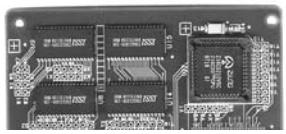


COMPATIBLE WITH BUS-65518

MIL-STD-1553 SIMULATOR AND TESTER CARD PCI INTERFACE

DESCRIPTION

DDC's BUS-65547 is a versatile, 32-bit PCI Bus circuit card designed for the test and simulation of MIL-STD-1553 systems. It provides full, intelligent interfacing between two serial dual redundant MIL-STD-1553 data buses and the PC. The BUS-65547 is based on the popular IDEA MKIII board and is divided into two identical halves which operate concurrently. Each half may operate as a Bus Controller (BC)/Remote Terminal (RT) or a Monitor (MT). User-friendly software controls each half of the BUS-65547 to simultaneously simulate a BC and multiple RTs, or a selectable, triggerable MT. The BUS-65547 may be either direct or transformer coupled to an external 1553 bus for use with user-provided 1553 devices.

The BC and RT evaluate each 1553 message in real time, and determine if any format errors have occurred. Separate registers are maintained for the last command word and the last status word of each emulated RT. In addition, the user may inject errors into any 1553 message issued by the emulated BC or RT. These errors include word count, bit count, zero crossing, parity, and, in the case of emulated RTs, response errors.

The intelligent MT captures the 1553 bus traffic. The user can define when MT operation is to begin and which messages (based on the RT address, T/$\overline{R}$ bit, and subaddress) are to be captured. Monitored information is displayed on a message-by-message basis.

32-bit Windows® 95/98 or NT menus and libraries are provided with the card. Support is included for Real Time extensions (RTX) by VenturCom for Windows NT.

FEATURES

- *Simulation and Test of MIL-STD-1553 Systems*
- *Plug and Play PCI Target*
- *512 Bytes of Shared RAM*
- *Simultaneous Emulation of BC, 31 RTs, and MT*
- *Selective Message Monitor*
- *Most Comprehensive Error Injection and Detection Capability in the Industry*
- *32-Bit Time Tag*
- *User-Friendly Menu Software for Windows 95/98 and NT*
- *Runtime Libraries for Windows*

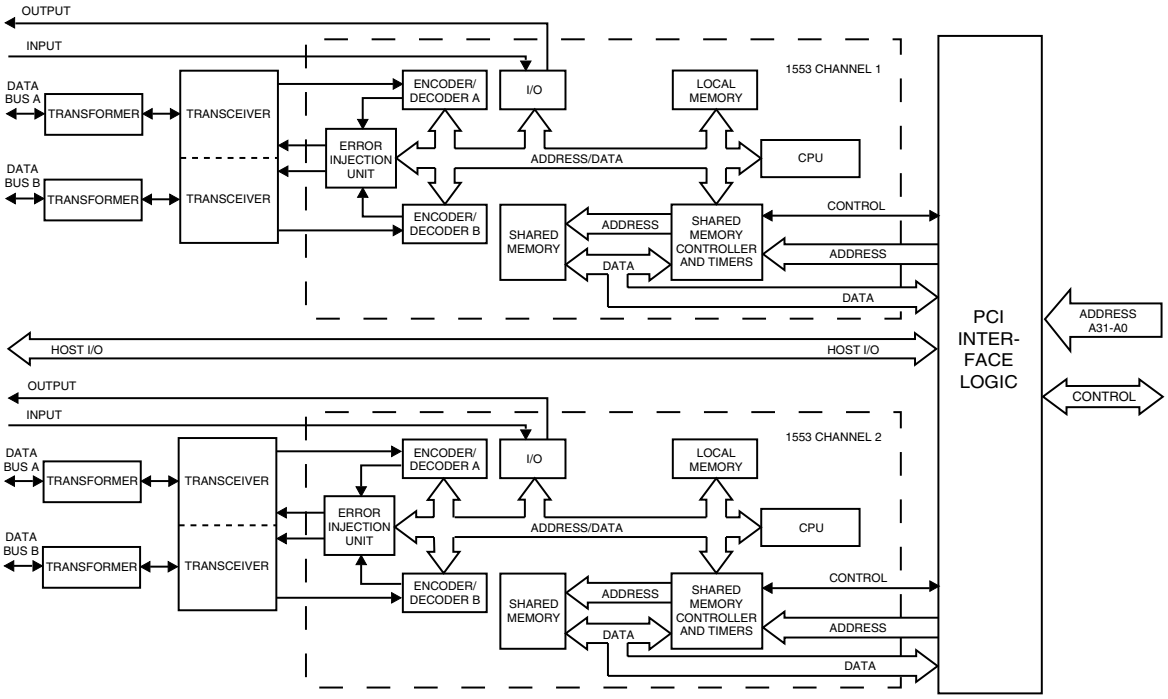


FIGURE 1. BUS-65547 BLOCK DIAGRAM

TABLE 1. BUS-65547 REQUIREMENTS AND CAPABILITIES
HARDWARE REQUIREMENTS Pentium Class computer with PCI Slot
SOFTWARE REQUIREMENTS Windows 95/98 or NT DDC's BUS-65547 Software diskettes
AVAILABLE OPERATIONS - Tests and simulates MIL-STD-1553 BC and up to 31 RTs. - Provides independent concurrent MT mode operation. - Error injection and detection capabilities.

TABLE 2. BUS-65547 HARDWARE SPECIFICATIONS			
PARAMETER	**UNITS**	**VALUE**	
POWER SUPPLIES		**Min**	**Max**
+5 V Supply Current Drain	mA	800	1000
+12 V Supply Current Drain	mA	30	400*
-12 V Supply Current Drain	mA	150	180
TEMPERATURE RANGE			
Operating (Case)	°C	0 to 50	
Storage	°C	-65 to 150	
PHYSICAL CHARACTERISTICS			
¾ size card; requires one 32-bit PCI slot.	in (mm)	12.3 x 4.2 (312.4 x 106.7)	

TABLE 3. I/O CHARACTERISTICS	
SIGNAL	**TYPE**
OUTPUT	
OUT(3-0)-1 - Channel 1 OUT(3-0)-2 - Channel 2	High 2.40 V min at -8 mA Low 0.55 V max at 64 mA
INPUT	
IN(3-0)-1 - Channel 1 IN(3-0)-2 - Channel 2	High 3.2 V min Low 1.0 V max
TRANSCEIVER HIO(3-0) HOST I/O	OUTPUT High Open collector with 1KΩ pullup to 5 V Low 0.55 V max at 64 mA INPUT High 3.2 V min Low 1.0 V max
MISCELLANEOUS **Vcc**	1KΩ pullup to 5 V

*(100% Duty Cycle at MAX Transmit Amplitude [28 V_{PP}])

GENERAL

The BUS-65547 (IDEA Card) is configured as two identical halves. Each half can simulate a MIL-STD-1553 Bus Controller (BC), multiple (up to 31) Remote Terminals (RTs), or an intelligent Bus Monitor (MT). The BUS-65547 may be either direct or transformer coupled to a dual redundant external 1553 bus, for operation with user-provided 1553 devices.

Each half of the BUS-65547's operation as BC/RT, or MT is implemented through the use of an embedded processor. The processors communicate with the PC via independent shared memories residing on the BUS-65547 card and through a common interrupt request line on the PCI backplane.

The 128K x 16 of static RAM for each half of the BUS-65447 is shared by the PC host and one of the processors with memory arbitration handled automatically by the BUS-65547. The shared RAM is allocated from PCI memory space. This eliminates the problems incurred when trying to allocate large blocks of memory from the ISA upper memory address space. An additional 128 bytes of RAM is provided for discrete host I/O.

The BUS-65547 does not require any jumpers or switches for system configuration. All memory and interrupt resources are selected automatically by the BIOS and the operating system.

Full error detection features are provided in all modes of operation. In addition, user specified errors – including bit count, word count and Manchester II errors – may be injected in both the BC and any of the emulated RT modes.

Operation of the BUS-65547 is controlled through the use of Windows-based user-friendly IDEA menus. The IDEA Windows menu provides a graphical multitasking interface. This new version of the menu provides an interactive software interface which allows for full control of the features and functionality of the BUS-65547.

HARDWARE INSTALLATION

HARDWARE CONFIGURATION

The BUS-65547 is a true PCI Plug and Play device, and as such does not require any jumpers or switches to set the base address or interrupt values. The configuration of the BUS-65547 is performed by the BIOS during boot up enumeration of the PC buses. When the operating system discovers the card inserted into the PCI slot, it will query the card for the configuration parameters. These parameters are stored in special PCI registers in the BUS-65547 and describe the usage of resources such as memory and interrupts. The selection and use of computer resources by a PCI card are not modifiable by the user. Once the BIOS establishes the resources to be used, the operating system will setup the card based on that choice.

BUS COUPLING MODE

The BUS-65547 interface to a MIL-STD-1553 bus can be either direct or transformer coupled. The jumpers on the card determine the coupling mode (see FIGURE 2). The default jumper configuration of the BUS-65547 provides for transformer coupling at the Triax connectors. If direct coupling is desired, the shorting plugs at jumper blocks JP12 through JP19 should be moved appropriately. Jump _D for direct (short stub) or _I for transformer coupled (long stub).

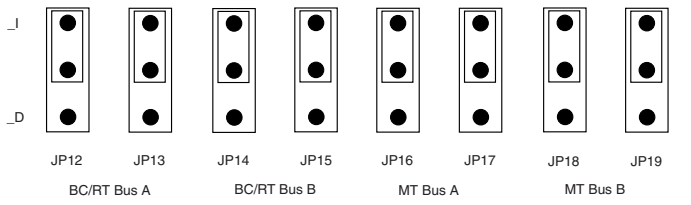


FIGURE 2. COUPLING MODE JUMPERS

D-TYPE CONNECTOR

The BUS-65547 contains two D-type connectors. The 9-pin connector provides the 1553 bus connection to both buses of two separate channels. The 25-pin connector provides access to the external trigger inputs and outputs. Refer to TABLE 5 for a description of the pins on the 9-pin connector and TABLE 6 for a description of the 25-pin connector. The external trigger inputs allow the BUS-65547 to synchronize its operation with external operations. This is most useful when it is desirable to start Monitor capture based on an external event. The outputs may be used to synchronize external equipment (e.g., an oscilloscope).

SOFTWARE INSTALLATION

The BUS-65547 PCI Card is supplied with Runtime Libraries, Windows Menu, and Driver software for Windows 95/98 and Windows NT. VenturCom's RTX is also supported with a driver and library that is supplied with the card. The basic installation procedure is described below. Please refer to the detailed installation procedures in later sections.

WINDOWS 95/98

Driver installation for Windows 95/98.

1. Insert the BUS-65547 into one of the PCI slots and start the computer. Windows 95/98 will detect new hardware and display the 'New Hardware Found' dialog. This dialog will also display the part identification for the card as "DDC: Idea 1553 PCI Device".

2. The system will then display the 'Update Device Driver Wizard' dialog containing the part number and identification. Insert the IDEA CD and click NEXT to continue and search for the device driver.

3. The operating system will first search the floppy drive(s), then it will search the hard drive. If the operating system does not locate the correct *.INF, use the Browse button and point the search to the '<cd drive>:\Win_95' folder. When the correct folder is selected the wizard will continue and display that it found an updated device driver for the specified part. If this is the correct part, click on the FINISH button to complete the installation of the driver.

4. After clicking the FINISH button, the system will complete the driver installation and display the last dialog.

5. Click on the OK button to complete the driver installation process.

At this point, an entry should be established in the Device Manager list. To verify that the driver has been installed, and the device is present in the Device Manager list, access the System Icon located in the Control Panel. When this icon is activated a System Properties dialog will displayed. Select the Device Manager tab. A list of device types will be displayed in the window. The IDEA device type should now be present and recognized by the device string 'IdeaPci' with a diamond icon. If this entry is expanded by clicking on the appropriate plus sign, the device 'DDC: Idea 1553 PCI Device' will be seen. Access the properties for this device by highlighting the entry and clicking on the Properties button.

When the properties dialog opens, it will default to the General Properties of the IDEA 1553 PCI Device (refer to FIGURE 3). This dialog will give general information about the driver such as the status of the device, the type, the manufacturer and the hardware version. If there are any problems with the setup or the device, a message will appear in the Status box describing a probable cause.

If the 'Driver' tab is clicked, a dialog will be displayed that describes any information about the driver (refer to FIGURE 4). This dialog shows the device name and date. The main feature of this dialog is the 'Update Driver' button. This feature will be used if the driver becomes corrupt or it is desirable to update the driver with a newer version. When the Update Driver button is clicked, the System Driver Wizard dialog will appear. This will provide the standard functionality to select a driver source disk or search for a new version.

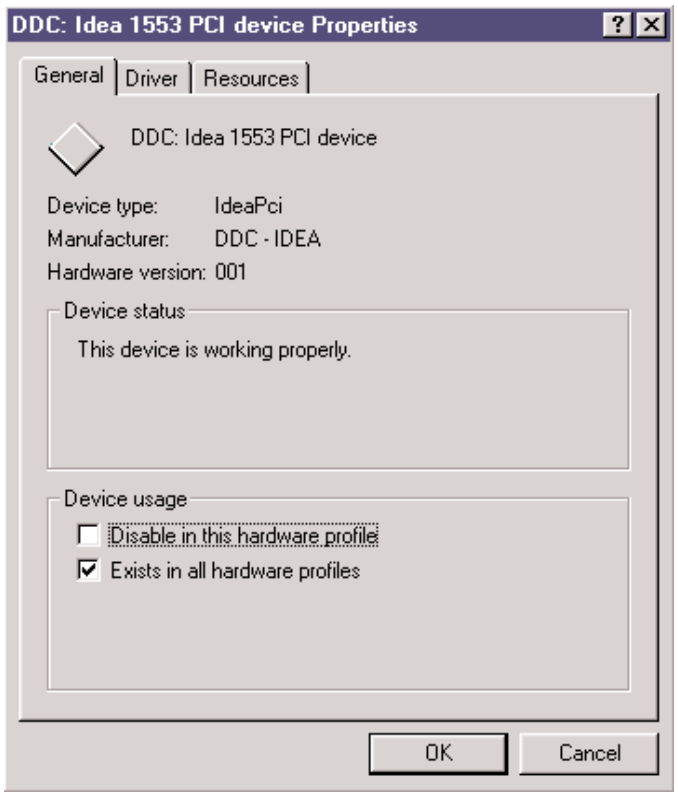


FIGURE 3. PCI DEVICE GENERAL PROPERTIES

FIGURE 4. PCI DEVICE DRIVER PROPERTIES

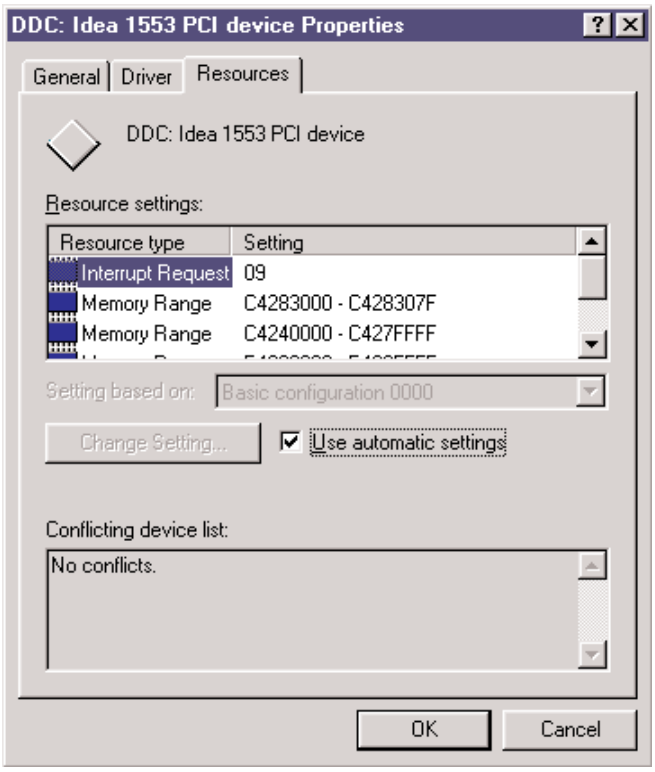


FIGURE 5. PCI DEVICE RESOURCE PROPERTIES

The third tab in the 'Idea 1553 PCI device Properties' displays the Resources the device uses in the system (refer to FIGURE 5). This dialog displays information about the Interrupt and Memory resources required and used by the DDC 1553 Idea Device. This card requires one interrupt resource to communicate to the system that an event occurred. This interrupt is used by the Monitor to move data from the card to a communication stack in host memory. The interrupt resource is also used to signal user applications that a hardware or software event occurred. The remaining resources required by the card are four Memory segments. These memory areas will be selected by the BIOS during the bus enumeration phase of the boot process.

Unlike previous card types (ISA and PCMCIA), there are no user assigned resources for the PCI cards. However, in some systems it is possible to assign the interrupt via settings in the BIOS setup. This is not always available, and is generally not needed. The BIOS is very good at determining and providing the required resources for PCI cards, and only in very rare cases are these resource settings unusable.

The second phase of the Windows 95/98 installation will create the directories and copy the necessary files from the source disk to the hard disk. The files include sample executables, *.LIB files, *.DLL files, and sample source. All of these files may be used in the creation of an application for the BUS-65547. The following steps should be performed in order to complete the Runtime Library installation. This text assumes that the drive 'D:' is the CD drive.

1. Insert the IDEA CD into the drive.

2. Select the START/RUN button and type < D:\SETUP.EXE> in the Run edit box, then click OK or press <ENTER>.

3. The setup.exe program runs the Installshield installation program. This program will present a Welcome screen for the 'MIL-STD-1553 IDEA for Windows 95'. Click the NEXT button to continue.

4. A destination directory selection dialog will be presented next. This dialog provides a default installation directory of C:\ID95103. The user may change the destination drive and directory by selecting the BROWSE button and then editing the drive/directory text or selecting an existing directory on the disk. Click the NEXT button to continue.

5. A 'Start Menu' Program Folder must be selected next. The default folder is 'IDEA 1553 for Windows 95' but may also be changed by selecting an existing folder or by typing a new folder name into the edit box. Click the NEXT button to continue.

6. At this point, the Installshield will create the necessary folders and then copy the files to the hard drive.

7. Finally, the 'Setup Complete' dialog will be displayed. There is an option to restart the computer now or to restart later. Choose an option and then click the FINISH button to return to Windows. Note that the IDEA software and library will not work correctly until the system has been rebooted and the registry has been updated.

WINDOWS NT

Driver installation for Windows NT.

The Setup.exe installation program will copy the system driver files, and control panel application for setting the device number. It will also create the directory structure that will be the target location for the library and all user applications. Follow the software installation instructions after inserting the BUS-65547 into one of the PCI slots.

1. Select the START/RUN button and type < D:\Win_NT\SETUP.EXE > in the Run edit box or select the appropriate directory using the Browse button, then click OK or press <ENTER>.

2. The setup.exe program runs the InstallShield installation program. This program will present a Welcome screen for the 'IDEA for Windows NT'. Click the NEXT button to continue.

3. A destination directory selection dialog will be presented next. This dialog provides a default installation directory of C:\IDNT103. The user may change the destination drive and directory by selecting the BROWSE button and then editing the drive/directory text or selecting an existing directory on the disk. Click the NEXT button to continue.

4. A 'Start Menu' Program Folder must be selected next. The default folder is 'IDEA for Windows NT' but may also be changed by selecting an existing folder or by typing a new folder name into the edit box. Click the NEXT button to continue.

5. At this point, the Installshield will create the necessary folders and copy the files to the hard drive.

6. Finally, the setup program will launch the Control Panel application 'IDEA Card Install'. The default card type is BUS-65517II. For this installation the card type should be changed to the BUS-65547. It should be noted that when the card type is changed to BUS-65547 the values for Interrupt, I/O, and Memory will be inactive. Since this is a PCI card, there are no user modifiable parameters to setup. These values are selected by the Operating System and by the BIOS, and can not be changed by the user.

7. The Setup Complete dialog gives the user the option of restarting the computer immediately or at a later time. If YES is selected, the computer will immediately shut down and restart. If NO is selected, it will be up to the user to ensure that the computer is restarted prior to using the IDEA software.

Once all of the above steps have been completed, the drivers will be ready for operation, and the sample programs may be run. At this time, you may also desire to write your own programs. The sample programs may be compiled, or altered and compiled as needed. These programs will aid in understanding the sequence of functions that are needed to produce an IDEA program. Please refer to the BUS-65547 RTL programming manual.

IDEA MENU

The IDEA Menu is an intuitive graphical interface that provides access to the full functionality of the BUS-65547 without having to write or compile a single line of code.

IDEA MENU INSTALLATION

The IDEA menu program is installed with the library and drivers. Once the installation process is complete, the user can run this program. Please refer to the section on Software Installation for complete details.

The paragraphs below describe the general use of the Menu software.

IDEA Menu Operation

For more information pertaining to the operation of the MIL-STD-1553 bus, please refer to the MIL-STD-1553 Designer's Guide.

Setup Editor

Operation of the IDEA menu starts with defining the setup of the 1553 bus environment that is to be tested. This setup is performed in the Setup Editor. This program may be started by selecting Start | Programs | Idea 1553 for Windows 95 | Setup Editor from the startup menu. When the Setup Editor starts, a default set of parameters for each menu item is created. This reduces the effort required to initialize a setup file.

Legality

This option allows you to determine whether or not RTs should check message legality. If the option is chosen, the RT will respond to illegal commands with ME=1 in the status.

The legality setup is used to specify which mode code commands are to be treated as legal commands. All 32 possible mode code commands are presented in this dialog with the capability to enable or disable the legality of the command. Checking a mode code indicates that the selected mode code is legal. If a mode command is left unchecked, then the command is treated as illegal.

Each mode code can be assigned up to two intermessage routines. To associate intermessage routines with a mode command, first choose the desired mode command name by clicking on the corresponding dialog box and then use the 2 combo boxes on the bottom of the screen to specify the intermessage routines.

The default selection includes all non-reserved mode codes.

Status Mask

The status mask is used to specify which of the RT status word bits should be used in determining an exceptional condition.

If a bit in the mask word is set to 1 and the corresponding bit in the status is also set to 1, the Monitor detects an exceptional condition. Exceptional conditions are counted in real-time and denoted in the stack. By default, all mask bits are set.

By default, all valid status word bits are enabled.

Response Timeout

The response timeout determines the amount of time that the emulated BC and the Monitor must wait for a Remote Terminal response before they declare a 'No Response' message error. The four entries in this dialog represent the maximum number of microseconds allowable between the parity bits of the last word of a message and the response by the Remote Terminal.

If the Remote Terminal responds any later than this value, the message will be declared as a faulty message with the error condition of NO_RESPONSE.

The default setting is 14 microseconds.

Transmit Amplitude

The transceivers of the BUS-65547 have variable output voltage control. The outputs can be varied by software control between 0.10 volts and 28 volts in 256 steps. A value of '0' indicates the lowest voltage level of 0.10 volts, while a value of 255 indicates the maximum output of at least 28 volts.

The default value for the setting is 200.

Terminal Activity Setup

One of the first settings that should be modified is the Global Definitions | Terminal Activity. The Terminal Activity dialog box (refer to FIGURE 6) is used to select the 1553 Bus terminals (BC and RTs) that will be emulated. All entries that are checked will be emulated by the BUS-65547. External terminals that are active on the bus must not be checked.

Bus Controller Setup

After specifying the terminals that will be emulated by the IDEA card, the message parameters for Bus Controller operation should be defined. There are two menu items that must be setup for proper Bus Controller operation. The first item is the Message Definition dialog (refer to FIGURE 7). This dialog provides the controls needed to create all of the possible 1553 bus messages. The default names for the messages are 'MSG_' with the index number of the message concatenated to it.

This name may be changed by typing the desired message name in the edit box 'NAME:'. The 'TYPE' button selects between one of the five possible message types: RX, TX, RT-RT, MODE receive, and MODE transmit. The arrow in the TYPE button will change direction, and the text will display the selected type. The BUS button contains a picture of a bus and the text A or B. This button selects the bus on which the message will be initially processed. The panels located on either side of the 'TYPE' button may be clicked to edit the Bus Controller or Remote Terminal properties. When these buttons are clicked, a dialog will pop up allowing the user to modify the terminal address, subaddress and word count for Remote Terminals. If the Bus Controller panel is clicked, the dialog allows the user to modify the data table contents.

The user specifies Time to Next Message by either selecting 'Default' or 'User' button. The default value is equal to the message length plus the shortest possible intermessage gap before the start of the next message. If 'User' is selected, the operator then manually types the desired message time.

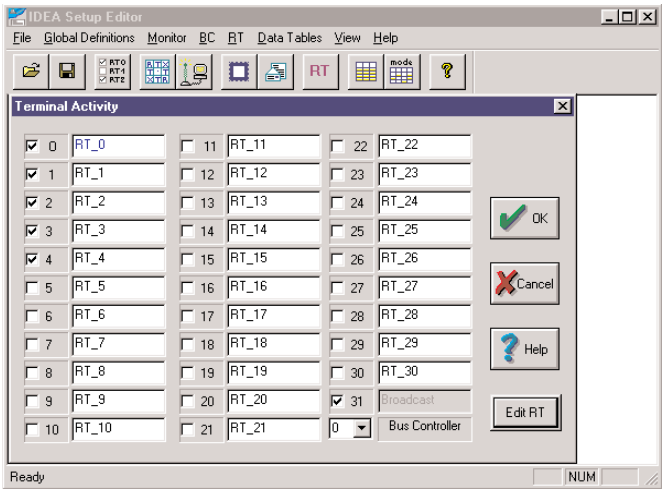


FIGURE 6. SETUP EDITOR: TERMINAL ACTIVITY

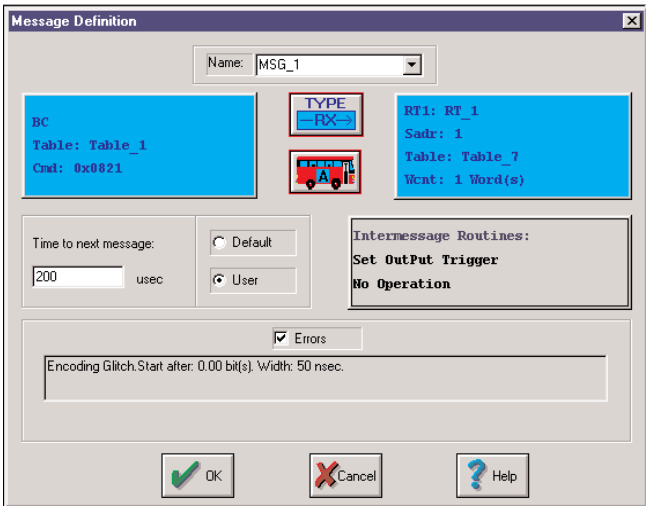


FIGURE 7. SETUP EDITOR: BC MESSAGE

The user can insert 1553 Bus format errors into a message. Checking the Errors box will open a Error Definition dialog (refer to FIGURE 8). Most of the entry definitions are obvious. 'Glitch' is defined as a period of time when the 1553 signal returns to the quiescent state during any point in the message. The Error Injection dialog allows the user to create any error that could occur on the bus.

After all of the messages for a session have been defined, the next operation needed for Bus Controller support is specify the sequence of messages by using the Frame dialog (refer to FIGURE 9).

The frame is a list of messages and special frame symbols. The frame may be up to 1000 messages long. Each message is selected from a drop-down list. The minor frame time can be entered in the Minor Frame Time edit box. This value is entered in microseconds. Each time a message is selected in the frame list, the message name and a graphic and text description of the message will be displayed at the top of the dialog.

Remote Terminal Setup

With all of the parameters now defined for Bus Controller emulation, the Remote Terminal emulation must be setup. This is performed from the Remote Terminal Definition dialog (refer to FIGURE 10). Within this dialog, the user is able to setup the operation and data requirements of the selected Remote Terminal. If the Remote Terminal is not checked as active in the Terminal Activity list under Global Definitions, then the settings for the Remote Terminal will not have any effect on the operation of the BUS-65547. Each Remote Terminal has a default set of data tables assigned to it when the Setup Editor is started. The Data Table Assignment button will cause the Data Table Assignment dialog to be displayed. From this dialog, the user may change the default data table selections for the selected Remote Terminal and subaddress combination.

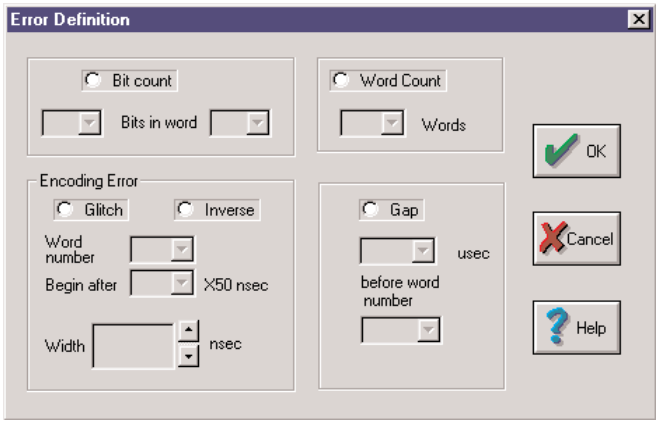


FIGURE 8. SETUP EDITOR: BC ERROR DEFINITION

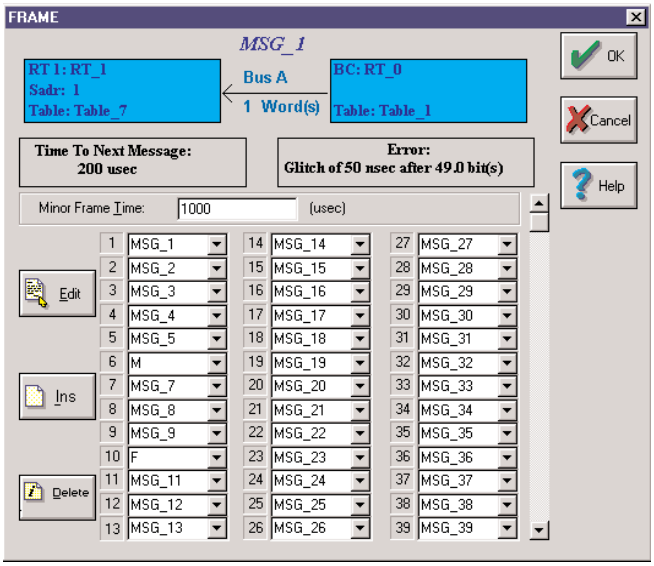


FIGURE 9. SETUP EDITOR: BC FRAME DIALOG

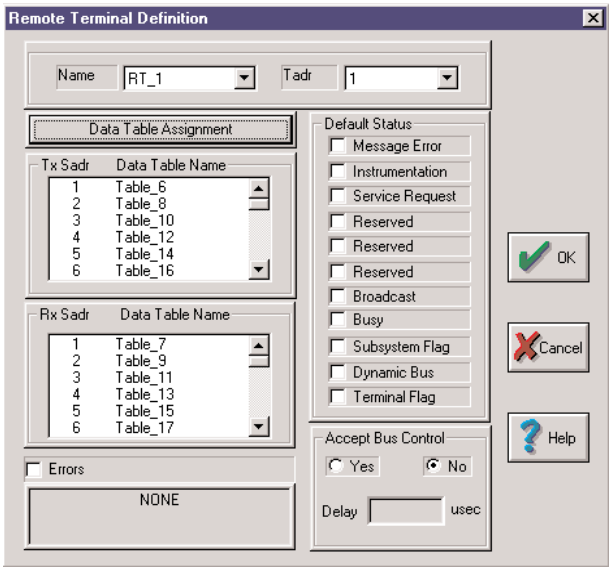


FIGURE 10. SETUP EDITOR: RT DEFINITION DIALOG

After all the data tables have been assigned, the contents of the data tables may be modified by double clicking on a transmit or receive data table entry. The Data Table Definition dialog (refer to FIGURE 11) allows the user to set the data table length. The data table can be set to any length between 1 and 32 words. Once the length is set, the data entries for each word may be set to desired values. Each data table that is associated to a Remote Terminal and stores the Intermessage Routines. This dialog will be used to set the desired routine for the selected Remote Terminal. This process should be repeated for each emulated Remote Terminal.

Monitor Setup

The BUS-65547 is capable of operating as a Monitor at the same time it is emulating the Bus Controller and up to all 31 Remote Terminals. The Monitor is capable of storing every message that appears on the bus from both the BUS-65547 emulated terminals and any physical terminals that are on the bus. The setup for the Monitor involves selecting which terminal address and subaddress combinations should be monitored, and what event, if any, should trigger the Monitor to start capturing data. Message Select is performed in the Monitor Message Selection dialog (refer to FIGURE 12). Messages are captured based on terminal address, subaddress and transmit/receive bit.

In FIGURE 12 it can be seen that receive messages to subaddress 4 for all terminal addresses will be monitored. The transmit messages for this same combination will be ignored. Likewise, the transmit messages directed to terminal address 2 for all sub-addresses will be monitored, and the receive messages to this same combination will be ignored. Finally, all other message combinations will be monitored. If the All button is selected, pressing the Tx, Rx, Rx/Tx or clear button will be applied to all of the combinations in the table. If a row or column is selected, the buttons will operate on that entire row or column. After selecting the messages to be monitored, the only step remaining is to decide when the Monitor should start to capture the data.

The Monitor Capture dialog (refer to FIGURE 13) allows setting the capture to occur immediately. This means as soon as the BUS-65547 starts running it will be monitoring the bus and storing the data to a disk file. This is convenient for general bus monitoring. The Trigger setting will cause the Monitor to wait for an external signal to be presented to the Monitor Input at the D-Connector (refer to TABLE 5). This mode of triggering is useful if an external event is available to pin point some message sequence on the bus. Starting the Monitor capture this way saves disk space and search time later. If a message sequence is known to start with some unique command, the Command trigger will be very useful. When the trigger option is enabled, the bus command elements may be specified. The terminal address, T/$\overline{R}$ bit and subaddress can be set to the desired message command. Each bit of these three fields may be either a '1', a '0' or an 'X'(don't care). The 'don't care' selection allows enabling an entire family of message commands to be used as the trigger. The final trigger selection will cause the Monitor to start capture based on a bus Exception. The 'Any' selection will

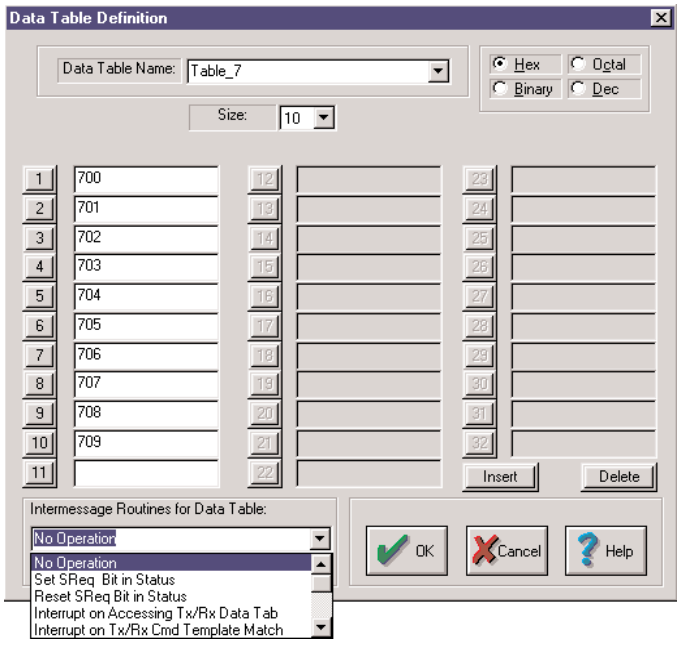


FIGURE 11. SETUP EDITOR: RT DATA TABLE EDIT DIALOG

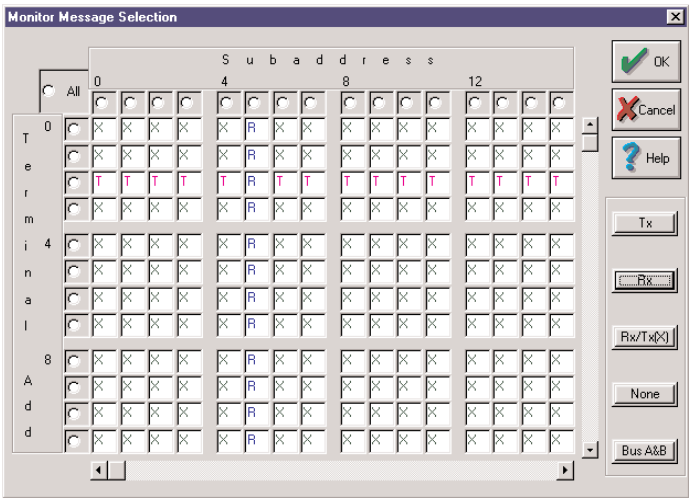


FIGURE 12. SETUP EDITOR: MONITOR MESSAGE SELECTION

cause the bus to start capture if any of the listed exceptions occur. If a specific exception is to be targeted, then the corresponding button may be checked.

At this point, all of the basic settings required to setup the BUS-65547 have been established, and the setup may now be saved to a file. This is performed by selecting FILE | SAVE and then entering the desired filename <file name.stp>.

IDEA Menu Operation

The IDEA Menu application is the actual menu (refer to FIGURE 14). This application opens a Setup file that was created in the Setup Editor and downloads the parameters to the BUS-65547 card. It also opens a Stack file into which the monitored information will be stored on disk. There is a menu item that allows the selection of the active IDEA card. This enables a user that has multiple IDEA cards in a single computer to select which one of the cards is to be run by this instance of the IDEA Menu application. Multiple instances of the IDEA Menu can run simultaneously if there is more than one IDEA card installed in the computer.

It is a simple matter to use the IDEA Menu program to run a 1553 Bus Controller session. When the program is started a blank window is presented to the user. The first operation is to select FILE | NEW. This will open a blank IDEA Menu project window similar in appearance to FIGURE 14. After the project window is opened, the user must select a setup file by clicking the SETUP | SELECT SETUP FILE menu item. When this item is selected the standard Windows File Open dialog will pop up and point to the ID95103 \ STP directory by default. The user may select any setup file presented here or in any other folder by double clicking the file, or by single clicking the file and then clicking OPEN. After selecting the setup file, a Stack file should be selected in the same way. Select the SETUP | SELECT STACK FILE menu item, and then select a stack file from the default directory ID95103 \ STK or any other directory in the same manner as the setup file selection.

If multiple cards are installed in the computer, the desired card may be selected by choosing one of the available cards from the SETUP | CARD NO menu selection.

At this point, operation as a Bus Controller can be initiated by clicking on the 'BC' button. This will cause the BUS-65547 to download the setup from the selected files and then run. When the '/BC' button is selected, a dialog will pop up requesting the number of times that the frame should be repeated. The selections are 'FOREVER' or a typed in number of times. Once the card is running, the display will look like that in FIGURE 14. The 'Terminal' column displays which Terminals are communicating on the 1553 bus, the Total Messages column displays the total number of messages addressed the specific terminal, and the rest of the columns display errors counts for each type of error that occurred in the messages.

The setup steps for Remote Terminal or Monitor operation are the same as for the Bus Controller with the exception that the '/RT' or '/MT' buttons will be pressed.

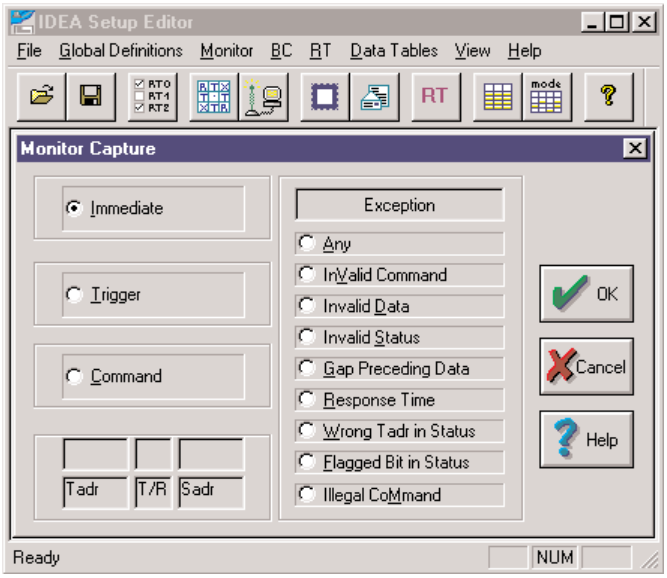


FIGURE 13. SETUP EDITOR: MONITOR CAPTURE DIALOG

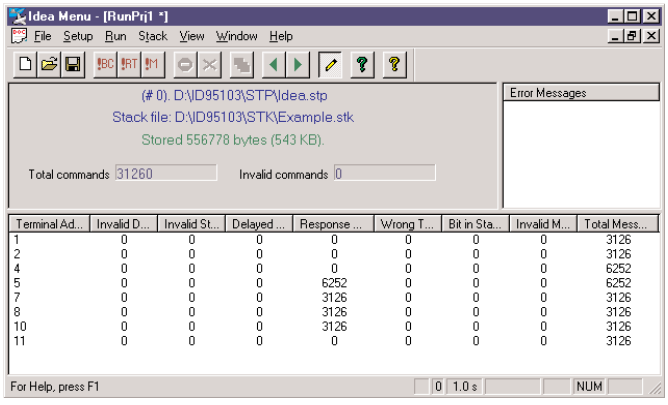


FIGURE 14. IDEA MENU

Stack View

The Stack View program displays previously captured stack files that have been saved to disk (refer to FIGURE 15). In order to view a previously saved stack file, the FILE | LOAD menu option should be selected. This will present the standard Windows File Open dialog with a file filter of '.STK'. The default folder for the open dialog is ':\id95103\stk'. The stack file uses a packed message format that closely represents the Monitor stack. The number of bytes used in the stack file for each message is variable. This can be calculated as 8 header words*2 plus the number of data words *2. The stack files from previous versions of the IDEA software are 100% compatible with the new software.

Stack Search

One of the tools provided with the Stack View program is the Search dialog (refer to FIGURE 16). This dialog provides a powerful and highly selective tool for searching the stack file for a specific message or group of messages. The search dialog can be directed to search based on terminal address, T/$\overline{R}$ bit, subaddress and word count. Each of the selection items may be cleared, set or the selection may be inverted with the click of a single button. If multiple entries for a given selection parameter are checked, then all messages that satisfy any of the checked items will be displayed with their command word highlighted in red. As an example, the search dialog shown in FIGURE 16 will search for and display all Receive messages that are addressed to Terminal addresses 0, 1 or 2 with a subaddress of 0 or 1 and any word count.

Successive searches with the same criteria may be performed by pressing 'F3', selecting the SEARCH | SEARCH AGAIN menu item or by clicking the Magnifying Glass button with the three dots.

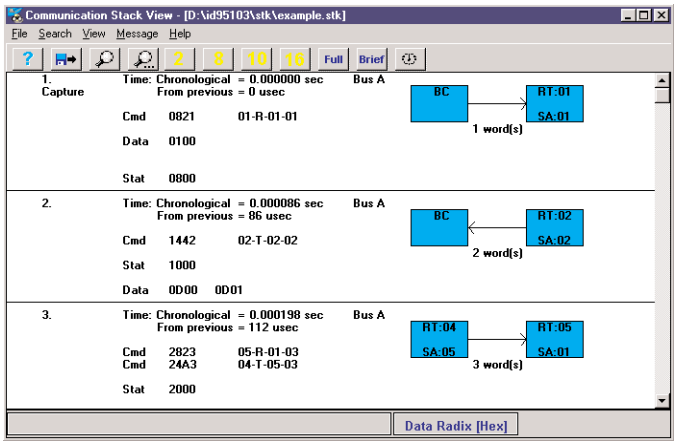


FIGURE 15. COMMUNICATION STACK VIEW

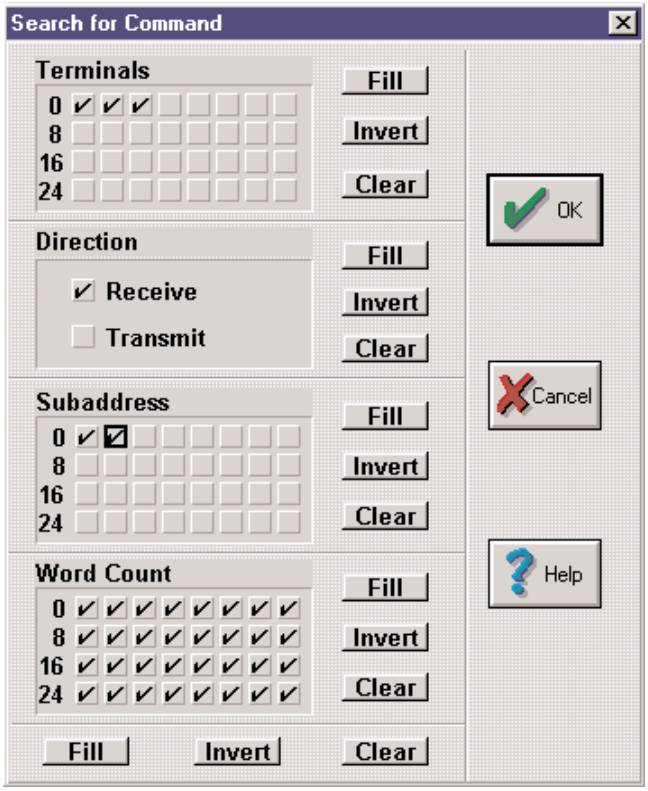


FIGURE 16. STACK VIEW: SEARCH

SAMPLE APPLICATION FOR BUS-65547 RTL

Software that is written for the BUS-65547 must use the Runtime Library to access the operation of the hardware. There are no registers or memory locations that are accessible by the user. Using the Runtime Library, the user has the ability to control all aspects of the 1553 bus operations. Each program will follow the same basic skeleton format of resetting the card, creating messages and frames, and then running. You will see this form in the sample code that follows. For a more detailed description of the Runtime Library and how to structure a program, please refer to the BUS-65547 Software Programming Manual. The example program below sets up the BUS-65547 as a BC. This program is designed to react to interrupts that are generated by the card based on the 1553 message flow. This example, and others for BC, RT and Monitor are available in the Samples subdirectory. Not all the code included in the sample program will be described here.

```
#include <stdio.h>
#include <conio.h>
#include "idea1553.h"
ViInt16 _VI_FUNCH bcrt_event_handler( ViInt16 intr_type , ViInt16 parameter);
// data tables
ViInt16 tx_tab = 1;
ViInt16 rx_tab = 2;
ViInt16 tx_len = 10;
ViInt16 rx_len = 10;
// addresses
ViUInt16 bc_adr = 0;
ViInt16  Cardno = 0;
ViStatus ind;
/* definition of user interrupt handler
decode the intr_type to identify the type of the interrupt */
ViInt16 _VI_FUNCH bcrt_event_handler( ViInt16 intr_type , ViInt16 parameter)
{
ViInt16 type, subtype;
  type = intr_type & (ViUInt16)0x00FF;
  subtype = intr_type >> 8;
       switch (type) {
              case IDEA_INT_BC_MSG:    // successful message
              switch (subtype) {
                case 1:  // msg #1
                       rx_cnt++;
                       break;
                case 2:  // msg #2
                       tx_cnt++;
                       break;
              }
                     break;
              case IDEA_INT_BC_ERR:    // communication error
              err_cnt++;
              break;
       }
return (0);
}
int main(void)
{
if (init()) {
```

SAMPLE APPLICATION FOR BUS-65547 RTL (CONT)

```
        printf("\nError during initialization. Abort...\n");
        return 1;
  } else
        printf("\nInitialization - Ok.");
  printf("\nPress any CR to exit.\n");
  printf("\nExpected: Rx cnt = %04ld  Tx cnt = %04ld  Err cnt = %04ld (no response on RT3)\n",
                          times, times, times);
  getchar();
ind = IdeaReadData(Cardno, tx_tab, buffer, tx_len);
  if (ind)
              print_error_message( Cardno, ind, "read_data");
done();
return 0;
}
int init(void)
/* init the IDEA card. return an error code. */
{
if (ind = IdeaResetCard (Cardno, IDEA_BC_AND_MON)) {
              print_error_message( Cardno, ind, " reset_idea_bcrt");
              return ind;
  }
/* configure TX table */
if (ind = IdeaDefTableSize(Cardno, tx_tab, tx_len)) {
              print_error_message( Cardno, ind, " def_table_size");
              return ind;
       }
/* configure RX table */
  if (ind = IdeaDefTableSize(Cardno, rx_tab, rx_len)) {
              print_error_message( Cardno, ind, " def_table_size");
              return ind;
       }
/* identify emulation of BC */
       if (ind = IdeaDefEmulateBc(Cardno, bc_adr, IDEA_YES)) {
              print_error_message( Cardno, ind, " def_emulate_bc");
              return ind;
       }
/* set the global minor frame time */
       if (ind = IdeaDefMinorFrameTime( Cardno, 10000)) {
              print_error_message (Cardno, ind, " def_minor_frame_time");
              return ind;
       }
/* write data to RT  data buffers*/
for (i = 0; i < rx_len; i++)
        buffer[ i]  = 0x1111;
       if (ind = IdeaWriteData( Cardno, rx_tab, buffer, rx_len, 1)) {
              print_error_message (Cardno, ind, " write_data");
              return ind;
```

SAMPLE APPLICATION FOR BUS-65547 RTL (CONT)

```
        }
/*create messages by filling in msg structure then call IdeaDefMsg()*/
        // message 1 - receive 10 word to RT1
        bc_page.comm_type      = IDEA_RECEIVE;
        bc_page.data_table_no = rx_tab;
        bc_page.bus             = IDEA_BUS_A;
        bc_page.cmd_1.tadr     = 1;
        bc_page.cmd_1.subadr   = 1;
        bc_page.cmd_1.wcnt     = 10;
        bc_page.time_to_next_message = 1000;
        bc_page.inj_error_ptr = IDEA_E_NONE;
        bc_page.first_intermessage_routine = IDEA_IMR_INT_ON_END_OF_MESSAGE;
        bc_page.second_intermessage_routine  = IDEA_IMR_NO_OPERATION;
        if (ind = IdeaDefMessage( Cardno, 1, &bc_page)) {
                print_error_message( Cardno, ind, " def_message 1");
                return ind;
        }
/*assign messages and Frame Symbols to frame*/
        frame_items[ 0]  = 1;              // BC->RT1 (sub1, 10w)
        frame_items[ 1]  = 2;              // BC<-RT2 (sub1, 10w)
        frame_items[ 2]  = 3;            // BC<-RT3 (sub1, 10w) - not exist
        frame_items[ 3]  = (ViChar)IDEA_END_OF_MAJOR; //
/*The function IdeaDefFrame() transfers the frame information to the IDEA card memory.*/
if (ind = IdeaDefFrame( Cardno, 4, frame_items)) {
                print_error_message( Cardno, ind, " def_frame");
                return ind;
        }
/*the IdeaDefIntMask() function must be called in order to enable the desired interrupts */
if (ind = IdeaDefIntMask ( Cardno, 0xFF00)) {  // enable BCRT interrupts
    print_error_message (Cardno, ind, "def int mask");
                return ind;
        }
/*The user interrupt routine defined locally in the program must be connected to the library by use of the
IdeaSetBcrtEvent() routine*/
if (ind = IdeaSetBcrtEvent (Cardno, bcrt_event_handler)) {
    print_error_message( Cardno, ind, "set bcrt event");
                return ind;
        }
/*IdeaRunBc() routine causes the IDEA card to run as all emulated Bus Controllers and Remote Terminals*/
        if (ind = IdeaRunBc( Cardno,1, times)) {
                print_error_message (Cardno, ind, " run_bc");
                return ind;
        }
/*Return to the calling routine 'No Errors'*/
return 0;
}
```

BUS CONTROLLER (BC) MODE

The BUS-65547 Bus Controller supports all MIL-STD-1553B message formats. Up to 1000 unique receive, transmit, mode code, and RT-to-RT messages may be defined at one time. Programmable attributes within a message are time to next message, bus (channel A or channel B), intermessage routines, and injected error. The time to next message defines the time from the start of the present message to the start of the next message. The time to next message is programmable up to 65,535 μsec in 1 μsec increments.

MINOR AND MAJOR FRAMES

The execution of messages is controlled by a message list referred to as a frame. The frame specifies the contents and timing of complete communication runs by the BC. Each entry in the frame is either a reference to a message or a special symbol. The entire frame is referred to as a major frame. The major frame is divided into minor frames of equal time duration.

The major frame time is based on a programmable 32-bit counter with 2 μsec resolution. The BUS-65547 supports major frames of up to 1000 messages for single buffered mode or up to 100 messages for double buffered mode, and the period is up to 143 minutes.

BC ERROR INJECTION

Error conditions may be injected on a message by message basis. The BUS-65547 supports three categories of injected BC errors: length errors, encoding errors, and gap errors. Length errors include both word count errors and bit count errors. Word counts of -32 to +1 words may be programmed. Bit counts of +3, +2, +1, -3, -2, or -1 bit may be programmed for any word within the message.

Encoding errors are implemented through the use of two simple yet powerful mechanisms for modifying the output of the BUS-65547's Manchester encoder. The two modifying functions are glitch and inverse. A glitch will force the output of the encoder to an idle bus condition for the specified period of time. An inverse will invert the output of the encoder for the specified period of time. The placement of the error is specified by the word number, starting time, and width. The error may be placed in any word within the message. The starting time is programmed in 50 ns increments from the beginning of the specified word. The width of the error is specified in 50 ns increments up to 3 μsec. This error injection scheme lends itself to generating a host of errors including invalid sync patterns, parity errors, and Manchester bi-phase errors.

A gap of 3, 4, or 5 μsec (measured mid-parity crossing to mid-sync crossing) may be inserted between any two words in a message. This allows for a "dead time" gap between words of 1, 2, or 3 μsec.

INSERTING ASYNCHRONOUS MESSAGES

The BUS-65547 allows an asynchronous message to be inserted while the card is running. This inserted message will be executed upon completion of the current message.

BC INTERMESSAGE ROUTINES

Upon completion of a BC message, the BUS-65547's on-board processor executes two intermessage routines. These routines are contained within the firmware for the on-board processor. The firmware is downloaded to the processor when the card is reset. Intermessage routines are used to implement automatic retries on failed messages as well as other “end of message” functions. TABLE 3 provides a summary of the BUS-65547's intermessage routines.

RESPONSE TIMEOUT

The BUS-65547 BC, RTs, and Monitor support programmable response timeout values of 14, 15, 18, and 20 μsec.

REMOTE TERMINAL (RT) MODE

The BUS-65547 can simultaneously simulate the operation of 31 unique remote terminals plus a broadcast address. The BUS-65547 maintains 31 independent "last status" and "last command" words allowing for full support of transmit last command and transmit status mode commands. The BUS-65547 supports command illegalization for each transmit or receive message based on Remote Terminal address and subaddress. In addition, individual mode commands may be illegalized.

RT ERROR INJECTION

Error conditions may be injected on an individual RT basis. The BUS-65547 supports five categories of injected RT errors: length errors, encoding errors, gap errors, status address errors, and response errors. Length errors include both word count errors and bit count errors. Word counts of -32 to +1 words may be programmed. Bit counts of +3, +2, +1, -3, -2, or -1 bit may be programmed for any word within the message.

Encoding and gap errors are implemented as in the BC. A status address error consisting of a wrong terminal address can be injected by the RT.

The BUS-65547 supports three types of response errors: no response, a late response, or a response on the wrong bus. No response errors may be programmed for a single channel (Bus A or Bus B) or for both buses. Injecting a no response error on

one bus provides a simple mechanism for testing bus controller retry conditions. A late response may be programmed in the range of 12 to 30 μsec in 1 μsec increments.

RT INTERMESSAGE ROUTINES

The RT section of the BUS-65547 also supports intermessage routines. Upon completion of an RT message the BUS-65547's on-board processor executes two intermessage routines. These routines are contained within the firmware for the on-board processor which is downloaded when the card is reset. The intermessage routine is specified by the data table that was used by the RT for a given message. Refer to TABLE 4 for a summary of the BUS-65547's intermessage routines.

TABLE 4. BC/RT INTERMESSAGE ROUTINES

NO OPERATION
RETRY CURRENT MESSAGE ON ALTERNATE BUS
RETRY CURRENT MESSAGE AND REMAIN ON ALTERNATE BUS
RETRY ON SAME BUS
INTERRUPT ON END OF MESSAGE
INTERRUPT ON FRAME SYMBOL
SET SERVICE REQUEST BIT IN STATUS
RESET SERVICE REQUEST BIT IN STATUS
INTERRUPT AFTER ACCESSING TX/RX DATA TABLE
INTERRUPT AFTER MODE COMMAND
INTERRUPT AFTER TX/RX COMMAND TEMPLATE MATCH
INTERRUPT AFTER MODE COMMAND TEMPLATE MATCH
TIME-TAG (STORE RTC IN A CIRCULAR QUEUE)
RETRY ON SAME BUS AND THEN ON ALTERNATE BUS
SET OUTPUT TRIGGER
RESET OUTPUT TRIGGER
WAIT FOR INPUT TRIGGER
NO RESPONSE ON BOTH BUSES
SET BUSY BIT IN STATUS
RESET BUSY BIT IN STATUS
SKIP NEXT MESSAGE
SKIP NEXT MESSAGE ONCE
BLOCK_DATA_BC
BLOCK_DATA_RT

BC/RT DATA TABLES

The BUS-65547 maintains 1024 data tables within the shared RAM on the card. Each data table may be up to 32 words in length. These data tables are common to both the BC and RTs Internal lookup tables map each Remote Terminal address, T/$\overline{R}$ bit, subaddress combination (RT mode) and message number (BC mode) to a given data table. Data tables may be read or written to in real time. Data tables may be either single or double buffered. Double buffering can be used to avoid contention when the PC's application and the 1553 bus access data tables simultaneously.

The BUS-65547 provides a block data mode in which the data table number associated with a given BC or RT message increments after completion of the message. The block data mode is implemented as a circular data structure. Each BC message and RT command (Remote Terminal address, T/$\overline{R}$ bit, and subaddress) has three data table numbers associated with it: first, last, and current. The current data table number will be incremented after completion of the message until the value of last is reached, at which point the current table number will rollover to the value of the first. The incrementing of the current data table number is accomplished through the use of an intermessage routine.

MONITOR (MT) MODE

The BUS-65547 contains an independent message Monitor with the ability to select messages in real time. Monitor selection is performed through the use of a lookup tabled based on the Remote Terminal address, T/$\overline{R}$ bit, and subaddress of command words. Monitored messages are stored into a 6K word circular buffer located in the shared RAM on the BUS-65547. Each entry in the monitor buffer contains an eight word header followed by a variable number of data words. Contained with the message header are the receive/transmit command(s), receive/transmit status(es), message format, Bus (A or B), a capture flag, word count (actual number of words in the message), a detected error field, and a 32-bit time tag (2 μsec resolution).

The transfer of the messages from the card's circular buffer to the host memory/disk is determined by the capture flag which is set upon detection of a predefined event. Capture events include immediate, command template match, exception, or trigger. The command template event is based on a 16-bit command word with a 16-bit mask. Exception events may be programmed for any exception, invalid command, invalid data, invalid status, gap preceding data, response time error, wrong Remote Terminal address error, status set condition or an illegal mode or broadcast command. The trigger event uses one of the monitor input pins on the 25-pin D-type connector as a trigger input.

INTERRUPTS

Interrupts to the PC are generated by both the BC/RT and the Monitor on a common output. The hardware interrupt level is selected automatically by the system during the boot up process and cannot be modified by the user. The BUS-65547 implements the interrupt on the INTA# line.

BC INTERRUPT GENERATION

Bus Controller interrupts may be enabled by a global interrupt mask for successful messages, communication errors, status set conditions, or on selected frame symbols (skip, break point, major frame, and minor frame symbols). The criteria for a status set condition is programmed globally through the use of a status mask. The status mask allows any of the 16 bits within an RT status word to be ignored. The status mask affects the generation of interrupts as well as the detected error field which is stored in the message structure.

Bus Controller interrupts are issued by intermessage routines associated with messages. Each interrupt request is accompanied by a two-word vector pushed onto a circular queue which is transparent to the user. The queue can hold up to 64 interrupt vectors; thus, the host computer is not required to immediately acknowledge the interrupt request.

RT INTERRUPT GENERATION

Remote Terminal interrupts may be enabled by a global interrupt mask for legal transmit/receive messages, legal mode commands, illegal transmit/receive messages, or illegal mode commands.

Remote Terminal interrupts are issued by intermessage routines associated with data tables allowing for selective interrupt generation on a message-by-message basis. Each interrupt request is accompanied by a two-word vector pushed onto a circular queue which is transparent to the user. The queue can hold up to 64 interrupt vectors; thus, the host computer is not required to immediately acknowledge the interrupt request.

MONITOR INTERRUPTS

Monitor interrupts may be generated after each message is received or after one third of the monitor's circular buffer has been filled (approximately 2K words). This allows for either real-time analysis or mass collection/storage of monitored data.

INTERFACE TO A MIL-STD-1553 BUS

The BUS-65547 provides options for both direct and transformer coupling to a MIL-STD-1553 bus. FIGURE 17 illustrates the interface from the IDEA Card to a MIL-STD-1553 bus. Connections for both transformer (long stub) and direct (short stub) coupling are indicated. Both coupling modes are implemented using the triax connectors and setting the appropriate jumper settings. In accordance with MIL-STD-1553, a transformer coupled connection to a 1553 bus requires the use of a bus coupler. In addition, the 1553 bus must be properly terminated. A bus coupler contains an impedance matching transformer as well as a pair of fault isolation resistors.

The transmitter output voltage of the BUS-65547 may be varied in 256 steps from 0.1 V to at least 28 Vpp, measured across a 77 ohm resistor attached to the stub (triax connector).

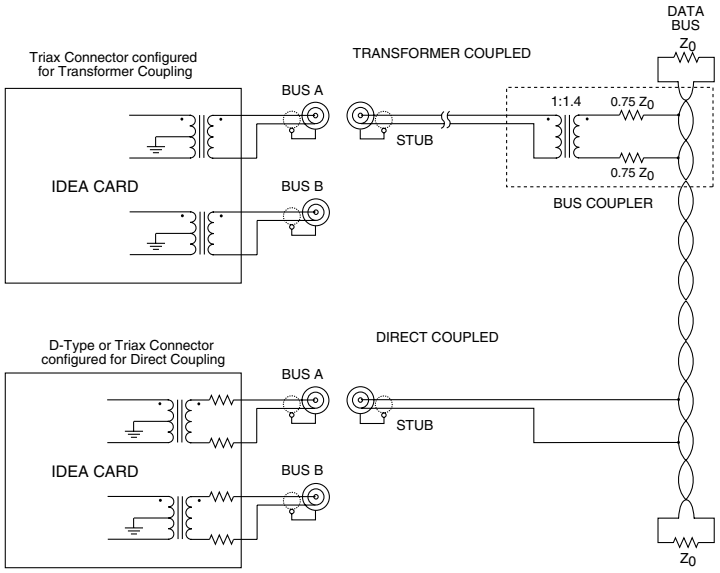


FIGURE 17. BUS-65547 INTERFACE TO A MIL-STD-1553 BUS

Figure 18 illustrates the BUS-65547 card connection to two Dual-Redundant Buses. Each Bus can be a BC/RT or MT. The configuration shown does not allow for BC/RT and MT on the same bus.

Figure 19 illustrates the BUS-65547 connection to one Dual-Redundant bus. Channel 1A BC/RT is tied to Channel 2A MT, providing a connection to a bus that is a BC/RT and MT.

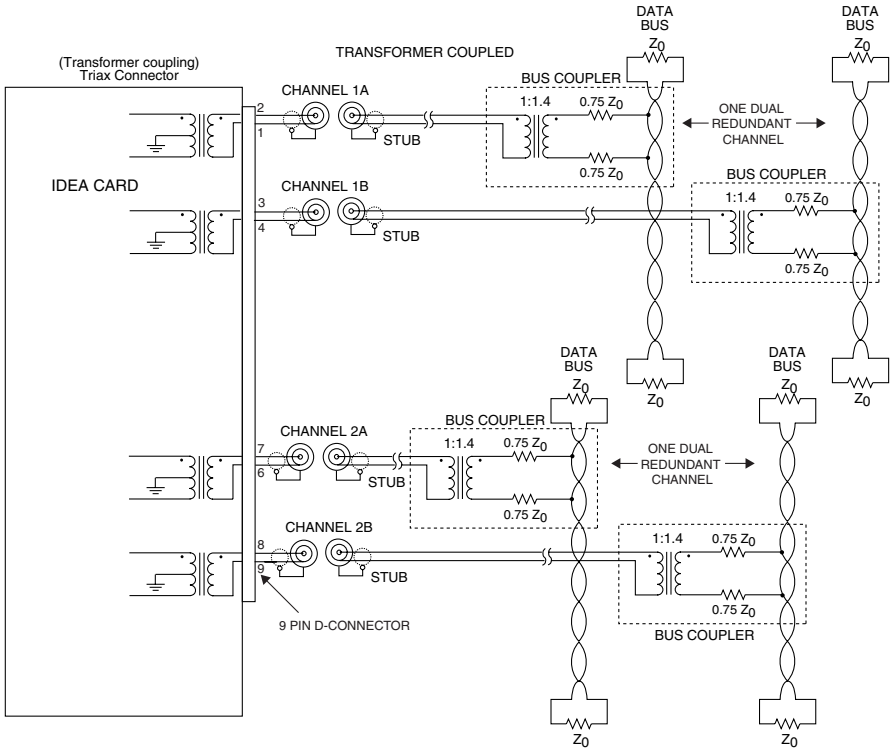


FIGURE 18. BUS-65547 INTERFACE TO TWO DUAL-REDUNDANT BUSES

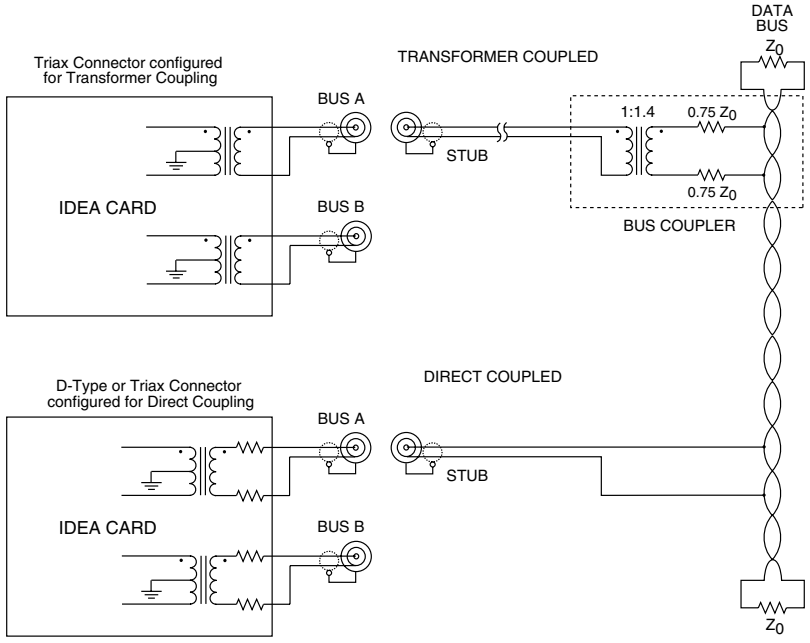
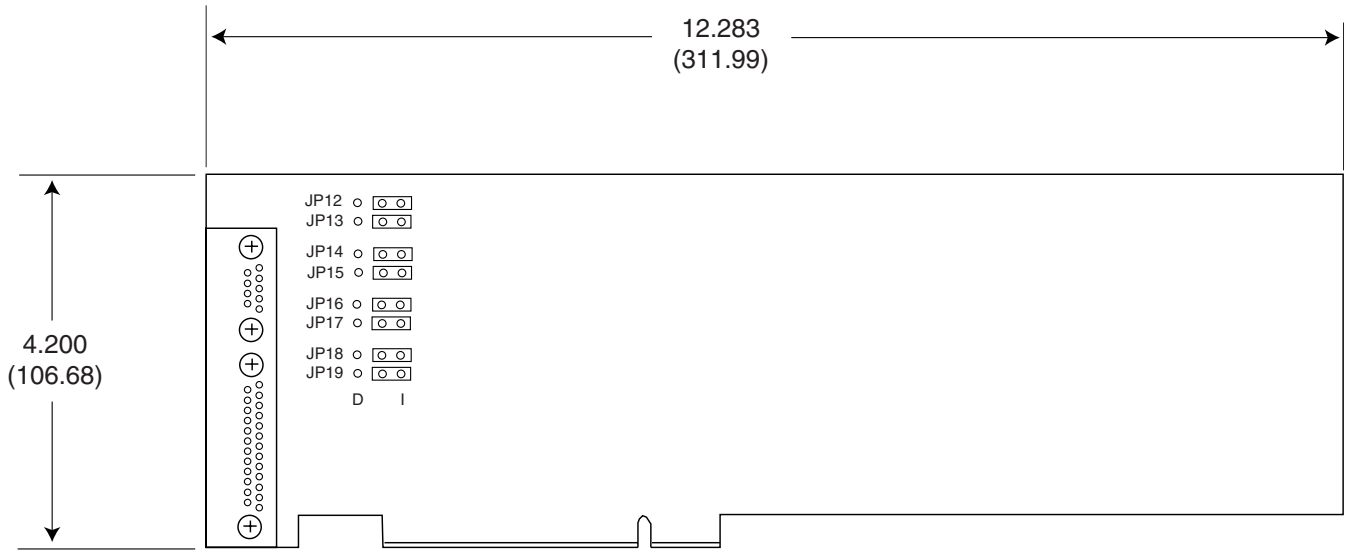


FIGURE 19. BUS-65547 INTERFACE TO ONE DUAL-REDUNDANT BUS

TABLE 5. 9-PIN CONNECTOR DESCRIPTIONS	
PIN	**FUNCTION**
1	Channel 1 A TX*
2	Channel 1 A TX
3	Channel 1 B TX
4	Channel 1 B TX*
5	Shield
6	Channel 2 A TX*
7	Channel 2 A TX
8	Channel 2 B TX
9	Channel 2 B TX*

TABLE 6. D-TYPE PIN DESCRIPTIONS			
PIN	**FUNCTION**	**PIN**	**FUNCTION**
1	HIO-0	14	GND
2	IN3-1	15	OUT3-1
3	IN2-1	16	OUT2-1
4	IN1-1	17	OUT1-1
5	IN0-1	18	OUT0-1
6	IN0-2	19	+5V
7	IN2-2	20	N/C
8	IN3-2	21	N/C
9	N/C	22	IN1-2
10	OUT3-2	23	HIO-1
11	OUT2-2	24	HIO-2
12	OUT1-2	25	HIO-3
13	OUT0-2		



DIMENSIONS ARE IN INCHES (MM).

FIGURE 20. BUS-65547 MECHANICAL OUTLINE

ORDERING INFORMATION

BUS-65547

└ MIL-STD-1553 Simulator and Tester (IDEA) Card.

Includes software:
C Libraries for Windows 95/98 and NT
Menu for Windows 95/98 and NT
Reconstructor for Parameter monitor for Windows 95/98 and NT
Drivers and Library support for VenturCom RTX

These products contain tin-lead solder.

DDC®
©Data Device Corporation
105 Wilbur Place, Bohemia, New York 11716-2426

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

Headquarters - Tel: (631) 567-5600 ext. 7257 or 7381, Fax: (631) 567-7358
West Coast - Tel: (714) 895-9777, Fax: (714) 895-4988
Southeast - Tel: (703) 450-7900, Fax: (703) 450-6610
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)8141-349-087, Fax: +49-(0)8141-349-089
Sweden - Tel: +46-(0)8-54490044, Fax +46-(0)8-7550570
Japan - Tel: +81-(0)3-3814-7688, Fax: +81-(0)3-3814-7689
World Wide Web - http://www.ddc-web.com

C-11/06-0

PRINTED IN THE U.S.A.