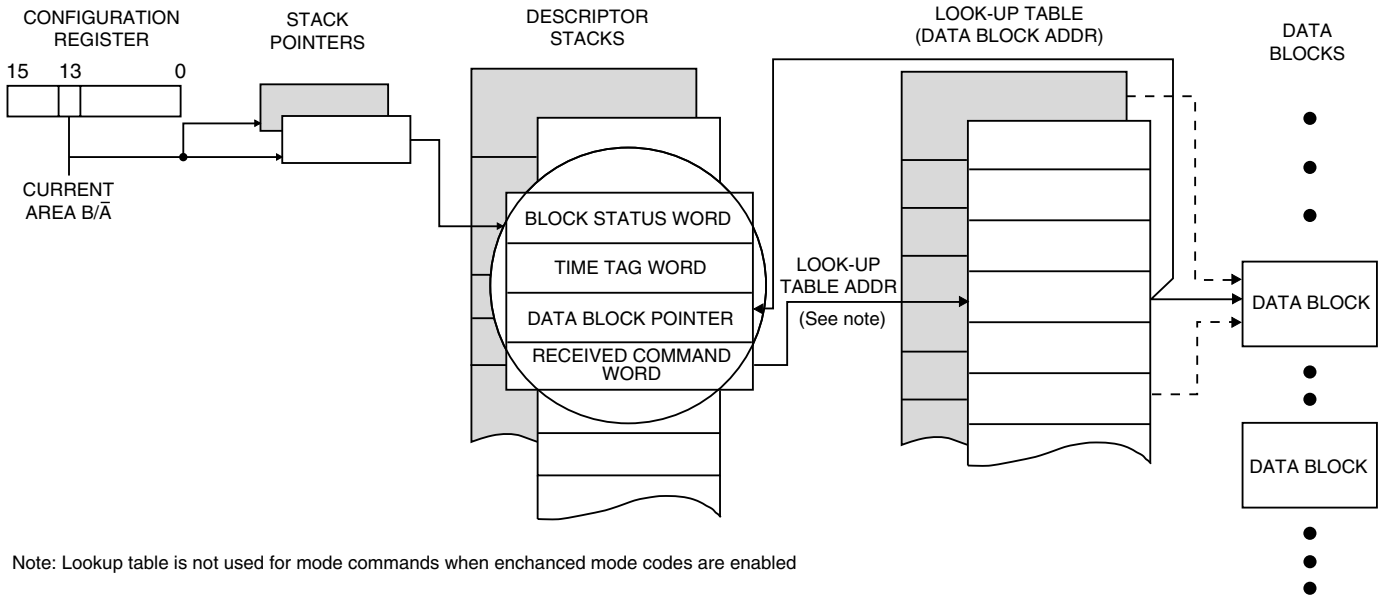
RT MEMORY MANAGEMENT

The Enhanced Mini-ACE provides a variety of RT memory management capabilities. As with the ACE and Mini-ACE, the choice of memory management scheme is fully programmable on a transmit/receive/broadcast subaddress basis.

In compliance with MIL-STD-1553B Notice 2, received data from broadcast messages may be optionally separated from non-broadcast received data. For each transmit, receive or broadcast subaddress, either a single-message data block, a double buffered configuration (two alternating Data Word blocks), or a variable-sized (128 to 8192 words) subaddress circular buffer may be allocated for data storage. The memory management scheme for individual subaddresses is designated by means of the subaddress control word.

For received data, there is also a global circular buffer mode. In this configuration, the data words received from multiple (or all) subaddresses are stored in a common circular buffer structure. Like the subaddress circular buffer, the size of the global circular buffer is programmable, with a range of 128 to 8192 data words.

The double buffering feature provides a means for the host processor to easily access the most recent, complete received block



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

FIGURE 6. RT SINGLE BUFFERED MODE

of valid Data Words for any given subaddress. In addition to helping ensure data sample consistency, the circular buffer options provide a means of greatly reducing host processor overhead for multi-message bulk data transfer applications.

End-of-message interrupts may be enabled either globally (following all messages), following error messages, on a transmit/receive/broadcast subaddress or mode code basis, or when a circular buffer reaches its midpoint (50% boundary) or lower (100%) boundary. A pair of interrupt status registers allow the host processor to determine the cause of all interrupts by means of a single read operation.

SINGLE BUFFERED MODE

The operation of the single buffered RT mode is illustrated in FIGURE 6. In the single buffered mode, the respective lookup table entry must be written by the host processor. Received data words are written to, or transmitted data words are read from the data word block with starting address referenced by the lookup table pointer. In the single buffered mode, the current lookup table pointer is not updated by the Enhanced Mini-ACE memory management logic. Therefore, if a subsequent message is received for the same subaddress, the same Data Word block will be overwritten or overread.

SUBADDRESS DOUBLE BUFFERING MODE

The Enhanced Mini-ACE provides a double buffering mechanism for received data, that may be selected on an individual subaddress basis for any and all receive (and/or broadcast) subaddresses. This is illustrated in FIGURE 7. It should be noted that the Subaddress Double Buffering mode is applicable for receive data only, not for transmit data. Double buffering of transmit messages may be easily implemented by software techniques.

The purpose of the subaddress double buffering mode is to provide data sample consistency to the host processor. This is accomplished by allocating two 32-word data word blocks for each individual receive (and/or broadcast receive) subaddress. At any given time, one of the blocks will be designated as the "active" 1553 block while the other will be considered as "inactive". The data words for the next receive command to that subaddress will be stored in the active block. Following receipt of a valid message, the Enhanced Mini-ACE will automatically switch the active and inactive blocks for that subaddress. As a result, the latest, valid, complete data block is always accessible to the host processor.

CIRCULAR BUFFER MODE

The operation of the Enhanced Mini-ACE's circular buffer RT memory management mode is illustrated in FIGURE 8. As in the single buffered and double buffered modes, 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 descriptor 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.

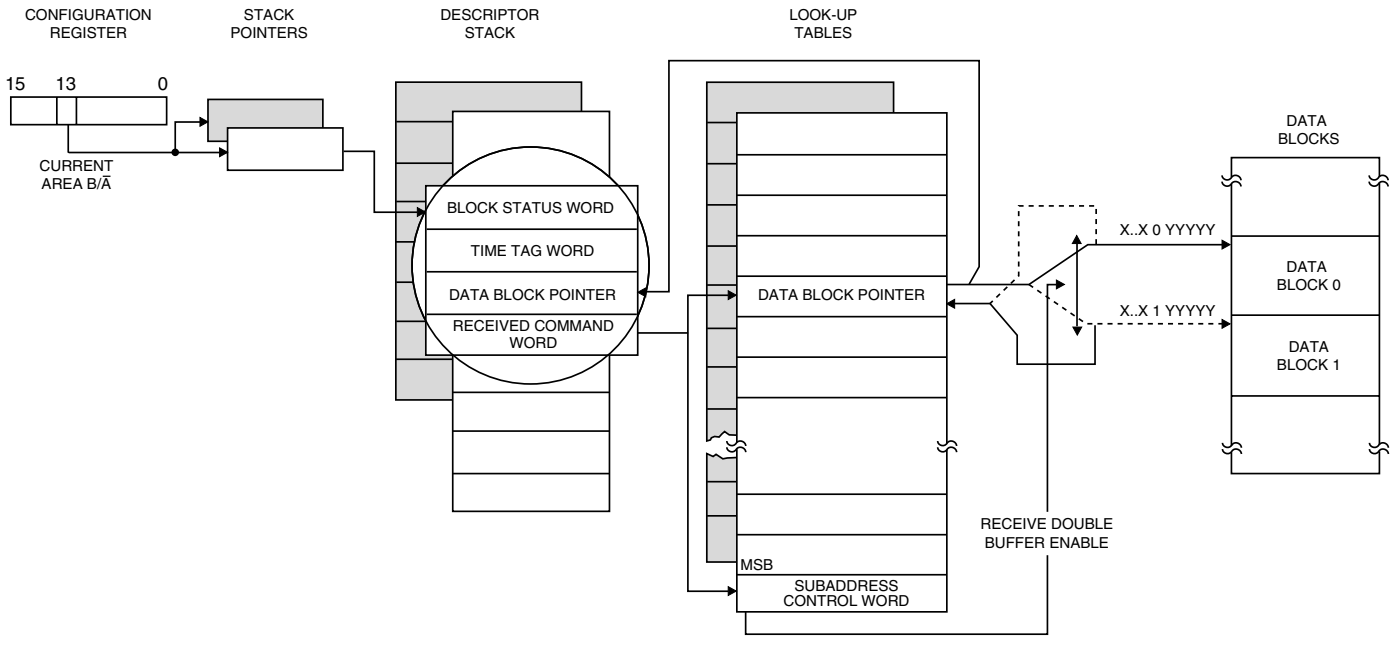


FIGURE 7. RT DOUBLE BUFFERED MODE

In general, the location after the last data word written or read (modulo the circular buffer size) during the message is written to the respective lookup table location during the end-of-message sequence. By so doing, data for the next message for the respective transmit, receive(/broadcast), or broadcast subaddress will be accessed from the next lower contiguous block of locations in the circular buffer.

For the case of a receive (or broadcast receive) message with a data word error, there is an option such that the lookup table pointer will only be updated following receipt of a valid message. That is, the pointer will not be updated following receipt of a message with an error in a data word. This allows failed messages in a bulk data transfer to be retried without disrupting the circular buffer data structure, and without intervention by the RT's host processor.

GLOBAL CIRCULAR BUFFER

Beyond the programmable choice of single buffer mode, double buffer mode, or circular buffer mode, programmable on an individual subaddress basis, the Enhanced Mini-ACE RT architecture provides an additional option, a variable sized global circular buffer. The Enhanced Mini-ACE RT allows for a mix of single buffered, double buffered, and individually circular buffered subaddresses, along with the use of the global double buffer for any arbitrary group of receive(/broadcast) or broadcast subaddresses.

In the global circular buffer mode, the data for multiple receive subaddresses is stored in the same circular buffer data structure. The size of the global circular buffer may be programmed for 128, 256, 512, 1024, 2048, 4096, or 8192 words, by means of Configuration Register #6. Individual subaddresses may be mapped to the global circular buffer by means of their respective subaddress control words.

The pointer to the Global Circular Buffer will be stored in location 0101 (for Area A), or location 0105 (for Area B).

The global circular buffer option provides a highly efficient method for storing received message data. It allows for frequently used subaddresses to be mapped to individual data blocks, while also providing a method for asynchronously received messages to infrequently used subaddresses to be logged to a common area. Alternatively, the global circular buffer provides an efficient means for storing the received data words for all subaddresses. Under this method, all received data words are stored chronologically, regardless of subaddress.

RT DESCRIPTOR STACK

The descriptor stack provides a chronology of all messages processed by the Enhanced Mini-ACE RT. Reference FIGURES 6, 7, and 8. Similar to BC mode, there is a four-word block descriptor in the Stack for each message processed. The four entries to each block descriptor are the Block Status Word, Time Tag Word, the pointer to the start of the message's data block, and the 16-bit received Command Word.

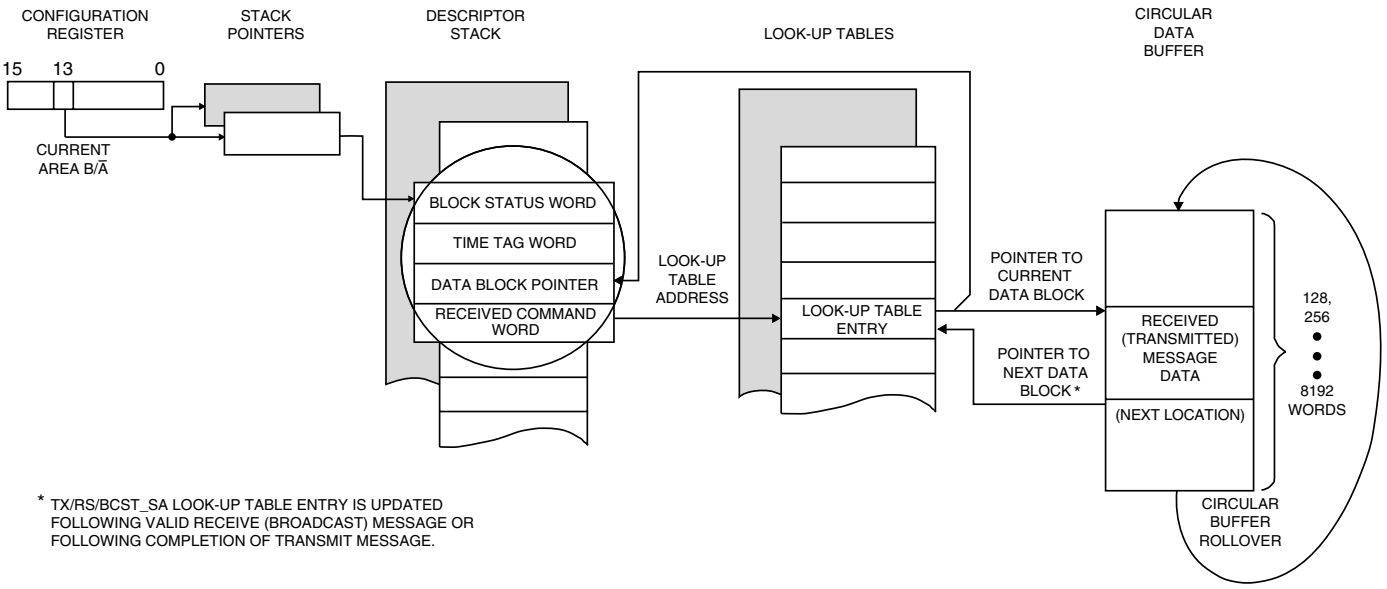


FIGURE 8. RT CIRCULAR BUFFERED MODE

The RT Block Status Word includes indications of whether a particular message is ongoing or has been completed, what bus channel it was received on, indications of illegal commands, and flags denoting various message error conditions. For the double buffering, subaddress circular buffering, and global circular buffering modes, the data block pointer may be used for locating the data blocks for specific messages. Note that for mode code commands, there is an option to store the transmitted or received data word as the third word of the descriptor, in place of the data block pointer.

The Time Tag Word provides a 16-bit indication of relative time for individual messages. The resolution of the Enhanced Mini-ACE's time tag is programmable from among 2, 4, 8, 16, 32, or 64 µs/LSB. There is also a provision for using an external clock input for the time tag (consult factory). If enabled, there is a time tag rollover interrupt, which is issued when the value of the time tag rolls over from FFFF(hex) to 0. Other time tag options include the capabilities to clear the time tag register following receipt of a Synchronize (without data) mode command and/or to set the time tag following receipt of a Synchronize (with data) mode command. For that latter, there is an added option to filter the "set" capability based on the LSB of the received data word being equal to logic "0".

RT INTERRUPTS

The BU-65553 offers a great deal of flexibility in terms of RT interrupt processing. By means of the Enhanced Mini-ACE's two Interrupt Mask Registers, the RT may be programmed to issue interrupt requests for the following events/conditions: End-of-(every)Message, Message Error, Selected (transmit or receive) Subaddress, 100% Circular Buffer Rollover, 50% Circular Buffer Rollover, 100% Descriptor Stack Rollover, 50% Descriptor Stack Rollover, Selected Mode Code, Transmitter Timeout, Illegal Command, and Interrupt Status Queue Rollover.

Interrupt for 50% Rollovers of Stacks, Circular Buffers. The Enhanced Mini-ACE RT and Monitor are capable of issuing host interrupts when a subaddress circular buffer pointer or stack pointer crosses its mid-point boundary. For RT circular buffers, this is applicable for both transmit and receive subaddresses. Reference FIGURE 9. There are four interrupt mask and interrupt status register bits associated with the 50% rollover function: (1) RT circular buffer; (2) RT command (descriptor) stack; (3) Monitor command (descriptor) stack; and (4) Monitor data stack.

The 50% rollover interrupt is beneficial for performing bulk data transfers. For example, when using circular buffering for a particular receive subaddress, the 50% rollover interrupt will inform the host processor when the circular buffer is half full. At that time, the host may proceed to read the received data words in the upper half of the buffer, while the Enhanced Mini-ACE RT writes received data words to the lower half of the circular buffer. Later, when the RT issues a 100% circular buffer rollover interrupt, the host can proceed to read the received data from the

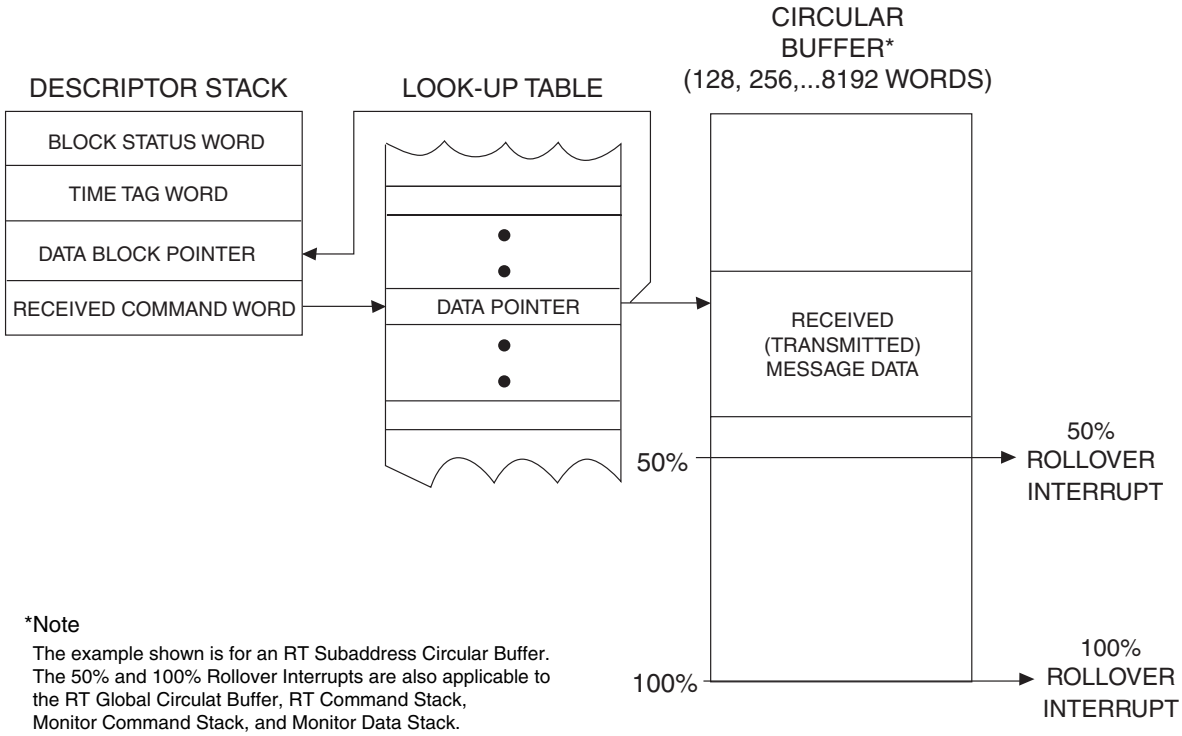


FIGURE 9. 50% AND 100% ROLLOVER INTERRUPTS

lower half of the buffer, while the Enhanced Mini-ACE RT continues to write received data words to the upper half of the buffer.

Interrupt status queue. The Enhanced Mini-ACE RT, Monitor, and combined RT/Monitor modes include the capability for generating an interrupt status queue. As illustrated in FIGURE 10, this provides a chronological history of interrupt generating events and conditions. In addition to the Interrupt Mask Register, the Interrupt Status Queue provides additional filtering capability, such that only valid messages and/or only invalid messages may result in the creation of an entry to the Interrupt Status Queue.

The pointer to the Interrupt Status Queue is stored in the INTERRUPT VECTOR QUEUE POINTER REGISTER (register address 1F). This register must be initialized by the host, and is subsequently incremented by the RT message processor. The interrupt status queue is 64 words deep, providing the capability to store entries for up to 32 messages.

The queue rolls over at addresses of modulo 64. The events that result in queue entries include both message-related and non-message related events. Note that the Interrupt Vector Queue Pointer Register will always point to the next location (modulo 64) following the last vector/pointer pair written by the Enhanced Mini-ACE RT, Monitor, or RT/Monitor.

Each event that causes an interrupt results in a two-word entry to be written to the queue. The first word of the entry is the interrupt vector. The vector indicates which interrupt event(s)/condition(s) caused the interrupt.

RT COMMAND ILLEGALIZATION

The Enhanced Mini-ACE provides an internal mechanism for RT Command Word illegalizing. By means of a 256-word area in shared RAM, the host processor may designate that any message be illegalized, based on the command word T/R bit, subaddress, and word count/mode code fields. The Enhanced Mini-ACE illegalization scheme provides the maximum in flexibility, allowing any subset of the 4096 possible combinations of broadcast/own address, T/R bit, subaddress, and word count/mode code to be illegalized.

RT ADDRESS

Fully software programmable by the host, by means of an internal register.

RT BUILT-IN TEST (BIT) WORD

The bit map for the Enhanced Mini-ACE's internal RT Built-in-Test (BIT) Word is indicated in TABLE 5.

OTHER RT FEATURES

The Enhanced Mini-ACE includes options for the Terminal flag status word bit to be set either under software control and/or

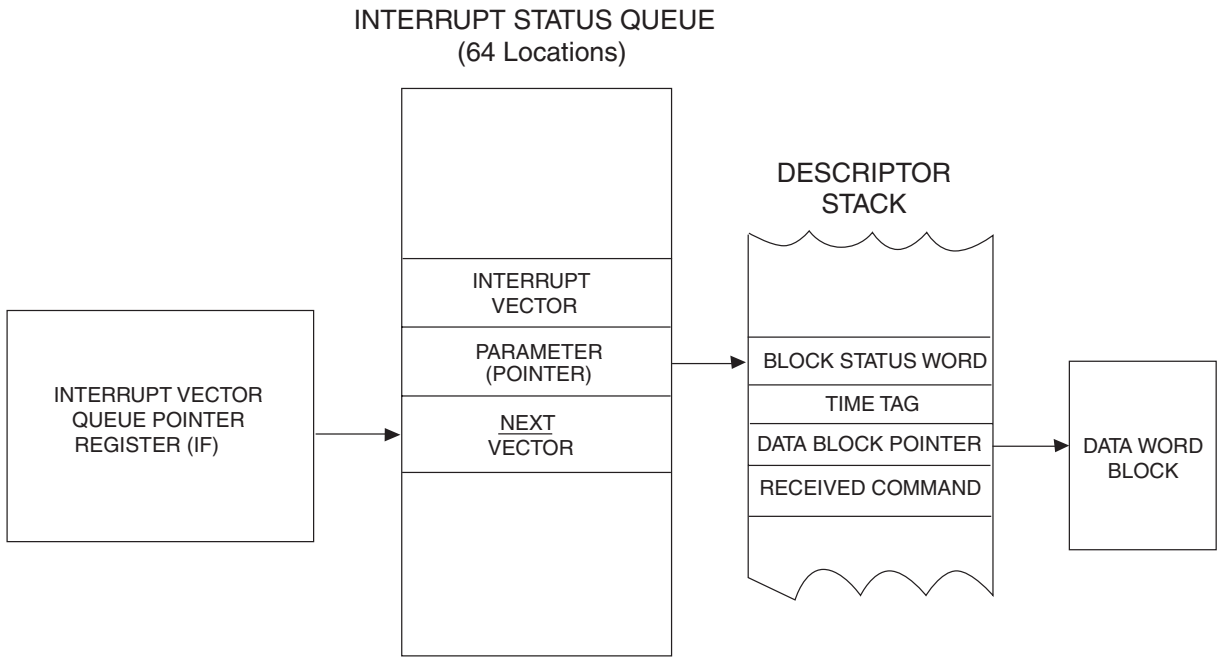


FIGURE 10. RT (AND MONITOR) INTERRUPT STATUS QUEUE (SHOWN FOR MESSAGE INTERRUPT EVENT)

automatically following a failure of the loopback self-test. Other software programmable RT options include software programmable RT status and RT BIT words, automatic clearing of the Service Request bit following receipt of a Transmit vector word mode command, options regarding Data Word transfers for the Busy and Message error (illegal) Status word bits, and options for the handling of 1553A and reserved mode codes.

TABLE 5. RT BIT WORD

ADDRESS BIT	DESCRIPTION
15 (MSB)	Transmitter Timeout
14	Loop Test Failure B
13	Loop Test Failure A
12	Handshake Failure
11	Transmitter Shutdown B
10	Transmitter Shutdown A
9	Terminal Flag Inhibited
8	BIT Test Fail
7	High Word Count
6	Low Word Count
5	Incorrect Sync Received
4	Parity/Manchester Error Received
3	RT-RT Gap/Sync/Address Error
2	RT-RT No Response Error
1	RT-RT 2nd Command Word Error
0 (LSB)	Command Word Contents Error

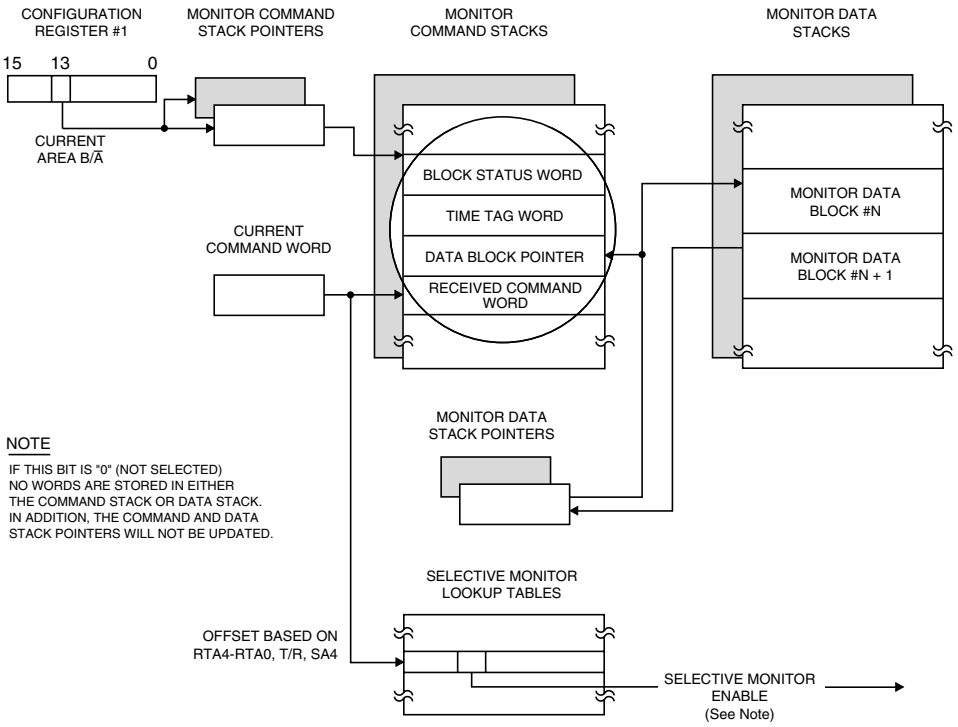


FIGURE 11. SELECTIVE MESSAGE MONITOR MEMORY MANAGEMENT

MONITOR ARCHITECTURE

The BU-65553 includes three monitor modes:

(1) A Word Monitor mode.
(2) A selective message monitor mode.
(3) A combined RT/message monitor mode.

For new applications, it is recommended that the selective message monitor mode be used, rather than the word monitor mode. Besides providing monitor filtering based on RT address, T/R bit, and subaddress, the message monitor eliminates the need to determine the start and end of messages by software.

WORD MONITOR MODE

In the Word Monitor Terminal mode, the BU-65553 monitors both 1553 buses. After the software initialization and Monitor Start sequences, the Enhanced Mini-ACE stores all Command, Status, and Data Words received from both buses. For each word received from either bus, a pair of words is stored to the Enhanced Mini-ACE's shared RAM. The first word is the word received from the 1553 bus. The second word is the Monitor Identification (ID), or "Tag" word. The ID word contains information relating to bus channel, word validity, and inter-word time gaps. The data and ID words are stored in a circular buffer in the shared RAM address space.

SELECTIVE MESSAGE MONITOR MODE

The BU-65553 Selective Message Monitor provides monitoring of 1553 messages with filtering based on RT address, T/R bit, and subaddress with no host processor intervention. By autonomously distinguishing between 1553 command and status words, the Message Monitor determines when messages begin and end, and stores the messages into RAM, based on a programmable filter (RT address, T/R bit, and subaddress).

The selective monitor may be configured as just a monitor, or as a combined RT/Monitor. In the combined RT/Monitor mode, the Enhanced Mini-ACE functions as an RT for one RT address (including broadcast messages), and as a selective message monitor for the other 30 RT addresses. The Enhanced Mini-ACE Message Monitor contains two stacks, a command stack and a data stack, that are independent from the BC/RT command stack. The pointers for these stacks are located at fixed locations in the RAM.

MONITOR SELECTION FUNCTION

Following receipt of a valid command word in Selective Monitor mode, the Enhanced Mini-ACE will reference the selective monitor lookup table to determine if this particular command is enabled. The address for this location is determined by means of an offset based on the RT Address, T/R bit, and Subaddress bit 4 of the current command word, and concatenating it to the monitor lookup table base address of 0500-0501 (hex). The bit location within this word is determined by subaddress bits 3-0 of the current command word.

If the specified bit in the lookup table is logic "0", the command is not enabled, and the Enhanced Mini-ACE will ignore this command. If this bit is logic "1", the command is enabled and the Enhanced Mini-ACE will create an entry in the monitor command descriptor stack (based on the monitor command stack pointer), and store the data and status words associated with the command into sequential locations in the monitor data stack. In addition, for an RT-to-RT transfer in which the receive command is selected, the second command word (the transmit command) is stored in the monitor data stack.

SELECTIVE MESSAGE MONITOR MEMORY MANAGEMENT

FIGURE 11 illustrates the Selective Message Monitor operation. Upon receipt of a valid Command Word, the Enhanced Mini-ACE will reference the Selective Monitor Lookup Table to determine if the current command is enabled. If the current command is disabled, the Enhanced Mini-ACE monitor will ignore (and not store) the current message. If the command is enabled, the monitor will create an entry in the Monitor Command Stack at the address location referenced by the Monitor Command Stack Pointer, and an entry in the monitor data stack starting at the location referenced by the monitor data stack pointer.

The format of the information in the data stack depends on the format of the message that was processed. For example, for a BC-to-RT transfer (receive command), the monitor will store the command word in the monitor command descriptor stack, with the data words and the receiving RT's status word stored in the monitor data stack.

The size of the monitor command stack is programmable, with choices of 256, 1K, 4K, or 16K words. The monitor data stack size is programmable with choices of 512, 1K, 2K, 4K, 8K, 16K, 32K or 64K words.

MONITOR INTERRUPTS

Selective monitor interrupts may be issued for End-of-message and for conditions relating to the monitor command stack pointer and monitor data stack pointer. The latter, which are shown in FIGURE 9, include Command Stack 50% Rollover, Command Stack 100% Rollover, Data Stack 50% Rollover, and Data Stack 100% Rollover.

The 50% rollover interrupts may be used to inform the host processor when the command stack or data stack is half full. At that time, the host may proceed to read the received messages in the upper half of the respective stack, while the Enhanced Mini-ACE monitor writes messages to the lower half of the stack. Later, when the monitor issues a 100% stack rollover interrupt, the host can proceed to read the received data from the lower half of the stack, while the Enhanced Mini-ACE monitor continues to write received data words to the upper half of the stack.

BUILT-IN SELF-TEST

The BU-65553 includes extensive, highly autonomous self-test capability. This includes both protocol and RAM self-tests. The Enhanced Mini-ACE protocol test is performed automatically following power turn-on. In addition, either or both of these self-tests may be initiated by command(s) from the BU-65553's PCMCIA host.

The protocol test consists of a toggle test of 95% of the terminal's logic gates. The test includes a comprehensive test of all registers, Manchester encoder and decoders, transmitter failsafe timer, and protocol logic. This test is performed in approximately 2.0 ms.

There is a separate built-in test for the Enhanced Mini-ACE's 64K X 17 RAM. This test consists of writing and then reading/verifying the two walking patterns "data = address" and "data = address inverted". This test takes about 40 ms to complete.

The Enhanced Mini-ACE built-in test may be initiated by a command from the PCMCIA host, via the START/RESET REGISTER. For RT mode, this may include the host invoking self-test following receipt of an Initiate self-test mode command. The results of the self-test are host accessible by means of the BIT status register. For the BU-65553's RT mode, the result of the self-test may be communicated to the bus controller by means of the Terminal flag status word bit and bit 8 of the RT BIT word ("0" = pass, "1" = fail).

If there is a failure of the protocol self-test, it is possible to access information about the first failed vector. This may be done by means of the Enhanced Mini- ACE's upper registers (register addresses 32 through 63). Through these registers, it is possible to determine the self-test ROM address of the first failed vector, the expected response data pattern (from the ROM), the register or memory address, and the actual (incorrect) data value read from register or memory. The on-chip self-test ROM is 4K X 24.

Note that the RAM self-test is destructive. That is, following the RAM self-test, regardless of whether the test passes or fails, the shared RAM is not restored to its state prior to this test. Following a failed RAM self-test, the host may read the internal RAM to determine which location(s) failed the walking pattern test.

SOFTWARE

The BUS-69090 series Enhanced Mini-ACE software is a suite of library functions and device drivers which provide comprehensive support of the BU-65553 card. It is strongly recommended that all new development be performed using the “Enhanced Mini-ACE Runtime Library.” The base library consists of a suite of function calls that serves to offload a great deal of low-level tasks from the application programmer. This includes register initialization, along with memory management software, and the means to implement an offline development environment.

As a means of supporting operation on multiple platforms, the BUS-69090 library is written in ANSI C, and leverages component object modeling (COM). The use of ANSI C and component object modeling provides portability to different operating systems and card types. As a result, the library may be easily ported to run on platforms based on a variety of microprocessors, running under different operating systems or -- in some cases -- no operating system.

The BUS-69090 software includes drivers for specific operating systems, including Linux and 32-bit Microsoft operating systems.

The library allows the user to specify a unique device number, along with memory size, base register address, base memory address, and mode of operation for any Enhanced Mini-ACE on a given card. The library initialization function results in configuring the Enhanced Mini-ACEs to a specific state, depending on the mode of operation. For each mode, advanced architectural features are enabled as part of the initialization. Depending on the mode of operation that is initialized, the user may access specific data structures. For example, for BC mode, there are separate functions for accessing op codes, messages, data blocks, and frames. There are separate functions which may be invoked to release all resources for a particular device.

For all function calls, the library checks all parameters for validity, with invalid parameters resulting in error codes.

MEMORY MANAGEMENT SOFTWARE

The library operates under an open/access/close model. Under this model, areas of Enhanced Mini-ACE and host RAM are allocated and de-allocated by means of low-level routines. When an area of host RAM is closed, it is relinquished back to the operating system. While these low-level functions may be invoked directly by an application, in general their operation is transparent to the application programmer. The library's memory manager module performs autonomous allocation of shared memory for stacks, data blocks, and other structures. This provides a high degree of flexibility for sizing various data structures.

HOST BUFFERING

For all modes of operation, the Enhanced Mini-ACE runtime library allows the programmer to log all messages processed. With the use of the Host Buffering feature, all messages will be automatically transferred from the Enhanced Mini-ACE memory into a user-definable host memory segment. When polling in a real time operating system such as VxWorks, simple logging techniques such as polling may be used to capture all messages.

Unfortunately, in non-deterministic systems in which the user has little or no control of how long it will take to read new messages off the Enhanced Mini-ACE stack, messages may be lost.

In systems such as Microsoft Windows which do not have the luxury of real time processing, the runtime library allows for a Host Buffer to be installed. The Host Buffer (HBUF) is a circular memory structure resident on the host which contains the log of all messages. Messages are transferred to the HBUF by means of interrupts which occur at 50% and 100% rollover of RT circular buffers, or of the Monitor command and data stacks.

BC MODE SOFTWARE

The memory management for BC mode allocates RAM space for the BC instruction list, individual message control/status blocks, data blocks, and the general purpose queue.

The library provides comprehensive support of the Enhanced Mini-ACE's advanced bus controller capabilities. This includes function calls and macros invoking the BC instruction set.

The Enhanced Mini-ACE runtime library encapsulates all Op Codes, data blocks, messages, and frames. Frame types include major, minor, and asynchronous. This allows the user to create the desired 1553 BC activity, without the overhead of memory management.

For BC mode, there are a number of linked list (LL) type constructs defined by the library. These include Op Code LL Items, Frame LL Items, Message LL Items, and Data Block LL Items. The library includes "create" and "delete" functions for these constructs.

The library also supports higher level bus controller functions. These implement higher level tasks such as minor and major frame timing control, conditional messaging, asynchronous message insertion, and interrupts after specific messages. There are also capabilities for interrupt-driven transfers of minor frame data and access to the general purpose queue.

RT MODE SOFTWARE

The library enables high-level operation for configuring and accessing the Enhanced Mini-ACE's RT. This includes routines for configuring the single, double, and circular subaddress buffering modes, and/or the global circular buffer mode. For each buffering mode, the library performs the necessary memory allocation and data block creation.

The library provides a mechanism for automatically reading and accessing the most recently received message. The library supports methods for synchronously and asynchronously accessing received message data using the single and double buffered methods respectively.

In addition, there is high-level support of subaddress illegalization and use of the busy bit (programmable by subaddress), enhanced mode code handling (interrupts and dedicated RAM locations for specific mode codes), along with functions allowing for accessing user-programmable status and BIT words.

The library includes constructs supporting bulk data transfers to or from an RT, using the Enhanced Mini-ACE circular buffer, and the 50% and 100% rollover interrupt features.

There is functionality for transferring one, multiple, or all RT messages to a host buffer. These functions store messages into consolidated data structures comprised of command and data words, and message status.

MESSAGE MONITOR SOFTWARE

The Enhanced Mini-ACE library's message monitor architecture includes functions for specifying the monitor command and data stack sizes. This includes automatic allocation of RAM space for these stacks. The monitor library includes a function for programming of the monitor "select" or "filter" table. This allows the application to specify which 1553 commands, defined by specific combinations of RT addresses, T-R bit, and subaddress, will be stored by the monitor.

The monitor library includes high-level tools to decode monitored messages. Transfers from the monitor command and data stacks to a host buffer are interrupt-driven, using the 50% and 100% stack rollover interrupt features. Similar to the RT, the monitor software includes the capability to transfer message data words and status information to the host RAM in a consolidated data structure.

INTERRUPT HANDLING

Enhanced Mini-ACE interrupt handling involves the use of a blocking system. Interrupts are handled by a very high priority background thread, which is part of the library. The library sets up the thread. This routine normally stays in a "blocking" state, awaiting an interrupt request. At this "blocking" point, the thread waits in a dormant state for an interrupt to occur, or an "exit" condition. An "exit" condition is either a BuClose, or is invoked by the operating system.

When the processor -- and then the operating system -- is signaled that an interrupt request has occurred, the driver reads card-specific registers to determine if a particular Enhanced Mini-ACE channel has requested interrupt service. Assuming that one of the Enhanced Mini-ACEs has issued an interrupt, the library checks to see which interrupt event(s) has occurred. If more than one interrupt event has occurred, these are processed sequentially, one at a time.

If the interrupt request was issued by a particular Enhanced Mini-ACE, then its "blocking" condition will be lifted. At this time, the library will then read the respective Enhanced Mini-ACE's

TABLE 6. A SAMPLING OF BC LIBRARY FUNCTIONS	
aceBCDataBlkCreate (DevNum, nDataBlkID, wDataBlkSize,*pBuffer, wBufferSize)	This function allocates a data block to be used by any message. The data block may be 1 to 32 words long, single or double buffered.
aceBCOpCodeCreate (DevNum, nOpCodeID, wOpCodeType, wCondition, wParameter1, wParameter2, dwReserved)	This function creates an op code/parameter word pair and appends it to the BC instruction list.
aceBCMsgCreateBCtoRT (DevNum, nMsgBlkID, nDataBlkID, wRT, wSA, wWC, MsgGapTime, dwMsgOptions)	This function creates the message control/status block for a BC-to-RT transfer message. There are separate functions for RT-to-BC transfers, RT-to-RT transfers, mode code messages, BC-to-RTs broadcast transfers, BC-to-RTs broadcast messages, and broadcast mode code messages.
aceBCFrameCreate (DevNum, nFrameBlkID, wFrameType, aOpCodeIDs, wOpCodeCount, wMnrFrmTime, wFlags)	This function creates a BC frame from an array of Op Code IDs. The frame may be either a minor or major frame.
aceBCStart (DevNum, nMjrFrmID, lMjrFrmCount, pMjrFrmNode, pMsgNode, pDataNode, pFrameNode)	This function initiates the BC to process a specified major frame.

TABLE 7. A SAMPLING OF RT LIBRARY FUNCTIONS	
aceRTDataBlkCreate (DevNum, nDataBlkID, wDataBlkType, *pBuffer, wBufferSize)	This function allocates an RT data block.
aceRTDataBlkMapToSA (DevNum, nDataBlkID, wSA, wMsgType, wIrqOptions, wLegalizeSA)	This function maps a data block defined using aceRTDataBlkCreate with a specified transmit, receive, or broadcast subaddress. The function may also be used to legalize or illegalize the specified subaddress.
aceRTGetHBufMsgDecoded (DevNum, MSGSTRUCT *pMsg, *pdwMsgCount, *pdwMsgLostStk, *pdwMsgLostHBuf, wMsgLoc)	This function reads and decodes a message from the host buffer (assuming that one is present), and places the decoded message into the MSGSTRUCT parameter.
aceRTDataBlkCircBufInfo (DevNum, nDataBlkID, *pUserRWOffset, *pAceRWOffset)	This function returns information about a circular buffer, including the last read or written location performed by the user and the last location read or written by the Enhanced Mini-ACE RT.

TABLE 8. A SAMPLING OF MONITOR LIBRARY FUNCTIONS	
aceMTEnableRTFilter (DevNum, wRT, wTR, dwSAMask)	This function may be used to enable monitor selection for a specific subaddress or all subaddresses, for a specific RT address or all RT addresses.
aceMTStkToHBuf (DevNum)	This function copies all messages from the (previously active) stack to the host buffer. Once the messages have been moved to the host buffer, they can be processed by the application using either aceMTGetHBufMsgsRaw() or aceMTGetHBufMsgsDecoded().
aceMTGetHBufMsgDecoded (DevNum, MSGSTRUCT *pMsg, *pdwMsgCount, *pdwMsgLostStk, *pdwMsgLostHBuf, wMsgLoc)	This function reads either the last unread or most recently received decoded message from the host buffer.

Interrupt Status Registers. Note that there is a separate thread for each Enhanced Mini-ACE. For each Interrupt Status Register bit that is set, the library interrupt service routine checks to see if the corresponding Interrupt Mask Register bit has been set. At this point, the library references a lookup table, which will then invoke one of several specific user routines associated with specific interrupt events. In responding to specific interrupt events, the user routines may then invoke other library functions.

After a particular interrupt event has been responded to, the thread will then perform a "manual" interrupt clear (if necessary). At that time, the thread will then revert to its "blocking" condition.

INTERFACE TO MIL-STD-1553 BUS

FIGURE 12 illustrates the interface from the BU-65553 to a 1553 bus for either transformer (long stub) or direct (short stub) coupling, plus the peak-to-peak voltage levels that appear at various points (when transmitting). Both coupling configurations require the use of an included cable assembly that interfaces directly to the PC Card. For the transformer (long stub) coupling configuration an external transformer, referred to as a coupling transformer, is required. In accordance with MIL-STD-1553B, the turns ratio of the coupling transformer is 1.0 to 1.4.

The standard cable provided with the BU-65553 (part number DDC-72316-1) provides for a transformer coupled connection. The mating connector required on the stub cable is a Trompeter PL75 or equivalent connector. Direct coupling requires the use of an optional cable assembly (DDC-72316-2).

TABLE 9. INTERFACE CABLE ASSEMBLY TERMINATION TABLE

TRANSFORMER COUPLED (DDC-72316-1)	
FROM	**TO**
P1	P2
9, 10	Center
13, 14	Outer
16, 17	Shield
P1	P3
21, 22	Center
25, 26	Outer
18, 19	Shield
DIRECT COUPLED (DDC-72316-2)	
FROM	**TO**
P1	P2
1, 2	Center
5, 6	Outer
16, 17	Shield
P1	P3
29, 30	Center
33, 34	Outer
18, 19	Shield

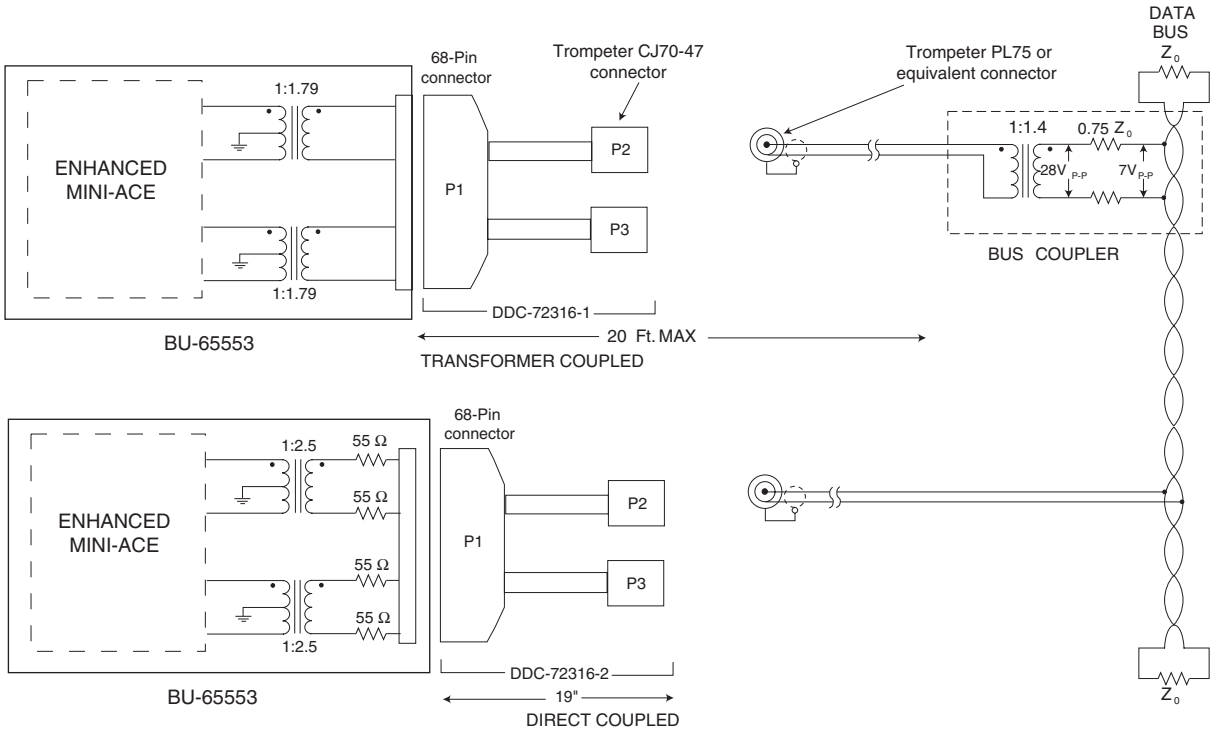


FIGURE 12. BU-65553 INTERFACE TO A MIL-STD-1553 BUS

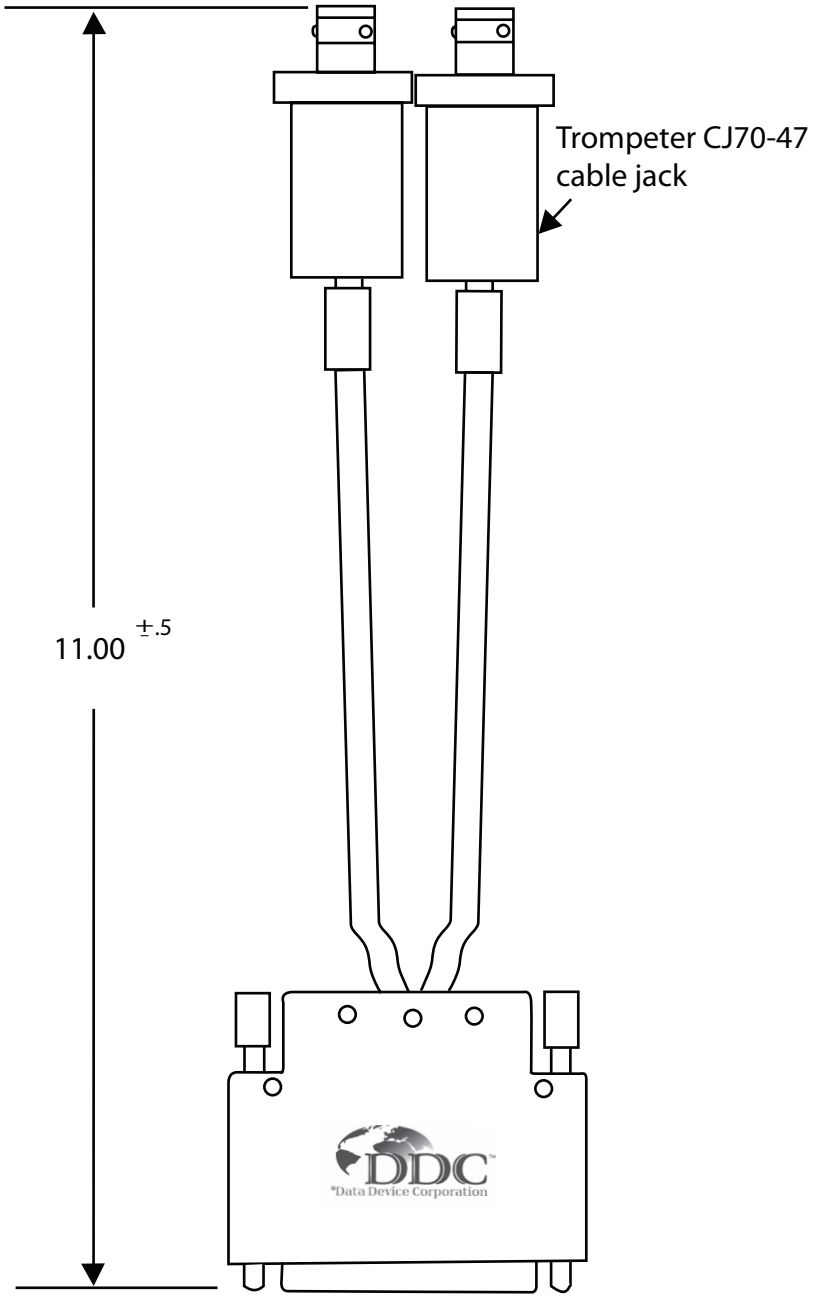


FIGURE 13. BU-65553 TRANSFORMER COUPLED CABLE (PN 72316-1)

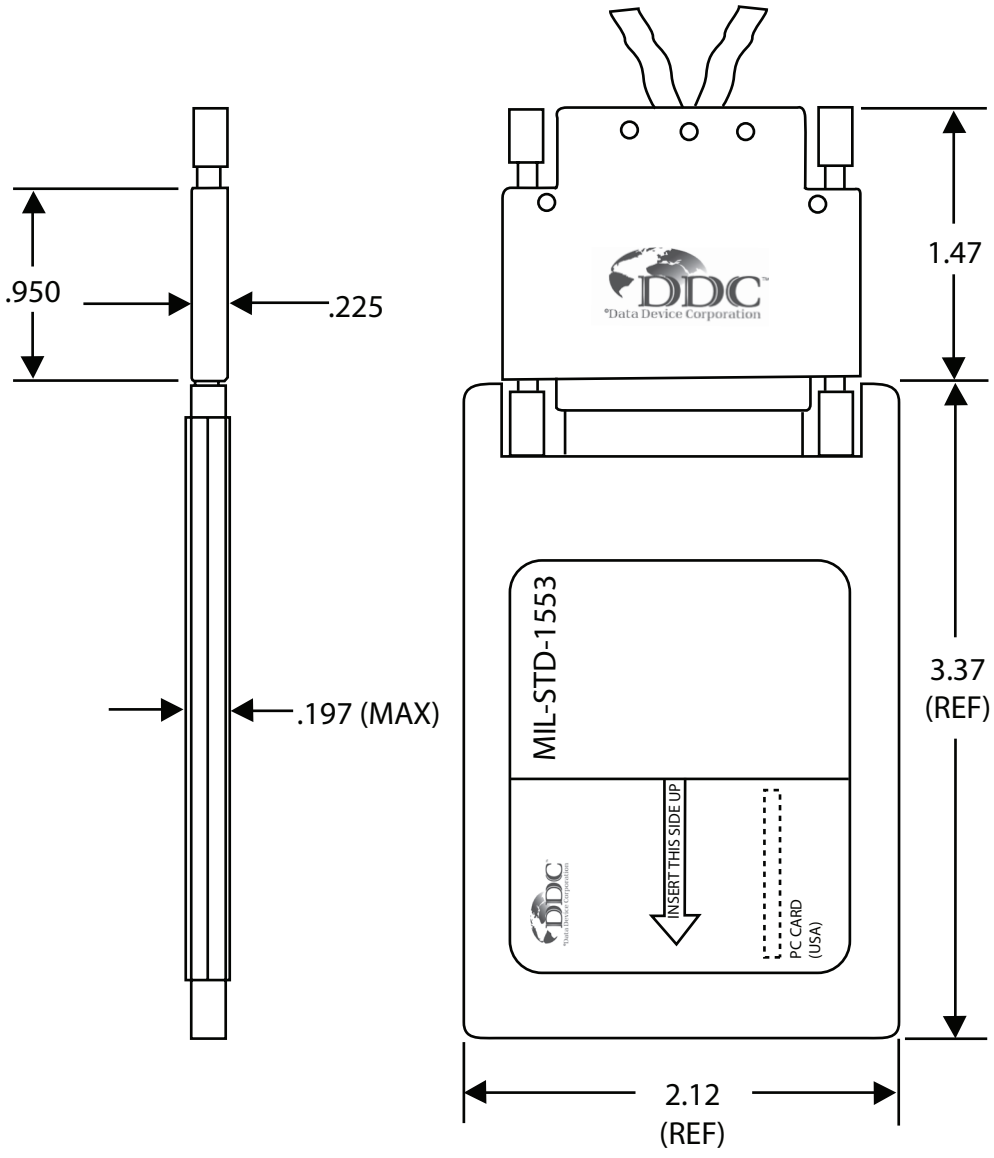


FIGURE 14. BU-65553 MECHANICAL OUTLINE

ORDERING INFORMATION

BU-65553 M 2 - 3 0 0

Test Criteria:
0 = None

Screening:
0 = Standard DDC Procedures

Temperature Range:
3 = 0°C to +55°C

Cable Type:
0 = Without Cable
1 = Direct Coupled (note 1)
2 = Transformer Coupled (note 2)

Package:
M = Module

Product Type:
BC/RT/MT with 64K RAM

Notes:
1. Includes DDC-72316-2 direct coupled cable assembly.
2. Includes DDC-72316-1 transformer coupled cable assembly.
3. The above products contain tin-lead solder.

INCLUDED SOFTWARE:

BUS-6908XS0

ACE Runtime Library / ACE Menu:
1 = Linux 32-bit Runtime Library (Kernel 2.4)
2 = Windows 9x 32-bit Runtime Library
3 = Windows NT/2000/XP 32-bit Runtime Library
4 = Windows 9x ACE Menu GUI
5 = Windows NT/2000/XP ACE Menu GUI

BU-69090SX

Enhanced Mini-ACE Runtime Library:
0 = Windows 9x/NT/2000/XP 32-bit Runtime Library
1 = Linux Library/Driver

OPTIONAL SOFTWARE:

BU-69404DM-64VM:
32-bit dataMars, Data Monitoring, Analysis, and Replay; with Virtual Panels and Multiplex

STANDARD DDC PROCESSING FOR DISCRETE MODULES/PC BOARD ASSEMBLIES		
TEST	**METHOD(S)**	**CONDITION(S)**
INSPECTION / WORKMANSHIP	IPC-A-610	Class 3
ELECTRICAL TEST	DDC ATP	—

The information in this data sheet is believed to be accurate; however, no responsibility is assumed by Data Device Corporation for its use, and no license or rights are granted by implication or otherwise in connection therewith.
Specifications are subject to change without notice.

Please visit our web site at **www.ddc-web.com** for the latest information.

©Data Device Corporation
105 Wilbur Place, Bohemia, New York, U.S.A. 11716-2426

For Technical Support - 1-800-DDC-5757 ext. 7771

Headquarters, N.Y., U.S.A. -Tel: (631) 567-5600, Fax: (631) 567-7358
United Kingdom - Tel: +44-(0)1635-811140, Fax: +44-(0)1635-32264
France - Tel: +33-(0)1-41-16-3424, Fax: +33-(0)1-41-16-3425
Germany - Tel: +49-(0)89-15 00 12-11, Fax: +49-(0)89-15 00 12-22
Japan - Tel: +81-(0)3-3814-7688, Fax: +81-(0)3-3814-7689
World Wide Web - http://www.ddc-web.com

CE

RECORD OF CHANGE

For BU-65553 Data Sheet

Revision	Date	Pages	Description
E	Feb., 2009	3	Added vibration, shock, and humidity values to the specification table.
F	Jan., 2010	26	Removed -25°C to +75°C “Y” option from ordering information page.

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