

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***

# Hardware Manual
# for the BU-65565/66 MIL-STD-1553
# PMC Four-Channel Interface Card
## MN-65565XX-001

The information provided in this Manual 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 http://www.ddc-web.com/ for the latest information.




**105 Wilbur Place, Bohemia, New York 11716-2426**

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

**Headquarters -** 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




DATA DEVICE CORPORATION
REGISTERED TO ISO 9001:2000
FILE NO. A5976

All rights reserved. No part of this Manual may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Data Device Corporation.

REV M – Sept, 2007
© 2000 Data Device Corp.

## RECORD OF CHANGE

| Revision | Date | Pages | Description |
|---|---|---|---|
| A | Sept., 2000 | All | Original Issue |
| B | June, 2002 | 9<br>19<br>46<br>49<br>56<br>68<br>69<br>79<br>83-88 | Card Pinouts section revised.<br>Chapter title revised.<br>Table 35 revised, and renumbered.<br>Table 36 revised, and renumbered.<br>Table 37 revised, and renumbered.<br>RT Address section revised.<br>RT Auto-boot section removed.<br>Acronyms revised.<br>Appendix C updated. |
| C | Aug., 2002 | 7,8<br>9<br>10,11<br>12<br><br>18-42<br>74 | PCI Interface Section<br>Signal Section<br>RT Address Inputs<br>$\overline{\text{SSFLG}}$/EXT_TRIG_X Inputs Section<br>Removed Appendix D<br>Added NEW Software Installation Section<br>Figure 32, Graphics Change |
| D | Mar., 2003 | Various | BU-65566 card information added.<br>Pn4 and J5 I/O connector information added.<br>Various edits and typos corrected. |
| E | June, 2003 | 12-14, 58-65 | Added NEW Table 1 and 2; Added 1MB Flash Memory Section. Added Tables 18-25 with Text |
| F | Nov., 2003 | Various | Amended system requirements, added QuickLogic details, typographical edits. |
| G | July, 2004 | 25, 26 | Replace Figures 5 and 6. |
| H | March, 2005 | Various | Added Table 4, renumbered all subsequent tables, updated Table 7, typographical edits. |
| J | May, 2005 | Pgs 16, 21, 24-27 | Added page 16, Table 4, Figures 3-6 |
| K | June, 2005 | Page 23 | Updated Table 5 |
| L | April, 2006 | 9, Various | Updated power description section, removed Appendix A and B, typographical edits. |
| M | Sept., 2007 | 9, 57, 58, 59 | Remove references to BAR3. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## TABLE OF CONTENTS

**LIST OF FIGURES** ..... **IV**

**LIST OF TABLES** ..... **V**

**HOW TO USE THIS MANUAL** ..... **VI**

Text Usage ..... vi
Symbols and icons ..... vi
Special Handling and Cautions ..... vii
Trademarks ..... vii

**INTRODUCTION** ..... **1**

System Requirements ..... 3
Technical Support ..... 3

**HARDWARE INSTALLATION** ..... **4**

Software Configuration and Operation ..... 4
Hardware Configuration and Operation ..... 7
Card Pinouts ..... 13
RT Address Inputs ..... 15
$\overline{\text{SSFLAG}}$ / EXT_TRIG_X Inputs ..... 17
Rear “Pn4” I/O Connector ..... 18
Front Panel I/O Connector ..... 21

**SOFTWARE INSTALLATION** ..... **27**

Overview ..... 27
Enhanced Mini-ACE Software Library Compatibility ..... 28
Software Driver Installation Overview ..... 29
*BU-69090S0* Enhanced Mini-ACE Runtime Library (RTL) ..... 36

**SOFTWARE OVERVIEW** ..... **42**

DLL Highlights ..... 42

**WRITING SOFTWARE FOR THE ENHANCED MINI-ACE** ..... **44**

Header Files ..... 44
Card Functions ..... 45
Low Level Programming ..... 50

**APPENDIX A** ..... **52**

BU-65565 PCI Register Definition ..... 52

**APPENDIX B** ..... **56**

BU-65566 PCI Register Definition ..... 56

**INDEX** ..... **65**

## LIST OF FIGURES


Figure 1. 64-Pin “Pn4” I/O Connector ..... 18
Figure 2. 50-Pin “J5” I/O Connector ..... 21
Figure 3. BU-65565M/C Mechanical Outline ..... 23
Figure 4. BU-65565B/F Mechanical Outline ..... 24
Figure 5. BU-65566G Mechanical Outline ..... 25
Figure 6. BU-65566M Mechanical Outline ..... 26
Figure 7. Found New Hardware ..... 30
Figure 8. Search for Driver ..... 31
Figure 9. Specify Driver Location ..... 31
Figure 10. Found Driver ..... 32
Figure 11. Finished Install of Driver ..... 32
Figure 12. 1553 Card Manager ..... 33
Figure 13. 1553 Card Manager Modify Device Number ..... 34
Figure 14. 1553 Card Manager Driver ..... 35
Figure 15. About DDC 1553 Card Manager ..... 35
Figure 16. Welcome Screen ..... 36
Figure 17. Software License Agreement ..... 37
Figure 18. Destination Folder ..... 37
Figure 19. Program Folder Location ..... 38
Figure 20. Select Components ..... 38
Figure 21. Setup Complete ..... 39
Figure 22. Select Device Number ..... 39
Figure 23. BC Op Code Format ..... 51

## LIST OF TABLES


Table 1. Flash Memory ..... 11
Table 2. Control Word Format ..... 12
Table 3. BU-65565M/C/B Pn4 Connector Pin Functions ..... 19
Table 4. BU-65566G/M Pn4 Connector Pin Functions ..... 20
Table 5. Supplied I/O Connector Components for BU-65565B/F ..... 21
Table 6. BU-65565B/F J5 Connector Pin Functions ..... 22
Table 7. Enhanced Mini-ACE Software ..... 28
Table 8. BU-65565 PCI Configuration Header ..... 52
Table 9. BU-65565 BAR0 Readback Value (after all ONEs are written to BAR0) ..... 53
Table 10. BU-65565 (BAR0) Enhanced Mini-ACE Memory Map ..... 53
Table 11. BU-65565 (BAR1) ACE Control Registers – 4K byte Total Space ..... 53
Table 12. BU-65565 Global Activity Register (BAR1 + 800h) ..... 54
Table 13. BU-65566 PCI Configuration Header ..... 56
Table 14. BU-65566 Base Address Register 0 (Memory Mapped, 4K bytes) ..... 57
Table 15. BU-65566 BAR1 Readback Value (after all ONEs are written to BAR1) ..... 57
Table 16. BU-65566 Base Address Register 1 (Memory Mapped, 512K bytes) ..... 58
Table 17. BU-65566 Base Address Register 2 (Memory Mapped, 4K bytes) ..... 58
Table 18. BU-65566 Global Activity Register (Read 0x800) ..... 59
Table 19. Interrupt Register (Read 0x804) ..... 60
Table 20. Global Write Control Register (Read/Write 0x808) ..... 61
Table 21. Flash Address Register (Read/Write 0xA00) ..... 61
Table 22. Flash Control/Data In Register (Read/Write 0xA04) ..... 62
Table 23. Flash Control/Data Out Register (Read/Write 0xA08) ..... 63
Table 24. Flash Error Code ..... 63
Table 25. FPGA Revision ID/Flash Unprotect Register (Read/Write 0xA0C) ..... 64

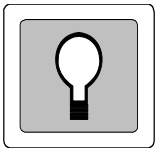
# HOW TO USE THIS MANUAL

This manual uses typographical and iconic conventions to assist the reader in understanding the content. This section will define the text formatting and icons used in the rest of the manual. This manual is formatted with a 'Scholar Margin' where many tips, symbols or icons will be located.

## Text Usage

- **BOLD** – text that is written in bold letters indicates important information and table, figure, and chapter references.
- ***BOLD ITALIC*** – will designate DDC Part Numbers.
- `Courier New –` is used to indicate code examples.
- <…> - Indicates user entered text or commands.

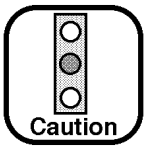
## Symbols and icons



The Idea/Tip icon will be used to identify a handy bit of supplementary information that may be useful to the user.



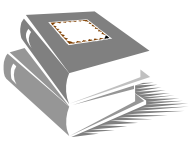
The Note icon signifies important supplementary information that will be useful to the user.



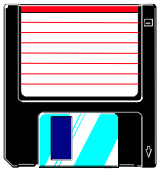

The Caution icon identifies important information that presents a possibility of damage to the product if not heeded.




Much stronger than a Caution, the Warning icon presents information pertaining to hazards that will cause damage to the product and possible injury to the user.



The Reference icon indicates that there is related material in this manual or in another specified document.



The Disk Icon describes information that is related to software.

## Special Handling and Cautions

The ***BU-65565/6XX-300*** uses state-of-the-art components, and proper care should be used to ensure that the device will not be damaged by Electrical Static Discharge (ESD), physical shock, or improper power surges and that precautions are taken to avoid electrocution.




- Turn off power to the computer hardware and unplug from wall.
- **NEVER** insert or remove card with power turned on.
- Ensure that standard ESD precautions are followed. As a minimum, one hand should be grounded to the power supply in order to equalize the static potential.
- Do not store disks in environments exposed to excessive heat, magnetic fields or radiation.

## Trademarks

All trademarks are the property of their respective owners.

# INTRODUCTION

The ***BU-65565*** is a multi-channel MIL-STD-1553 PMC card supporting one to four dual redundant 1553 channels on a convection-cooled or conduction-cooled card. The design of the ***BU-65565*** leverages the full capabilities of the new DDC Enhanced Mini-ACE terminal. These features include a highly autonomous bus controller, a remote terminal providing a wide variety of buffering options, and a selective message monitor. Each of the installed Enhanced Mini-ACE channels incorporates 3.3 volts and 64K words of RAM.

The conduction-cooled version of the ***BU-65565*** includes a black anodized aluminum plate, thermal vias connected to chassis ground, and a chassis ground plane to provide improved thermal conduction.

Features of the board include:

- Each channel independently programmed for BC, RT, Monitor or RT/Monitor
- Programmable BC operation
- Conduction or convection cooled
- Transformer-coupled bus connection
- Plug and play configuration of memory and interrupt resources
- 64K words of shared static RAM per installed channel

The ***BU-65566*** card evolved from the ***BU-65565*** PMC card, and adds secondary thermal interfaces and a 66MHz bus interface. The card is available for commercial, industrial, or ruggedized military environments as a conduction cooled PMC card with primary and secondary thermal interfaces as per the VITA-20-2001 specification. Secondary thermal interfaces reduce thermal rise and provide compatibility with a larger array of commercially available single board computers. The card provides a target 32-bit PCI interface that operates at clock speeds of up to 66MHz for applications where a higher bus speed is desired. If a 33MHz PMC card were placed on a 66MHz bus, this would force the speed of the bus to slow down to 33MHz. Although 1553 may not require the additional bandwidth of a 66 MHz bus, the ***BU-65566*** 66MHz PMC card allows the base CPU PCI bus to run at 66 MHz, allowing other high-speed devices on the bus to take advantage of the higher bandwidth.

Both cards support the **free *BU-69090*** series Enhanced Mini-ACE software, which includes a high level C API library, and drivers for Windows 9x/NT/2000/XP, Linux, and VxWorks. The library interface supports all of the Enhanced Mini-ACE's advanced architectural features. The easy-to-use high level functions in the library abstract all register accesses and memory allocation such that no low level knowledge of the hardware is required.

The BC's highly autonomous message sequence control engine provides a means for offloading the host processor, which can be a critical factor in embedded systems and real time applications. The library supports the full use of the BC instruction set and allows you to manage the creation of data blocks, while easily creating BC opcodes, messages, and frames. New features in the library allow users to dynamically insert asynchronous messages while the BC is running! A software circular Host Buffer can be easily created in the library to log all messages and data on the 1553 data bus.

The Enhanced Mini-ACE RT offers the choice of multiple memory management schemes for data buffering to ensure data integrity. Other features of the ***BU-65566*** include fully autonomous built-in self-tests of protocol logic and RAM, and built in RAM parity checking.

The library supports an offline development environment that allows you to easily generate binary image files and C header files. This greatly serves to minimize the code size, reduce computational resources, and reduce software validation and documentation. The library also supports simulated access mode. This operating mode allows you to start programming your card in BC, RT, MT, or RT/MT mode without having any hardware. The library will allocate host memory as if it were Enhanced Mini-ACE memory.

Both cards are available with DDC's optional ***dataMARS*** and ***dataSIMS*** software. ***dataSIMS*** is an advanced monitoring and simulation tool that provides a graphical user interface for acquisition, display, simulation, and storage of real time bus traffic. The software provides engineering unit conversions and advanced simulation capability such as applying waveforms, constants, formulas, or user defined code to data. ***dataMARS*** has all of the same features as ***dataSIMS*** except for the simulation capability.

### WHAT IS INCLUDED IN THIS MANUAL?



This manual contains a complete description of hardware installation and use, and an introduction to the software functions.

The library software provides a level of abstraction such that it is not necessary to understand the operation of the chip set.

## System Requirements

- PMC bus supporting 3.3 Volts and/or 5 Volts
- Windows® 9x/2000/XP, Windows NT®, Linux®, or VxWorks®
- For Non-Standard ***BU-65565*** cards, only 5 Volts are required on the PCI Bus

## Technical Support

In the event that problems arise beyond the scope of this manual, you can get in touch with DDC by calling:

**Customer support:**

1-800-DDC-5757, ext. 7771

**Headquarters:**

1-631-567-5600, ext. 7771
Fax: 1-631-567-5758 to the attention of DATA BUS Applications.

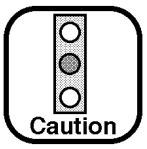
**Regional Offices:**

Southeast:
Tel: (703) 450-7900 Fax: (703) 450-6610
West Coast:
Tel: (714) 895-9777 Fax: (714) 895-4988
Europe:
Tel: 44 (1635) 811140 Fax: 44 (1635) 32264
Asia/Pacific:
Tel: 81 (3) 3814-7688 Fax: 81 (3) 3814-7689

DDC also has an Internet World Wide Web site, which allows customers to easily download new revisions of software and documentation. The Internet address is **www.ddc-web.com**.

# HARDWARE INSTALLATION

The ***BU-65565*** card is a PMC target device that operates at bus speeds of up to 33MHz that may be inserted into any compatible slot. The ***BU-65566*** card is a PMC master/target device that operates at bus speeds of up to 66MHz and has both primary and secondary thermal interfaces as per the VITA-20-2001 specification that may be inserted into any compatible slot.




When installing the cards, the following should be observed:

**NEVER** insert or remove the cards with the power turned on.




**ALWAYS** take proper precautions to guard against static damage. Use a wrist strap if available, or ensure proper static grounding by touching the power supply cover **WITH POWER OFF**.

Insert the cards at a slight angle and then gently press into the PMC site connector. Secure with proper hardware.

Make sure that adjacent cabling and wiring do not hinder the airflow around the convection cooled card.

These cards do not contain any configuration jumpers or switches. Connecting to the appropriate pins at the I/O connectors makes selection of direct or transformer coupling. In addition, there are no jumpers needed for card installation, as these cards are true Plug and Play devices. Interrupts and memory selections will be performed by the operating system as the boot system progresses.

## Software Configuration and Operation

The cards are supplied with a C Application Programming Interface (API) library and drivers for Windows 9x/2000/XP, Windows NT, Linux, and VxWorks. The VxWorks software was developed by DDC on a Motorola MVME2700 card, which utilizes a PowerPC 750 microprocessor. A version of the VxWorks software for the Intel platform is also available free of charge. The device drivers are developed using the WindRiver Tornado integrated development environment. The source code for the driver and library are available to you so that the driver can be tailored to your actual hardware that is hosting VxWorks. The Linux driver was built for the Intel platform and is a loadable module that can be recompiled against your kernel version. A make file is included to allow you to easily rebuild the

driver against your kernel source. The drivers for Windows 9x/2000/XP and Windows NT are plug and play drivers. No user intervention is required other than running the included setup program to install the drivers under your Windows operating system.

The driver software will make the required calls to the operating system in order to acquire the correct address and interrupt resource information. These resources will be used by the driver during initialization of the card, and for read, write and interrupt functions during normal operation. The ability to gain configuration in this manner enables a hands-off configuration of the card in any system.

## MEMORY ALLOCATION

For each mode of operation - BC, RT, and Monitor - the library and driver operate under an open/access/close model, in which areas of RAM are autonomously allocated and de-allocated by means of low-level library routines. While these 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 frame structures. This provides a high degree of flexibility for sizing the various data structures.

The memory management functions make use of handles consisting of starting addresses, sizes of memory blocks, and information delineating whether particular areas of shared RAM are unused, used, or protected. For each mode, there are functions to transfer data between shared RAM data blocks and buffers in host memory. In addition, there are functions to access consolidated data structures providing both message status information, as well as 1553 message words.

## MODE-SPECIFIC SOFTWARE

### *BC Mode*

There is comprehensive support of the enhanced bus controller capabilities, allowing the user to leverage the library function calls and macros invoking the BC instruction set. Some of the functions support higher level tasks such as minor and major frame timing control and asynchronous message insertion. For the enhanced BC, the software also supports the offline development and compilation of BC message scenarios on (for

example) a desktop PC. The binary images created from these may then be downloaded to the target processor environment.

The Enhanced Mini-ACE Library encapsulates all opcodes, data blocks, messages, and frames (major, minor, and asynchronous). This allows the user to create the desired 1553 BC activity, without the overhead of memory management. The library includes a function that creates two files that allow the user to view the opcode sequence generated by the library. The first file is a binary file that contains an image of Enhanced Mini-ACE memory, and the second file is a 'C' header file that shows all locations to structures/frames within memory. This header file contains a readable ASCII representation of the entire mapping of the Enhanced Mini-ACE as it would be setup based on the operations entered into the program. It also includes functions that allow a program to access and modify the structure of the Enhanced Mini-ACE while it is running.

### *RT Mode*

There is high-level operation for configuring and utilizing the single-buffer, double buffer, and circular-buffered sub-address memory management schemes. This includes methods for accessing synchronously and asynchronously received message data. There is also a mechanism provided for automatically reading and accessing the most recently received message. In addition, there is high-level support of sub-address illegalization and use of the busy bit, enhanced mode code handling, along with functions allowing for accessing user programmable status and BIT words.

### *Message Monitor Mode*

Functions that are provided for programming the message monitor include command and data stack sizes and programming of the monitor "filter" table (which addresses/T-R/sub-addresses to monitor). Additionally, there are high-level tools that decode monitored messages, and transfer status information and message words to host RAM in a consolidated stack data structure.

## DOWNLOADABLE IMAGE

One of the new features provided with the new Enhanced Mini-ACE software library is the ability to create an image of the configured device. An additional header file is also generated that contains all of the address offsets into the image and the necessary functionality to access all messages and frames.

This new feature will help in reducing development time by allowing the programming and testing for the Enhanced Mini-ACE to be performed on a PC with any card that contains an Enhanced Mini-ACE. When the code is complete and tested, an image of the Enhanced Mini-ACE can be saved to a file that can then be downloaded to the Enhanced Mini-ACE on the card. When the header file is included with the system software, all operations that must be performed on the device are possible without the need of including the entire library in the compilation.

## Hardware Configuration and Operation

The ***BU-65565/66*** cards are PCI devices, and as such do not require any jumpers or switches to set the Base address or interrupt values. The job of configuration for Plug-and-Play PCI is performed by the operating system. During the initial power on boot process, the operating system performs an enumeration of the PCI bus and allocates a resource configuration that satisfies the card requirements. The operating system will save the configuration information in the PCI registers that the card is required to provide. These registers are configured at the factory to contain information that identifies the card type and manufacturer, the required memory sizes and interrupt resources. The operating system will perform some simple tests on the resource registers in order to identify the memory sizes required, and then reload the registers with the actual resource address that has been reserved for the card. When the driver loads it will access these registers and identify what configuration the operating system has been provided for the card. After identifying each of the installed cards, the device driver will enumerate each of the Enhanced Mini-ACE's and create a configuration structure that defines the allocated address and interrupt. This information is directly available to the Enhanced Mini-ACE library and is necessary for read, write and interrupt operations.

### POWER

As a means of minimizing power consumption and dissipation, the design of both of the standard ***BU-65565*** and ***BU-65566*** cards use +3.3 volts for the PCI interface and 1553 Enhanced Mini-ACE logic, and +5 volts for the 1553 transceivers and RAM.

The standard ***BU-65565*** card uses +3.3 and +5.0 volts from the PMC connector. For applications where +3.3 volts is not available, DDC can supply a non-standard version of the ***BU-***

***65565*** card that requires only +5.0 volts from the PMC connector (consult factory).

The default configuration of the ***BU-65566GX*** card provides for an on-board voltage regulator that locally generates +3.3 volts from the +5.0 volts on the PMC connector. On the other hand, the ***BU-65566MX*** requires both +5.0 volts and +3.3 volts from the PMC connector.

DDC can supply a non-standard version of the ***BU-65566GX*** card where the voltage regulator is removed and the card uses both +3.3 volts and +5.0 volts from the power supply pins on the PMC connector (consult factory).

## PCI INTERFACE

The ***BU-65565***'s PCI interface is a fully compliant target (slave) agent, as defined by the PCI Local Bus Specification Revision 2.2, using a 32-bit interface that operates at clock speeds of up to 33 MHz, in a 3.3 volt or 5 volt signaling environment. The ***BU-65566's*** PCI interface is a fully compliant target agent using a 32-bit interface that operates at clock speeds of up to 66MHz on a 3.3V PCI bus as defined by the PCI bus specification. The interface supports PCI interrupts on the INTA line and contains a 64 deep by 32 wide FIFO to accelerate burst write transfers from the PCI host.

When using the ***BU-65565/66*** with the supplied Enhanced Mini-ACE runtime library and drivers there is no need to reference the PCI configuration registers. These registers are listed in this manual for the advanced user to aid in the development of custom software without using the supplied drivers and Enhanced Mini-ACE runtime library. This may be necessary if a custom operating environment is being used or an unsupported operating system is the target interface.

Both cards contain only a single set of PCI configuration registers such that all of the Enhanced Mini-ACE's memory and register space may be addressed through a single PCI function.

### *BU-65565 Card Only*

Internal registers implement the Subsystem Vendor and Device ID, and control the Fail-Safe operation of the device. There are two Base Address Registers used to implement the Enhanced Mini-ACE memory space (BAR0) and register space (BAR1). The Base Address Register mapping is located in PCI configuration register space.

The Enhanced Mini-ACE registers and memory mapping are located in PCI memory space, allowing for full PCI access to all 1553 terminals. The ***BU-65565*** configuration registers, the Enhanced Mini-ACE RAM (64K X 16 each) (BAR0) and Enhanced Mini-ACE registers (BAR1) can be accessed in 32 bit-words. The BAR1 space has the Enhanced Mini-ACE registers unpacked: The upper 16 bits of the 32 bit read will be all zeros and the upper 16 bits during a 32 bit write will be ignored. The BAR0 space has the RAM packed: A 32 bit PCI access will transfer TWO 16 bit words internally.

The ***BU-65565*** supports 32-bit and 16-bit read and write operations only. 8-bit accesses are illegal, but will not generate a target abort.

### *BU-65566 Card Only*

Internal registers implement the Subsystem Vendor and Device ID, and control the Fail-Safe operation of the device. There are four Base Address Registers (BAR) used on the design of this card. All BAR mapping is located in PCI configuration register space. BAR 0 is 4K bytes long and is used for the Quick Logic FPGA control registers. BAR 1 is 512K bytes long and is used for the Enhanced Mini-ACE memory space mapping of all available channels. BAR 2 is 4K bytes long and is used for the Enhanced Mini-ACE register space mapping of all available channels.

The ***BU-65566*** configuration registers, the Enhanced Mini-ACE RAM (64K x 16 each, BAR 1) and the Enhanced Mini-ACE registers (BAR 2) can be accessed in 32-bit words. The BAR 2 space has the Enhanced Mini-ACE registers unpacked: the upper 16-bits of a 32-bit read will be all zeros and the upper 16-bits of a 32-bit write will be ignored. The BAR 1 space has the RAM packed: a 32-bit PCI access will transfer two 16-bit words internally.

The ***BU-65566*** supports 32-bit and 16-bit read and write operations only. All 8-bit accesses are illegal, but will not generate a target abort.

## INTERRUPTS

The Enhanced Mini-ACEs may issue interrupt requests over the PCI bus. PCI Interrupts are generated on the INTA# output signal to the PCI host. The interrupts from each of the channels are functionally OR'd together to provide a single interrupt. The operations to be taken during the interrupt cycle will be

determined by both the device driver requirements and a service routine that is specified by the user application. The device driver will determine which of the Enhanced Mini-ACEs on the card generated the interrupt, read the interrupt status registers and queue, transfer this information to an interrupt structure and then signal the appropriate application that the interrupt occurred. The Enhanced Mini-ACE library provides the necessary routines for generating user Interrupt Service Routines and then hooking them into the device driver.

## REGISTER AND MEMORY ADDRESSING

The PCI interface on the cards contains a set of "Type 00h" PCI configuration registers that are used to map the device into the host system. The PCI configuration register space is mapped in accordance with PCI revision 2.2 specifications. As stated above, the ***BU-65565*** requires only the first two Base Address Registers. For a description of the specific use of the PCI registers as defined by the ***BU-65565***, refer to Appendix A. For a detailed description of the specific use of the PCI registers as defined by the ***BU-65566***, refer to Appendix B.

## 1MB FLASH MEMORY

The card contains a 512K x 16 (1MB) Flash memory that can be used to configure the devices to auto initialize. If using the Enhanced Mini-ACE Runtime Library Software package that came with your card then please refer to the Enhanced Mini-ACE Runtime Library Software Manual for proper initialization and operation of the Flash Memory.

The following section contains a detailed description of the structure and usage of the on-board Flash memory. Following a reset of the device, the card will check the flash memory to see if it needs to perform an auto initialization (auto-init) of any Enhanced Mini-ACE devices. The card will perform auto-init of the Enhanced Mini-ACE devices only when the first 3 flash memory locations contain the SAI1, SAI2, and SAI3 words as specified below. If the auto-init is requested then the card will read instructions from the Flash memory and perform the specified write operations as commanded. If the auto-init is not requested then control of the device is passed onto the host.

The auto-init function will stop when an End Auto Init (EAI) word is seen (in a control word position) or address 77FFFh is reached and no EAI word has been seen. Following completion of initialization of the Enhanced Mini-ACEs, control of the device is returned to the host.

Flash memory is 512K x 16(1MB) as follows:

**Table 1. Flash Memory**

| Sector | Size (K words) | Address Range |
|---|---|---|
| ***S0*** | 32 | 00000h-07FFFh |
| ***S1*** | 32 | 08000h-0FFFFh |
| ***S2*** | 32 | 10000h-17FFFh |
| ***S3*** | 32 | 18000h-1FFFFh |
| ***S4*** | 32 | 20000h-27FFFh |
| ***S5*** | 32 | 28000h-2FFFFh |
| ***S6*** | 32 | 30000h-37FFFh |
| ***S7*** | 32 | 38000h-3FFFFh |
| ***S8*** | 32 | 40000h-47FFFh |
| ***S9*** | 32 | 48000h-4FFFFh |
| ***S10*** | 32 | 50000h-57FFFh |
| ***S11*** | 32 | 58000h-5FFFFh |
| ***S12*** | 32 | 60000h-67FFFh |
| ***S13*** | 32 | 68000h-6FFFFh |
| ***S14*** | 32 | 70000h-77FFFh |
| ***S15*** | 16 | 78000h-7BFFFh |
| ***S16*** | 4 | 7C000h-7CFFFh |
| ***S17*** | 4 | 7D000h-7DFFFh |
| ***S18*** | 8 | 7E000h-7FFFFh |

**Notes for Table 1:**
1) Sectors S0-S14 are available for the auto initialization function.
2) Size of the auto-init image will vary with differing configurations.
3) Any sectors not used for auto-init functionality are available for user storage.
4) Sectors S15 - S18 are dedicated for user storage and can not be used for auto-init functionality.
5) Memory locations cannot be re-written. Entire sector must be erased prior to writing a different value to a memory location.
6) FPGA will write/erase protect auto-init data such that an erroneous write to the Flash memory does not corrupt auto-init image for future boots.

The format of the auto-init stored in Flash memory is as follows:

SAI1 = Start Auto Init Word1 (0xFEED)

SAI2 = Start Auto Init Word2 (0xDEAD)

SAI3 = Start Auto Init Word3 (0xBEEF)

Control = Defines Write Instruction

Address = Enhanced Mini-ACE Address to be Written

Data = Word to be written to the Enhanced Mini-ACE

EAI = End Auto Init Word (0xC0DE)

The format of the Control Word is as follows:

**Table 2. Control Word Format**

<table>
<tr><th>Control Bits</th><th>Description</th></tr>
<tr><td>(15:8) -</td><td>
# of successive data words in flash to be written consecutively to MPrime(s) Valid Range = 0 to 255<br><br>
Note : 0 (0 x 00) in bits (15:8) will result in no data words being written to the Enhanced Mini-ACE(s).<br>
In that case, the FPGA expects the flash will have a Control Word (with 0 data words in [15:8]) followed by a new Control Word for the next instruction as follows:<br>
Control (15:8) = 0<br>
Address<br>
Control New Control, No Data<br>
Address<br>
Data<br>
Data<br><br>
(7:5) – Must be set to “000” ( to differentiate from EAI word)<br>
(4) – ‘0’ – FPGA will write Enhanced Mini-ACE REGISTER<br>
‘1’ – FPGA will write Enhanced Mini-ACE MEMORY<br>
(3) – Channel 4 Write Enable<br>
(2) – Channel 3 Write Enable<br>
(1) – Channel 2 Write Enable<br>
(0) – Channel 1 Write Enable<br><br>
Example:<br>
512 x 16 (1 MB) Flash
<table>
<tr><td>SIA1</td><td>0 x 00000</td></tr>
<tr><td>SIA2</td><td>0 x 00001</td></tr>
<tr><td>SIA3</td><td>0 x 00002</td></tr>
<tr><td>Control<br>Address<br>Data</td><td>Single Write Instruction</td></tr>
<tr><td>Control<br>Address<br>Data<br>Data<br>Data<br>Control</td><td>Multiple Write Instruction (Burst)</td></tr>
<tr><td>V</td><td>(Etc.)</td></tr>
<tr><td>Data</td><td>Last data word to be written</td></tr>
<tr><td>EAI</td><td>Indicates auto-init is done</td></tr>
<tr><td>V</td><td>(Not Used)</td></tr>
<tr><td colspan="2">End of Sector EAI resides in<br>Available for User Storage</td></tr>
<tr><td>Dedicated User Storage</td><td>0 x 7800 – 0x7FFF (S15-S18)</td></tr>
</table>
</td></tr>
</table>

## Card Pinouts

This section delineates the pinouts for the cards. The only connector described here is the one that contains the 1553 signals.

### SIGNAL NAMING CONVENTION FOR I/O CONNECTORS

#### *1553 Bus Signals*

1553_#X – Where "X" indicates the channel, (A-D, 1:4), and "#" is bus A-B (1:2). + Or – indicates the phase of the 1553 BUS signal, (positive or negative).

Examples:

1553_1D + is 1553, channel 4, bus A, positive
1553_2A – is 1553, channel 1, bus B, negative

#### *RT Address Signals*

RT_ADD_X# - Where "X" indicates one of 5 RT address bits (0:4) or the parity bit (P), and "X" indicates the channel, (A-D).

Examples:

RT_ADD_D0 is the LSB of channel 4s RT Address
RT_ADD_AP is the parity bit for channel 1.

#### $\overline{\text{SSFLAG}}$ */ External Triggers*

$\overline{\text{SSFLG}}$ **/** EXT_TRG_X - Where "X" indicates the channel, (A-D, 1:4).

Examples:

$\overline{\text{SSFLG}}$ / EXT_TRIG_A is the external trigger for channel 1.
$\overline{\text{SSFLG}}$ / EXT_TRIG_D is the external trigger for channel 4.

### *External Tx Inhibit*

This pin allows the user to individually disable the transmit capability of one MIL-STD-1553 dual redundant channel. The entire channel will not transmit on bus A or B. This is accomplished by connecting this signal high, and can be done for each channel (1-4) independently. If this pin is left unconnected then the transmitters are by default enabled to transmit. This pin is only available on the ***BU-65566G/MX*** model cards that are Rev E or greater. The revision of the card can be found by the Pn 2 connector as shown on the mechanical outline drawing in Figure 5 & Figure 6.

### *External $\overline{\text{Reset}}$*

This pin allows the user to individually reset each MIL-STD-1553 channel (1-4). To reset a channel you must drive this signal low. If the pin is left unconnected you will only be able to reset channels via software. This pin is only available on the ***BU-65566G/MX*** model cards that are Rev E or greater. The revision of the card can be found by the Pn 2 connector as shown on the mechanical outline drawing in Figure 5 & Figure 6.

## RT Address Inputs

The RT addresses for each channel can be set either via hardware or software. The RT address pins are 5V tolerant and are weakly internally pulled up. If the RT address is assigned via these pins it is recommended that these pins be actively driven low/high, OR be connected directly to VCC/GND OR be pulled up (resistors less than 10K ohms)/shorted to GND.

The INTERNAL RT_AD_LAT (RT Address Latch) signal will go from Low to High shortly after the deassertion of the PCI Reset signal. This will cause the RT address bits (which are brought out to the front/rear connector, depending upon your card), to be latched into the Enhanced Mini-ACE parts on the card.

Remember that the Enhanced Mini-ACE C API library abstracts all individual register and memory operations from the application programmer to make card programming a simple task. To set the RT address with the Enhanced Mini-ACE C API library, simply call the aceRTSetAddress() function. All of the functions in the Enhanced Mini-ACE API library are described in complete detail in the ***BU-69090*** Enhanced Mini-ACE Runtime Library Software Manual.

If not using the supplied library software, the RT address can be programmed in the following ways:

1) The RT address may be “relatched” from RTAD4-RTAD0 and RTADP under software control. This operation is performed as follows:

(a) Program ENHANCED MODE (bit 15 of Configuration Register #3) to logic “1”.

(b) Program RT ADDRESS SOURCE (bit 5 of Configuration Register #6) to logic **“0**”.

(c) Program LATCH RT ADDRESS WITH CONFIGURATION REGISTER # 5 (bit 3 of Configuration Register #4) to logic **“1**”.

(d) Perform a write operation to Configuration Register #5. This will cause the Enhanced Mini-ACE’s RT address to be latched from the values presented on RTAD4-RTAD0 and RTADP. Note that for this operation, the logic values written by the host (on D5-D0) **do not** affect the value of the Enhanced Mini-ACE’s internal RT address.

(e) To prevent the RT address from being inadvertently overwritten by a subsequent write to Configuration Register

#5, it is recommended to then program LATCH RT ADDRESS WITH CONFIGURATION REGISTER # 5 (bit 3 of Configuration Register #4) to logic **"0**".

2) The second method allows for a fully software programmable RT address. This operation is performed as follows:

(a) Program ENHANCED MODE (bit 15 of Configuration Register #3) to logic "1".

(b) Program RT ADDRESS SOURCE (bit 5 of Configuration Register #6) to logic "1".

(c) Program LATCH RT ADDRESS WITH CONFIGURATION REGISTER # 5 (bit 3 of Configuration Register #4) to logic "1".

(d) Perform a write operation to Configuration Register #5. The value of the Enhanced Mini-ACE's internal RT address bits 4-0 will be latched from processor data bus bits 5-1 (D5-D1), and internal RTADP to be latched from processor data bus bit 0 (D0).

(e) To prevent the RT address from being inadvertently overwritten by a subsequent write to Configuration Register #5, it is suggested to then program LATCH RT ADDRESS WITH CONFIGURATION REGISTER # 5 (bit 3 of Configuration Register #4) to logic "0".



Refer to the "RT Address Inputs" section of the "Enhanced Mini-ACE User's Guide" for more information.

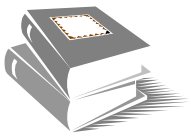
## $\overline{\text{SSFLAG}}$ / EXT_TRIG_X Inputs

These pins are an optional input for each Enhanced Mini-ACE part. These inputs are 5V tolerant and are internally pulled up. They can serve as either the subsystem flag in RT mode (with standard status word) or external trigger in BC/Word monitor mode.

In RT mode with standard status word, if this input is asserted low, the Subsystem Flag bit will be set in the Enhanced Mini-ACE RT Status Word. If the $\overline{\text{SSFLAG}}$ input is logic "0" while bit 8 of Configuration Register #1 has been programmed to logic "1" (cleared), the Subsystem Flag RT Status Word bit will become logic "1," but bit 8 of Configuration Register #1, SUBSYSTEM FLAG*, will return **logic "1"** when read. That is, the sense on the $\overline{\text{SSFLAG}}$ input **has no effect** on the $\overline{\text{SSFLAG}}$ register bit. This input has no meaning in RT mode with alternate status word.

In the non-enhanced BC mode, this signal operates as an External Trigger input. In BC mode, if the external BC Start option is enabled (bit 7 of Configuration Register #1), a low to high transition on this input will issue a BC Start command, starting execution of the current BC frame.

In the enhanced BC mode, during the execution of a Wait for External Trigger (WTG) instruction, the Enhanced Mini-ACE BC will wait for a low-to-high transition on EXT_TRIG before proceeding to the next instruction.



Refer to the Configuration Register 1 section of the “Enhanced Mini-ACE User’s Guide” for more information.

## Rear "Pn4" I/O Connector

Models ***BU-65565M/C/B*** and ***BU-65566G/M*** include provisions for a rear I/O connection via connector "Pn4". This connector (Soldered on the board) is a 64-pin "Mezzanine IEEE 1386 Board-to-Board Plug, Dual Row, SMT Vertical Stacking" (Molex P/N 71436-2164). Note that the connector is symmetrical and has no inherent pin numbering. Pin numbering designations for "Pn4" have been assigned by DDC and are as shown in Figure 1.

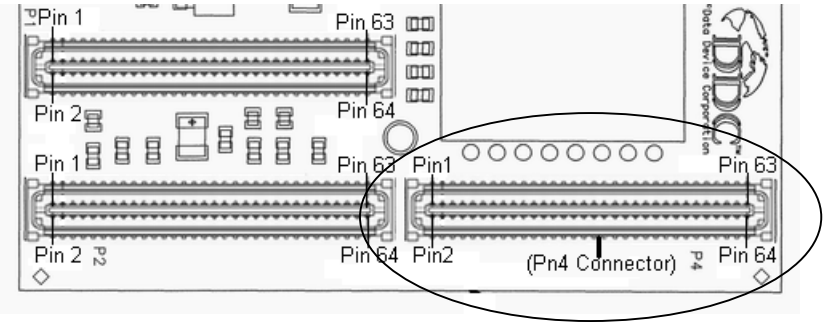



**Figure 1. 64-Pin "Pn4" I/O Connector**

**(viewed from component side of card)**

| Table 3. BU-65565M/C/B Pn4 Connector Pin Functions | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | RT_ADD_B0 | 33 | RT_ADD_D3 |
| 2 | $\overline{\text{SSFLG}}$/EXT_TRIG_A | 34 | RT_ADD_D2 |
| 3 | $\overline{\text{SSFLG}}$/EXT_TRIG _B | 35 | RT_ADD_A0 |
| 4 | $\overline{\text{SSFLG}}$/EXT_TRIG _C | 36 | 1553_1B- |
| 5 | RT_ADD_B1 | 37 | GND |
| 6 | $\overline{\text{SSFLG}}$/EXT_TRIG _D | 38 | 1553_1B+ |
| 7 | GND | 39 | RT_ADD_A1 |
| 8 | 1553_1D+ | 40 | 1553_2D- |
| 9 | RT_ADD_B2 | 41 | RT_ADD_D1 |
| 10 | RT_ADD_C4 | 42 | 1553_1C+ |
| 11 | RT_ADD_C3 | 43 | RT_ADD_A2 |
| 12 | 1553_1D- | 44 | 1553_1C- |
| 13 | RT_ADD_B3 | 45 | RT_ADD_D0 |
| 14 | RT_ADD_C2 | 46 | 1553_2C+ |
| 15 | RT_ADD_C1 | 47 | RT_ADD_A3 |
| 16 | 1553_2A- | 48 | 1553_2C- |
| 17 | RT_ADD_B4 | 49 | RT_ADD_DP |
| 18 | 1553_2A+ | 50 | 1553_1A- |
| 19 | RT_ADD_C0 | 51 | RT_ADD_A4 |
| 20 | RT_ADD_CP | 52 | 1553_1A+ |
| 21 | RT_ADD_BP | 53 | NC |
| 22 | 1553_2B- | 54 | NC |
| 23 | GND | 55 | NC |
| 24 | 1553_2B+ | 56 | NC |
| 25 | 1553_2D+ | 57 | NC |
| 26 | RT_ADD_D4 | 58 | NC |
| 27 | NC | 59 | NC |
| 28 | NC | 60 | NC |
| 29 | RT_ADD_AP | 61 | NC |
| 30 | NC | 62 | NC |
| 31 | GND | 63 | NC |
| 32 | NC | 64 | NC |

| Table 4. BU-65566G/M Pn4 Connector Pin Functions | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | RT_ADD_B0 | 33 | RT_ADD_D3 |
| 2 | $\overline{\text{SSFLG}}$/EXT_TRIG_A | 34 | RT_ADD_D2 |
| 3 | $\overline{\text{SSFLG}}$/EXT_TRIG _B | 35 | RT_ADD_A0 |
| 4 | $\overline{\text{SSFLG}}$/EXT_TRIG _C | 36 | 1553_1B- |
| 5 | RT_ADD_B1 | 37 | GND |
| 6 | $\overline{\text{SSFLG}}$/EXT_TRIG _D | 38 | 1553_1B+ |
| 7 | GND | 39 | RT_ADD_A1 |
| 8 | 1553_1D+ | 40 | 1553_2D- |
| 9 | RT_ADD_B2 | 41 | RT_ADD_D1 |
| 10 | RT_ADD_C4 | 42 | 1553_1C+ |
| 11 | RT_ADD_C3 | 43 | RT_ADD_A2 |
| 12 | 1553_1D- | 44 | 1553_1C- |
| 13 | RT_ADD_B3 | 45 | RT_ADD_D0 |
| 14 | RT_ADD_C2 | 46 | 1553_2C+ |
| 15 | RT_ADD_C1 | 47 | RT_ADD_A3 |
| 16 | 1553_2A- | 48 | 1553_2C- |
| 17 | RT_ADD_B4 | 49 | RT_ADD_DP |
| 18 | 1553_2A+ | 50 | 1553_1A- |
| 19 | RT_ADD_C0 | 51 | RT_ADD_A4 |
| 20 | RT_ADD_CP | 52 | 1553_1A+ |
| 21 | RT_ADD_BP | 53 | $\overline{\text{CH.1RESET}}$ |
| 22 | 1553_2B- | 54 | CH. 1 TX INHIBIT |
| 23 | GND | 55 | NC (NO USER CONNECTION) |
| 24 | 1553_2B+ | 56 | $\overline{\text{CH.4RESET}}$ |
| 25 | 1553_2D+ | 57 | CH. 4 TX INHIBIT |
| 26 | RT_ADD_D4 | 58 | NC (NO USER CONNECTION) |
| 27 | $\overline{\text{CH.2RESET}}$ | 59 | NC (NO USER CONNECTION) |
| 28 | CH. 2 TX INHIBIT | 60 | NC (NO USER CONNECTION) |
| 29 | RT_ADD_AP | 61 | NC (NO USER CONNECTION) |
| 30 | $\overline{\text{CH.3RESET}}$ | 62 | NC (NO USER CONNECTION) |
| 31 | GND | 63 | NC (NO USER CONNECTION) |
| 32 | CH. 3 TX INHIBIT | 64 | NC (NO USER CONNECTION) |

## Front Panel I/O Connector

Models ***BU-65565B/F*** include provisions for a front-panel I/O connection via connector “J5”. This connector (Soldered on the board) is a 50-pin “SCSI Shielded Type” (Molex P/N 15-87-6050). Pin numbering designations for “J5” are as shown in Figure 2.

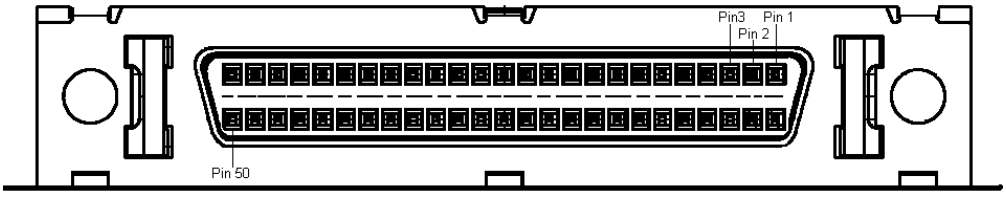



**Figure 2. 50-Pin “J5” I/O Connector**

**(viewed from Front panel of card)**

Models ***BU-65565B/F*** are shipped with “J5” mating connector and backshell as listed in Table 5.

**Table 5. Supplied I/O Connector Components for BU-65565B/F**

| Description | Manufacturer # | Quantities |
|---|---|---|
| 1.27mm (.050") Plug Assembly Amplimite | AMP # 749111-4 | 1 |
| KIT, UNASSEMBLED, 50 POSITION, STRAIGHT EXIT, UNKEYED, AMPLIMITE® .050 Series<br><br>ALTERNATE<br><br>SCSI-2 Backshell Modular Sub-Assembly | AMP # 749193-2<br><br>ALTERNATE<br><br>Molex # 15-87-0050 | 1 |

| Table 6. BU-65565B/F J5 Connector Pin Functions | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | RT_ADD_B4 | 26 | RT_ADD_C4 |
| 2 | RT_ADD_B1 | 27 | RT_ADD_C3 |
| 3 | 1553_2B- | 28 | RT_ADD_C0 |
| 4 | RT_ADD_BP | 29 | $\overline{\text{SSFLAG}}$/EXT_TRIG_C |
| 5 | 1553_2B+ | 30 | RT_ADD_B3 |
| 6 | RT_ADD_CP | 31 | RT_ADD_C2 |
| 7 | 1553_1B | 32 | RT_ADD_B0 |
| 8 | $\overline{\text{SSFLAG}}$/EXT_TRIG_D | 33 | RT_ADD_D2 |
| 9 | 1553_1B+ | 34 | RT_ADD_C1 |
| 10 | $\overline{\text{SSFLAG}}$/EXT_TRIG_B | 35 | GND |
| 11 | RT_ADD_D0 | 36 | GND |
| 12 | GND | 37 | RT_ADD_D3 |
| 13 | RT_ADD_B2 | 38 | RT_ADD_D4 |
| 14 | GND | 39 | RT_ADD_A1 |
| 15 | 1553_2C- | 40 | GND |
| 16 | RT_ADD_A2 | 41 | GND |
| 17 | 1553_2C+ | 42 | RT_ADD_AP |
| 18 | $\overline{\text{SSFLAG}}$/EXT_TRIG_A | 43 | RT_ADD_D1 |
| 19 | RT_ADD_A0 | 44 | RT_ADD_A4 |
| 20 | RT_ADD_A3 | 45 | 1553_1A- |
| 21 | 1553_1C- | 46 | 1553_1A+ |
| 22 | RT_ADD_DP | 47 | 1553_2A+ |
| 23 | 1553_1C+ | 48 | 1553_2A- |
| 24 | 1553_1D+ | 49 | 1553_2D+ |
| 25 | 1553_1D- | 50 | 1553_2D- |

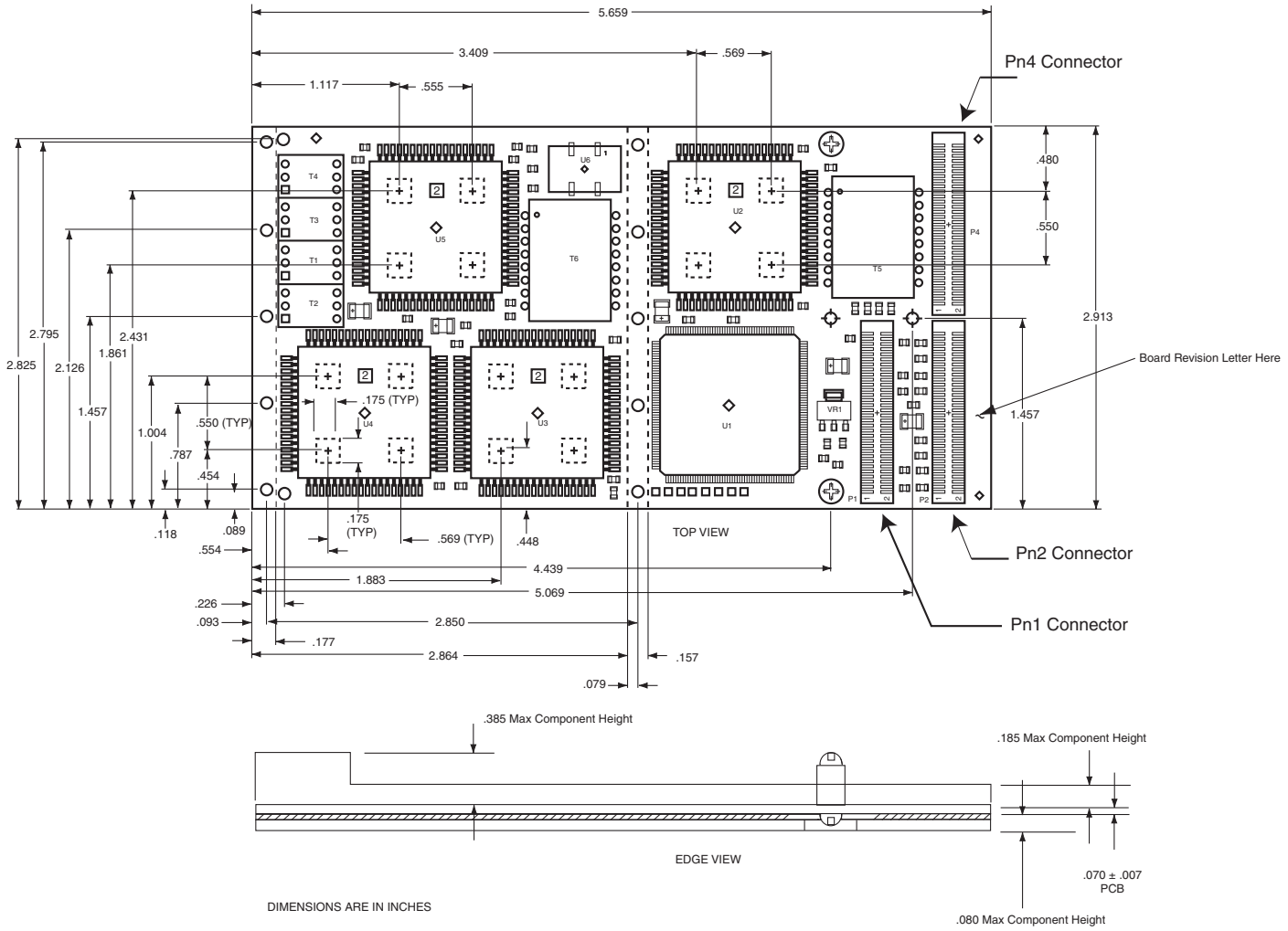



**Figure 3. BU-65565M/C Mechanical Outline**

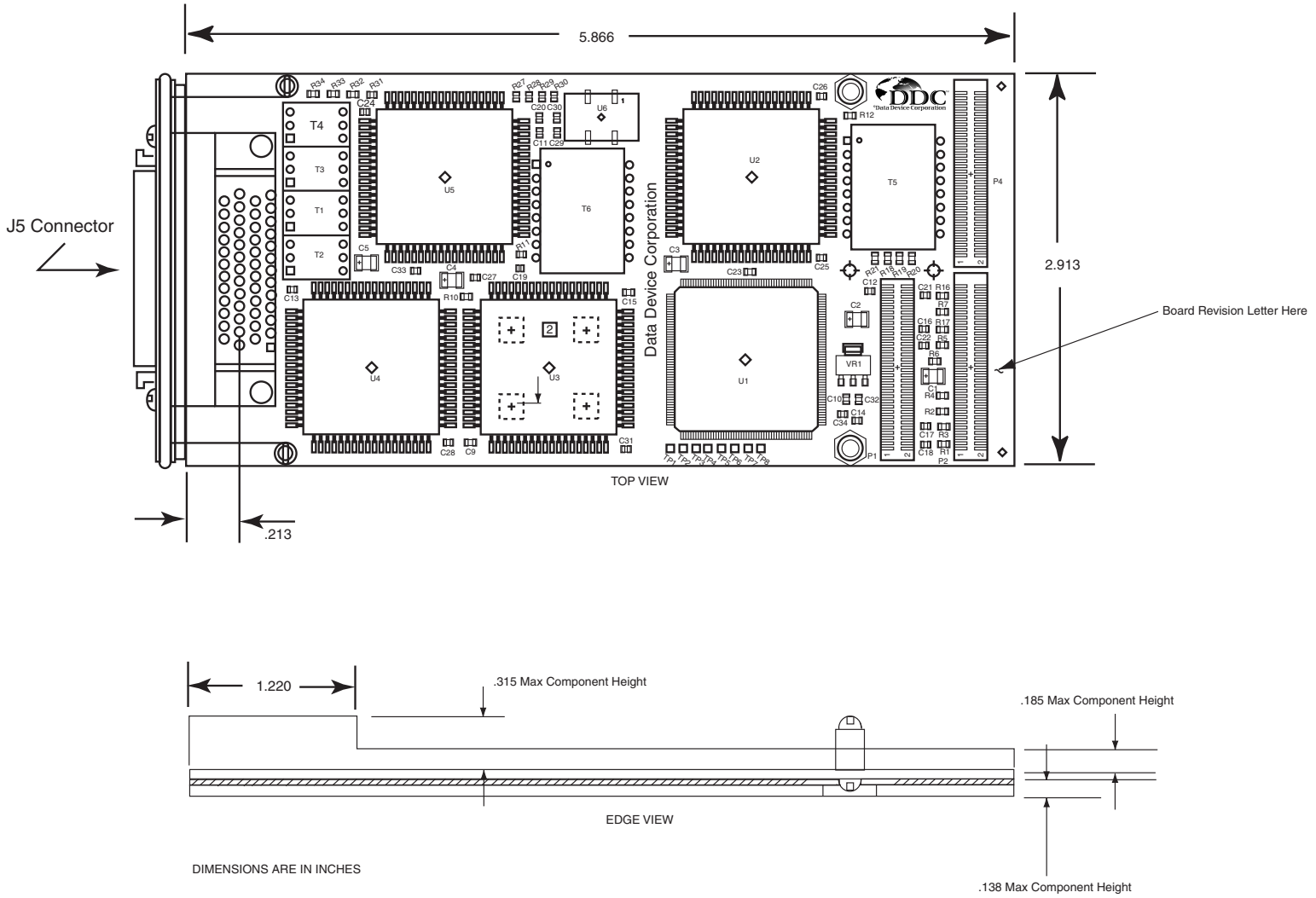



**Figure 4. BU-65565B/F Mechanical Outline**

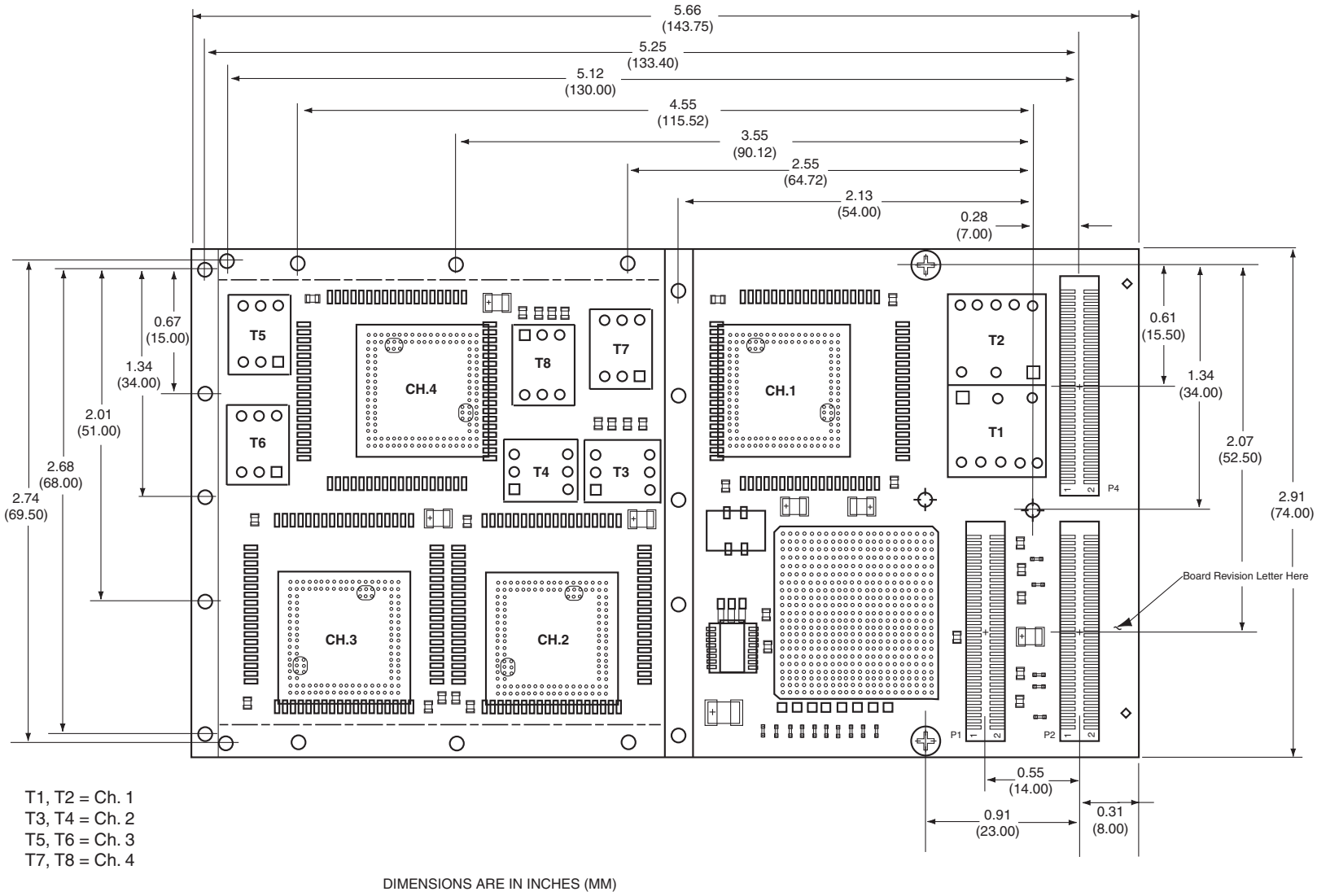



**Figure 5. BU-65566G Mechanical Outline**

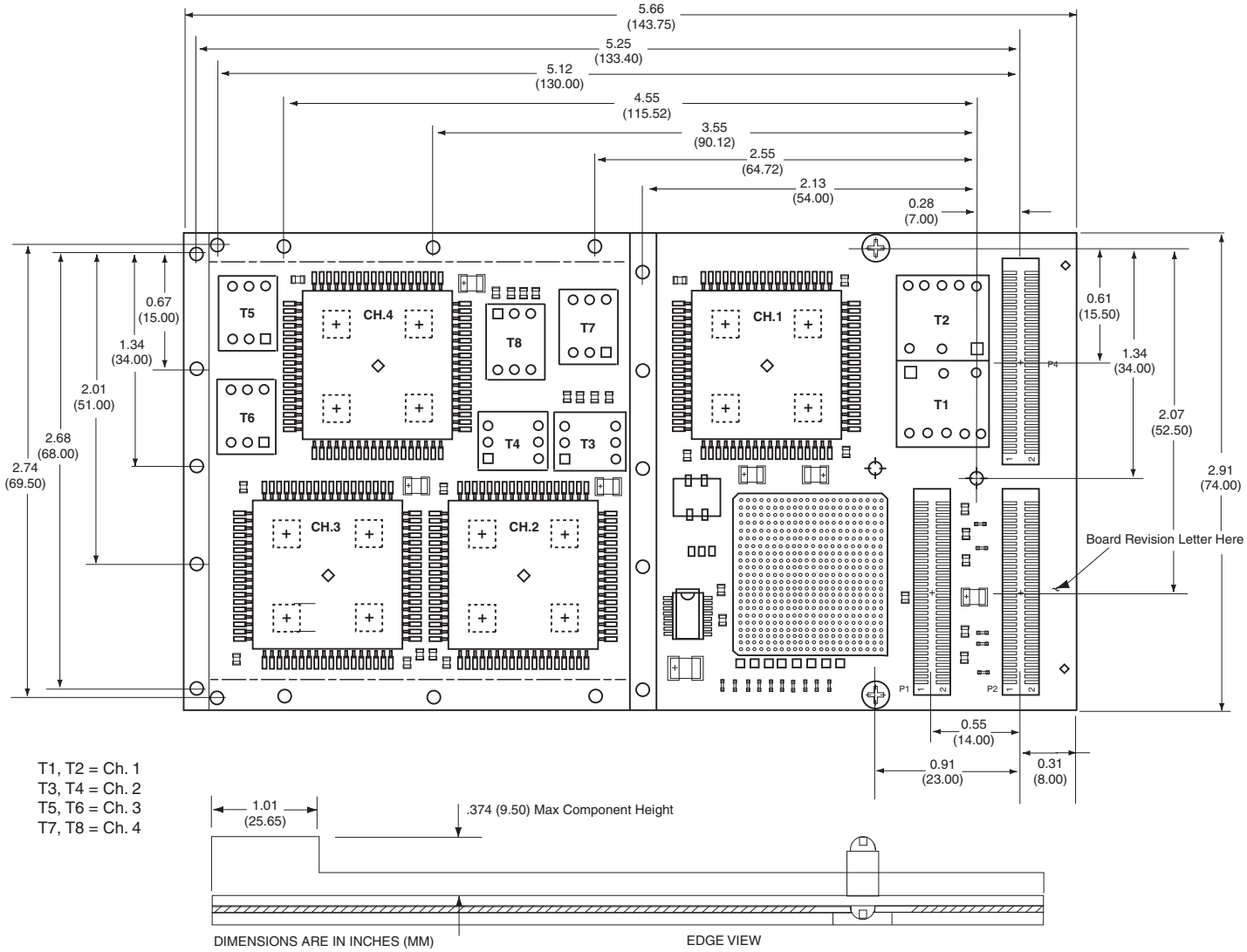



**Figure 6. BU-65566M Mechanical Outline**

# SOFTWARE INSTALLATION

## Overview



The card is supplied with Runtime Libraries (RTL) for Windows 9x/XP/2000, Windows NT, WindRiver's VxWorks®, and Linux.

The VxWorks RTL will be supplied as a single driver/library. This file can be linked to the VxWorks Kernel with a full rebuild of the system or it can operate as a module that is loaded and unloaded by the user.

Extracting the tar file using an appropriate utility performs the installation of the software. The library will be placed in the /lib/modules directory. A script file is provided with the software to handle dynamic loading of the module.

### *Run tar cvf acelib.tar*

This software is a runtime library that provides the user with a hardware abstraction layer for the DDC Enhanced Mini-ACE PCI series MIL-STD-1553 Card. This software layer includes routines that dramatically reduce software development time by providing a high level interface to the PCI card. Demo programs are included with the library to demonstrate the full capabilities of the card.

### *Windows 9x/2000/XP and Windows NT*

Installation of the software is handled completely by InstallShield. This install script will make all necessary registry modifications and will create the required directory structures.

## Enhanced Mini-ACE Software Library Compatibility

To use the enhanced properties of the ***BU-65565*** or ***BU-65566*** card, the 32-Bit Enhanced Mini-ACE Runtime Library series ***BU-69090*** libraries must be used. The libraries and requirements are described in Table 7.

**Table 7. Enhanced Mini-ACE Software**

| Part Number | Description | Requirements |
|---|---|---|
| ***BU-69090S0*** | 32-Bit Enhanced Mini-ACE Runtime Library and Device Drivers for Windows | Win 9x/NT/2000/XP<br>Pentium 450 MHz* or better CPU<br>2.5 Meg Disk Space |
| ***BU-69090S1*** | 32-Bit Enhanced Mini-ACE Runtime Library and Device Drivers for Linux | Linux |
| ***BU-69090S2*** | 32-Bit Enhanced Mini-ACE Runtime Library and Device Drivers for WindRiver VxWorks | WindRiver VxWorks 5.4 |
| ***BU-69090S5*** | 32-Bit Enhanced Mini-ACE Runtime Library and Device Drivers for Green Hills® Integrity® | Green Hills Integrity 4.x |

* Slower processors may be used, but limitations in data processing, host buffering and interrupt processing can exist.

## Software Driver Installation Overview



16-bit not supported

The Enhanced Mini-ACE C API library is supplied in three different distributions with drivers for Windows 9x/2000/XP Windows NT, Linux, and VxWorks. The ***BU-69090S2*** contains the drivers and Enhanced Mini-ACE C API library for WindRiver's VxWorks on a Motorola or Intel platform. The ***BU-69090S1*** contains the drivers and Enhanced Mini-ACE C API library for Linux distributions on an Intel platform.

Note that the 16-bit versions of the library and menu are not supported for this hardware.

The basic installation for Windows 9x/2000/XP is listed below. Refer to page 33 for Windows NT installation instructions.

Windows 9x/2000/XP will recognize new hardware and ask for a driver disk. Windows Device drivers are available in the root path of the Data Bus Software CD or on the "Device Driver Disk" if floppy disks were provided.

Insert DDC's Data Bus Software CD into the appropriate drive or, if using floppy disks, insert disk 1 into the appropriate drive, and run the setup.exe program.

When asked for the location of the device drivers, click on the CD-ROM drive letter if using the Data Bus Software CD, or on the floppy disk drive letter if using floppy disks.

Supply the necessary information during setup.

### DRIVER INSTALLATION – VXWORKS



*For detailed installation and use instructions in a VxWorks environment, refer to the readme.txt file included with the **BU-69090S2** software package.*

### DRIVER INSTALLATION – LINUX



*For detailed installation and use instructions in a Linux environment, refer to the install.txt and readme.txt files included with the **BU-69090S1** software package.*

## DRIVER INSTALLATION – WINDOWS 9X/2000/XP

If you have Windows 9x/2000/XP use the following installation instructions. If you are using Windows NT refer to page 33.

***Note:*** *User must have administrator rights when using Windows 2000/XP.*

1) Install the card as described in the **HARDWARE INSTALLATION** section.

2) Turn on the computer.

3) The **Found New Hardware Wizard** window will appear after boot-up:




**Figure 7. Found New Hardware**

4) Click on the **Next** button.




**Figure 8. Search for Driver**

5) Click on **Search for a suitable driver for my device**.

6) Click on the **Next** button.

7) Insert the Data Bus Software CD or, if using floppy disks, the “Device Drivers” disk into the appropriate drive.

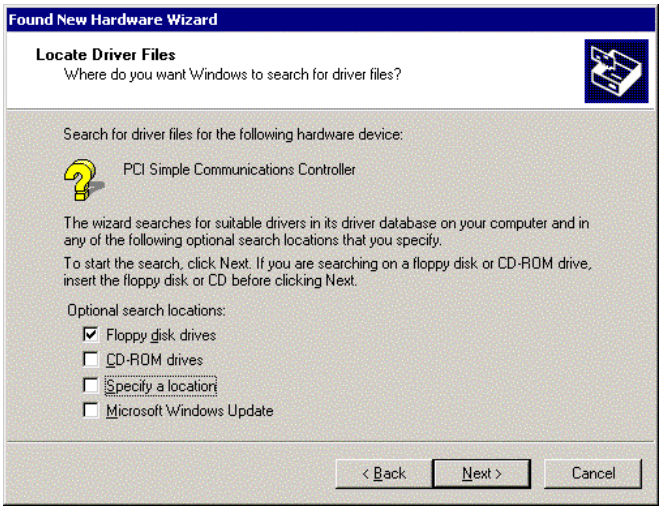



**Figure 9. Specify Driver Location**

8) Select the Checkbox for the appropriate drive type option, or select the **Specify a location** option to browse for the driver. Figure 9 illustrates a dialog box for a floppy disk drive type option.




**Figure 10. Found Driver**

9) Once the driver has been located, click on the **Next** button to install the device driver.




**Figure 11. Finished Install of Driver**

10) Click on the **Finish** button to confirm the installation.

11) Continue configuring the device by installing the Enhanced Mini-ACE Runtime Libraries. Refer to page 36.

## DRIVER INSTALLATION – WINDOWS NT

### *Installation*

The installation for the drivers and Enhanced Mini-ACE library is slightly different for Windows NT than it is for Windows 9x/2000/XP. Windows NT 4.0 will not automatically recognize the card even though it is a PnP PCI card. The drivers will be installed with the rest of the Enhanced Mini-ACE Runtime Library when the installation program is run from the CD.

***Note:*** *User must have administrator rights when using Windows NT.*

After the hardware is installed in the PC, install the driver and library by selecting the appropriate package and part number from the Data Bus CD. This installs the system files, the library files and the control panel DDC 1553 Card Manager. When the setup program is running, a dialog will be presented to the user to select the card type to install. If the computer has multiple cards, select one type now. Other cards may be installed later using the DDC 1553 Card Manager, which is located under Control Panel.

After the Setup program is finished, reboot the computer.

### *Final Installation Procedures*

The final steps in installing the new Enhanced Mini-ACE software require the device number selection for each installed card. This is performed via the control panel 1553 CARD MANAGER. Access the control panel and double click on the 1553 CARD MANAGER. When the applet is opened, the status of any installed cards is displayed.

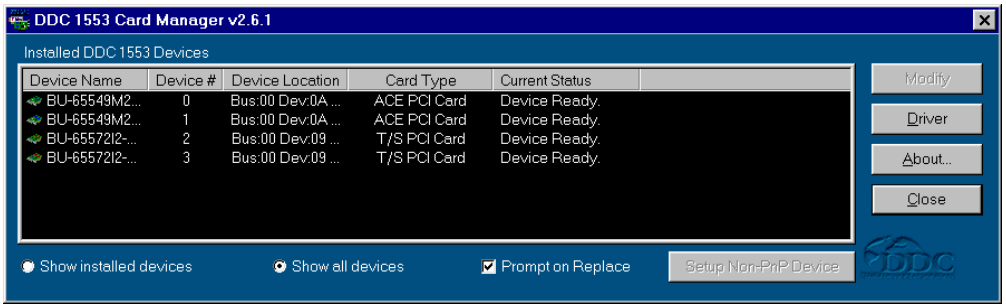



**Figure 12. 1553 Card Manager**

There are four buttons to the right side of the dialog box, and a status window in the center. The four buttons are **MODIFY**, **DRIVER**, **ABOUT**, **CLOSE**. The 3rd column, device location, consists of the PCI Bus, PCI Device and board channel number.

***Note:*** *The **Modify** button presents a dialog that allows the device number to be selected and modified. A different device number can be selected for each channel on the card. **A device must be selected before the Modify operation can be completed**.*

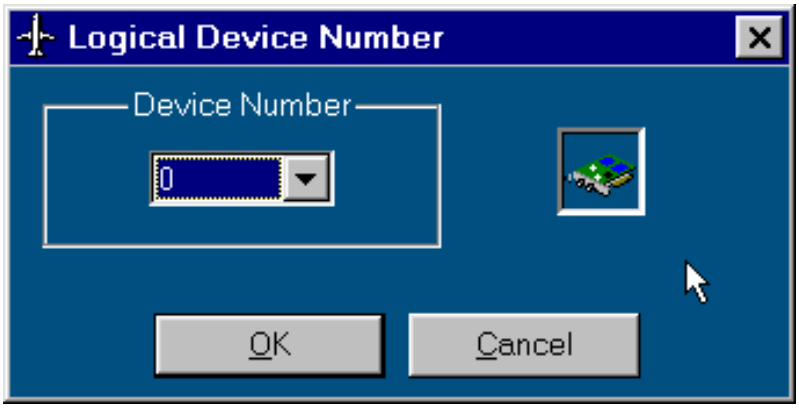



**Figure 13. 1553 Card Manager Modify Device Number**

The **Driver** button presents a dialog that shows the status of the installed drivers. Note that the version numbers are subject to change.

The ***BU-65565*** PMC card uses the Enhanced Mini-ACE PMC/PCI Card Device Driver.

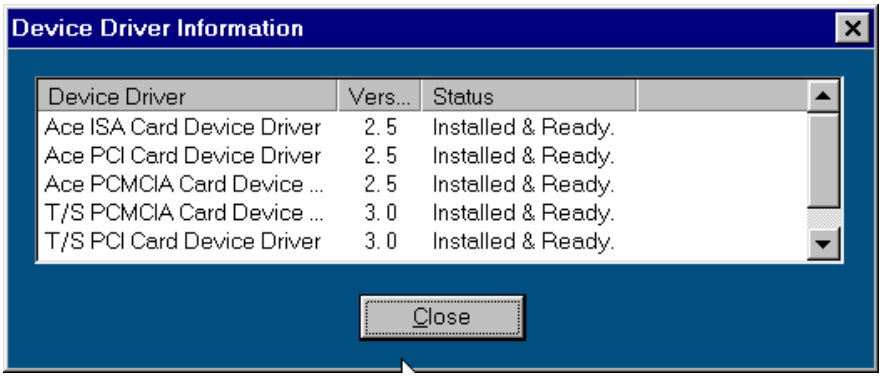



**Figure 14. 1553 Card Manager Driver**

The **About** button shows the version of the 1553 Card Manager.




**Figure 15. About DDC 1553 Card Manager**

Pressing the 1553 CARD MANAGER **Close** button will cause all changes to be saved, and the 1553 CARD MANAGER dialog to be closed. This action will return to the previous process.

## *BU-69090S0* Enhanced Mini-ACE Runtime Library (RTL)

### ENHANCED MINI-ACE RTL INSTALLATION - WINDOWS 9X/2000/XP

***CD-ROM Installation:***

- Insert the Data Bus Software CD into the appropriate drive and allow to autorun.
- Select the card type, then the software type.
- Click on the Install Software icon, then refer to the InstallShield wizard instructions below.

***Floppy Disk Installation:***

- From the ***BU-69090S0*** installation disk set, insert disk 1 into the appropriate drive.
- Run the setup.exe program, then refer to the InstallShield wizard instructions below.

InstallShield Wizard Instructions:

1) The InstallShield wizard will launch the Welcome screen. Click Next to continue.

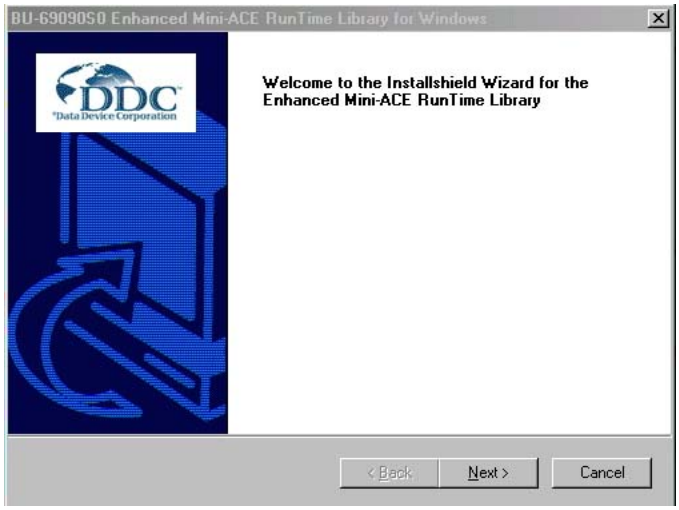



**Figure 16. Welcome Screen**

2) After reading the License Agreement click Yes to accept the terms, or No to disagree with them. (Choosing No will end the installation).

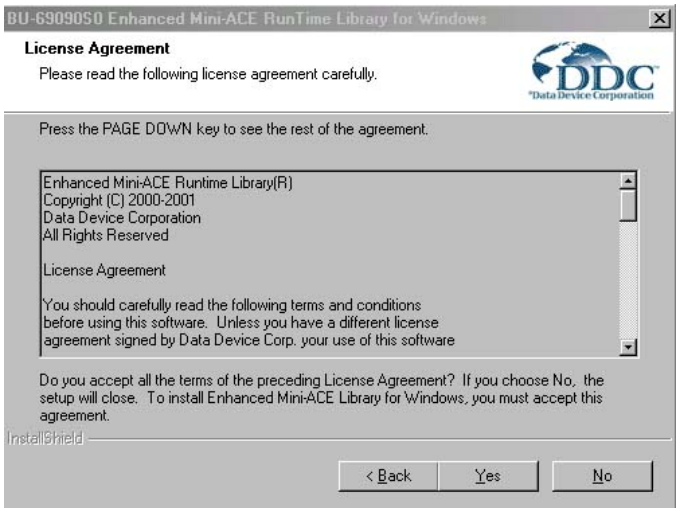



**Figure 17. Software License Agreement**

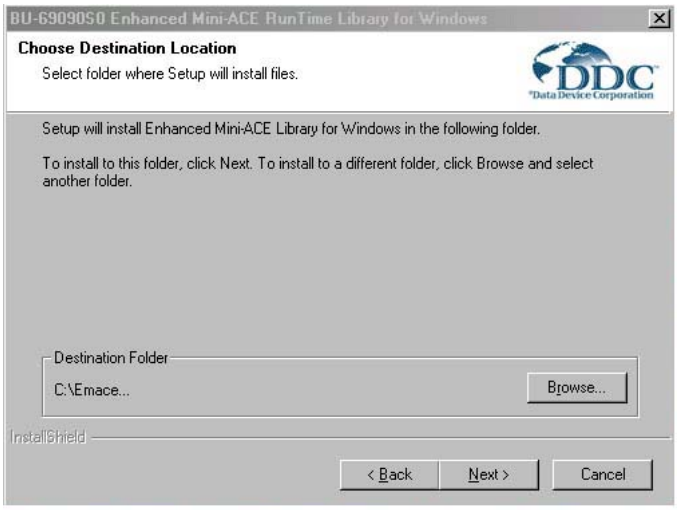



**Figure 18. Destination Folder**

3) Click Next to choose the default installation folder, or Browse to locate a custom folder destination.

4) Click Next to choose the default Program Folder location.

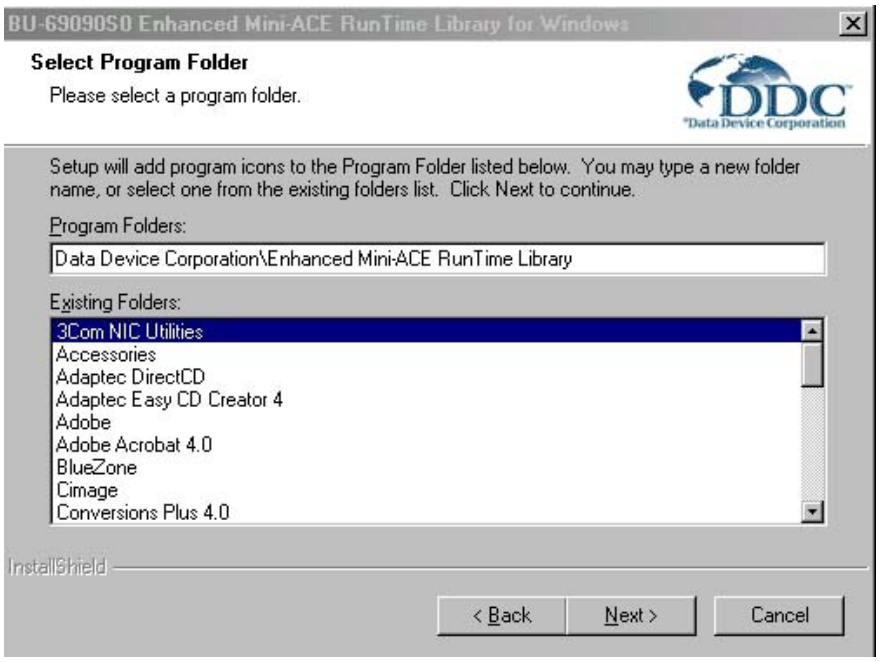



**Figure 19. Program Folder Location**

5) Select or clear the various components available for custom installation. The default is all checked.

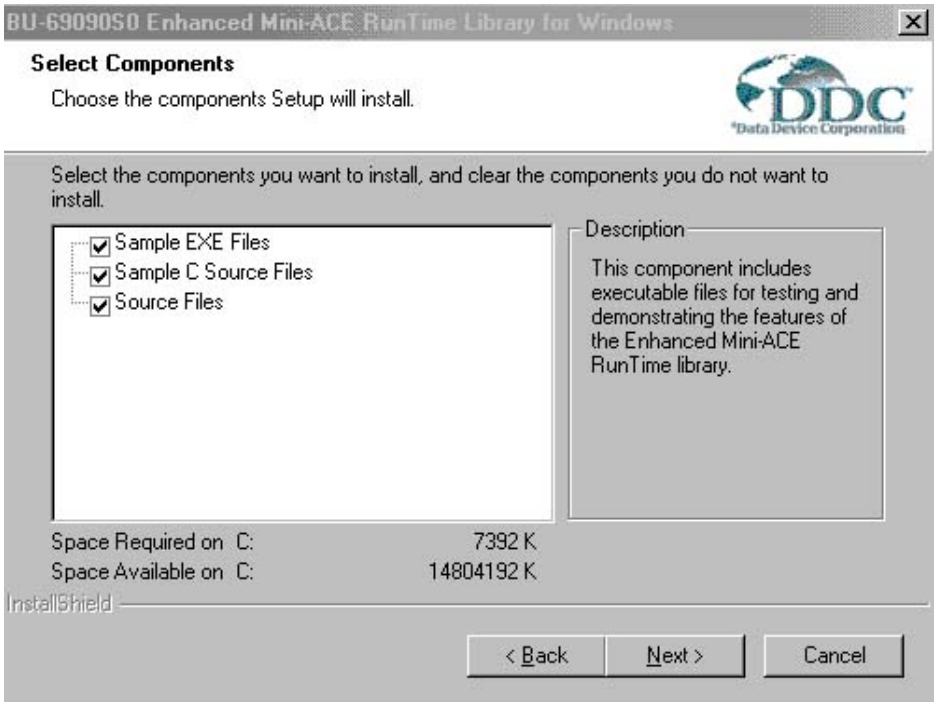



**Figure 20. Select Components**

6) Click Finish when installation is complete.

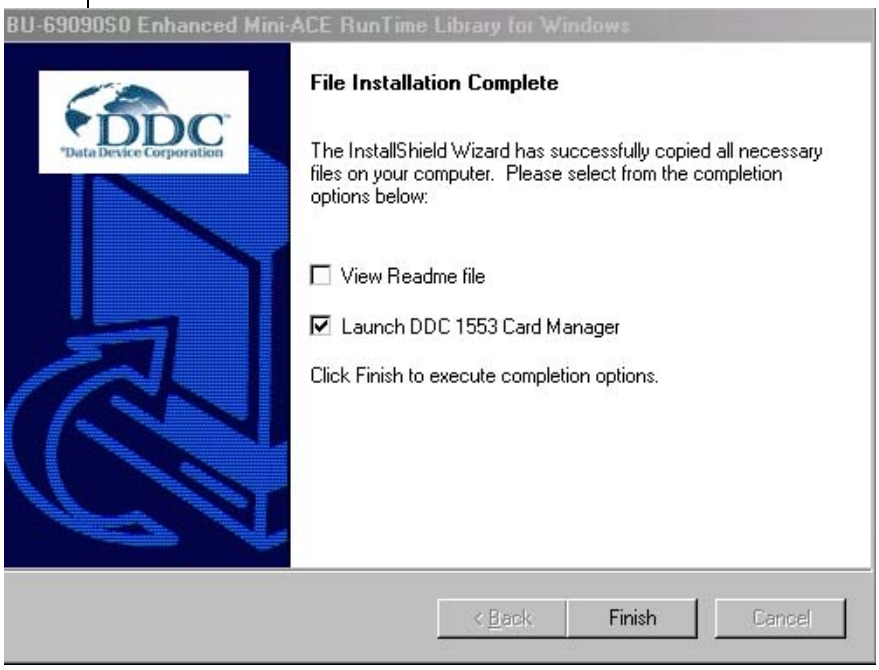



**Figure 21. Setup Complete**

7) The DDC 1553 CARD MANAGER will be launched. Double click the ***BU-65565/66*** card entry in the list to display the Logical Device Number dialog box. Any device that you wish to actively use must be assigned a logical device number. It is customary to assign device numbers starting with Device Number 0. Select a device number then click OK. Click Close when finished.

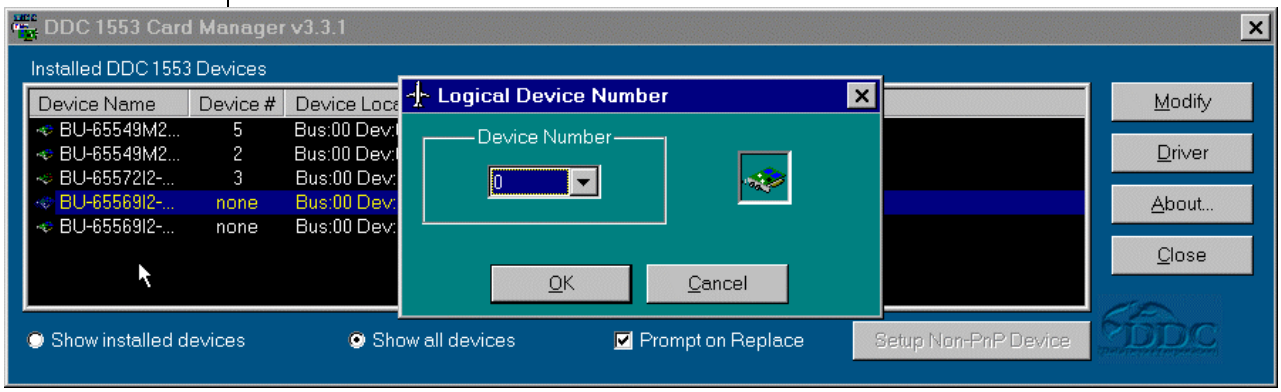



**Figure 22. Select Device Number**

8) Allow the installation program to reboot the computer.

## ENHANCED MINI-ACE RTL INSTALLATION - WINDOWS NT

If you are installing the Enhanced Mini-ACE RTL in a Windows NT environment refer to Page 33 for installation instructions.

## TESTING THE INSTALLATION

Once the hardware and software is installed, you can test the installation by running one of the sample programs that is shipped with the software. Follow the steps below to test the installation.

### *Testing the Enhanced Mini-ACE RTL*

From a Command prompt, change to the directory '<disk>:/ /EMACE <*version* >/exe'.

Run the Tester.exe program. This program requires a logical device number. A 0 indicates that the hardware is defined as 0 in the 1553 CARD MANAGER. This parameter will be entered as required per the installation. If you have more than one device installed in the system, then running Tester with the appropriate parameter, 0, 1, 2 or 3 should test each of the channels or cards individually.

If all elements of the test passed, the 1553 card/software is ready for use.

If one or more elements did not pass, refer to the section below on Troubleshooting the Installation or visit the DDC Web site at **www.ddc-web.com**.

## TROUBLESHOOTING THE INSTALLATION

Usually the installation will be successful, and the self-test will pass. There are, however, some situations that can cause problems during the installation. The most common are detailed below.

An error is returned when an attempt is made to run the SELFTEST or any of the other sample programs. This fault is almost always related to an incorrectly assigned device number. Be sure that a device number was correctly assigned through the DDC Card Manager.

If an error is encountered and the device number appears to be correctly assigned, check the operating system resources to verify that the card has been given a valid memory region by the operating system. Errors have sometimes been observed in Windows 2000 or Windows XP operating systems. The BIOS setting for a PnP operating system is sometimes set to YES, which can cause a problem. This BIOS option was originally created by manufacturers to distinguish between Windows NT and Windows 9x based operating systems. Since Windows 2000 and Windows XP are extensions of the Windows NT kernel this BIOS option must be set for NO under these operating systems. The operating system, as well as all hardware on your system, will still maintain PnP compatibility; it will not be necessary to manually configure resources for PnP cards.

# SOFTWARE OVERVIEW

## DLL Highlights

The DDC MIL-STD-1553 Win32 DLL is a collection of routines that will allow a Windows Win32 programmer access to the following features in a simple, easy to understand format:

### INITIALIZATION, INTERRUPTS AND BUFFERS

- Initialize the hardware
- Setup interrupt service routines
- Create and initialize data buffers

### BUS CONTROLLER OPERATION

- Create Messages
- Create Frames
- Setup BC operating parameters such as Retry, Error Response, and Frame Timing
- Run the BC Frame one, many or infinite times

### REMOTE TERMINAL OPERATION

- Initialize RT response data
- Set the RT response timing
- Specify RT Status Word contents

### MONITOR OPERATION

- Define monitor filter tables to disable the storage of unwanted messages
- Define the expected RT Response times that will be seen on the bus
- Create a monitor buffer and routines that control message storage.

## SELF TEST OPERATION

- Protocol tests
- Memory tests
- Register tests
- Loop tests
- Internal Built-in Self-Test (BIST)

# WRITING SOFTWARE FOR THE ENHANCED MINI-ACE

The source code for the sample programs listed in the previous section is available in the installation directories on the hard drive. These files provide a good platform for learning how to use the library and for troubleshooting compiler setup. It may sometimes be difficult to determine compile or link errors from coding errors. By attempting to recompile the sample files provided with the library, a baseline for coding is provided, and any errors would be due to compiler settings.

The library has the capability of creating a binary image file of the programmed Enhanced Mini-ACE. This feature enables a programmer to write and test code on a PC platform using one of DDC's Enhanced Mini-ACE cards, and then create an image of the programmed device that may be downloaded to the card. When the binary image is created the library will also create a header file. This file will include an entire mapping of the Enhanced Mini-ACE data structures, an initialization sequence that reflects the programmed steps, and a set of access routines that can be used to access and modify the device during use. This operation eliminates the need to carry the entire library and Enhanced Mini-ACE program source in the system program. When the system boots up, it will only have to download each of the Enhanced Mini-ACE's that are installed on the card, and they will be ready to run.

In this section, an introduction to programming the ***BU-65565*** Enhanced Mini-ACE's will be presented.

## Header Files

The header files are functionally divided into 4 main groups. The first group deals with the configuration and initialization of the hardware. This group includes all structures that are required for configuring the device, creation and tracking of memory buffers, and error reporting. The second group supports all structures and data types necessary for Bus Controller operation. The third group supports Remote Terminal, and finally the fourth group supports Monitor operations. Each of these groups will be discussed briefly below.

### CONFIGURATION

This group of files includes Config.h, Errordef.h, Stdemace.h, and Testop.h. Stdemace.h is the only header file that should be included within each of the files of program. This header file will include all other header files needed for programming. Errordef.h provides the error definitions that will be returned to a calling program and the prototype of a function that can be used to convert the returned error to a text string. Config.h provides the definition of the device configuration structure that will be used to identify the device and its present state. This file also

contains defined macros that will be used during the configuration and initialization of the card. Finally, the Testop.h header file contains defined macros and function prototypes that will be used to initiate self-test and to run specific tests for the Enhanced Mini-ACE devices on the card.

### BUS CONTROLLER

This group includes two header files. Bcop.h includes all of the defined macros and function prototypes for all functions that pertain to Bus Controller operation. This includes BC initialization, message and frame creation and tracking the operation of the BC once it is started. The macros included in this header file make message and frame creation simple by allowing the user to simply specify high level concepts of the operation. The functions from the library will actually perform all memory allocation and structure linking.

The Msgop.h file contains the defined macros and message structures that are needed to process messages. The message structures defined in this file will actually be used in both the RT and Monitor functionality.

### REMOTE TERMINAL

There are actually two header files in this category. The Rtop.h file contains the defined macros and function prototypes that are involved with Remote Terminal operation only. These operations include setting the RT address, allocating RT resources, and processing messages received by the Remote Terminal.

A second file included in this group is the Rtmop.h header file. This file provides the necessary operations for use with the Remote Terminal plus Monitor operation. This mode provides the extra capability to process RT operations and at the same time run as a monitor. All functionality that is needed for these operations is included in Rtmtop.h.

### MONITOR

The last group contains only one header file. The Mtop.h file defines all of the macros, structures and function prototypes needed to process Monitor operations.

## Card Functions

The basic steps that must be performed when writing a program for the card will be discussed in the next sections. Each of the prominent operations of the Enhanced Mini-ACE will be addressed. For details on programming your card refer to the ***BU-69090*** Enhanced Mini-ACE Runtime Library Software manual.

## BUS CONTROLLER



The Bus Controller operation of the Enhanced Mini-ACE is simplified by the use of the library. The Enhanced Mini-ACE is a highly autonomous BC that provides either the previous Mini-ACE operation or the new Enhanced Mini-ACE operation. This document will describe the new Enhanced Mini-ACE's operation.

The first step to programming the Enhanced Mini-ACE is to initialize the device. This is based on a Logical Device number that is associated to one of the installed Enhanced Mini-ACE's on the card. Based on the operating system, this device number may be assigned by the user or by the operating system. It is not important where the device number is assigned, just that it is directly associated with an Enhanced Mini-ACE and will be used to identify that device throughout the entire program.

Once the device is initialized with a statement similar to the following:

```
aceInitialize(DevNum,ACE_ACCESS_CARD,ACE_MODE_BC,0,0,0);
```

data blocks will have to be initialized. This operation reserves memory on the Enhanced Mini-ACE to be used for message data. These data blocks are not initially attached to any type of message or operation. They are only reserved for later use. This operation takes the following functionality:

```
aceBCDataBlkCreate(DevNum,DBLK1,32,wBuffer,32);
```

After defining all of the necessary data blocks, they must then be attached to messages. Each message is created using a macro that is defined in the Bcop.h file. The input parameters include the Device ID, a message ID, a data block that was previously defined, and the BC command elements RT address, T/R bit, Sub-Address, and Word Count. The basic structure of this command is as follows:

```
aceBCMsgCreateBCtoRT(
        DevNum,     /* Device number */
        MSG1,       /* Message ID to create */
        DBLK1,      /* Message will use this data block */
        5,          /* RT address */
        1,          /* RT subaddress */
        10,         /* Word count */
        0,          /* Default message timer */
        ACE_BCCTRL_CHL_A);      /* use chl A options */
```

The op-codes used to program the Enhanced Mini-ACE BC must be setup next. There are functions that are provided for this operation and they will take the following form:

```
aceBCOpCodeCreate(DevNum,OP1,ACE_OPCODE_XEQ,ACE_CNDTST_ALWAYS,M
SG1,0,0);
```

This op-code indicates that device DevNum will be creating a new opcode 'OP1', which is a XEQ (execute message) op-code, it should always be unconditionally executed, and should use message MSG1. This operation must be performed for each of the messages that will be processed by the BC.

After defining all BC messages, the messages must be attached to a BC frame. This is performed with the following function:

```
aceBCFrameCreate(DevNum,MJR1,ACE_FRAME_MAJOR,aOpCodes,1,0,0);
```

Each of the op-codes created in previous steps will be entered into a list array. In this case the array is named OpCodes. The frame will be attached to device DevNum, it will be designated MJR1, it is typed as a major frame, and it has a length of one op-code in the array.

After all minor and major frames are defined, the final step is to run the scenario. This is performed by executing the command:

```
aceBCStart(DevNum,MJR,90);
```

This command tells the Enhanced Mini-ACE identified as DevNum to run the major frame MJR1 ninety times.

Finally, the BC should be closed and all structures should be de-allocated with the command sequence:

```
aceBCStop(DevNum);

aceFree(DevNum);
```

Refer to the ***BU-69090*** Enhanced Mini-ACE Runtime Library Software Manual for more details on programming the card in BC mode.

## REMOTE TERMINAL

Programming the ***BU-65565*** for Remote Terminal operation is similar to previous versions of the DDC Terminal cards. The Enhanced Mini-ACE provides some new RT capability such as an interrupt that will be activated when the RT data stack is half full. This is in addition to the standard RT data stack full interrupt. This new interrupt allows for greater control of data handling.

The most basic setup for the Remote Terminal operation is described in this section. For more detail and examples, refer to the sample program source provided with the library.

The first step that must be performed when programming the RT is to initialize the Enhanced Mini-ACE. This is performed with the function:

```
aceInitialize(DevNum,ACE_ACCESS_CARD,ACE_MODE_RT,0,0,0);
```

Again, you will notice that the operation is associated directly to a logical device 'DevNum'. This same paradigm is used with every function for all operations of the Enhanced Mini-ACE. This function is initializing the card to operate in RT mode. After initializing the device, the RT address should be set to the desired address using the function:

```
aceRTSetAddress(DevNum,5);
```

This example sets the device 'DevNum' to RT address 5.

Many of the RT operations are made more efficient when run as an interrupt driven program instead of the polling operation. The interrupt conditions must be defined for the RT. This can be done with the following command:

```
aceSetWin32IrqConditions(DevNum,TRUE,ACE_IMR1_RT_CIRCBUF_ROVER

| ACE_IMR2_RT_CIRC_50P_ROVER | ACE_IMR1_TT_ROVER,

RTInterruptHandler);
```

This command enables the circular buffer rollover, the circular buffer 50% rollover and the time tag rollover. If any of these conditions occur, the user defined interrupt handler 'RTInterruptHandler' will be called. This is a pointer to a function that is provided by the user, and will be called for each interrupt that is enabled. It is up to this interrupt handler to decode the interrupt type and act accordingly.

The next step would be to define a data block and attach it to a circular buffer. Not all Remote Terminals will need a circular buffer for correct operation. In many cases single buffer or double buffer operation may be enough. The following sequence may be used to define a data block and attach it to a circular buffer:

```
aceRTDataBlkCreate(DevNum,DATA_BLOCK_ID_TX,ACE_RT_DBLK_C_128,
wBufferTx, DATA_BLOCK_SIZE);

aceRTDataBlkMapToSA(DevNum,DATA_BLOCK_ID_TX,1,

ACE_RT_MSGTYPE_TX,    ACE_RT_DBLK_CIRC_IRQ,TRUE);
```

These commands create a circular buffer of 128 words that will be used as a transmit buffer. The buffer is then mapped to Sub-Address 1. Finally, after the above steps have been performed, the RT is ready to be run. This is performed with the statement:

```
aceRTStart(DevNum);
```

This tells the device DevNum to start running as an RT.

When all RT operations are complete, the last operations that should be performed would be to halt the RT and to remove all allocated data structures. This is performed with the following commands:

```
aceRTStop(DevNum);

aceFree(DevNum);
```

Refer to the ***BU-69090*** Enhanced Mini-ACE Runtime Library Software Manual for more details on programming the card in RT mode.

## MONITOR

The least change between the Enhanced Mini-ACE and the Mini-ACE is in the Monitor operation. This mode of operation will be used to capture data that is presented on the 1553 bus by other devices and either analyze it or store it to a disk file. As with the previous discussions, the monitor mode uses the open, run, close paradigm. To start running the ***BU-65565*** in the monitor mode, the user application must first initialize the device with the statement:

```
aceInitialize(DevNum,ACE_ACCESS_CARD,ACE_MODE_MT,0,0,0);
```

Monitor initialization will automatically create and attach a circular buffer to the device. The size of this buffer will be determined by the size of the memory available on the Enhanced Mini-ACE device. There are no other setups required when running in the monitor mode, and at this point the monitor can be started using the following command:

```
aceMTStart(DevNum);
```

After running the monitor, the device should be stopped and closed as was performed for the BC and the RT. This is accomplished by the following command sequence:

```
aceMTStop(DevNum);

aceFree(DevNum);
```

These two statements halt the monitor and then release any allocated resources.

Refer to the ***BU-69090*** Enhanced Mini-ACE Runtime Library Software Manual for more details on programming the card in MT mode.

## COMPILING REQUIREMENTS

The Enhanced Mini-ACE library was written in ‘C’. It is intended to follow ANSI C standards as closely as possible. This should make it usable in any ‘C’ or ‘C++’ development environment. There are no special compiling or linking switches that must be set. If the operating and development environment is DOS or Windows based, the appropriate header, library and DLL files will be copied to the install

directories. If the development environment is VxWorks, the header files will be delivered as in the Windows version, but the DLL/library (library executable) will be located in the device driver module.

In all environments, the user application is required to include the header file STDEMACE.H. This file will in turn include all of the other required header files. The development project (make file) must include a reference to the library file. The EMACE_32.LIB file contains all of the references needed for the linking of the application.

### GENERATING PC INTERRUPTS

The card is capable of generating PC Interrupts. The Enhanced Mini-ACEs may issue interrupt requests over the PCI bus. PCI Interrupts are generated on the INTA# output signal to the PCI host. The interrupts from each of Enhanced Mini-ACE(s) are functionally OR'd together to provide a single interrupt. There are 4 user-defined interrupts and a general-purpose queue that can store information pertaining to an interrupt condition. This construction allows the host hardware to access information that was placed on the queue while servicing the interrupt.

### DIAGNOSTIC FUNCTIONS

The ***BU-65565/66*** incorporates the latest Enhanced Mini-ACE devices. These devices have the capability to perform a complete internal self-test upon being powered up. This test provides comprehensive testing of the internal protocol logic. A separate test verifies the operation of the Enhanced Mini-ACE's internal RAM. Since the self-tests are fully autonomous, they eliminate the need for the host to write and read stimulus and response vectors. All self-test functionality is provided in the Testop.h file and the associated functions of the library.

## Low Level Programming

For details on low-level programming without using the Runtime Library refer to the Enhanced Mini-ACE User's Guides.

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 23, 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.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Odd Parity | OpCode Field | | | | | 0 | 1 | 0 | 1 | 0 | Condition Code Field | | | | |

**Figure 23. BC Op Code Format**

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.

# APPENDIX A

The following tables describe the content and use of the PCI registers as they are defined by the ***BU-65565***. For a complete description of the generic use of these registers, refer to any PCI system architecture manual.

## BU-65565 PCI Register Definition

**Table 8. BU-65565 PCI Configuration Header**

| Address | Word Contents | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 31 | | | 24 | 23 | | | 16 | 15 | | | 8 | 7 | | | 0 |
| 00h | Device ID = 901, 902, 903, or 904h | | | | | | | | Vendor ID = 4DDCh | | | | | | | |
| 04h | Status Register | | | | | | | | Command Register | | | | | | | |
| 08h | Class Code = 078000 h | | | | | | | | | | | | Rev ID = 01 | | | |
| 0Ch | BIST (not implemented) | | | | Header Type 00h | | | | Latency Timer | | | | Cache Line Size | | | |
| 10h | Base Address Register 0 (for Enhanced Mini-ACE RAM) | | | | | | | | | | | | | | | |
| 14h | Base Address Register 1 (for Enhanced Mini-ACE Registers) | | | | | | | | | | | | | | | |
| 18h- 24h | Base Address Registers 2 through 5 (not used) 00000000h | | | | | | | | | | | | | | | |
| 28h | Card Bus CIS pointer (not used) 00000000h | | | | | | | | | | | | | | | |
| 2Ch | Subsystem Device and Subsystem Vendor ID Same as Configuration Register 0, Alias Reads to Configuration Register User configurable | | | | | | | | | | | | | | | |
| 30h | Expansion ROM Base Address (Not Used, bit 0 = 0) | | | | | | | | | | | | | | | |
| 34h-38h | Reserved | | | | | | | | | | | | | | | |
| 3Ch | Max Lat. 00h | | | | Min Gnt 00h | | | | Interrupt Pin 01h | | | | Interrupt Line R/W | | | |

**Vendor ID** is 4DDCh, which is DDC's vendor ID.

**Device ID** field is used to indicate the device being used.

**Device ID** = 901h (1 channel)

902h (2 channels)

903h (3 channels)

904h (4 channels)

**Status Register** will be 0280h after reset, indicating that the ***BU-65565*** is fast back-to-back capable and has medium device select timing.

**Base Address Registers** are used to identify Enhanced Mini-ACE memory space (BAR0) and Enhanced Mini-ACE register space (BAR1). BARs 2 through 5 are not used and will return all zeroes when read.

**BAR0** is used to define Enhanced Mini-ACE memory space.

When all ONEs are written to BAR0, the following values will be read back. Besides defining the address space window size and base, the lower 4 bits of BAR0 will read back as zeroes, indicating that these are Memory BARs. The memory can be mapped anywhere in the 32-bit memory space and the memory is NOT prefetchable.

**Table 9. BU-65565 BAR0 Readback Value (after all ONEs are written to BAR0)**

| # of Enhanced Mini-ACEs | BAR0 readback value |
|---|---|
| 1 | FFFE0000 (128k byte window) |
| 2 | FFFC0000 (256k byte window) |
| 3 | FFF80000 (512k byte window) |
| 4 | FFF80000 (512k byte window) |

The BAR0 address space is partitioned according to the following table.

**Table 10. BU-65565 (BAR0) Enhanced Mini-ACE Memory Map**

| ADDRESS OFFSET | DEFINITION |
|---|---|
| 00000 – 1FFFC | ACE 1 Memory Space |
| 20000 – 3FFFC | ACE 2 Memory Space (if available) |
| 40000 – 5FFFC | ACE 3 Memory Space (if available) |
| 60000 – 7FFFC | ACE 4 Memory Space (if available) |

Note that the Enhanced Mini-ACE memory is packed two Enhanced Mini-ACE memory words per PCI doubleword location. Therefore, a 32-bit PCI read of BAR0 offset 0h will return ACE 1 memory location 0 (lower 16 bits) and ACE 1 memory location 1 (upper 16 bits).

**BAR 1** is used to access Enhanced Mini-ACE registers and the Enhanced Mini-ACE -PCI interface (Global Activity) register. BAR1 will read back as FFFFF000h (4K bytes address window, minimum size recommended in PCI spec) when all ONEs are written to it. Since the lower 4 bits of BAR0 read back as zeroes, this indicates that these are Memory BARs. The memory can be mapped anywhere in the 32-bit memory space and the memory is NOT prefetchable.

**Table 11. BU-65565 (BAR1) ACE Control Registers – 4K byte Total Space**

| PCI ADDRESS OFFSET | NAME | DEFINITION /ACCESSIBILITY |
|---|---|---|
| 000-0FC | ACE1 | ACE 1 REGISTER SPACE |
| 100-1FC | ACE2 | ACE 2 Register Space (if ACE 2 present) |
| 200-2FC | ACE3 | ACE 3 Register Space (if ACE 3 present) |
| 300-3FC | ACE4 | ACE 4 Register Space (if ACE 4 present) |
| 800 | ACE-PCI IF | GLOBAL ACTIVITY |

Enhanced Mini-ACE REGISTER SPACE (ACE1 – ACE4): This address space allows access to Enhanced Mini-ACE registers. The lower four 256-byte segments of allocated address space allow access to a total of 64 Enhanced Mini-ACE registers per segment. Note that the Enhanced Mini-ACE registers are packed ONE Enhanced Mini-ACE register per PCI doubleword location. The internal Enhanced Mini-ACE registers are only 16-bits wide: the upper 16 bits of a Dword (32 bit) read will be all zeroes. Register access is on a 32-bit boundary (e. g., BAR1 + 000h = ACE1 REG 0, BAR1 + 004h = ACE1 REG 1, BAR1 + 008h = ACE1 REG 2, etc). All reads of Enhanced Mini-ACE registers are via DRR (Delayed Read Request) transactions on the PCI bus.

**Table 12. BU-65565 Global Activity Register (BAR1 + 800h)**

| **Global Activity Register (Read 800h)** | |
|---|---|
| **BIT** | **DESCRIPTION** |
| 31 (MSB) | RESERVED (0) |
| 30 | RESERVED (0) |
| 29 | RESERVED (0) |
| 28 | RESERVED (0) |
| 27 | RESERVED (0) |
| 26 | RESERVED (0) |
| 25 | NUMBER OF ACE'S PRESENT – BIT 1 (MSB) |
| 24 | NUMBER OF ACE'S PRESENT – BIT 0 (LSB) |
| 23 | RESERVED (0) |
| 22 | RESERVED (0) |
| 21 | RESERVED (0) |
| 20 | RESERVED (0) |
| 19 | ACE 4 INTERRUPT ACTIVE |
| 18 | ACE 3 INTERRUPT ACTIVE |
| 17 | ACE 2 INTERRUPT ACTIVE |
| 16 | ACE 1 INTERRUPT ACTIVE |
| 15 | RESERVED (0) |
| . | . |
| . | . |
| . | . |
| 0 (LSB) | RESERVED (0) |

NOTE: this register is directly readable and does not generate a DRR PCI transaction.

NUMBER OF Enhanced Mini-ACEs PRESENT: Bits 25 (MSB) and 24 (LSB) provide a binary representation of the number of ACE terminals presently installed and operating. The ***BU-65565*** card supports up to four ACE terminals and may be configured for 1, 2, 3, or 4 Enhanced Mini-ACE devices.

Enhanced Mini-ACE INTERRUPT ACTIVE: Bits 19 to 16 reflect the interrupt status of the Enhanced Mini-ACE's. All Enhanced Mini-ACE interrupts must be programmed for level operation. When an Enhanced Mini-ACE interrupt occurs, the corresponding Enhanced Mini-ACE INTERRUPT ACTIVE bit gets set to a ONE. Bits 19 to 16 are cleared when the respective Enhanced Mini-ACE clears its interrupt request line.

# APPENDIX B

The following tables describe the content and use of the PCI registers as they are defined by the ***BU-65566***. For a complete description of the generic use of these registers, refer to any PCI system architecture manual.

## BU-65566 PCI Register Definition

**Table 13. BU-65566 PCI Configuration Header**

| Address | Word Contents | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 31 | 24 | 23 | 16 | 15 | 8 | 7 | 0 |
| 00h | Device ID = 0E06h | | | | Vendor ID = 4DDCh | | | |
| 04h | Status Register | | | | Command Register | | | |
| 08h | Class Code = 078000 h | | | | | | Rev ID = 01 | |
| 0Ch | BIST (not implemented) | | Header Type 00h | | Latency Timer | | Cache Line Size | |
| 10h | Base Address Register 0 (for FPGA) | | | | | | | |
| 14h | Base Address Register 1 (for Enhanced Mini-ACE RAM) | | | | | | | |
| 18h | Base Address Register 2 (for Enhanced Mini-ACE Registers) | | | | | | | |
| 1Ch | Base Address Register 3 (Not Used) | | | | | | | |
| 20 - 24h | Base Address Register 4 (Not Used) | | | | | | | |
| 28h | Card Bus CIS pointer (Not Used)<br>00000000h | | | | | | | |
| 2Ch | Subsystem Device and Subsystem Vendor ID<br>Same as Configuration Register 0, Alias Reads to Configuration Register<br>User configurable | | | | | | | |
| 30h | Expansion ROM Base Address (Not Used, bit 0 = 0) | | | | | | | |
| 34h-38h | Reserved | | | | | | | |
| 3Ch | Max Lat.<br>00h | | Min Gnt<br>00h | | Interrupt Pin<br>01h | | Interrupt Line<br>R/W | |

**Vendor ID** is 4DDCh, which is DDC's vendor ID.

**Device ID** field is used to indicate the device being used.

**Device ID** = 0E06h

**Status Register** will be 0280h after reset, indicating that the ***BU-65566*** is fast back-to-back capable and has medium device select timing.

**Base Address Registers** are used to identify Enhanced Mini-ACE memory space (BAR1) and Enhanced Mini-ACE register space (BAR2). BAR0 is used for FPGA specific registers.

**BAR0** is used to define the control registers on the FPGA used in the construction of this card.

**Table 14. BU-65566 Base Address Register 0 (Memory Mapped, 4K bytes)**

| ADDRESS OFFSET | DEFINITION /ACCESSIBILITY |
|---|---|
| 0x000-0x0FF | QL5064 Control Registers |
| 0x100-0xFFF | Reserved |

The BU-65566 card will default to PCI interrupts disabled at power-up, which differs from the BU-65565 card series. PCI interrupts are disabled because of the QuickLogic® device used for the PCI interface on the card.

QuickLogic User Mask: This bit will enable/disable PCI interrupts. A value of 1 will enable the interrupt capability. The bit may be accessed (depending on 8,16, or 32 bit PCI transfers) at the following locations:

BAR0

8-bit PCI write: Bit 0 @ Offset 0x8F

16-bit PCI write: Bit 8 @ Offset 0x8E

32-bit PCI write: Bit 24 @ Offset 0x8C

NOTE: Only this bit should be changed in all of BAR0. All other bits should be left at their default values.

**BAR1** is used to define Enhanced Mini-ACE memory space.

When all ONEs are written to BAR1, the following values will be read back. Besides defining the address space window size and base, the lower 4 bits of BAR1 will read back as zeroes, indicating that these are Memory BARs. The memory can be mapped anywhere in the 32-bit memory space and the memory is NOT prefetchable.

**Table 15. BU-65566 BAR1 Readback Value (after all ONEs are written to BAR1)**

| # of Enhanced Mini-ACEs | BAR1 readback value |
|---|---|
| 1 | FFFE0000 (128k byte window) |
| 2 | FFFC0000 (256k byte window) |
| 3 | FFF80000 (512k byte window) |
| 4 | FFF80000 (512k byte window) |

| Table 16. BU-65566 Base Address Register 1 (Memory Mapped, 512K bytes) | |
|---|---|
| **ADDRESS OFFSET** | **DEFINITION /ACCESSIBILITY** |
| 0x00000-0x1FFFC | Channel 0 Memory Space |
| 0x20000-0x3FFFC | Channel 1 Memory Space |
| 0x40000-0x5FFFC | Channel 2 Memory Space |
| 0x60000-0x7FFFC | Channel 3 Memory Space |

**BAR 2** is used to access Enhanced Mini-ACE registers and the Enhanced Mini-ACE -PCI interface (Global Activity) register. BAR2 will read back as FFFFF000h (4K bytes address window, minimum size recommended in PCI spec) when all ONEs are written to it. Since the lower 4 bits of BAR1 read back as zeroes, this indicates that these are Memory BARs. The memory can be mapped anywhere in the 32-bit memory space and the memory is NOT prefetchable.

| Table 17. BU-65566 Base Address Register 2 (Memory Mapped, 4K bytes) | |
|---|---|
| **ADDRESS OFFSET** | **DEFINITION /ACCESSIBILITY** |
| 0x000-0x0FC | Channel 0 Register Space |
| 0x100-0x1FC | Channel 1 Register Space |
| 0x200-0x2FC | Channel 2 Register Space |
| 0x300-0x3FC | Channel 3 Register Space |
| 0x400-0x7FC | Reserved (P-prime compatible) |
| 0x800-0x803 | 800Reg |
| 0x804-0x807 | Interrupt Reg |
| 0x808-0x80B | Global Write Control |
| 0x900-0x9FC | Global Write Registers |
| 0xA00-0XA03 | Flash Address Register |
| 0xA04-0XA07 | Flash Control/Data In Register |
| 0xA08-0XA0B | Flash Control/Data Out Register |
| 0xA0C-0XA0F | FPGA Revision ID/Flash Unprotect Register |

Enhanced Mini-ACE REGISTER SPACE (ACE1 – ACE4): This address space allows access to Enhanced Mini-ACE registers. The lower four 256-byte segments of allocated address space allow access to a total of 64 Enhanced Mini-ACE registers per segment. Note that the Enhanced Mini-ACE registers are packed ONE Enhanced Mini-ACE register per PCI double word location. The internal Enhanced Mini-ACE registers are only 16-bits wide: the upper 16 bits of a Dword (32 bit) read will be all zeroes. Register access is on a 32-bit boundary (e. g., BAR2 + 000h = ACE1 REG 0, BAR2 + 004h = ACE1 REG 1, BAR2 + 008h = ACE1 REG 2, etc). All reads of Enhanced Mini-ACE registers are via DRR (Delayed Read Request) transactions on the PCI bus.

The 800Reg is a status register used to report information regarding interrupts, Enhanced Mini-ACE Auto Initialization, and card configuration.

**Table 18. BU-65566 Global Activity Register (Read 0x800)**

| BIT | DESCRIPTION |
|---|---|
| 31 (MSB) | RESERVED (0) |
| 30 | RESERVED (0) |
| 29 | RESERVED (0) |
| 28:24 | FIRST FLASH MEMORY BLOCK ADDRESS AVAILABLE TO USER FOR NON-VOLATILE STORAGE |
| 23 | RESERVED (0) |
| 22 | ENHANCED MINI-ACE AUTO INIT FUNCTION REACHED TERMINAL ADDRESS |
| 21 | ENHANCED MINI-ACE AUTO INIT FUNCTION IN PROGRESS |
| 20 | ENHANCED MINI-ACE AUTO INIT FUNCTION COMPLETED |
| 19 | CHANNEL 4 OF ENHANCED MINI-ACE WAS WRITTEN BY AUTO INIT FUNCTION |
| 18 | CHANNEL 3 OF ENHANCED MINI-ACE WAS WRITTEN BY AUTO INIT FUNCTION |
| 17 | CHANNEL 2 OF ENHANCED MINI-ACE WAS WRITTEN BY AUTO INIT FUNCTION |
| 16 | CHANNEL 1 OF ENHANCED MINI-ACE WAS WRITTEN BY AUTO INIT FUNCTION |
| 15 | RESERVED (0) |
| 14 | RESERVED (0) |
| 13 | RESERVED (0) |
| 12 | RESERVED (0) |
| 11 | CHANNEL 4 ENHANCED MINI-ACE INTERRUPT IS ASSERTED |
| 10 | CHANNEL 3 ENHANCED MINI-ACE INTERRUPT IS ASSERTED |
| 9 | CHANNEL 2 ENHANCED MINI-ACE INTERRUPT IS ASSERTED |
| 8 | CHANNEL 1 ENHANCED MINI-ACE INTERRUPT IS ASSERTED |
| 7 | RESERVED (0) |
| 6 | RESERVED (0) |
| 5 | RESERVED (0) |
| 4 | CARD JUMPER INPUT (NOT USED) |
| 3 | FLASH MEMORY INSTALLED ON CARD |
| 2 | CARD MODEL |
| 1:0 (LSB) | NUMBER OF 1553 CHANNELS |

First Flash Memory Block Address Available to User for Non-Volatile Storage: Bits 28 to 24 indicate the first available memory location in the flash memory that is available to the user for non-volatile user defined storage.

Enhanced Mini-ACE Auto Init Function Reached Terminal Address: Bit 22 indicates if the Auto Init Function has reached the terminal address. If the terminal address has been incorrectly reached then this bit will contain a 1. If this bit contains a 0 then the Auto Init function has not reached the terminal address.

Enhanced Mini-ACE Auto Init Function In Progress: Bit 21 indicates if the Auto Init Function of the card is currently in process or not. If the Auto Init Function is currently in process then this bit will contain a 1.

Enhanced Mini-ACE Auto Init Function Completed: Bit 20 will contain a 1 when the Auto Init Function of the card has completed. If this function does not complete or is not yet completed then this location will contain a 0.

Channel x of Enhanced Mini-ACE was Written by Auto Init Function: Bits 19 to 16 will indicate if a specific channel on your Enhanced Mini-ACE card was written to by the Auto Init Function. If the Auto Init Function wrote to none of the channels, then these bits will contain 0's.

Channel x Enhanced Mini-ACE Interrupt is Asserted: Bits 11 to 8 will indicate if an interrupt on a specific Enhanced Mini-ACE channel is currently asserted. If the interrupt is not asserted on any channel then these bits will contain 0's.

Flash Memory Installed On Card: Bit 3 will indicate if your card has the Flash Memory on board or not. If the Flash Memory is on your card then this bit will contain a 1.

Card Model: Bit 2 will indicate the specific card model that you have. A 1 in this location indicates a BU-65566M model card (ceramic devices installed). A 0 in this location indicates a BU-65566G model card (BGA devices installed).

The Interrupt Register provides registered versions of the Enhanced Mini-ACE interrupt signals. These bits are set when the Enhanced Mini-ACE interrupt is asserted. The bits are cleared when the Enhanced Mini-ACE interrupt is disserted or a read of this register address is performed. A read of this register also results in the removal of the interrupt form the PCI bus. Following a read of this register address, any asserted Enhanced Mini-ACE interrupts still need to be cleared even though the PCI interrupt has been removed.

**Table 19. Interrupt Register (Read 0x804)**

| BIT | DESCRIPTION |
|---|---|
| (MSB) 31:4 | Reserved (0) |
| 3 | Interrupt Occurred on Channel 4 |
| 2 | Interrupt Occurred on Channel 3 |
| 1 | Interrupt Occurred on Channel 2 |
| 0 (LSB) | Interrupt Occurred on Channel 1 |

Interrupt Occurred on Channel x: Bits 3 to 0 of this register will indicate the channel that an interrupt occurred on. If no interrupts occurred then all these bits will contain 0's.

The Global Write Control Register is a control register used to determine which channels on your card will be written to during a global write. A global write is defined as a write to one or more Enhanced Mini-ACE register in parallel. This advanced capability allows for any Enhanced Mini-ACE register to be written to in parallel on multiple channels.

**Table 20. Global Write Control Register (Read/Write 0x808)**

| BIT | DESCRIPTION |
|---|---|
| (MSB) 31:16 | Reserved (0) |
| 15:4 | Not Used (Will read back as written) |
| 3 | Channel 4 Will be Written by a Global Write |
| 2 | Channel 3 Will be Written by a Global Write |
| 1 | Channel 2 Will be Written by a Global Write |
| 0 (LSB) | Channel 1 Will be Written by a Global Write |

Channel x Will be Written by a Global Write: Bits 3 to 0 of this register will control which channels will be written during a global write.

The Global Write Registers address space (0x900-0x9FC) is utilized when one wishes to write to an Enhanced Mini-ACE register location in multiple channels at the same time. The Global Write Control Register that is described above will indicate which channels will be written to. This address space directly corresponds to the Channel 0 Register Space (0x000-0x0FC), the Channel 1 Register Space (0x100-0x1FC), the Channel 2 Register Space (0x200-0x2FC), and the Channel 3 Register Space (0x300-0x3FC).

The Flash Address Register (0xA00-0xA03) is used to specify which Flash address will be read, written, or erased.

**Table 21. Flash Address Register (Read/Write 0xA00)**

| BIT | DESCRIPTION |
|---|---|
| (MSb) 31:19 | Reserved (0) |
| 18:0 (LSb) | Flash address to be read, written, or erased |

Flash address to be read, written, or erased: Bits 18 to 0 contain the Flash address memory location.

The flash memory used on the card has the following address memory map:

| Sector | Size (Kwords) | Address Range |
|---|---|---|
| S0 | 32 | 00000h-07FFFh |
| S1 | 32 | 08000h-0FFFFh |
| S2 | 32 | 10000h-17FFFh |
| S3 | 32 | 18000h-1FFFFh |
| S4 | 32 | 20000h-27FFFh |
| S5 | 32 | 28000h-2FFFFh |
| S6 | 32 | 30000h-37FFFh |
| S7 | 32 | 38000h-3FFFFh |
| S8 | 32 | 40000h-47FFFh |
| S9 | 32 | 48000h-4FFFFh |
| S10 | 32 | 50000h-57FFFh |
| S11 | 32 | 58000h-5FFFFh |
| S12 | 32 | 60000h-67FFFh |
| S13 | 32 | 68000h-6FFFFh |
| S14 | 32 | 70000h-77FFFh |
| S15 | 16 | 78000h-7BFFFh |
| S16 | 4 | 7C000h-7CFFFh |
| S17 | 4 | 7D000h-7DFFFh |
| S18 | 8 | 7E000h-7FFFFh |

The Flash Control/Data In Register (0xA04) is a read/write control register used to initiate flash transfers, specify what type of flash transfer to perform (read, write, or erase), and specify the data word to be written during the write operations.

**Table 22. Flash Control/Data In Register (Read/Write 0xA04)**

| BIT | DESCRIPTION |
|---|---|
| (MSB) 31:20 | Reserved (0) |
| 19 | Perform Flash Operation |
| 18:16 | Flash OpCode |
| 15:0 (LSB) | Flash Data In |

Perform Flash Operation: Bit 19 instructs the card's FPGA to perform the specified flash operation when set to a value of 1. Following the completion of the flash instruction, this bit must be cleared to a 0 before another transaction can be requested.

Flash OpCode: Bits 18 to 16 specify which type of flash operation to perform. A value of 000 will instruct the card to read the flash location, a value of 001 will instruct the card to write the flash location, a value of 010 will instruct the card to erase a sector. If any other value is input to this field then a flash error code will be returned to the user.

Flash Data In: Bits 15 to 0 specify the data word to be written to the flash memory location during the write operation.

Note:

- Any attempt to set a bit in the flash that has already been cleared by a previous write will result in a Flash Write Failure.
- Flash memory locations must be erased (returned to 0xFFFF) before being re-written.
- The flash memory is erased on sector boundaries and a sector erase operation will erase the entire sector that the specified flash address falls in.

The Flash Control/Data Out Register (0xA08) is a read only status register providing information regarding flash transfers.

**Table 23. Flash Control/Data Out Register (Read/Write 0xA08)**

| BIT | DESCRIPTION |
|---|---|
| (MSB) 31:20 | Reserved (0) |
| 19 | Flash Operation Done |
| 18:16 | Flash Error Code |
| 15:0 (LSB) | Flash Data Out |

Flash Operation Done: When bit 19 contains a 1 then this specifies that the requested flash operation has been completed. Following desertion of Perform Flash Operation in the Flash Control/Data In Register, this bit should be polled to make sure that it has been cleared before requesting another flash operation.

Flash Error Code: Bits 18 to 16 contain the errors returned from the flash operation. The following error codes are defined:

**Table 24. Flash Error Code**

| Error Code | Description |
|---|---|
| 000 | Success |
| 010 | Invalid Flash OpCode |
| 011 | Flash Write Failure |
| 100 | Flash Sector Erase Failure |
| 101 | Protected Address Used for Auto Init Function |

Flash Data Out: Bits 15 to 0 specify the data word read during the read operation.

The FPGA Revision ID/Flash Unprotect Register (0xA0C) is a read/write status register used to provide the host software with the revision ID of the FPGA on the card. This register is also a control register used to unprotect the protected portion of flash memory. Following an Auto Initialization function, the flash memory locations that store the Auto Init data are write/erase protected by the card's FPGA. A user must set the Flash Unprotect bit if a modification to the existing Auto Init data is desired.

**Table 25. FPGA Revision ID/Flash Unprotect Register (Read/Write 0xA0C)**

| BIT | DESCRIPTION |
|---|---|
| (MSB) 31:16 | FPGA Revision ID (ex. Rev. 2.3 = 0x0203) (READ ONLY) |
| 15:1 | Reserved (0) |
| 0 (LSB) | Unprotect |

FPGA Revision ID: Bits 31 to 16 contain the card's FPGA version.

Unprotect: If this bit is a 1 then the flash memory locations storing Auto-Init data are unprotected and can be written or erased. If this bit is a 0 then the flash memory locations storing Auto-Init data are protected and cannot be written or erased.

# INDEX


***B***

binary image ..... 44
Bus Controller (BC) ..... 5, 42, 45, 46
  programming ..... 46

***C***

card functions
  Bus Controller (BC) ..... 46
  Monitor (MT) ..... 49
  Remote Terminal (RT) ..... 47
compiling requirements ..... 49
  Stdemace.h ..... 50
configuration
  hardware ..... 7

***D***

DLL ..... 42
driver/library ..... 27

***E***

electrical static discharge ..... vii
ESD ..... *See* electrical static discharge

***H***

hardware installation ..... 4
Header Files ..... 44
  Bus Controller Group ..... 45
  Configuration Group ..... 44
  Monitor Group ..... 45
  Remote Terminal Group ..... 45
headquarters ..... 3

***I***

icon
  caution ..... vi
  disk ..... vii
  idea/tip ..... vi
  note ..... vi
  reference ..... vi
  warning ..... vi
installation
  hardware ..... 4
  software ..... 29
    driver ..... 29, 30
  testing ..... 40
  troubleshooting ..... 41

***J***

jumper blocks ..... 4

***M***

Monitor (MT)
  programming ..... 49

***P***

PCI Register definitions ..... 52, 56
Plug-and-Play configuration ..... 7

***R***

receiving ..... 42
Remote Terminal (RT)
  programming ..... 47

***S***

self test ..... 50
software
  driver/library ..... 27
software installation
  Runtime libraries ..... 27

***T***

TAR file ..... 27
technical support ..... 3
testing
  installation ..... 40
transmitting ..... 42

***W***

Web site ..... 3

**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